FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

A MULTIMEDIA INDEXING AND RETRIEVAL FRAMEWORK FOR MULTIMEDIA DATABASE SYSTEMS

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Chengcui Zhang

2004

To: Dean R. Bruce Dunlap
College of Arts and Sciences

This dissertation, written by Chengcui Zhang, and entitled A Multimedia Indexing and Retrieval Framework for Multimedia Database Systems, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

_____
Yi Deng

_____
Jainendra K. Navlakha

_____
Nagarajan Prabakar

_____
Mei-Ling Shyu

_____
Shu-Ching Chen, Major Professor

Date of Defense: May 21, 2004

The dissertation of Chengcui Zhang is approved.

_____
Dean R. Bruce Dunlap
College of Arts and Sciences

_____
Dean Douglas Wartzok
University Graduate School

Florida International University, 2004

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude and appreciation to my dissertation advisor Professor Shu-Ching Chen for his guidance, support, suggestions and encouragement while this dissertation was being conducted. I am also indebted to Professors Yi Deng, Jainendra K Navlakha, Nagarajan Prabakar of the School of Computer Science, and Professor Mei-Ling Shyu of the Department of Electrical and Computer Engineering, University of Miami, for accepting the appointment to the dissertation committee, as well as for their suggestions and support.

The financial assistance I received from the School of Computer Science is gratefully acknowledged.

I would like to thank all my friends and colleagues whom I have met and known while attending Florida International University. In particular, I would like to thank Min Chen, Na Zhao and Guo Chen for their support, encouragement, and generous help. My special thanks go to Paresh Gupta for his help with English and presentation. Finally, my utmost gratitude goes to my husband, parents, and sister, for their support and encouragement with which this work would have been impossible.

ABSTRACT OF THE DISSERTATION

A MULTIMEDIA INDEXING AND RETRIEVAL FRAMEWORK

FOR MULTIMEDIA DATABASE SYSTEMS

by

Chengcui Zhang

Florida International University, 2004

Miami, Florida

Professor Shu-Ching Chen, Major Professor

The main challenges of multimedia data retrieval lie in the effective mapping between low-level features and high-level concepts, and in the individual users' subjective perceptions of multimedia content.

The objectives of this dissertation are to develop an integrated multimedia indexing and retrieval framework with the aim to bridge the gap between semantic concepts and low-level features. To achieve this goal, a set of core techniques have been developed, including image segmentation, content-based image retrieval, object tracking, video indexing, and video event detection. These core techniques are integrated in a systematic way to enable the semantic search for images/videos, and can be tailored to solve the problems in other multimedia related domains. In image retrieval, two new methods of bridging the semantic gap are proposed: 1) for general content-based image retrieval, a stochastic mechanism is utilized to enable the long-term learning of high-level concepts from a set of training data, such as user access frequencies and access patterns of images. 2) In addition to whole-image retrieval, a novel multiple instance learning framework is proposed for object-based image retrieval, by which a user is allowed to more effectively search for images that contain multiple objects of interest. An enhanced image segmentation algorithm is developed to extract the object information from images. This segmentation algorithm is further used in video indexing and retrieval, by which a robust video

shot/scene segmentation method is developed based on low-level visual feature comparison, object tracking, and audio analysis. Based on shot boundaries, a novel data mining framework is further proposed to detect events in soccer videos, while fully utilizing the multi-modality features and object information obtained through video shot/scene detection.

Another contribution of this dissertation is the potential of the above techniques to be tailored and applied to other multimedia applications. This is demonstrated by their utilization in traffic video surveillance applications. The enhanced image segmentation algorithm, coupled with an adaptive background learning algorithm, improves the performance of vehicle identification. A sophisticated object tracking algorithm is proposed to track individual vehicles, while the spatial and temporal relationships of vehicle objects are modeled by an abstract semantic model.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

LIST OF DEFINITIONS

DEFINITION PAGE

Chapter 1. Introduction and Motivation

With the advances in the multimedia technologies for media capture, storage, and transmission, the production of digital multimedia content has increased tremendously in recent years, which leads to a strong need in efficient and effective storage and retrieval of multimedia data. However, traditional Database Management Systems (DBMSs) cannot handle multimedia data effectively because of the differences between the characteristics of traditional data (plain text data, for example) and multimedia data. Lack of techniques for efficient content-based access and multimedia data mining hinders the availability of these data to the general users.

In brief, multimedia data has the following characteristics [Lu00]:

1) Multimedia data is storage-consuming. For example, a 20-minute video (in MPEG format) of medium frame size (320×240) with medium quality requires above 100 MB storage.
2) Multimedia data is rich in information. An important issue related to multimedia data is that they may require significant levels of intermediate processing or interpretation, such as image or acoustic signal processing, which is not required for processing traditional textual data. The results of intermediate processing usually contain a set of parameters representing the multimedia content. For example, in order to describe the color content of images, the color histogram is often used and contains more than 64 columns per image for a reasonably good representation.
3) Unlike text data which has clear semantic structure, multimedia data are represented either by a set of spatially ordered pixel values (images), or by temporally sequenced visual samples and audio samples (video), which prevents the automatic content recognition by computers. In addition, the meaning of multimedia data is highly subjective since different people may have entirely different interpretations about the same image/video.

Due to the special characteristics of multimedia data which are quite different from traditional data, new multimedia indexing and retrieval techniques are required. In response to such an increasing need, the multimedia database management systems (MDBMSs) have emerged and attracted a great deal

of attention in recent years. There are many applications that can be supported directly or indirectly by MDBMSs, such as the video on demand application, query by example for image and audio databases, medical image analysis and management, multimedia education, face recognition, etc. To be feasible, a multimedia database management system has to combine DBMS, information retrieval, and content-based retrieval techniques. Parts of multimedia data, such as the text annotation data (authors, date of creation, keywords, etc.) for a multimedia document, are structured and can be handled by using traditional DBMS and information retrieval techniques. For the content description of multimedia data, content-based retrieval together with the corresponding indexing techniques is used to offer the user an efficient way of finding and retrieving the multimedia data qualified for the matching criteria of users' queries from the database. It involves automatically extracting the features for the unique characteristics of each image/video, and a matching process between the query media and the media data stored in the database.

Previous research in content-based retrieval has focused on developing automatic tools for extracting low-level features (e.g., color, texture, and shape) [Smith99, Smeulders00] and searching image/video databases using low-level similarity measures [Dimitrova00, Santini99]. However, it has been well recognized that a significant gap exists between such low-level feature indexing techniques and high-level semantic queries. A major reason is that while users want to retrieve images/videos by the high-level (semantic) concepts (e.g., "Give me those images that contain Bill Clinton"), most of the existing automatic algorithms can only extract low-level features (e.g., color, texture, shape, motion, etc.). Bridging the gap between low-level features and high-level concepts is the most challenging problem in multimedia indexing and retrieval [Smeulders00]. To address this issue, an advanced indexing and retrieval scheme, which maps low-level features to high-level concepts is needed.

Recently, there is a growing interest in developing new techniques to learn a user's high-level query concepts through active learning [Naphade01, Hanjalic01, Cox00, Dimitrova03] using users' relevance feedback [Rui98]. Another approach focuses on comprehending how people view images as similar, on the basis of perception [Chang02]. Others have tried to go even further addressing the

multimedia content indexing and retrieval problem at a knowledge discovery and data mining level [Assfalg02]. In this dissertation, our effort fits into this new trend of research which combines active learning and data mining techniques for multimedia database indexing and retrieval. However, while data mining and learning techniques can assist users to find their information more readily and to enable the automatic annotation of the large amount of image/video data, there are challenges that hinder their popularity. The challenges include how to integrate low-level feature matching into high-level semantic learning, the overhead incurred during users' relevance feedback, the "cold-start" problem in long term (collective) learning, the scalability and pre-filtering issues in multimedia data mining, etc. Therefore, there is a need to develop new approaches in order to improve the performance of the existing learning and data mining algorithms for content-based image/video indexing and retrieval.

The remainder of this chapter is organized as follows. In next section, the significance of an integrated multimedia indexing and retrieval framework is discussed in further details. In Section 1.2, the proposed solution for constructing such a framework is given. In Section 1.3, the main contributions of this dissertation are presented. The scope and limitations of the framework are discussed in Section 1.4. Section 1.5 gives the outline of the dissertation.

1.1 Significance of an Integrated Multimedia Indexing and Retrieval Framework

In this dissertation, an integrated multimedia indexing and retrieval framework as well as the approach for multimedia database management are proposed. The media studied in this dissertation include image, video, and audio data. Three major characteristics distinguish this framework from other types of multimedia information management systems. The first distinction is its comprehensive coverage and support of basic image/video management operations, including image/video parsing, indexing, and retrieval. More importantly, all these components are integrated in a systematic way, in which a set of core multimedia technologies form the basis of the proposed solutions, and can be applied and tailored to solve the problems in other multimedia related domains. In this dissertation, we demonstrate their applicability by using a real world application – video surveillance applications for Intelligent

Transportation Systems (ITS). The same set of technologies are tailored in order to meet the special needs of this application and are used to detect vehicle objects from surveillance videos, model their spatio-temporal relationships for video database indexing and retrieval, and so on. The second difference is its ability to assist users intelligently in finding their desired information and progressively adapt to the user's high-level query concepts. In particular, both long-term learning for the general users and instant learning for the individual users are considered in this framework. Long-term learning aims to improve a satisfactory level of general users in a long run, while instant learning is able to retrieve multimedia data that are highly appealing to the user's preference in a short time. This would not only improve the retrieval performance, but is also likely to increase the popularity of multimedia database management systems. Third, this framework also integrates multimedia data mining techniques to discover the semantic events in a large amount of video data. Consider a typical event-based query where a user is acquiring all the goal events in a large soccer video database. These soccer videos in the database can be of various production styles and be held in different places. In this case, it would be very difficult and time-consuming to derive the appropriate rules for event detection. In order to benefit from these video data sets, multimedia data mining techniques can be used to extract implicit, previously unknown, and potentially useful information from such a large database [Han92]. However, directly applying data mining techniques to the video features will not yield satisfactory results due to the noise and the small ratio of interesting events (for example, soccer goal events only constitute <1% of the total video data). In order to solve this problem, in this dissertation, we add an additional data cleaning phase prior to the data mining phase, and use heuristic rules in the data cleaning phase to reduce the noise and narrow down the search space. On the other hand, the computation intensive procedure of selecting appropriate features as well as their thresholds is left to the data mining phase.

1.2 Proposed Solution

Our expectation is that semantic retrieval of multimedia data will increase the ability to automatically annotate multimedia content and will lead to a high level of semantic abstraction. Therefore, the

objectives of this dissertation are to explore new paradigms and to develop innovative algorithms for the systems that support data indexing and efficient retrieval of multimedia data, especially image and video data from multimedia databases. The user will be allowed to perform content-based image retrieval (CBIR) through a general image retrieval interface (whole-image retrieval) or an object-based retrieval interface. In the general content-based image retrieval, we enable the long-term learning (collective learning) through the use of a stochastic mechanism. In addition to the whole-image retrieval, the object-based image retrieval enables the user to phrase queries in terms of objects, and it enables the computer to search the image database for similar objects. For example, object-based image retrieval enables the computer to extract similar surfaces from a GIS aerial photo database according to a given coherent surface. It has been recognized that the object-based method is able to retrieve images that are considered poor matches by the whole-image retrieval [Maxwell01], which can complement the general content-based image retrieval. Hence, it is more desirable to support this type of content-based image retrieval. In this dissertation, an innovative method using machine learning techniques (multiple instance learning) and unsupervised image segmentation techniques is developed for retrieving multiple objects of interest. It should be pointed out that the indexing and retrieval of static images is also the base for video data management. For example, after video parsing, the output video segments/clips are represented visually by their key frames which are static images. While query by key frame is a basic query type in video retrieval, the approaches to realize and optimize content-based retrieval for image databases can be definitely applied to video key-frame retrieval. In other words, the techniques and approaches developed for content-based image retrieval are not restricted to image databases; instead, they can be directly applied to video databases.

Another focus of this dissertation is the effective integration of multi-modality data (visual and audio features) for semantic video indexing and retrieval (for example, event-based indexing and retrieval). In order to do that, a video is first segmented into more coherent segments like video shots/scenes. A robust video shot/scene segmentation method is developed in this study based on low-

level visual feature comparison, object tracking, and audio analysis. As previously mentioned, multimedia data are usually poorly structured and highly varied in content. To handle these characteristics, data mining techniques have been applied in video features. In this dissertation, a novel framework based on data mining techniques is proposed to detect the events in soccer videos, while fully utilizing the multi-modality features and object information obtained through video shot/scene detection.

The major research problems in this study are summarized as follows:

1) Research Problems in *Content-Based Image Retrieval*: (1) more accurate content-based retrieval results appealing to users' high-level concepts; and (2) more flexibility in specifying image database queries to meet the requirements of different users (for example, object-based retrieval instead of global image retrieval).
2) Research Problems in *Video Parsing*: (1) more accurate video scene/shot detection; (2) the higher-level video event recognition based on video scenes/shots and audio clues; and (3) automatic object detection, feature extraction, and object tracking while parsing the video data.
3) Research Problems in *Video Indexing and Retrieval*: (1) the appropriate video indexing method to model video clips including their visual features and the spatio-temporal characteristics of the video objects; and (2) the high-level semantic indexing and annotation for video clips such as the detection of interesting events/highlights in news or sports videos.

In response to the above research problems, a set of methodologies, including image segmentation, image retrieval, object tracking, video segmentation, video event detection using data mining, etc., have been developed. Brief discussions on these methodologies are given as follows:

- For salient object extraction from image/video data, the approach will be an enhanced image segmentation method using wavelet analysis and unsupervised classification. Image segmentation is a process of segmenting an image into a set of non-overlapping regions with each of them having homogeneous features. The outputs of image segmentation are the classified regions (also called segments) which are expected to correspond to different types of objects. For example, the

foreground objects and background objects.

- For the general content-based image retrieval, the subjectivity of users' high-level concepts will be considered and integrated into the retrieval process via relevance feedback. Also, the user's access history to the database, including access pattern and access frequency, will be recorded and then utilized in the retrieval process by using a stochastic process. The pre-filtering techniques can be applied in the retrieval process in order to reduce the search space. As for the object-based image retrieval, using the object information produced by the proposed image segmentation algorithm, a multiple instance learning framework is developed to collect users' relevance feedbacks and learn users' high-level concepts represented by multiple objects of interests through neural networks.
- For video segmentation, the techniques from image processing and pattern recognition, such as color histogram, pixel-wise comparison, and object tracking, etc., are integrated to detect the video shot boundaries. Based on that, with the aid of audio content analysis for each video clip, the higher-level video segmentation such as scene/event detection can be realized. In fact, there is no commonly agreed-upon definition for a video event. Instead, a video event can only be defined within the context of a specific domain, so the scope of video event recognition in this dissertation targets one specific application domain - soccer game videos. Particularly, a novel multi-modal data mining framework using the decision tree logic is developed for soccer event (goals, corner kicks, etc.) detection.
- In order to extend the techniques developed for multimedia databases to some practical application domains such as traffic video surveillance, more auxiliary techniques, such as background subtraction, adaptive background learning, vehicle identification, occlusion handling in object tracking, and spatio-temporal modeling, are developed for this purpose. It should be pointed out that all the above mentioned techniques are based on the object information produced by the enhanced image segmentation method tailored for traffic surveillance videos.

- For performance evaluation, the criteria, such as *precision* and *recall*, are used to justify the accuracy for image retrieval, video segmentation (shot/scene detection), and video event detection. In addition, complexity analysis in terms of processing time is conducted to evaluate the efficiency of the proposed video shot detection. Furthermore, the cross-validation method is used to evaluate the robustness of the proposed data mining framework for soccer event detection.

1.3 Contributions

The main contributions of this dissertation are as follows:

1) First, in object segmentation, an effective yet efficient image segmentation method called WavSeg is proposed in this dissertation. This method is based on wavelet analysis and unsupervised classification. The good balance of effectiveness and efficiency of this method enables image segmentation as an important component in content-based image retrieval and video indexing and retrieval.
2) Second, in the general content-based image retrieval, a new mechanism called Markov Model Mediator (MMM) [Shyu01a] is used to facilitate the searching and retrieval process for content-based image retrieval. Different from the common methods, this stochastic mechanism carries out the searching and similarity computing process dynamically, taking into consideration not only the low-level image features but also other characteristics of images such as their access frequencies and access patterns. The mining of user access patterns helps the system to gradually adapt to general user's needs. Thus, it enables the long-term learning (collective learning) in content-based image retrieval. Also, the MMM mechanism can be deemed as a new alternative in the efforts towards bridging the 'semantic gap' between the low level image features and high level user concepts.
3) Third, in object-based image retrieval, a new method is proposed to effectively discover a specific user's concept patterns when multiple objects of interest (e.g., foreground and background

objects) are involved in content-based image retrieval. The proposed method incorporates Multiple Instance Learning into the user relevance feedback in a seamless way to discover where the specific user's most interested objects/regions are and how to map the local image features of that region(s) to that user's high-level concepts. A three-layer neural network is used to model the underlying mapping progressively through the user feedback and learning procedure. The input object information of this method is obtained through the proposed image segmentation technique.

4) Most of the existing work in video database modeling and management focuses on either the *Video Parsing* or *Video Indexing* and *Retrieval*, but few of them touch both of the two areas. However, without the *Video Parsing* step providing the available video features and video segments (scene/shots), the *Video Indexing and Retrieval* step would have no knowledge about what kinds of video features are available given the current state of art in computer vision, pattern recognition, and image processing. It is often the case that research works focusing on *Video Indexing* and *Retrieval* make too many assumptions about the available video features. On the other hand, analyzing the user's requirements for *Video Retrieval* can in turn give a guide to *Video Parsing* in video feature extraction and the granularities of video segmentation. The work in this dissertation covers both of the two areas, in which an event-based video indexing framework for soccer videos is proposed based on the results of video parsing – video shots and the corresponding video features (visual and audio features) produced during video parsing. The visual and audio features extracted through shot/scene detection are fully explored and utilized in shot-based event detection in a seamless way. This novel multi-modal framework also employs data mining techniques to handle the heterogeneous video features and aid the detection of various soccer events like soccer goals, corner kicks, etc. Our expectation is that semantic-based video indexing and retrieval will increase the ability to automatically extract the video semantic contents and will lead to a higher level of semantic abstraction.

5) The proposed multi-modal framework will definitely benefit the multimedia research society in that it aims to fully utilize and integrate the two totally different media data (visual and audio features) into the video database management systems. The information derived from the two media data can compensate for each other and produce more accurate results in both video segmentation and video event detection. In addition, the feature extraction approaches and indexing techniques developed for handling visual and audio features in video data can be directly applied to and benefit the image databases and audio databases.
6) The automatic process for *Video Parsing,* together with the semi-automatic process for *Video Indexing,* will greatly reduce the manual labor, which makes it possible for making practical use of multimedia databases in multimedia education.
7) Moreover, in order to demonstrate the potential of the proposed framework, some of the techniques and methodologies developed for this framework, such as object extraction and object tracking, are applied to a few practical application domains, such as traffic surveillance. In fact, to our best knowledge, there is no such framework like the one proposed in this dissertation, which focuses on a set of key techniques that are closely related to each other (such as object extraction, object segmentation, and object indexing), and the same set of core techniques can be applied to a wide scope of areas from content-based image retrieval and video indexing, to the multimedia database for Intelligent Transportation Systems.

1.4 Scope and Limitations of the Framework

The proposed framework has the following assumptions and limitations:

- Inaccurate object segmentation results are expected because of the current state of the art techniques in computer vision, pattern recognition and image processing. Similar to the 'semantic gap' issue in content-based retrieval, using low level features alone cannot provide accurate descriptions/representations for semantic objects. In fact, sometimes the segmentation algorithms

tend to split objects into several regions, and sometimes multiple objects may be merged into one segment. For example, a dog in one image corresponds to one single region, while in another image a dog object is split into two segments. Although we try to improve the object segmentation method by estimating a relatively 'good' initial partition for it, the results are not perfect, considering the gap between low level feature representation and the high level semantics of the objects. A user in the loop is needed if the performance of object extraction is the major concern.

- Similar to the limitation in image segmentation, the 'semantic gap', also exists in content-based image retrieval. Although we can use techniques such as relevance feedback, region-based retrieval, data clustering, and data mining to reduce the gap, the perfect retrieval results with perfect rankings according to a given query image, are not permitted due to the current state of art techniques in computer vision [Smeulders98] and beyond the scope of this dissertation. As such an example, different users may have totally different perceptions about the same image. As shown in Figure 1.1, given an image with both tiger object and grass object, one user may be interested in the tiger object, while another one may have more interest in the grass. It is a general agreement in the CBIR society that it is impossible to build a fully-automatic, general-purpose CBIR system which can match users' high level concepts perfectly.

Figure 1.1 Example of "*Semantic Gap*" - One user is only interested in the tiger object, while another may be interested in just the lawn [Carson02].

- Video parsing and indexing will also suffer from the limitation in object segmentation. In addition, even given perfect object segmentation, it is still a challenging task to distinguish 'foreground' objects from 'background' objects. Although some indications, such as the object

variance and spatial layout, can be used to help distinguish them, one hundred percent identification is impossible due to the combined effects of camera motion and object motion [Smeulders98]. Also, even though we combine several techniques in shot/scene detection and expect them to compensate for each other, there are still some extreme cases where the proposed framework cannot work well. Such cases include the very long gradual transition (for example, long dissolve between two video shots), and big objects in fast motion.

- As for object tracking used in both video indexing and specific application domains (for example, Intelligent Transportation Systems), there are two assumptions made on the size and shape of the objects being tracked. Considering the situation when an overlapping happens between two objects separate from each other in a previous or a later frame. Assume that there are no major differences in the sizes and the shapes of those two objects, and the sizes and shapes of the same object do not change a great deal in the consecutive frames. Under these two assumptions, the proposed algorithm can handle the situation of two objects' occlusion very well, and recover the object locations as well. Although there are assumptions being made in this method, it also has the advantages of efficiency and without much loss of accuracy, which is critical for real-time processing.

1.5 Outline of the Dissertation

The organization of the dissertation is as follows. In Chapter 2, the literature reviews are given in the areas of content-based retrieval for image and video data as well as the indexing methods and data structures for efficient retrieval.

Chapter 3 describes the proposed multimedia indexing and retrieval framework for multimedia database systems. Each component of the framework is discussed in details.

In Chapter 4, we present the proposed content-based retrieval framework for image databases. The approach of image segmentation and dynamic process for general image retrieval is described. In

addition, as an extension to general image retrieval, a prototype framework for object-based image retrieval is also included in this chapter.

In Chapter 5, a video parsing and indexing framework using joint video and audio clues are presented. An object tracking method based on image segmentation is also developed for video shot detection. Furthermore, a video scene detection method is developed based on the results of shot detection. In addition, a novel data mining framework for video event detection is presented, taking the soccer videos as an example. A comparative study on performance evaluation is given to show the potential of the framework.

In Chapter 6, a practical application, traffic video surveillance, is introduced. Then how to apply the multimedia processing techniques on this application is also discussed, together with the current experimental results presented.

In Chapter 7, the conclusions are given along with the proposed future work.

Chapter 2. Literature Review

In this chapter, the existing approaches and methodologies in content-based retrieval for image and video databases are summarized.

2.1 Content-Based Image Retrieval

The term CBIR (Content-Based Image Retrieval) has been widely used to describe the process of retrieving desired images from a large collection on the basis of features (such as color, texture and shape) that can be automatically extracted from the images themselves. The features used for retrieval can be either primitive or semantic, but the extraction process must be predominantly automatic. In contrast to the text-based approach, CBIR operates on a totally different principle, retrieving stored images from a collection by comparing features automatically extracted from the images themselves. The commonest features used are mathematical measures of color, texture or shape. A typical CBIR system allows users to formulate queries by submitting an example of the type of image being sought, or offers some alternatives such as selection from a palette or sketch input. The system then identifies those stored images whose feature values match those of the query most closely and displays these images on the screen.

2.1.1 Syntactic Features (Global Features) Used in CBIR Systems

According to [Web5], image features used in content-based retrieval can be classified into two groups: syntactic features and semantic features. Syntactic features include color, texture, and shape, etc. Semantic features refer to objects (humans, animals, buildings, artworks, etc.) and topics (pollution, demonstration, etc.).

A. Color features:

Several methods for retrieving images on the basis of color similarity have been described in the literature, but most are variations on the same basic idea. Each image added to the collection is analyzed

to compute a *color histogram* which shows the proportion of pixels of each color within the image. The color histogram for each image is then stored in the database. At search time, the user can either specify the desired proportion of each color, or submit an example image from which a color histogram is calculated. The matching technique most commonly used is *histogram intersection*. Variants of this technique are now used in a high proportion of current CBIR systems. Methods of improving the original technique include the use of *cumulative color histograms*, combining histogram intersection with some element of spatial matching [Stricker96], and the use of region-based color querying [Carson97]. Other color features include color layout, dominant color, etc.

The above-mentioned methods based on color features can be applied into different color spaces as listed below:

- RGB Color Space: This acronym stands for Red-Green-Blue. It is device-dependent and normally used on monitors. RGB is called primary colors because a color is produced by adding the three components, red, green and blue.
- HSL/HSV Color Space: This acronym stands for Hue, Saturation, and Luminosity. Hue is the perception of the nuance. It is the perception of what one sees in a rainbow. The perception of Saturation is the vividness and purity of a color. For example, a sky blue has different saturation from a deep blue. Luminosity, also called brightness, is the perception of an area to exhibit more or less light. Although the representation of the colors in the RGB space is quite adapted for monitors, HSV space is preferred for a human being.
- CIE-Lab/Luv Color Space: The CIE defined the Lab/Luv spaces in order to get more uniform and accurate color models. **L** defines lightness, **a** denotes red/green value, and **b** the yellow/blue value.

B. Texture features:

The ability to retrieve images on the basis of texture similarity may not seem very useful. But the ability to match on texture similarity can often be useful in distinguishing between areas of images with similar color. A variety of techniques have been used for measuring texture similarity. From these it is possible to calculate measures of image texture such as the degree of *contrast, coarseness*, *directionality* and *regularity*, or *periodicity, directionality* and *randomness* [Liu98]. Alternative methods of texture analysis for retrieval include the use of Gabor filters [Ma98] and fractals [Kaplan98]. A recent extension of the technique is the texture thesaurus developed by Ma and Manjunath [Ma98], which retrieves textured regions in images on the basis of similarity to automatically derive code words representing important classes of texture within the collection.

C. Shape features:

Unlike texture, shape is a fairly well defined concept, and there is considerable evidence that natural objects are primarily recognized by their shape. A number of features characteristic of object shape, but independent of size or orientation, are computed for every object identified within each stored image. Two main types of shape feature are commonly used - *global* features such as aspect ratio, circularity and moment invariants and *local* features such as sets of consecutive boundary segments. Alternative methods proposed for shape matching have included elastic deformation of templates [Zhong00], comparison of directional histograms of edges extracted from the image, and *shocks*, skeletal representations of object shape that can be compared by using graph matching techniques [Tirthapura98]. Queries to shape retrieval systems are formulated either by identifying an example image to act as the query, or as a user-drawn sketch.

D. Other features:

Several other types of image feature have also been proposed. Most of these rely on complex transformations of pixel intensities which have no obvious counterpart in any human description of an

image. The most well-researched technique of this kind uses the *wavelet transform* to model an image at several different resolutions. Promising retrieval results have been reported by matching wavelet features computed from query and stored images [Liang98]. Another method giving interesting results is *retrieval by appearance*. Two versions of this method have been developed, one for whole-image matching and one for matching selected parts of an image. The part-image technique involves filtering the image with Gaussian derivatives at multiple scales [Ravela98], and then computing differential invariants; the whole-image technique uses distributions of local curvature and phase.

2.1.2 Relevance Feedback (RF)

In Content-Based Image retrieval, the biggest challenge is to bridge the semantic gap between human's high-level semantic concepts. Translating or converting the query posed by a human to the low level global features seen by the computer illustrates the problem in bridging the semantic gap. However, the semantic gap is not merely translating high level features to low level features. Due to the emotional and intellectual aspects in users' queries, understanding the meaning behind the query becomes essential. In order to reduce this gap, one of the most widely used approaches is called relevance feedback (RF).

Relevance feedback is an interactive process in which the user judges the quality of the retrieval results returned by the system by marking those images that the user perceives as truly relevant. This information is then used to refine the original query. In the past few years, the RF approach to image retrieval has been an active research field. This powerful technique has been proved successful in many application areas. Various ad hoc parameter estimation techniques have been proposed for RF. The method of RF is based on the most popular vector model [Buckley95] used in information retrieval. RF techniques do not require a user to provide accurate initial queries, but rather estimate the user's ideal query by using positive and negative examples (training samples) provided by the user. The fundamental goal of these techniques is to estimate the ideal query parameters (both the query vectors and the associated weights) accurately and robustly. Most of the previous RF researches [Rui98, Aksoy00,

Chang99] are based on low-level image features such as color, texture and shape and can be classified into two approaches: query point movement and re-weighting techniques [Lu00a]. However, this process should be dealt with in a real-time manner in the loop because the metric dynamically depends upon the user's feedback and the context. More recently, [Tong01] proposed a support vector machine active learning algorithm for conducting effective relevance feedback for image retrieval. The algorithm selects the most informative images to query a user and quickly learns a boundary that separates the images that satisfy the user's query concept from the rest of the dataset.

2.1.3 Region-Based Image Retrieval

Rather than dealing with the global features over the whole image, the region-based retrieval systems segment an image into several homogeneous regions, and then the features for each region can be extracted and compared. As a result of continuous effort towards this area, some region-based image retrieval systems have been proposed. For example, Blobworld [Carson02] is an early region-based image retrieval system that segments the images into blobs based on color and texture features, and queries the blobs by using some high-dimensional index structure. For each image, a similarity score is given by a fuzzy combination over the similarity scores between the query blobs and their most similar blob in that image. However, the multi-region queries remain unclear and unaddressed in this work. The SIMPLicity [Wang01] system uses the integrated region matching technique (IRM) to allow many-to-many matching between regions in two images. WALRUS [Natsev99] is another region-based retrieval system, segmenting images by using wavelets. The use of wavelets for segmentation has been promising. In its retrieval process, the sum of the sizes of all the retrieved regions for each image is calculated, and only those images with their matched region sizes exceeding some threshold are returned. In [Jing02], an indexing schema customized especially for region-based image retrieval was proposed.

Recently, the research in integrating the two major techniques, relevance feedback and region-based retrieval, has gained much attention. The representative is the RF-based multiple instance learning mechanism proposed in [Huang02] which seamlessly integrates the RF and the single region-based

retrieval. However, some issues still remain open in region-based retrieval [Jing02]: 1) the image similarity measure between images based on region similarity; 2) the scalability of region-based retrieval systems; and 3) the relevance feedback strategy in region-based systems.

2.1.4 Prototype Content-Based Image Retrieval Systems

Among a number of CBIR systems and techniques, IBM's QBIC system [Flickner95], Virage's VIR engine [Virage], VisualSEEk [Smith96], Metaseek [Beigi98], PhotoBook [Pentland94], and Blobworld [Carson02], are some representative systems. [Venters] gave a very good review of the current content-based image retrieval systems.

- Blobworld

The Blobworld system [Carson02], developed at the University of California, Berkeley, supports color, shape, spatial, and texture matching features. The system is region-based and automatically segments each image into regions, which correspond approximately to objects or parts of objects in an image. The system allows querying at object level rather than global image properties by retrieving image regions that correspond approximately to objects. The segmentation results are displayed to users for both query image and returned images so that the correspondence between query region and returned regions is revealed. The Blobworld system forms part of the Berkeley Digital Library Project.

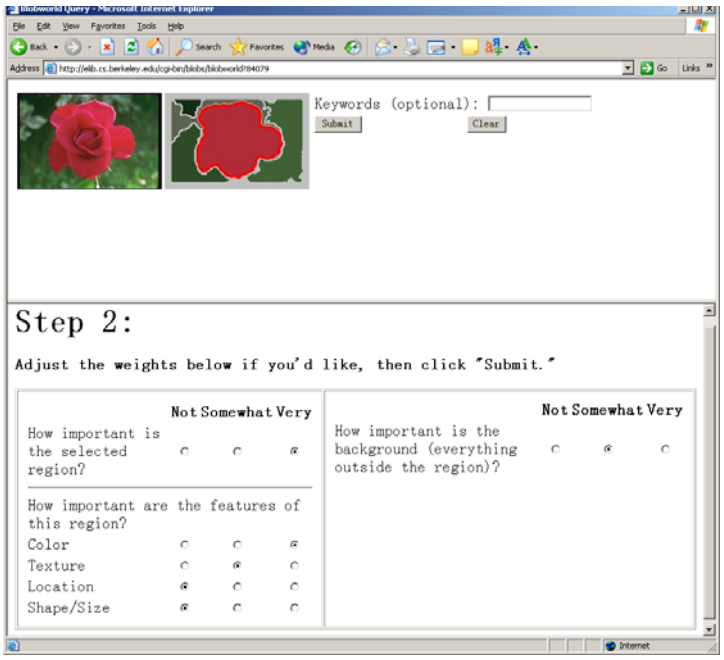


Figure 2.1 The query interface of Blobworld system (http://elib.cs.berkeley.edu/photos/blobworld/).

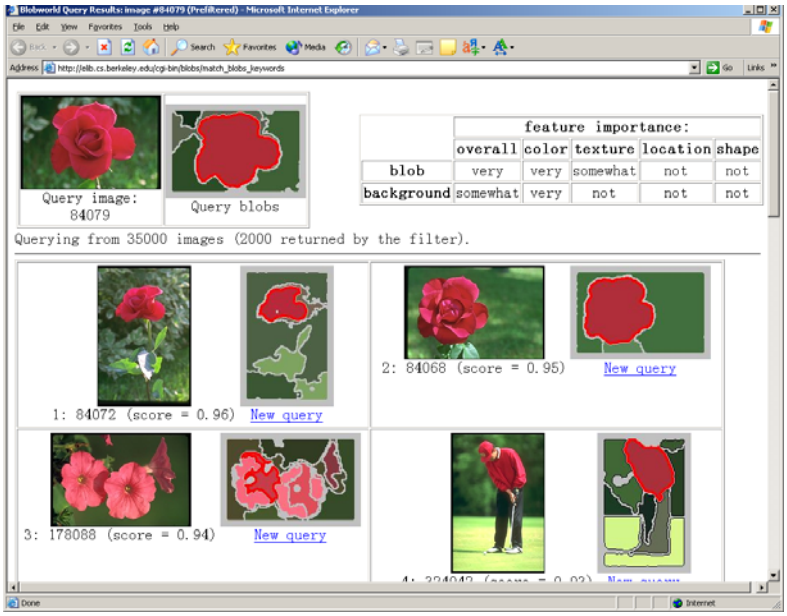


Figure 2.2 The query result display interface of Blobworld system (http://elib.cs.berkeley.edu/photos/blobworld/).

- VisualSEEk

VisualSEEk [Smith96], which was developed at the Department of Electrical Engineering, Columbia University, supports color and spatial location matching features. VisualSEEk supports several methods for conducting queries that specify the colors, sizes and arbitrary spatial layouts of color regions, which include both absolute and relative spatial locations. The system uses a novel approach for region extraction and representation based upon color set back-projection where salient color regions from images are automatically extracted.

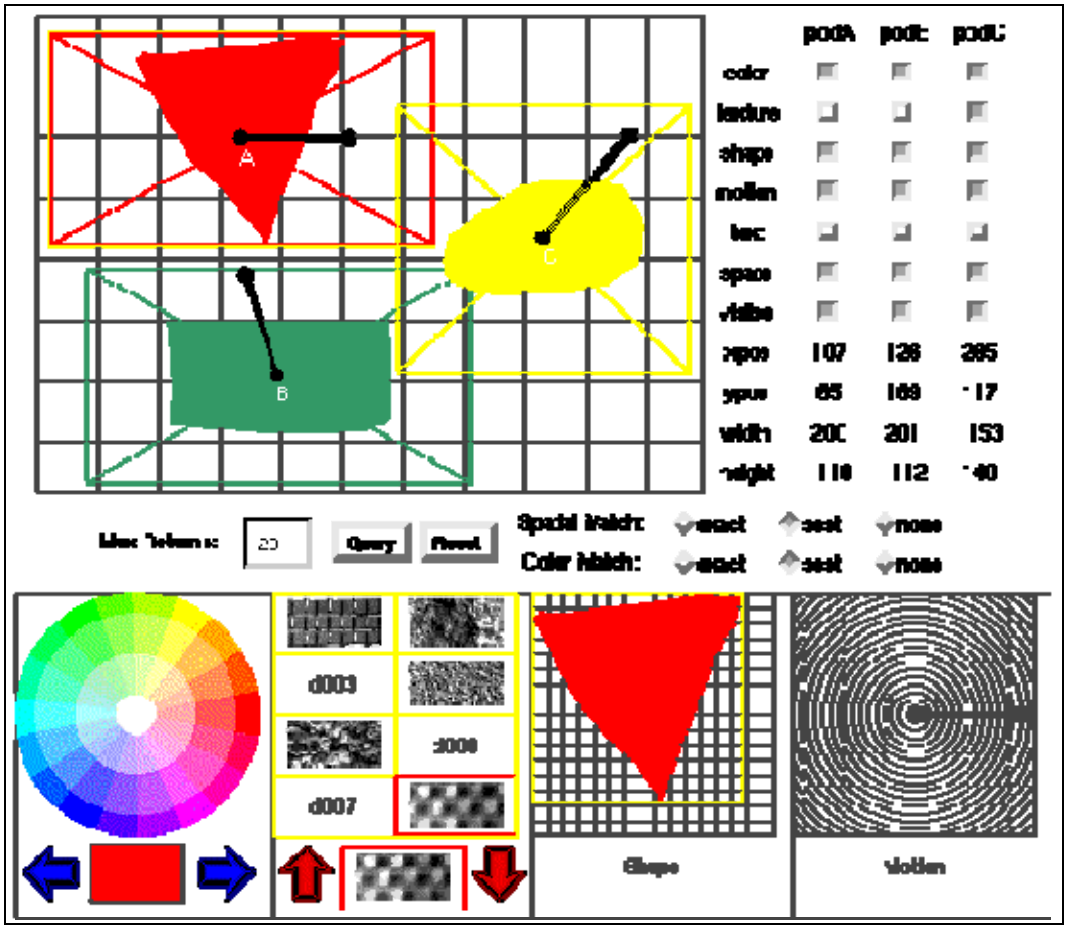


Figure 2.3 The interface of VisualSEEk system.

- PhotoBook

The Photobook system, developed at the Massachusetts Institute of Technology Media Laboratory, supports color, shape and texture matching features. The system calculates features vectors for the image characteristics, which are then compared to compute a distance measure utilizing one of the systems matching algorithms, including Euclidean, mahalanobis, divergence, vector space angle, histogram, Fourier peak, wavelet tree distances and user-defined matching algorithms via dynamic code loading. Photobook is freely available to the academic community.

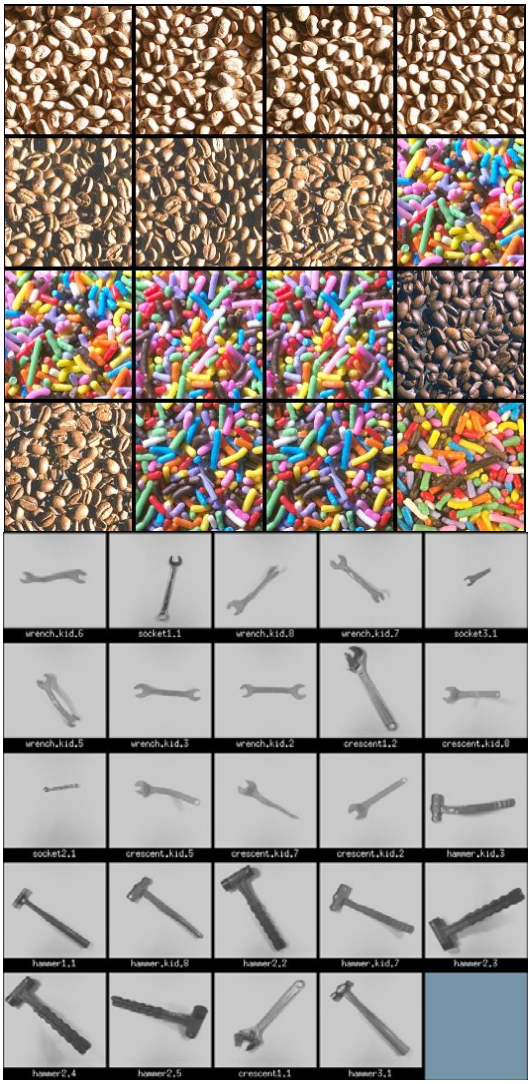


Figure 2.4 The texture retrieval of PhotoBook system (http://web.media.mit.edu/~tpminka/photobook/).

- ImageScape

The ImageScape system, developed at the Institute of Advanced Computer Science, Leiden University, supports query by sketch and is a World Wide Web image search system.

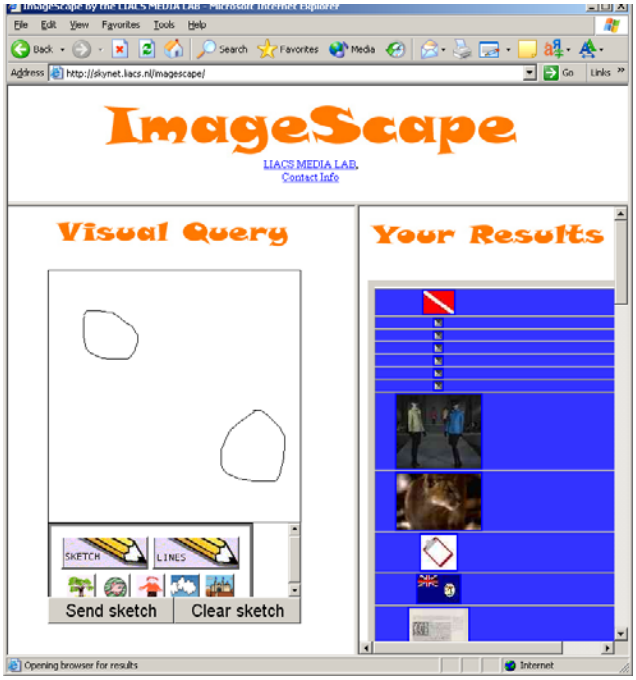


Figure 2.5 The interface of ImgeScape visual query system (http://skynet.liacs.nl/imagescape/).

- MARS

MARS (Multimedia Analysis and Retrieval System) [Rui98] was developed by the University of Illinois at Urbana-Champaign. It employs relevance feedback in image retrieval. The weighting of image features is directed by the user dynamically. Therefore, the image representation could adapt to different applications and different users.

- iFind

Microsoft Research system iFind [Lu00a] also explicitly uses relevance feedback in image retrieval. This system attempts to get away from just low-level image features by addressing the semantic content in images. Images are associated with keywords, and a semantic net is built for image access based on these, integrated with low-level features. Keywords are linked to images in the database, with weights assigned to each link. The degree or relevance, the weight, is updated on each relevance feedback round. An image

can be associated with multiple keywords, and these keywords can be obtained manually or extracted from the HTML tag associated with an image.

2.2 Video Parsing, Indexing, and Retrieval

2.2.1 Video Parsing

As a fundamental phase for automatic video indexing and retrieval, video parsing has emerged in response to the application demand for finer grain access to video. Due to the recent rapid advances in communication and multimedia computing technologies, digital video has become very popular in many applications such as education, training, video conferencing, video on demand (VOD), and news services. The requirements for efficiently accessing the mass amount of video data are becoming more and more important. Traditionally, when users want to search certain content in videos, they need to fast forward or rewind to get a quick overview of interest on the videotape. This is a sequential process and users do not have a chance to choose or jump to a specific topic directly. How to organize video data and provide the visual content in compact forms becomes important in multimedia applications [Yeo97]. Therefore, users can browse a video sequence directly based on their interests so that they can get the desired information quicker and the amount of data transmission can be reduced. Also, users should have the opportunity to retrieve the video materials using database queries. Since video data contains rich semantic information, database queries should allow the users to get high level content such as *scenes* or *shots*.

- Video shot detection

A video *shot* is a video sequence that consists of continuous video frames for one camera action. The goal of video shot detection is to separate the video into a set of shots that can be used as the basic units for video indexing and browsing. Usually, the existing techniques for shot change detection can be grouped into the methods operating on uncompressed video data and those operating on compressed video data.

Gargi et al. [Gargi98] gave a survey on video indexing, as well as the video segmentation

techniques used in the uncompressed data domain. In the uncompressed domain, the shot change detection algorithms process the uncompressed video, and a similarity measure between the successive frames is defined [Nagasaka95, Zhang93]. The basic idea in pixel-level comparison is to compute the differences in the values of the corresponding pixels between two successive frames. This method is very simple, but the disadvantage is that it is very sensitive to object and camera movements. In our method, we embed this method combined with histogram comparison into the techniques of object tracking and image segmentation in order to overcome its shortcomings, and at the same time to improve the efficiency. Another kind of comparison technique used in the uncompressed domain is the block-wise comparison, where each frame is divided into several blocks that are compared with their corresponding blocks in the successive frame. Instead of pixel-by-pixel matching, block-wise comparison methods use the local characteristics (such as the mean and variance intensity values) of the blocks to reduce the sensitivity to object and camera movements. This method is more robust, but it is still sensitive to fast object movement or camera panning. Moreover, since the mean and variance values of a block are not good enough to represent the block's characteristics, it is highly possible to introduce incorrect matching between two blocks that have the same mean and variance values but with totally different contents [Xiong98].

Hampapur et al. [Hampapur94] presented a model-driven approach where the models for video edit effects are developed and shot boundary extraction is carried out based on these models. Basically they computed the so-called chromatic images by dividing the change in the gray level of each pixel between two images by the gray level of that pixel in the second image. During dissolves and fades, this chromatic image assumes a reasonably constant value. Unfortunately, this technique is very sensitive to camera and object motion.

A further improved method to reduce the sensitivity to camera and object movements is the histogram-based comparison. Since the object moving between two successive frames is relatively small, their histograms will not have big differences. Therefore, it is more robust to small rotations and slow

variations [Pass99, Swain93]. However, the histogram-based method has its potential problems. That is, two successive frames will possibly have similar histograms but with different contents. Another approach based on the low-level features of images was proposed by Zabih et al. [Zabih95]. Their proposed approach used the intensity edges between successive frames to detect shot cuts. However, as the authors have pointed out, the weakness of their approach is the false positives due to the limitations of the edge detection method.

Some recent works also try to bring the object tracking technique to video segmentation. [Gunsel98] tried to employ object tracking into the scene cut detection, but the detection and tracking of the semantic objects of interest need to be specified manually by the user, and a bunch of template frames containing the semantic objects of interest were selected by the users in order to track the semantic objects of interest, which is not feasible for automatic and unsupervised processing. Moreover, the method proposed in [Gunsel98] is domain-specific instead of a more general framework. For example, it only focuses on TV news video, and the objects of interest are only channel logos and anchorpersons. Generalized block matching methods that allow affine transformations in intensity have been used in this paper for object tracking purposes. However, affine transformation is still sensitive to luminance changes. In [Vinod97], the objects were also manually selected by the users and tracked based on simple color similarity.

There are also many shot change detection algorithms in the compressed domain, especially in MPEG format videos. Since the encoded video stream already contains many features such as the DCT (discrete cosine transform) coefficients and motion vectors, it is suitable for video shot change detection. In [Arman93], the DCT coefficients of **I** frames were used as the similarity measure between the successive frames. Yeo and Liu [Yeo97] used the dc-images to compare the successive frames. In [Hwang98], Hwang and Jeong utilized the changes of directional information in the DCT domain to detect the shot breaks automatically. Lee et al. [Lee00] further improved the DCT coefficient-based method. Although fast and efficient video analysis can be achieved in the compressed data domain, it

should be noted that this advantage is based on the complex compressing process done before people can obtain the compressed video data. Moreover, since this kind of method is based on some specific compressed video data format such as MPEG, it is not general enough. The research work in uncompressed video data still remains important and necessary in the literature. In addition, there are few works addressing automatic semantic object tracking in the compressed video data domain.

In addition, not many of the approaches in the literature handle explicit performance analysis in terms of video processing rate (frames/second), especially in the uncompressed data domain [Ngo00, Truong00]. Even in the compressed data domain, only a few approaches have addressed this issue. For example, in [Lee00], the processing rate is given as ranging from 10.3~11.4 frames/sec. The authors in [Hwang98] claimed that their proposed method is practical without providing a performance analysis. [Yeo95] did not explicitly analyze the processing rate in terms of frames/sec, but reported the relative performance analysis table based on the video sequence which was taken several years ago when gradual transitions and special edit effects were not widely used.

- Video scene detection

A video *scene* usually consists of a sequence of related video shots that follow certain semantic rules [Jiang00]. Unlike video shot detection, video scene detection needs to consider the semantic meaning of related shots based on video features and audio features. In the past few years, research in the area of video scene detection has focused on either visual or audio information alone. However, using audio or visual information alone often cannot provide a satisfactory solution. For example, two shots belonging to the same scene may have similar audio features but with totally different visual features due to the different shooting angles and/or scales. In this case, by using visual information alone it is virtually impossible to know whether two shots are semantically related or not. Thus more and more research has been done trying to consider both audio and visual information in video scene detection.

Despite several initial successes, finding a good way to combine audio and video information is still challenging. In [Yoshitaka01], the candidate scene boundaries are extracted from the video data based on

the extraction of visual effects such as dissolve or fade in/out. However, for audio data, they only analyze the starting and ending audio frames of each shot using the average power of subbands. Since the audio features within a short time duration often cannot represent the characteristics of the whole shot, monitoring the audio changes within a short time around the video shot boundaries cannot guarantee to identify the audio changes between the neighboring video shots. For example, when an audio change occurs before the video shot change, the method in [Yoshitaka01] cannot work well. In [Sundaram00], the authors used a finite-memory model to separately segment the audio and video data into scenes, and then applied two ambiguity windows to merge the audio and video scenes. In [Muramoto00], the authors did not combine the audio and video segmentation results. A hierarchical segmentation approach was proposed in [Huang98] to detect scene breaks and shot breaks at different hierarchical levels.

2.2.2 Video Indexing and Retrieval

Indexing video data is essential for providing content-based retrieval. Video indexing is typically conducted either by manual annotation or by the features (visual and audio) automatically extracted from video parsing phase. The indexing effort is directly proportional to the granularity of video access. An excellent survey about video indexing is provided in [Brunelli99].

Existing work on content-based video retrieval and video indexing can be grouped into four main categories:

- *Low level indexing* provides access to video based on low level features like color, texture etc. The global features are extracted from the video data and organized on some distance metric. Then similarity-based matching is used to retrieve the video data from the database. Their primary limitation is the lack of semantics attached to the features, also known as ‘semantic gap’.
- *Middle level indexing* uses the salient video objects to represent the spatio-temporal characteristics of video clips. The spatial and temporal relationships among video objects are indexed and modeled in order to answer the spatio-temporal related queries [Chang98, Chen02]. Since salient video objects contain both low level features and some good indications for high

level concepts, we refer to this kind of indexing technique as middle level indexing.

- *High level indexing* uses a set of predefined index terms to annotate video segments output by video parsing/segmentation phase. Considerable manual labor for indexing and annotation is involved in this process. The index terms are organized by high level ontological categories like static, action, etc. This approach is suitable for small quantities of video data.
- *Domain Specific Indexing* techniques are effective in their intended domain of application (news video indexing, for example). However, the limitation of these techniques lies in their narrow range of applicability.

As for content-based video retrieval and navigation, it is important that both retrieval and navigation appeal to the user's visual intuition. There are basically two types of queries in video retrieval. In visual query, the low level features (color, texture, and temporal variance of video shots or their representative frames, etc.) of video clips are used. In concept query, the presence of specific video objects or events is used to find out the video shots/scenes conforming to the requirements. Since fully automatic object extraction is still impossible, some extent of user interaction is necessary in this process. Recently, there has been considerable research on content-based video modeling and retrieval of video data based on video objects [Wasfi99, Chen01, Chen02]. Video objects are the physical objects that appear in the video data. Users of a video database may want to retrieve video data through queries on video objects' properties and spatio-temporal relationships among the video objects. The typical queries based on video objects can be categorized into query for spatial feature only, query for spatial relationship among objects, query for temporal feature only, query for temporal relationships among objects, and query for spatio-temporal relationship among objects.

Other approaches attempt to deal with higher level semantic aspects of video, as opposed to lower-level visual or audio feature. Much effort has gone into applying data mining or knowledge discovery techniques to classifying videos, detecting interesting events/highlights in videos, and so on. [Zhou02] gives a good summary of efforts to provide intelligent systems for video analysis and indexing.

2.2.3 Prototype Systems for Video Indexing and Retrieval

The Informedia project [Wactlar99] of Carnegie Mellon University aims at providing a digital video library for use in education, training, and entertainment. Their approach uses combined speech, language and image understanding technology to automatically transcribe, segment and index the linear video data. In Informedia I, they apply a speaker-independent speech recognizer to automatically transcribe video soundtracks into full-text information for future retrieval purpose. Informedia II allows for rapid retrieval of individual video paragraphs, as well as the capability for matching of similar faces and images.

The IBM CueVideo project at IBM Almaden also does searching of video content based on speech recognition, smart browsing and other video analysis techniques [CueVideo]. It also supports query by media other than text query. For example, the user query can include actual video shots and ask the system to retrieve video shots similar to the query. The techniques underlying include color correlograms matching for a video shot (a multidimensional data structure incorporating color histograms for frames spread throughout a shot) and automatic pre-classification of video shots into a semantic hierarchy [Smeaton02].

The VDBMS (Video Data Based Management System) project [VDBMS] developed at Purdue University aims to provide a full range of functionality for video database management. Currently it supports both search-by-content and search-by-streaming for video data. Two query operators, rank-join and stop-after algorithms, are implemented for video data. As videos may be considered streams of consecutive image frames, video query processing can be viewed as continuous queries over video data streams. From this viewpoint, a method for defining and processing video streams is also implemented through the query execution engine. Several algorithms are developed for video query processing expressed as continuous queries over video streams, such as fast forward, region-based blurring and left outer join, in which the window-join algorithm is a core operator for continuous query systems. The query interface of VDBMS is shown in Figure 2.6.

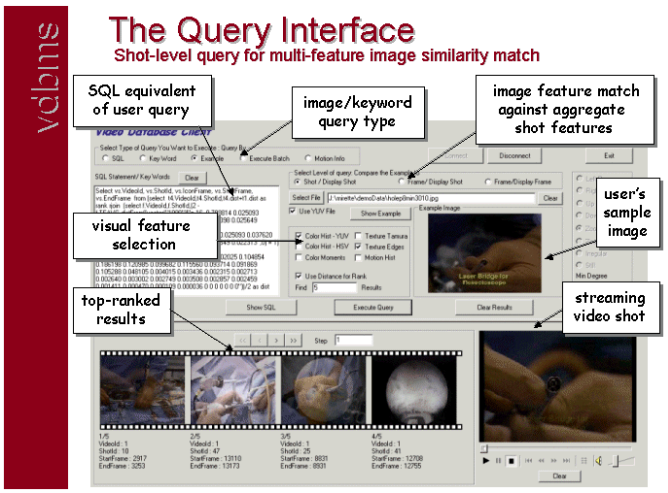


Figure 2.6 The query interface of VDBMS.

2.3 Techniques and Data Structures for Efficient Multimedia Database Retrieval

Efficiency is another important issue for multimedia database retrieval. When the number of stored objects and/or the number of dimensions of the feature vectors are large, linearly searching in full feature space is too slow and not feasible for large-scale databases. The situation becomes even worse when a multimedia database system normally uses a set of different feature vectors. The current MPEG-7 standard specifies many feature vectors (called descriptors). Techniques and data structures are required to organize feature vectors and fasten the search process so that the relevant feature vectors can be located quickly. The main aim of these techniques and data structures is to divide the multidimensional feature space into many subspaces so that only one or a few subspaces need to be searched for each query [Lu00]. Data structures can also be optimized for a certain type of query if it is known that only one specific type of query will be commonly used for a particular application. Different techniques and data structures differ in how subspaces are formed and how relevant subspaces are chosen for each query. Three approaches have been widely used for the sake of retrieval efficiency:

1) Feature space reduction and search space reduction: For feature space reduction, principal component analysis (PCA) and wavelet transform are two commonly used techniques to generate compact representations for original feature space. For search space reduction, there are various pre-filtering

processes [Lu02], such as filtering with structured attributes, methods based on triangle inequality, and filtering with color histograms [Hafner95].

2) Indexing and data structures for organizing image feature vectors. For indexing, quite a few data structures, approaches and techniques have been proposed to manage an image database and hasten the retrieval process, such as VA-file [Weber98], MB^{+}-trees [Dao96], 3DR-tree [Theodoridis98], STR-tree [Pfoser00], TB-tree [Pfoser00], and *k*-d tree [Lew00]. In [Faloutsos94], the system extracts and stores color, shape and texture features from each image added to the database, and uses R*-tree indices to improve search efficiency. White and Jain [White96] concluded that the VAMSplit R-tree provides better overall performance than the R*-tree, SS-tree, and optimized k-d tree variants. The QBIC system [Flickner95], for instance, uses pre-filtering technique and efficient indexing structure like R-trees to accelerate its searching performance. The latest version of the QBIC system incorporates more efficient indexing techniques, an improved user interface, the ability to search gray-level images, and a video storyboarding facility [Niblack98]. Heisterkamp and Peng [Heisterkamp03] proposes a novel KVA-File (kernel VA-File) that extends VA-File to kernel-based retrieval methods. As another example, the ImageScape system [Lew00] uses *k*-d tree as its indexing structures. However, even for R*-tree, it was not scalable to dimensions higher than 20. A new data structure, Dynamic Inverted Quadtree is introduced in [Vassilakopoulos95]. The authors claimed to outperform earlier tree data structures such as Fully Inverted Quadtree in terms of space and searching efficiency. But the overheads of using these complex index structures are considerable. For video retrieval, [Theodoridis98] uses 3DR-tree to index salient objects in a video database by treating the time as another dimension in the R-tree. The search computation complexity of almost all data structures increases exponentially with the number of feature vector dimensions. Thus, the number of dimensions of the feature vectors should be chosen to be as low as possible.

3) The more recent approaches, which seem to offer better prospects of success, are the use of similarity clustering of images and the use of neural networks. Clustering techniques allow hierarchical access

for retrieval and provide a way of browsing the database as a bonus [Jin98, Vellaikal98]. In [Zhang95], they proposed to use self-organization map (SOM) neural networks to construct the tree indexing structure in image retrieval. SOM is unsupervised and has the nature of dynamic clustering. Moreover, it has the potential of supporting arbitrary similarity measures.

Chapter 3. Overview of the Framework

Although multimedia data (image, video, audio, etc.) are different in terms of their processing, there are some similarities in the ways to handle these data. In fact, different processing methods are needed for different types of media data. Thus, a multimedia application needs more than one processing component and to have more than one output format in order to handle different media data. However, similarities exist in the management of these various multimedia data. First, the raw multimedia data are a sequence of bytes that have to be preprocessed in order to extract, store, and index the underlying semantics for content-based retrieval. Second, the semantics (such as salient objects) are extracted by a manual or an automatic processing on the raw multimedia data. Third, some media data have temporal or both spatio-temporal characteristics, which have to be modeled and indexed in support of spatio-temporal queries.

In this chapter, a general framework for multimedia database systems is presented. The framework uses several techniques in information retrieval and multimedia data (image, video, audio) indexing.

Figure 3.1 illustrates the overall components and processes of the proposed framework for multimedia database (especially image and video database) management systems. The two major components of this framework, as shown in

Figure 3.1, include video component and image component, and each component can be further divided into three steps: parsing, indexing, and retrieval. It should be pointed out that the basic composition unit of a video scene/shot is a video frame (key frame), which is actually a special instance of a general image. It can take a general image as the abstract type (parent type) and inherit all the attributes from image type and add a time dimension to model its temporal features. In this framework, key frames are used to represent the contents of a shot. By associating shots with key frames, the video component is connected with the image component.

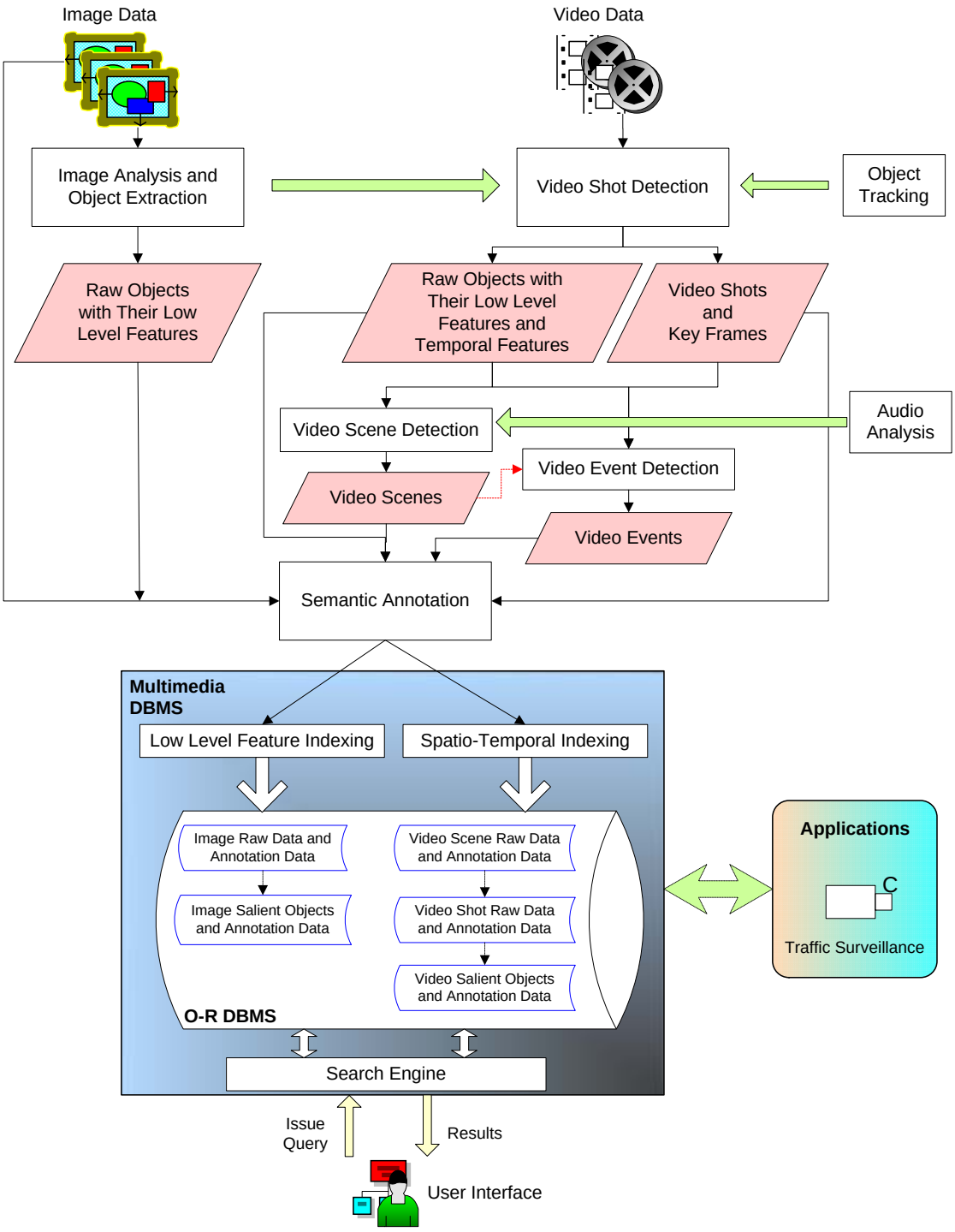


Figure 3.1 The proposed indexing and retrieval framework for multimedia database management systems.

3.1 Image Component

In image component, we model the image data as two parts: the *global part* and *salient object* part. By this way, we can distinguish the global features of an image from its finer granularity representation, the object-level features. In the *global part*, besides the raw image data, the global low level features (color, texture, shape, etc.) of the images are extracted, together with the annotation data (meta-data) describing the semantics of the whole image. In the *salient object* part, an image is segmented into several segments/objects, which have their own local low level features and annotations. As an example of such a representation model, an image may contain grass, white horse, and sky. Then the global color feature of such an image will probably contain 'green', 'white', and 'blue' color components. The global annotation of such an image may be 'natural scene'. However, as for the object level representation, the grass, horse, and sky may be represented by three different objects and with totally different color/texture features and annotations. In many cases, object information is more useful for content-based retrieval when a user's interest is on one or two objects inside the images. However, global features and global annotations are still useful when there is no significant object inside the images, or they can be used just for pre-filtering purpose. More details about object extraction, image feature indexing, and image annotation are discussed in the following subsections.

3.1.1 Object Extraction

Although there is no general definition for *salient objects*, usually they can be defined given a specific application domain. A similar concept to salient objects is 'foreground objects'. However, given the current state of computer vision and pattern recognition, it is almost impossible to achieve perfect segmentation for individual objects, or the identification of 'foreground' objects. Unlike global feature extraction, which is a fairly mature topic, image segmentation for object extraction is still far from perfect for automatic processing. As a consequence, user interactions are needed in this process. In this dissertation, to reduce the manual labor as much as possible, we adopt an unsupervised image

segmentation method which can achieve reasonably good results with no user interaction needed. This segmentation method is the Spontaneous Partition and Parameter Estimation (SPCPE) algorithm originally proposed by [Sista99]. However, this algorithm cannot always converge to an optimal condition due to the random initial partition it used. In order to solve this problem, a wavelet and template based method, WavSeg, is proposed in this dissertation to improve the initial partitions for SPCPE algorithm.

3.1.2 Image Annotation

As mentioned above, inaccurate segmentation results are expected due to the current state of art of image segmentation techniques. For example, an object may consist of several segments, while one big segment may correspond to more than one object. User interactions have to be integrated in order to identify the 'real' salient objects from a set of raw objects. After the salient objects are selected, the annotation data (meta-data) need to be added for both images and their salient objects. As for the image meta-data standards, a number of organizations have developed several of them. These include VRA Core [VRA], Visual Arts Data Service (VADS) [VADS], CNI/OCLC Image Metadata Workshop [CNI], NISO/CLIR/RLG Technical Metadata for Images Workshop [NISO], etc.

Manual annotation of image data is not the focus of this dissertation. However, through the automatic process of content-based image query and retrieval, we can expect that the existing image annotations in an image database can be propagated in an automatic or semi-automatic way as the similarity relationships among images in the database become more revealed and known to the database management system.

3.1.3 Content-Based Image Retrieval

The user query interface typically consists of a query formulation part and a result presentation part. Image queries can be specified in many ways. One way is to query the image in terms of annotation

words, or in terms of global image features that are extracted from the images, such as a color and texture. Another way is 'Query-by-example', in which the similarity matching is based on the global annotation or global low-level feature of the whole image. Instead of querying global features, a user can also conduct object-based queries. Relevance feedback is needed in providing positive or negative feedback about the retrieval results, so that the system can refine the search.

In this study, we aim to support both general content-based image retrieval and object-based retrieval.

- The general content-based retrieval system employs the Markov model mediator (MMM) [Shyu01a, Shyu00a] mechanism to retrieve images, which functions as both the searching engine and image similarity arbitrator to facilitate the functionality of an MDBMS. This stochastic-based mechanism provides the capability to learn the intra-database affinity and users' high-level concepts based on access patterns and access frequencies, without any user's interaction. In particular, this component also includes a pre-filtering phase to reduce the search space. Several experiments were conducted and the experimental query-by-example image query results of the proposed retrieval system were reported. The fact that the proposed stochastic content-based CBIR system utilizes the MMM mechanism and supports both spatial and color information offers more flexible and accurate results for user queries. The experimental results exemplify this point, and the overall retrieval performance of the presented system is promising.
- In object-based image retrieval, we propose a framework that incorporates Multiple Instance Learning into the user relevance feedback in a seamless way to discover the concept patterns of users, especially where the user's most interested region(s) and how to map the local feature vector of that region(s) to the high-level concept patterns of users. This underlying mapping can be progressively discovered through the feedback and learning process. The role user plays in the retrieval system is to guide the system learning process to his/her own focus(focuses) of attention. The retrieval performance is tested under a couple of conditions.

3.2 Video Component

In video component, video data can be viewed as having four tiers (scene, shot, key frame, salient objects) as shown in Figure 3.2, where a video clip consists of a set of video scenes/events, each scene contains several shots, each shot is represented by a few key frames with each key frame containing a set of salient objects. In video parsing, a video clip is first segmented into video shots, and then the key frames are extracted for these shots. Based on the video shots, key frames and audio content, video scenes can be identified. Further, the detection of video events/highlights can be carried out by using the rich information obtained through video parsing. The techniques used in video component are discussed in the following subsections.

3.2.1 Video Shot Detection

In this study, focusing on the uncompressed video data, we propose an innovative shot detection method using pixel-level comparison, histogram comparison, unsupervised image segmentation algorithm, and object tracking. All the above techniques can compensate for each other and produce more accurate criteria for shot change detection. It is worth mentioning that the image segmentation step used in image component can be directly applied into the video shot detection process. In addition, with only small modifications of the segmentation algorithm, the processing speed has been greatly increased, making real-time processing possible. As being shown in Chapter 5, the experimental results for shot detection are significantly better when compared with a well-recognized and credited algorithm called twin-comparison algorithm [Zhang93]. The proposed method also has the advantages of unsupervised process, fast parsing, and reusable objects information produced by image segmentation. Also, the object tracking method adopted in this step can generate the necessary information for building object trajectories and reasoning the relative spatio-temporal relationships among salient objects.

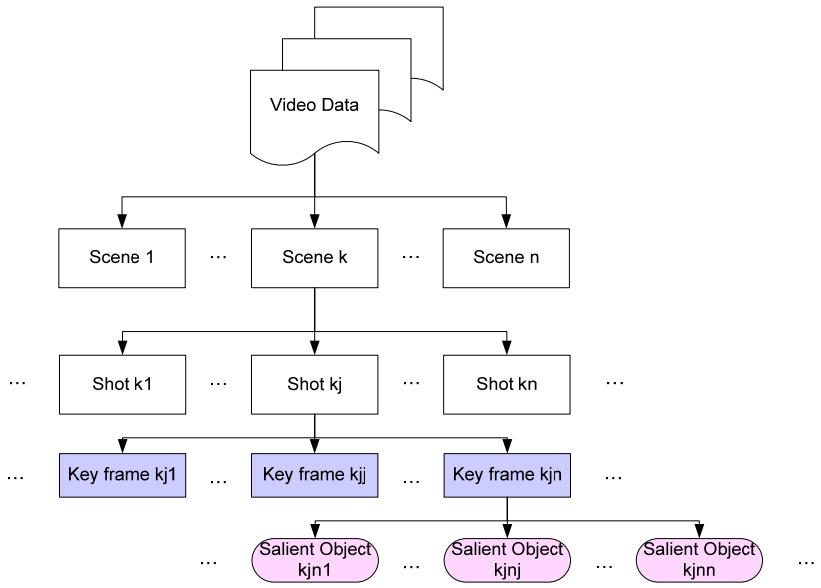


Figure 3.2 The proposed indexing and retrieval framework for multimedia database management systems.

3.2.2 Key Frame Selection and Spatio-Temporal Indexing for Salient Objects

By video shot detection, not only the shot boundaries but also the key frames and salient objects are extracted. It should be pointed out that a human in the loop is necessary to correct and select the salient objects needed. Since a video shot is represented by its key frames, key frame selection based on object-based representation is an important issue for video data. Choosing key frames based on regular time intervals may miss some important segments and segments may have multiple key frames with similar contents (redundancy). One image in each shot also may not capture the temporal and spatial relations of semantic objects; however, showing all key frames may confuse users when too many key frames are displayed at the same time. In this dissertation, we propose a key frame selection approach based on the temporal and spatial relations of salient objects in each shot. The temporal and spatial relations of salient objects are captured by the image segmentation algorithm and are modeled by the Multimedia Augmented Transition Networks (MATNs) and multimedia input strings [Chen01], where the MATN and multimedia string serve as the internal data structures for indexing salient objects.

3.2.3 Video Scene Detection

Video scene has higher-level semantic content than video shots. Sometimes, a video scene may correspond to a video event such as shooting the football. Automatic video scene change detection is a challenging task. Using audio or visual information alone often cannot provide a satisfactory solution. However, how to combine audio and visual information efficiently still remains a difficult issue since there are various cases in their relationship due to the versatility of videos. In this study, based on the proposed shot detection method, an effective scene change detection method [Chen02a] that adopts the joint evaluation of the audio and visual features is proposed. Experiment results show that the proposed method is better than those that separately segment the audio and video data into scenes and then integrate them. The future extension of this method can be made by adding domain knowledge for specific applications such as soccer game and news video; thus it is able to identify the events for specific application domains.

It should be pointed out that, in the proposed framework, video can be annotated at all the different levels through the hierarchy tree as shown in Figure 3.2.

3.2.4 Video Indexing and Retrieval

The proposed framework aims to support a variety of video database queries at different levels (scene, shot, key frame, salient objects). The basic query type is key frame query, which will exactly follow the query process for general image databases, with the only difference lying in the returned results (the system will return video shots/scenes, instead of static images). Yet another type of query is related to the semantic content of video clips.

In this framework, in addition to shot/scene based indexing for video data, we also support the semantic video indexing which is based on events/highlights. In particular, an effective data mining framework is developed for automatic extraction of interesting events in soccer videos. The extracted soccer events (goals, corner kicks, etc.) can be used for high-level indexing and retrieval of soccer videos. The proposed multimedia data mining framework first analyzes the soccer videos by using joint

multimedia features (visual and audio features). Then the data cleaning step is performed on raw video features with the aid of domain knowledge, and the cleaned data are used as the input data in the data mining process using the decision tree logic. The proposed multi-modal framework fully exploits the rich semantic information contained in visual and audio features for soccer video data, and incorporates the data mining process for effective detection of soccer events. This framework has been tested using soccer videos with different styles as produced by different broadcasters. The results are promising and can provide a good basis for analyzing the high-level structure of video content.

3.3 Special Applications

In the proposed framework, the Intelligent Transportation System (ITSs) is selected as a special application for multimedia database management systems. The goal here is to build a multimedia database for Intelligent Transportation Systems (ITSs). In this study, a learning-based automatic framework is proposed to support the multimedia data indexing and querying of spatio-temporal relationships of vehicle objects in a traffic video sequence. The spatio-temporal relationships of vehicle objects are captured via the proposed unsupervised image/video segmentation method and object tracking algorithm, and modeled by using a multimedia augmented transition network (MATN) model and multimedia input strings. An efficient and effective background learning and subtraction technique is employed to eliminate the complex background details in the traffic video frames. It substantially enhances the efficiency of the vehicle segmentation process and the accuracy of the segmentation results to enable more accurate video indexing and annotation. Four real-life traffic video sequences collected from different road intersections are used in the study experiments. The results show that the proposed framework is effective in automating data collection and access for complex traffic situations.

Chapter 4. Content-Based Retrieval for Image Databases

The objective of a CBIR system is to offer the user an efficient way in finding and retrieving those images that are qualified for the matching criteria of the users' queries from the database. Most of the existing CBIR systems retrieve images in the following manner. First, they build the indexes based on the low-level features such as color, texture and shape for the images in the database. The corresponding indexes of a query image are also generated upon the time the query is issued. Second, they search through the whole database and measure the similarity of each image to the query image. Finally, the results are presented to the user in a sorted order of the similarity matching level. In order to bridge the semantic gap between low level features and high-level concepts, relevance feedback (RF) and region-based retrieval are two major techniques in dealing with this issue. In particular, region-based retrieval is more preferred in the sense that it is more appealing to users' perception about images. In this chapter, first an object/region extraction method is introduced, together with the discussions of its problem and the proposed solutions. Second, a general-purpose image retrieval framework integrating global features and region features is proposed, where the user access pattern and access frequency are also considered as a good indication to bridge the 'semantic gap'. In addition, based on the object information extracted by the first step, a multiple instance learning framework for multi-object image retrieval is proposed, and the preliminary results are presented.

4.1 Object Extraction

4.1.1 Unsupervised Image Segmentation

In this study, we use an unsupervised segmentation algorithm to partition the video frames. First, the concepts of a class and a segment should be clarified. A class is characterized by a statistical description and consists of all the regions in an image that follows this description; whereas a segment is an instance of a class. This is illustrated in Figure 4.1. The light gray areas and dark gray areas in the right segmentation mask map represent two different classes, respectively. Considering the light gray class, there are in total four segments within this class (the CDs, for example). Notice that each segment is

bounded by a bounding box and has a centroid, which are the results of segment extraction. The details of object extraction after image segmentation will be discussed in Section 4.1.2.

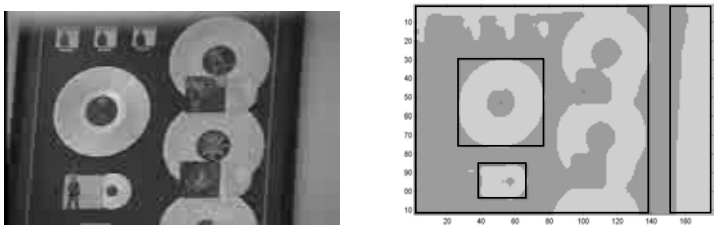


Figure 4.1 Examples of *classes* and *segments.* The original video frame is on the left and the segmentation mask map of the left frame is on the right.

The SPCPE (Simultaneous Partition and Class Parameter Estimation) algorithm [Sista99] is an unsupervised video segmentation method to partition video frames. A given class description determines a partition. Similarly, a given partition gives rise to a class description, so the partition and the class parameter have to be estimated simultaneously. In practice, the class descriptions and their parameters are not readily available. An additional difficulty arises when images have to be partitioned automatically without the intervention of the user. Thus, we do not know a priori which pixels belong to which class. In the SPCPE algorithm, the partition and the class parameters are treated as random variables.

Suppose there are two classes -- *class1* and *class2*. Let the partition variable be $\boldsymbol{c} = \{\boldsymbol{c_1}, \boldsymbol{c_2}\}$, and the classes be parameterized by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$. Also, suppose all the pixel values y_{ij} (in the image data Y) belonging to class k *(k=1,2)* are put into a vector $\boldsymbol{Y_k}$. Each row of the matrix Φ is given by (*1, i, j, ij*) and a_k is the vector of parameters $(a_{k0}, \ldots, a_{k3})^T$.

$$y_{ij} = a_{k0} + a_{k1} i + a_{k2} j + a_{k3} ij, \ \forall (i, j)\, y_{ij} \in c_k \quad (1)$$

$$\boldsymbol{Y_k} = \Phi\, a_k \quad (2)$$

$$\hat{a}_k = (\Phi^T \Phi)^{-1} \Phi^T \boldsymbol{Y_k} \quad (3)$$

The best partition is estimated as that which maximizes the a posteriori probability (MAP) of the partition variable given the image data Y. Now, the MAP estimates of $\boldsymbol{c} = \{\boldsymbol{c_1}, \boldsymbol{c_2}\}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ are

given by

$$(\hat{c},\hat{\theta}) = Arg\max_{(c,\theta)} P(c,\theta \mid Y)$$

$$= Arg\max_{(c,\theta)} P(Y \mid c,\theta)P(c,\theta) \quad (4)$$

Let $J(c,\theta)$ be the functional to be minimized. With the above assumptions, this joint estimation can be simplified to the following form:

$$(\hat{c},\hat{\theta}) = Arg\min_{(c,\theta)} J(c_1,c_2,\theta_1,\theta_2) \quad (5)$$

$$J(c_1,c_2,\theta_1,\theta_2) = \sum_{y_{ij}\in c_1} -\ln p_1(y_{ij};\theta_1) + \sum_{y_{ij}\in c_2} -\ln p_2(y_{ij};\theta_2) \quad (6)$$

The problem of segmentation thus becomes the problem of simultaneously estimating the class partition and the parameter for each class. About the parameter estimation, we can use equation (3) to directly compute the parameter for each assigned set of class labels without any numerical optimization methods. About the class partition estimation, we assign pixel y_{ij} to the class that gives the lowest value of $-\ln p_k(y_{ij} \mid \theta_k)$. The decision rule is:

$$y_{ij} \in \hat{c}_1 \text{ if } -\ln p_1(y_{ij}) \le -\ln p_2(y_{ij}) \quad (7)$$

$$y_{ij} \in \hat{c}_2 \text{ otherwise} \quad (8)$$

Just as shown in Figure 4.2(a), the algorithm starts with an arbitrary partition of the data in the first video frame and computes the corresponding class parameters. Using these class parameters and the data, a new partition is estimated. Both the partition and the class parameters are iteratively refined until there is no further change in them. We note here that the functional *J* is not convex. Hence its minimization may yield a local minimum, which guarantees the convergence of this iterative algorithm. Figure 4.2(b)-(e) show the segmentation results for the image shown on top at different iterations. The initial partition (randomly generated) is given in Figure 4.2(b), and the segmentation result after one iteration is given in Figure 4.2(c), which demonstrates how fast SPCPE converges to a local minima.

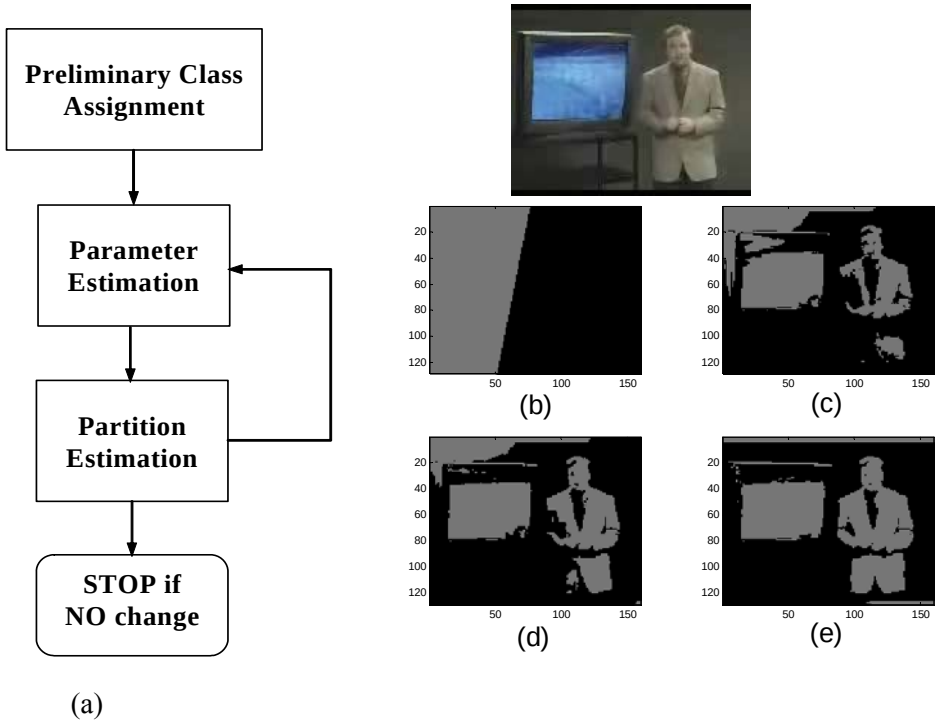


Figure 4.2 (a) The flowchart of the SPCPE algorithm; (b) Initial random partition; (c-e) Object segmentation results after 1, 2 and 3 iterations.

It should be pointed out that the SPCPE algorithm could not only simultaneously estimate the partition and class parameters, but also estimate the appropriate number of the classes in the meantime by some easy extension of the algorithm. Moreover, it can handle multiple classes rather than two. In our experiment, we just use two classes in segmentation since two classes are efficient and reasonably good for our purpose in this application domain.

4.1.2 Line Merging Algorithm (LMA) for Extracting Disconnected Objects/Segments

Contrasted to the traditional way to do disconnected segment extraction such as the *seeding and region growing* method used in [Sista99], a computationally simple and fast method called *line merging algorithm (LMA)* is proposed to extract the segments from the segmented frames. The basic idea is to scan the segmented frame either row-wise or column-wise. If the number of rows (columns) is less than the

number of columns (rows), then row-wise (column-wise) is used, respectively. For example, as shown in Figure 4.3, suppose the pixels with value '*1*' represent the segment we want to extract, we scan the segmented frame row by row. By scanning the first row, we get two lines and let each line represent a new segment so that we have two segments at the beginning. In scanning rows 2 to 4, we merge the new lines in each row with the lines in previous rows to form the group of lines for each segment. At row 5, we get one line and find out that it can be merged with both of the two segments, which means we must merge the two previously obtained segments to form a new segment so that we have only one big segment now. Similarly, at row 8, two lines belong to the same segment because they can be merged with the same line in row 7.

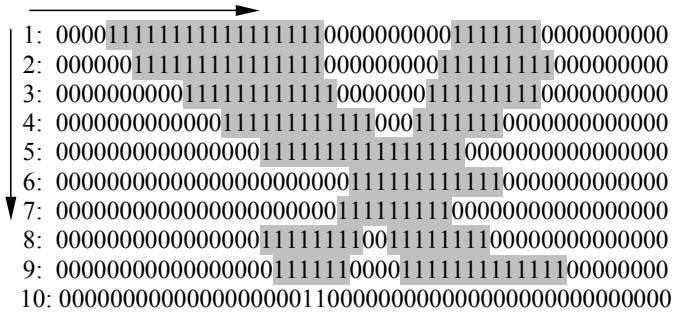


Figure 4.3 The segmentation mask map.

The pseudo codes for *line merging algorithm (LMA)* are listed below:

```
Algorithm: GetSegments(V, i, A[i])→ to get the new lines of each
row.
V: the input vector of segmented frame of row 'i';
'i': the current row we are scanning;
A[i]: a list to store the segments.
```

```
GetSegments(V, i, A[i])
   1) Number_of_segments = -1;
```

```
2) Segment D[col/2]; /* D is the temporary variable to store
   the line segments in row i. The maximal size of D is col/2.
   */
3) for j from 1 to col
4)   if V[j] == 1
5)       if j == 1 /* if the first line segment is at the
   beginning of the current row, add it to array D and
   increase the number of line segments. */
6)            number_of_segments++;
7)            D[number_of_segments].data = data; /* data
   contains the i and j values */
8)       else if V[j-1] == 0 /* detect a new line segment and
   add it to array D */
9)            number_of_segments++;
10)                D[number_of_segments].data = data;
11)           else D[number_of_segments].data += data;
            /* collect all the pixels belonging to the same
line segment together. */
12)           end if;
13)       end if;
14)       for k from 0 to number_of_segments /* copy the line
   segments in D to the data structure in A[i]. */
15)            A[i].Add(D[k]);
16)      end for;
```

Algorithm: GetBoundingBox(m[row][col])→ to combine A[i] and A[i-1] by checking each line in A[i] and A[i-1] and combining those lines which belong to the same segment.
m[row][col]: the input matrix of segmented frame of size row by column.

GetBoundingBox(m[row][col])

```
1) number_of_objects =0; /* initially there is zero object
   identified. */
2) for k1 from 1 to row
3)       GetSegments(m[k1][col], k1, A[k1]) /* get the line
   segments in current row */
4)       for k2 from 1 to A[k1].size
         /* between the current row and the previous row,
check and merge the corresponding line segments in them which
belong to the same object to one big segment. */
5)             for k3 from 1 to A[k1-1].size
6)                   if Segment Sk1 in A[k1] ∩ Segment Sk2 in
                      A[k1-1] != null
7)                        combine Sk1 and Sk2 into one
                            segment
8)             end for
9)       end for
10)     end for
```

Compared with the *seeding and region growing* method, the proposed algorithm extracts all the segments and their bounding boxes as well as their centroids within one scanning process, while the *seeding and region growing* method needs to scan the input data for indeterminate times depending on the layout of the segments in the frame. Moreover, the proposed algorithm needs much less space than the *seeding and region growing* method.

4.1.3 WavSeg-An Enhancement to SPCPE

The original SPCPE segmentation method starts with a randomly generated initial partition. Hence, different initial partitions yield different local minima. Unfortunately, random initial partition often leads to unreasonable segmentation results. An intuitive solution for this is to select the smallest local minimum among a set of local minima although it may not be the global minimum. In the proposed solution [Chen01a], a number of local minima (e.g., *20*) are computed and the smallest local minimum is used.

Since the computational requirement for each local minimum is very little, the overall computation needed for the best local minimum is not much. Two methods are used to generate those twenty initial partition candidates. By the *straight-line partition method*, the area of the original texture images is partitioned by an arbitrarily generated straight-line across the whole image area. Different areas separated by the straight-line represent different classes. Figure 4.4 gives four examples of the randomly generated straight-line partitions (the number of classes is two).

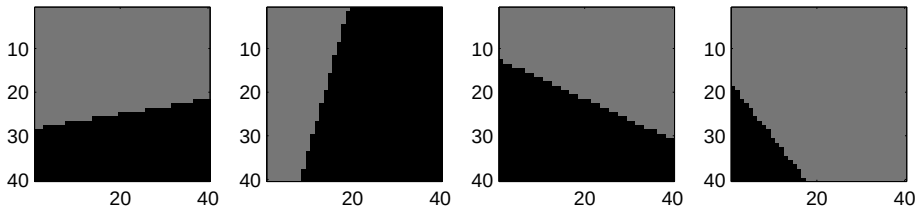


Figure 4.4 Four examples of randomly generated initial partitions for the proposed segmentation method.

In many cases, the randomly generated straight-line partitions are good enough to get the desired initial partition, but in many other cases it cannot work well. In order to obtain a good initial partition as quickly as possible, we integrate the method of *predefined templates* into the generation of the initial partitions. As shown on Figure 4.5, eight predefined templates are selected as candidates in the selection of the desired initial partition.

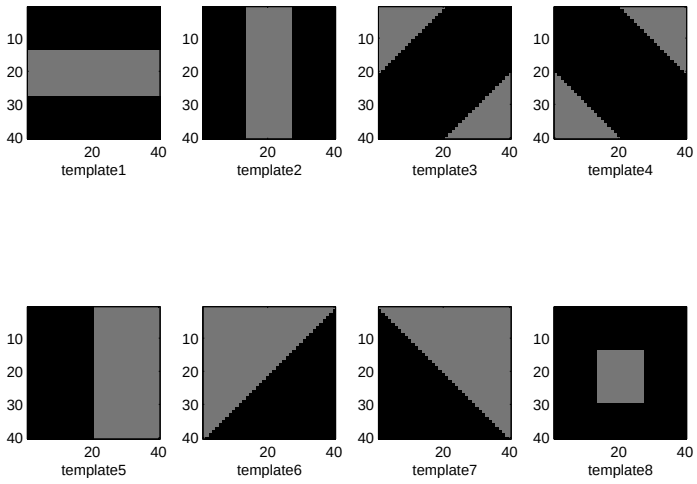


Figure 4.5 The eight predefined initial partition templates.

In this method, an important issue about the initial partition is how to select the "best" one among those candidates. The criteria for evaluating the candidates involve two aspects. One is the local minimum, and the other is the standard deviation of each class within the target image. Two candidates are chosen when each of them has either the lowest local minimum or the lowest standard deviation. Then, the global minima of these two candidates are computed and the one with the lower global minimum is chosen as the final partition. Our experiments show that, by using the two combined techniques, the segmentation results can be improved.

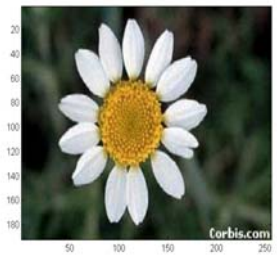
However, the predefined templates still could not capture the natural contour and the distribution of objects in a target image. Thus, in [Zhang04], we propose a fast yet effective image segmentation method called WavSeg to partition the images. In WavSeg, a wavelet analysis in concert with the SPCPE algorithm is used to segment an image into regions. Our initial testing showed two promising phenomena about wavelet analysis:

1. First, the transformed low frequency image (much smaller than the original image) is more suitable for image segmentation (data clustering). In fact, wavelet transformations are known for their decorrelation properties. In addition to preserving the structure of the original model, observations in the wavelet domain are less correlated and thus suitable for object segmentation.
2. Second, the significant points obtained can guide the segmentation process to achieve a local minimum which is near the global minimum in most cases. The significance of this is that by using wavelet transformation, the data clusters obtained could be more likely to approximate the optimal clustering situation.

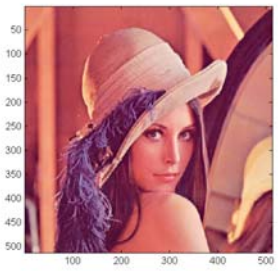
By using wavelet transform and choosing proper wavelets (Daubechies wavelets), the high-frequency components will disappear in larger scale subbands, and therefore the possible regions will be

clearly evident. In our experiments, the images are pre-processed by Daubechies wavelet transform because it is proven to be suitable for image analysis. The decomposition level is 1. Then by grouping the salient points from each channel, an initial coarse partition can be obtained and passed as the input to the SPCPE segmentation algorithm. Actually, even the coarse initial partition generated by wavelet transform is much closer to some global minimum in SPCPE than a random initial partition, which means a better initial partition will lead to better segmentation results. In addition, wavelet transform can produce other useful features such as texture features in addition to extracting the region-of-interest within one entry scanning through the image data. Based on our initial testing results, the wavelet based SPCPE segmentation framework (WavSeg) outperforms the random initial partition based SPCPE algorithm in average. It is worth pointing out that WavSeg is fast. The processing time for a 240*384 image is only about 0.33 sec in average.

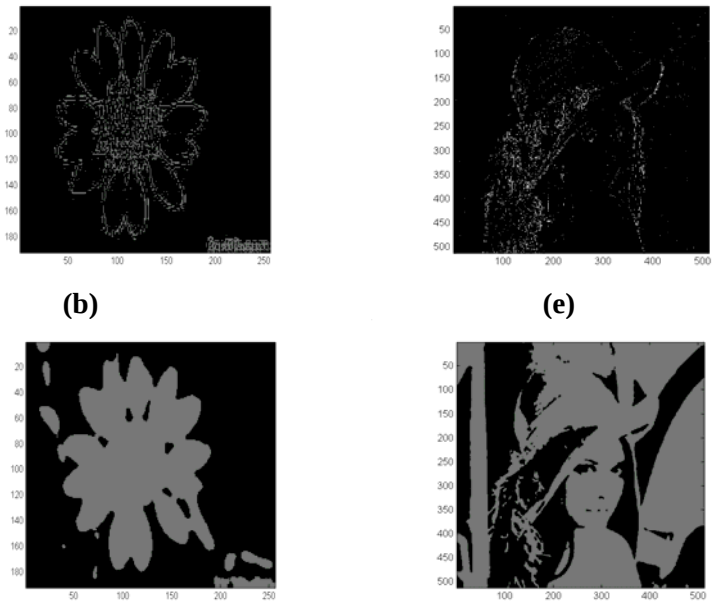
Figure 4.6(a)-(c) are the experimental segmentation results for image *flower.jpg*, while Figure 4.6(d)-(f) are the results for image *lena.jpg*. Figure 4.6(a) and (d) are original images. Figure 4.6(b) and (e) show the salient pixels detected by multi-resolution wavelet analysis, from which we can see that even the preliminary results can capture the most salient characteristics of the target images. Figure 4.6(c) and (f) show the segmentation results based on the detected salient points using WavSeg.



(a)



(d)



(b) **(e)**

(c) **(f)**

Figure 4.6 Preliminary results of WavSeg.

4.2 General-Purpose Content-Based Image Retrieval

In this section, we present a mechanism called the *Markov Model Mediator* (MMM) to facilitate the efficient searching and effective retrieval process for content-based image retrieval (CBIR). The *Markov Model Mediator* (MMM) [Shyu02] serves as the retrieval engine of the CBIR systems and uses affinity-based similarity measures. This mechanism is effective in capturing user's subjective concepts in that it not only takes into consideration the global image features, but also learns high-level concepts from the history of a user's access pattern and access frequencies on the images in the database, which differentiates it from the common methods in content-based image retrieval. The advantage of our proposed mechanism is that it exploits the richness in the structured description of visual contents as well as the relative affinity measurements among the images. Consequently, it provides the capability to bridge the gap between the low-level features and high-level concepts. This mechanism is also efficient in that it integrates principal component analysis (PCA) to significantly reduce the image search space at very low cost before performing exact similarity matching.

In our previous studies, the MMM mechanism was applied to multimedia database management [Shyu00b, Shyu00c] and document management on the World Wide Web (WWW) [Shyu01a, Shyu00a]. The MMM mechanism adopts the Markov model framework and the concept of the mediators. The Markov mechanism is one of the most powerful tools available to scientists and engineers for analyzing complicated systems, whereas a mediator is defined as a program to collect and combine information from one or more sources, and finally yield the resulting information [Wiederhold92]. Markov models have been used in many applications. Some well-known examples are Markov Random Field Models in [Frank86], and Hidden Markov Models (HMMs) [Rabiner86]. Some research works have been done to integrate the Markov model into the content-based image retrieval. Lin *et al*. [Lin97] used a Markov model to combine the spatial and color information. In their approach, each image in the database is represented by a pseudo two-dimensional hidden Markov model (HMM) in order to adequately capture both the spatial and chromatic information about that image. [Wolf97] used the hidden Markov model (HMM) to parse the video data. In [Naphade01], the hidden Markov model was employed to model the time series of the feature vector for the cases of events and objects in their probabilistic framework for semantic level indexing and retrieval.

In this proposed CBIR framework, the MMM mechanism is applied to the dynamic content-based image retrieval process. In this mechanism, a user's perceptions are captured by using a set of training data. Unlike relevance feedback, the training data set we used contains not only user access patterns, but also user access frequencies of the images in the image database. Also, the user queries contained in the training data set may have different query images, which is different from RF where there is only one query image in each retrieval cycle and all the user feedback across iterations is only for refining the retrieval results for one query image. In contrast to this, the proposed mechanism is able to mine the affinity between images across different queries as well as to discover the inner-query image affinities from the training data set. In addition, instead of on-line learning user's preferences, interpretations or retrieval requirements in RF, we calculate the relative affinities of images in the image database off-line,

which serve as the factors of the high-level concepts in our system. Further, the proposed MMM mechanism also provides a way for reducing the search space by using principal component analysis (PCA). PCA is a classic statistical technique that has been widely applied in many areas including data mining, face recognition, web mining [Moore97], and multimedia mining [Su01], etc. In this study, the original image feature space is transformed and projected into the PCA space where the covariance of any pair of principal components is 0, which means the less redundant, less noisy, and more compact representations for the original feature space. Then the pre-filtering process is conducted in the PCA space by using only two principal components at very low cost. Our experimental results show that the proposed mechanism is efficient in terms of retrieval time and storage without sacrificing much accuracy.

We begin with the review of the key components of the MMM mechanism. Then the reduction of search space by using principal component analysis as well as the complete process for image retrieval are introduced, followed by the analysis and experiments in applying the MMM mechanism to content-based image retrieval. In order to test the performance of the proposed framework, a training subsystem has been implemented to collect the user access patterns and access frequencies, which is integrated into our system [Chen03a]. The performance comparisons with the brute-force method are discussed as well. Finally, a brief conclusion is given.

4.2.1 Framework Architecture

The architecture of the proposed framework is shown in Figure 4.7. As can be seen from this figure, our proposed framework is divided into three major components based on their functionalities, namely image feature extraction process, training process, and retrieval process. In our framework, not only the low-level features (e.g., color) and the mid-level features (e.g., object locations), but also the high-level concepts learned from the off-line training process are used in the image retrieval process. Moreover, instead of conducting the exact similarity matching process in the whole database scope, a pre-filtering process using PCA is applied to reduce the search space. Hence, the retrieval process consists of the pre-filtering process and similarity matching process. In the following three subsections, each component and

the relationships among the components are presented in details.

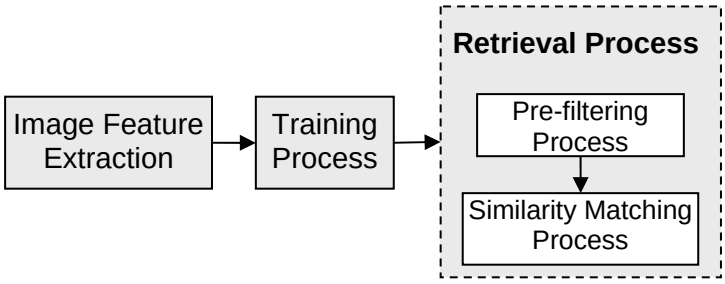


Figure 4.7 Architecture of the proposed framework.

4.2.2 Markov Model Mediator (MMM)

Markov model mediator (for short, MMM) [Shyu02] is a probabilistic-based mechanism that adopts the Markov model framework and the mediator concept. The MMM mechanism is defined as follows.

Definition 1: An MMM is represented by a 6-tuple $\lambda = (\mathcal{S}, \mathcal{F}, \mathcal{A}, \mathcal{B}, \mathcal{P}, \Pi)$, where $\mathcal{S}$ is a set of images called states; $\mathcal{F}$ is a set of distinct features of the images; $\mathcal{A}$ denotes the affinity matrix, where each entry (i, j) actually indicates the affinity between image i and j; $\mathcal{B}$ is the feature matrix; $\mathcal{P}$ is the principal component matrix; and Π is the initial state probability distribution.

Each image database in our CBIR system is modeled by an MMM, where $\mathcal{S}$ consists of all the images in the image database and $\mathcal{F}$ includes all the distinct features for the images in $\mathcal{S}$. $\mathcal{A}$ represents the affinity among all the images in the database based on user's preference, and the relationship of the images are modeled by the sequences of the MMM states connected by transitions. $\mathcal{B}$ consists of the normalized image feature vectors for all the images. $\mathcal{P}$ is the principal component matrix derived from the original feature space. The last tuple Π indicates how likely an image would be accessed without knowing the query image. A training data set consisting of the access patterns and access frequencies of the queries issued to the database is used to train the model parameters $\mathcal{A}$, $\mathcal{B}$, $\mathcal{P}$, and Π for an MMM.

4.2.2.1 Model Parameters

In each MMM, its model parameters $\mathcal{A}$, $\mathcal{B}$, $\mathcal{P}$, and Π need to be formulated and constructed. For this purpose, a set of training data is used.

1. Training Data Set

The training data set is used to generate the training concepts off-line for an MMM mechanism to construct its model parameters $\mathcal{A}$, $\mathcal{B}$, $\mathcal{P}$, and Π matrices. The source of training data set is actually the history of user access patterns and access frequency on the image database. Access pattern, in brief, denotes the co-occurrence relationship among images accessed by user queries, while access frequency denotes how often each query was issued by the users. Definition 2 gives the information available in the training data set.

Definition 2: The training data set consists of the following information:

- The value n_i that indicates the number of images in database d.
- A set of queries $Q = \{q_1, q_2, ..., q_q\}$ that are issued by the users to the database in a period of time.

  Let $use_{k,m}$ denote the access pattern of image m ($1 \le m \le n_i$) with respect to query q_k per time period,

$$use_{k,m} = \begin{cases} 1 & \text{if image } m \text{ is accessed by } q_k \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

- The value of $access_k$ denotes the access frequency of query q_k per time period.

Img1 Img2 Img3

Figure 4.8 Three sample images (Img1 - Img3).

Table 4.1 The query access frequencies ($access_k$) and access pattern ($use_{k,m}$) of the sample images.

q_k	$access_k$	Img1	Img2	Img3	…
q_1	4	1	1	0	…
q_2	2	0	1	1	…
q_3	$access_3$	0	0	0	…
…	…	…	…	…	…

Table 4.1 gives some example queries issued to the image database with their corresponding access frequencies. The usage patterns of the three sample images in Figure 4.8 versus the example queries are also shown in Table 4.1. In this table, the entry $(k, m) = 1$ indicates that the m^{th} image is accessed by query q_k. For example, suppose q_1 is a user-issued query related to retrieving some country house scenes. Then Img1 and Img2 are accessed together in q_1, with their corresponding entries in the usage pattern matrix having value 1. Let q_2 denote some query related to the concept of '*red flowers*', then Img2 and Img3 will probably be accessed together in this query. In fact, the usage pattern matrix captures the users' subjective concepts about the images. Moreover, the access frequency indicates the user's preference among different queries. In this example, since the user may have more interest in country scenes when he/she sees Img2, the access frequency of q_1 could be larger than that of q_2. Consequently, after the system training, the Img1 is probably more likely to be retrieved than Img3, given that the Img2 is selected as the query image.

2. Matrix $\mathcal{A}$: The Affinity Matrix

Based on the information in the training data set, we can capture the relationships among the images in the database based on the high-level concepts. That is, the more frequently two images are accessed together, the more closely they are related. In order to capture the relative affinity measurements among all the images, an assisting matrix $\mathcal{AFF}$ is defined, which is constructed by having the $aff_{m,n}$ be the relative affinity relationship between two images m and n using the following definition.

Definition 3: The relative affinity measurement ($aff_{m,n}$) between two images m and n indicates how frequently these two images are accessed together, where

$$aff_{m,n} = \sum_{k=1}^{q} use_{k,m} \times use_{k,n} \times access_k \tag{10}$$

The affinity matrix $\mathcal{A}$ is then constructed by having $a_{m,n}$ as the element in the $(m, n)^{th}$ entry in $\mathcal{A}$, where

$$a_{m,n} = \frac{aff_{m,n}}{\sum_{n \in d} aff_{m,n}} \tag{11}$$

As shown in this formulation, matrix $\mathcal{A}$ is obtained via normalizing $\mathcal{AFF}$ per row and represents the conditional probability that refers to as the affinity matrix for an MMM.

3. Matrix $\mathcal{B}$: The Feature Matrix

The feature matrix $\mathcal{B}$ consists of normalized image feature vectors for all images. Since our focus is to evaluate the performance of the retrieval mechanism and the reduction of searching space rather than to explore the most appropriate features for image retrieval, in this study each image has a feature vector of only twenty-one elements. Within the twenty-one features, twelve are for color descriptions and nine are for location descriptions. Since the color feature is closely associated with image scenes and it is more robust to changes due to scaling, orientation, perspective and occlusion of images, it is the most widely used visual feature in image retrieval. In our CBIR system, color information is obtained for each image from its HSV color space. The HSV color space is chosen for two reasons. First, it is perceptual, which makes HSV a proven color space particularly amenable to color image analysis [Cheng01]. Secondly, the benchmark results showed that the color histogram in the HSV color space performs the best. The color features considered are 'black' (black), 'white' (w), 'red' (r), 'red-yellow' (ry), 'yellow' (y), 'yellow-green' (yg), 'green' (g), 'green-blue' (gb), 'blue' (b), 'blue-purple' (bp), 'purple' (p) and 'purple-red' (pr) according to the combinations of different ranges of the hue (H), saturation (S), and the intensity values (V). Colors with the number of pixels less than 5% of the total number of pixels are regarded as non-important and the corresponding positions in the feature vector have the value 0. Otherwise, we put the

corresponding percentage of that color component to that position.

For image segmentation and object location information, we use the enhanced SPCPE algorithm to segment each image into background segments and foreground segments. Since foreground segments are more likely corresponding to salient objects or object-of-interest, we use their location information in this study. The minimal bounding rectangle (MBR) concept in R-tree [Hafner95] is adopted so that each object is covered by a rectangle. The centroid point of each object is used for space reasoning so that any object is mapped to a point object. Figure 4.9 shows three sample images and their corresponding segmentation results. The gray segments correspond to the foreground segments (elephant, face, building, etc.), while the white areas correspond to the background segments. In order to get the relative spatial location for each foreground segment, each image is divided into 3×3 equal-sized reference regions. The image can be divided into a coarser or finer set of regions if necessary. As shown in Figure 4.9(d-f), the nine regions are ordered from left to right and top to bottom: L1, L2, L3, L4, L5, L6, L7, L8, and L9. When there is an object in the image whose centroid falls into one of the nine regions, the value 1 is assigned to that region. Objects with their areas less than 2% of the total area are ignored.

Table 4.2 illustrates the normalized feature vectors for the three sample images in Figure 4.9. We consider that the color and location information are of equal importance, such that the sum of normalized color features should be equal to that of location features (0.5 each). By normalizing the feature vector this way, the sum of the probabilities that the features are observed from a given image should be 1. For example, originally Img1 has color component 'ry' (red-yellow) of percentage 46%, which becomes 23% (46%×0.5) after the normalization. As shown in Figure 4.9(d), Img4 also has three objects residing in locations L4, L5, and L6, respectively. After normalization, L4=L5=L6=0.167 and L4+L5+L6=0.5. An alterative way to normalizing the object location features is to assign a weight to each location feature based on the percentage of object area. It should be pointed out that the definition of matrix $\mathcal{B}$, as that of the other variables in the MMM mechanism, is general enough in that any normalized vector-based image feature set can be plugged into matrix $\mathcal{B}$ without any specific prerequisites.

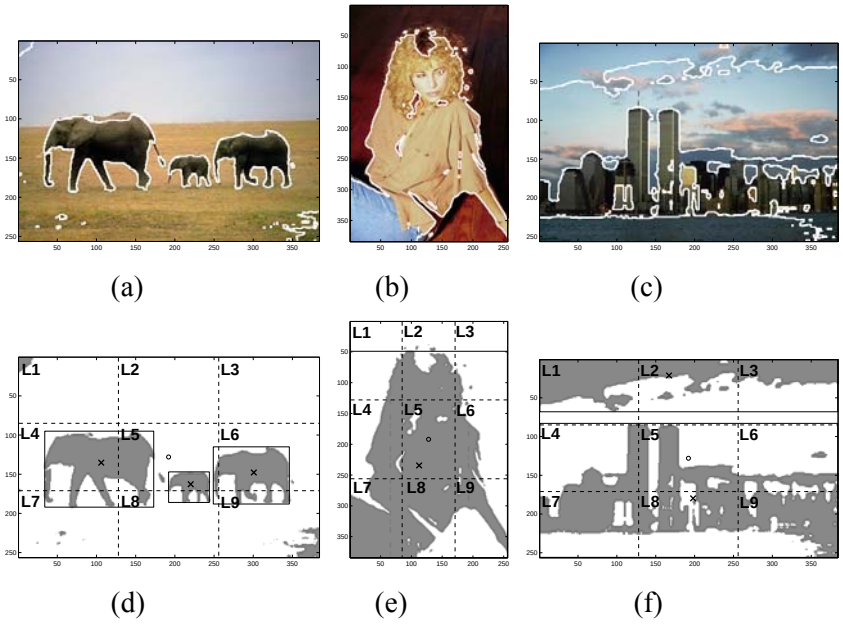


Figure 4.9 Object locations and their corresponding regions. (a-c) Three sample images (Img4 – Img6) with their segmentation boundaries; (d-f) The corresponding segmentation maps and object locations for (a)-(c).

Table 4.2 $\mathcal{B}$ matrix - Normalized image feature vectors of the sample images.

	Color Features												Spatial Features								
	black	**w**	**red**	**ry**	**y**	**yg**	**g**	**gb**	**b**	**bp**	**p**	**pr**	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**	**L8**	**L9**
Img4	0	0.22	0	0.23	0.05	0	0	0	0	0	0	0	0	0	0	0.17	0.17	0.17	0	0	0
Img5	0.25	0.08	0	0.04	0.07	0.06	0	0	0	0	0	0	0	0	0	0.25	0.25	0	0	0	0
Img6	0.03	0.32	0	0	0.04	0	0	0	0.10	0	0	0	0	0.25	0	0	0	0	0	0.25	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

4. Matrix $\mathcal{P}$: Principal Component Matrix

The principal component matrix $\mathcal{P}$ *is* derived from the original feature matrix $\mathcal{B}$ by applying principal component analysis (PCA). Creating this smaller set of new features reduces the dimensionality of original feature space, which is the main purpose of PCA [Jobson92]. The resulting principal component matrix $\mathcal{P}$ will be further used in the proposed retrieval process, serving as the pre-filtering feature space, thus to reduce the number of images that need to perform more expensive exact similarity matching. The

details will be covered in the next section.

- *Principal Component Analysis (PCA)*:

  Principal components are particular linear combinations of the p random variables $X_1, X_2, \ldots, X_p$, with three important properties [Shyu03]: (1) the principal components are uncorrelated, (2) the first principal component has the highest variance, the second principal component has the second highest variance, and so on, and (3) the total variation in all the principal components combined equals the total variation in the original variables $X_1, X_2, \ldots, X_p$. We can visualize the principal components as a new coordinate system obtained by rotating the original system with $X_1, X_2, \ldots, X_p$ as the coordinate axes. The new axes represent the directions with maximum variability and provide a simpler and more parsimonious description of the covariance structure [Johnson98]. It has been shown that the new coordinate system with such properties can be easily obtained from eigen analysis of the covariance matrix or the correlation matrix of $X_1, X_2, \ldots, X_p$.

  Let the original data $\boldsymbol{X}$ be an $n \times p$ data matrix of n observations on each of p variables ($X_1, X_2, \ldots, X_p$), $\boldsymbol{S}$ be a $p \times p$ sample covariance matrix of $X_1, X_2, \ldots, X_p$, and (λ_1, e_1), (λ_2, e_2), …, (λ_p, e_p) be the p eigenvalue-eigenvector pairs of the matrix $\mathbf{S}$. The i^{th} principal component is given by

$$\begin{aligned} y_i &= \mathbf{e}_i'(\mathbf{x} - \bar{\mathbf{x}}) \\ &= e_{i1}(x_1 - \bar{x}_1) + e_{i2}(x_2 - \bar{x}_2) + \ldots + e_{ip}(x_p - \bar{x}_p), \quad i = 1,2,\ldots,p \end{aligned} \tag{12}$$

  where

  $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$

  $\mathbf{e}_i' = (e_{i1}, e_{i2}, \ldots, e_{ip})$ is the i^{th} eigenvector,

  $\mathbf{x}' = (x_1, x_2, \ldots, x_p)$ is any observation vector on the variables $X_1, X_2, \ldots, X_p$,

  $\bar{\mathbf{x}}' = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p)$ is the sample mean vector of variables $X_1, X_2, \ldots, X_p$.

  The i^{th} principal component has sample variance λ_i, and the sample covariance of any pair of

principal components is 0. In addition, let s_{ii} be the sample covariance of the variable X_i, then the total sample variance in all variables $X_1, X_2, \ldots, X_p$ is

$$\sum_{i=1}^{p} s_{ii} = \lambda_1 + \lambda_2 + \ldots + \lambda_p \tag{13}$$

which is the total sample variance in all the principal components. This means that instead of working with the original variables $X_1, X_2, \ldots, X_p$, we can instead work with the principal components and get the same result. There is no loss of information since all of the variation in the original data is accounted for by the principal components.

To reproduce the total system's variability, p principal components are required. However, often much of this variability can be accounted for by a small number k of the principal components. In such case, the first k principal components can replace the initial p variables; and the original data set, which consists of n measurements on p variables, is then reduced to a data set consisting of n measurements on k principal components. There will be almost as much information in the k components as there is in the original p variables, if k is chosen appropriately.

PCA can be carried out on the $p \times p$ sample correlation matrix $\boldsymbol{R}$ of the variables $X_1, X_2, \ldots, X_p$ in the same fashion as with the covariance matrix. If (λ_1, e_1), (λ_2, e_2), …, (λ_p, e_p) are the p eigenvalue-eigenvector pairs of the matrix $\boldsymbol{R}$, then the i^{th} principal component takes the form

$$y_i = \mathbf{e}_i' \mathbf{z} = e_{i1} z_1 + e_{i2} z_2 + \ldots + e_{ip} z_p, \quad i = 1,2,\ldots,p \tag{14}$$

where $\mathbf{z}' = (z_1, z_2, \ldots, z_p)$ is the vector of standardized observations defined as

$$z_k = \frac{(x_k - \bar{x}_k)}{\sqrt{s_{kk}}}, \quad k = 1,2,\ldots,p \tag{15}$$

The principal components from the sample correlation matrix still have the same properties as before. That is, the i^{th} principal component has sample variance λ_i and the sample covariance of any pair of principal components is 0, and the total sample variance in all the principal components is now

$$\lambda_1 + \lambda_2 + \ldots + \lambda_p = p \qquad (16)$$

which is the total sample variance in all standardized variables Z_1, Z_2, …, Z_p.

Note that the principal components from the sample covariance matrix $\boldsymbol{S}$ and the sample correlation matrix $\boldsymbol{R}$ usually will not be the same. Besides, they are not simple functions of the others. If the variables are measured on scales with widely different ranges or if the units of measurement are not commensurate, it is better to perform PCA on the sample correlation matrix. This is because when some variables are in much bigger magnitude than others, they will receive heavy weight in the leading principal components. Standardizing the variables, which leads to the PCA on the sample correlation matrix, makes the magnitude of all variables in the same order and alleviates the problem.

- *Obtaining Matrix $\mathcal{P}$*:

In this study, the feature matrix $\mathcal{B}$ *contains* 12 color features and 9 spatial location features forming 21 variables, X_1, X_2, …, X_{21}, which are normalized and lie in the range of 0.0 to 0.5. Therefore, principal components are obtained from the sample covariance matrix $\boldsymbol{S}$ computed from matrix $\mathcal{B}$. In order to lower the processing time and to reduce the search space, only the first two principal components are used as the search space for image pre-filtering. In retaining only a few principal components, we sacrifice some of the information in the original feature space in an exchange for the efficiency of the search. For this data set, the first two principal components account for 50.12% of the total variation in the data. Although the percentage of total variance explained is not high, the two principal components seem adequate and provide reasonable retrieval results. In general, the number of principal components to be used can be set differently according to different requirements. For instance, we may want to reduce the dimension to at least 10% of the original number, or we may select the first few principal components that can explain at least 75% of the total variance. Su, et al. [Su01] also used PCA to reduce the dimensionality of the feature spaces. They allowed varying dimensions for different types of features based on a goodness measure for a feature type.

5. Matrix Π: The Initial State Probability Distribution

The preference of the initial states for queries can be obtained from the training data set. For any image $m \in d$, the initial state probability is defined as the fraction of the number of occurrences of image m with respect to the total number of occurrences for all the images in the image database d from the training data set.

$$\Pi = \{\pi_m\} = \frac{\sum_{k=1}^{q} use_{k,m}}{\sum_{l=1}^{n_i} \sum_{k=1}^{q} use_{k,m}} \tag{17}$$

4.2.3 The Proposed Process for Image Retrieval

The need for efficient information retrieval from the databases is strong. However, as we mentioned earlier, usually the cost for query processing is expensive and time-consuming. In addition, the results may not be very satisfactory due to the lack of mapping between the high-level concepts and the low-level features. The proposed MMM mechanism offers a way to perform the searching process more efficiently and correctly. The proposed approach for content-based image retrieval is conducted in two steps:

- Apply the pre-filtering process on the principal component subspace to select a small set of candidate images at low cost, thus to reduce the search space.
- Perform the actual retrieval process that computes the similarity functions on the original feature space only for those candidate images.

4.2.3.1 Pre-Filtering to Reduce the Search Space

Once the number of principal components has been decided, which is two in this study, the principal component scores for each image in the database are computed and stored together with each image. Before performing the actual image retrieval process, the first step is to get all the images that have principal component scores close to those of the query image in the principal component subspace, while filtering out those images that are unlikely to be the similar images to the query image. Ideally, we would

like to search for some k nearest neighbors of the new image, where k is a specified integer. This requires the computation of the distance in the form of

$$D_j = \frac{(y_{j1} - y_1)^2}{2\lambda_1} + \frac{(y_{j2} - y_2)^2}{2\lambda_2} \tag{18}$$

where

D_j is the distance between image j in the database and the query image,

y_{j1} and y_{j2} are the scores of principal components 1 and 2 of image j,

y_1 and y_2 are the scores of principal components 1 and 2 of the query image.

The computation is time consuming for a big database (10,000 images in this study). In fact, the need to calculate the distance, based on Gaussian assumptions, between the query images and all of the images in the database in real time is one of the drawbacks of the method proposed by Su, et al. [Su01]. For the sake of real time processing, instead of expensive distance computation, we propose to only sort the images in the database according to the values of principal component scores 1 and 2. Given a desired candidate pool size c, for example one hundred, the top one hundred images that have principal component score 1 closest to that of the query image will be identified. Repeat the same process for principal component score 2. The intersection of the two sets of images is the desired candidate image pool. If the size of the intersection set is less than c, then increase the scan scope in components 1 and 2 until the candidate pool is filled. This approach is reasonable because the principal components are uncorrelated. The nearest neighbors found from individual principal component distributions will not be much different from the joint distribution.

It is worth mentioning that PCA is particularly useful when the image features are highly correlated so that we can easily point out the strong relationship in the variables. Based on the experiments conducted on matrix $\mathcal{B}$, from the last two principal components that have almost no variation, we see that the variables are clearly separated into two groups, X_1 to X_{12} which give one type of information (corresponding to 12 color features) and X_{13} to X_{21} (corresponding to 9 spatial features) another type. A

representative of each group is present in the first two principal components with the highest weight, X_{17} (L5 in $\mathcal{B}$ matrix) the principal component score 1 and X_2 ('white' in $\mathcal{B}$ matrix) in the principal component score 2. That is, other variables in each group, more or less, provide duplicate information to its representative.

4.2.3.2 Retrieval Process

Based on the concepts learned from the training data set, we capture the most matched images through a dynamic programming algorithm that conducts a stochastic process in calculating the current edge weights and the cumulative edge weights.

Assume N is the total number of images in the database, and the features of the query image S_q is denoted as $\{o_1, o_2, \ldots, o_T\}$, where T is the total number of non-zero features of the query image S_q. In our case, $1 \le T \le 21$ since there are 21 features in total.

Definition 4: $W_t(i)$ is defined as the edge weight from the edge S_i to S_q at the evaluation of the t^{th} feature (o_t) in the query, where $1 \le i \le N$ and $1 \le t \le T$.

Definition 5: $D_t(i)$ is defined as the cumulative edge weight from the edge S_i to S_q at the evaluation of the t^{th} feature (o_t) in the query, where $1 \le i \le N$ and $1 \le t \le T$.

Based on definitions 4 and 5, the dynamic programming algorithm is given as follows.

At $t = 1$,

$$W_1(i) = \pi_{S_i}(1 - | b_{S_i}(o_1) - b_{S_q}(o_1) | / b_{S_q}(o_1)) \tag{19}$$

$$D_1(i) = W_1(i) \tag{20}$$

The values of $W_{t+1}(i)$ and $D_{t+1}(i)$, where $1 \le t \le T$, are calculated by using the values of $W_t(i)$ and $D_t(i)$.

$$W_{t+1}(i) = D_t(i) a_{S_q S_i}(1 - | b_{S_i}(o_1) - b_{S_q}(o_1) | / b_{S_q}(o_1)) \tag{21}$$

$$D_{t+1}(i) = (\max_i D_t(i)) + W_{t+1}(i) \tag{22}$$

Then the similarity function is defined as:

$$SIMI\ (i) = \sum_{t=1}^{T} W_t(i) \tag{23}$$

As we mentioned before, $\mathcal{A} = \{a_{S_i S_j}\}$ denotes the conditional probability, $\mathcal{B} = \{b_{S_i}(o_k)\}$ is the probability of the feature observed from an image, and $\Pi = \{\pi_{S_i}\}$ is the initial state probability distribution. The complete image retrieval process including pre-filtering step is shown in Table 4.3.

In Step 5, since we already obtained matrices $W_1(i)$ and $D_1(i)$ from Step 2, and the second feature O_2 is known, the content of $W_2(i)$ and $D_2(i)$ can be determined. Following the same manner, all the pairs of W and D vectors can be obtained. The value of $SIMI(i)$ (obtained in Step 7) is the sum of the edge weights $W_1(i)$, $W_2(i)$, … , $W_T(i)$. In other words, it indicates the matching percentage of the i^{th} image in the image database to the query image with respect to the features $\{o_1, o_2, \ldots, o_T\}$.

Table 4.3 Image retrieval steps using our proposed model.

1. Issue the query image q, and obtain its first two principal component scores.
2. Apply pre-filtering step based on the two principal component scores to produce a candidate image pool c.
3. Obtain the query image q's feature vector $\{o_1, o_2, \ldots, o_T\}$, where T is the total number of non-zero features of the query image q.
4. Upon the first feature o_1, calculate $W_1(i)$ and $D_1(i)$ for each image in candidate pool c according to Equations (11) and (12).
5. Move on to calculate $W_2(i)$ and $D_2(i)$ according to Equations (13) and (14).
6. Continue to calculate the next values for the W and D vectors until all the features in the query have been taken care of.
7. Calculate $SIMI(i)$ for each image in c.
8. Rank the images by sorting their corresponding values in $SIMI(i)$. The larger the value is, the stronger the relationship exists between the candidate image and the query image.

In contrast to the common methods that either have difficulties in capturing the high-level concepts or try to learn the concepts in real-time, our method provides the capability of training the data set off-line. On the other hand, the proposed method is efficient in terms of storage and retrieval. Given a query image q issued by a user, the pre-filtering step will filter out most of the images by using only two component scores and only the data in the row q of matrix $\mathcal{A}$ are used, which is a vector of small size. Second, since most of the values in the entries of the $\mathcal{B}$ matrix are zeros, we can use a sparse matrix to store it. In addition, normally the non-zero features contained in one query image are no more than six, which enables us to load less than half of the whole $\mathcal{B}$ matrix. Thus we can retrieve the results more efficiently without sacrificing much accuracy.

4.2.4 Experiments

4.2.4.1 Experimental Image Database System

The image set in our database contains 10,000 images of 72 semantic categories with various dimensions. Both the color information and object location information of the images are considered and the query-by-example strategy is used to issue queries in our experiments. The following experiments are the results based on 149 training queries, which cover 80% of the images in the database.

4.2.4.2 Implementation of Training System

The training subsystem for this framework is implemented and integrated into our system [Chen03a], a prototype multimedia management system developed by our research group aiming to support a comprehensive set of functionalities and components for multimedia database management systems. Figure 4.10 shows the interface of the training subsystem.

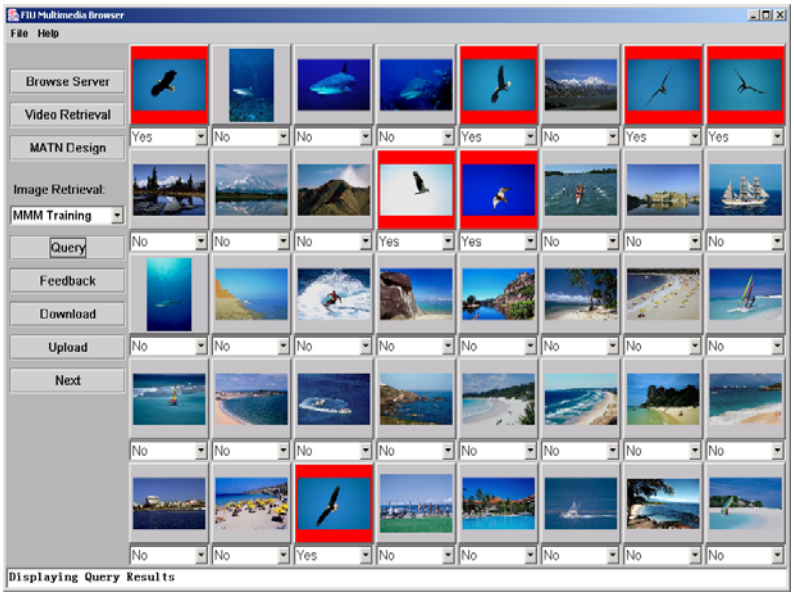


Figure 4.10 The interface of the training subsystem.

The detailed training process is described as follows: Firstly, the user selects one query image. After clicking the "*Query*" button, a query message is sent to the server through UDP. The query results will be sent back after the server fulfills the query process. It is worth mentioning that for training purposes, any available image retrieval methods can be implemented on the server side. Upon receiving the results, the user selects the images that he/she thinks are related to the query image by right-clicking on the image canvases, and clicks the "*Feedback*" button to send the feedbacks back to the server. When the server receives and identifies this feedback message, it updates the user access patterns and access frequencies accordingly. Then the user can continue the training process or exit.

4.2.4.3 Experiments

To test the performance and efficiency of our proposed mechanism, 80 randomly chosen images belonging to five distinct categories are used as the query images. Table 4.4 lists the descriptions for each category as well as the number of images it includes.

Table 4.4 The category distribution of the query image set

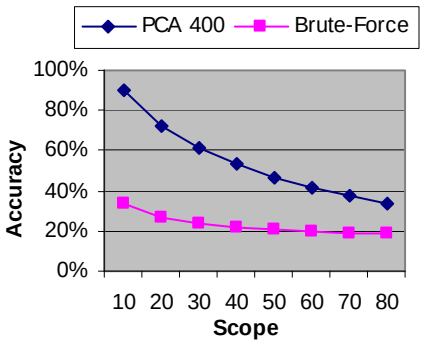
Category	Explanation	Number of Images
Landscape	Land, Sky, Mountain	16
Flower	Flower	16
Animal	Elephant, Panther, Tiger	16
Vehicle	Car, Bus, Plane, Ship	16
Human	Human	16

In our first experiment, the overall performance of our proposed mechanism is compared with that of the brute-force method which does not integrate the information of user access pattern and access frequency and performs the full sequential search through the image database based on the feature matrix $\mathcal{B}$. In this experiment, we select the size of candidate image pool c as 400 for the sake of speed and memory requirements, which means there are only 400 images that need to do exact similarity matching after pre-filtering step by using the principal component analysis. Thus, the candidate image pool constitutes only 4% of the total number of images in the database. In fact, the larger the candidate pool is used, the more accurate results can be achieved as we will show in our second experiment.

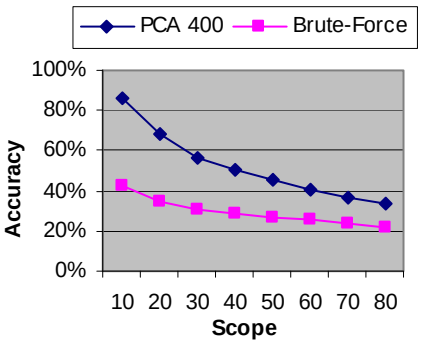
Accuracy-scope curve is used here to evaluate the image retrieval performance. Scope specifies the number of images returned to the user; while accuracy is defined as the percentage of the retrieved images that are actually relevant to the query image. Figure 4.11(a)-(f) shows the accuracy comparison of our proposed mechanism ('PCA 400') and the brute-force method ('Brute-Force'). The results in Figure 4.11(a) are calculated by using the averages of all the 80 query images, while Figure 4.11(b)-(f) shows the results for each category. As can be seen from this figure, our approach outperforms the brute-force method in all cases, which implies that the user access pattern and access frequency have a great potential to capture the subjective aspects of the user concepts. In addition, it is worth mentioning that our mechanism dramatically decreases the search space from 10,000 to 400 images by using only the first two principal components that only account for 50.12% of the total variation of the feature data, and the cost for pre-filtering is trivial compared to the exact similarity matching process.

In our second experiment, the accuracy after PCA search space reduction is studied according to the

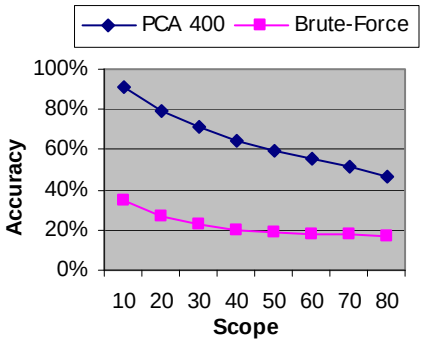
candidate pool size. In this experiment, the candidate pool size varies from 20 to 800. The accuracy used here is the average accuracy within the top 20 images. It makes sense since the users typically expect the first screen of retrieved images to contain a high proportion of the relevant images [Tong01], and the retrieval of every relevant image is less concerned in most cases.



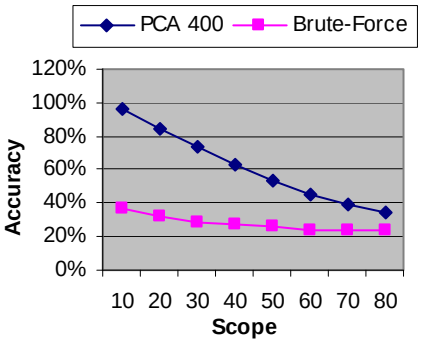
(a) Summary



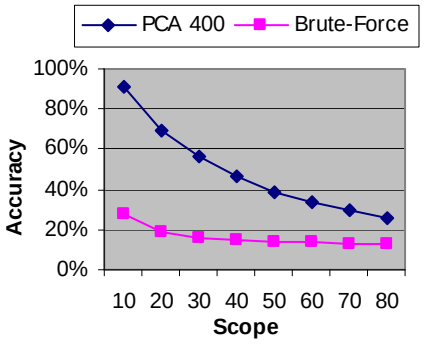
(b) Landscape



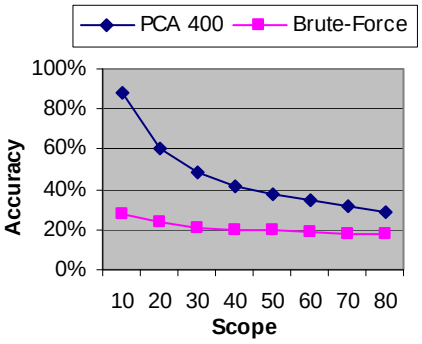
(c) Flower



(d) Animal

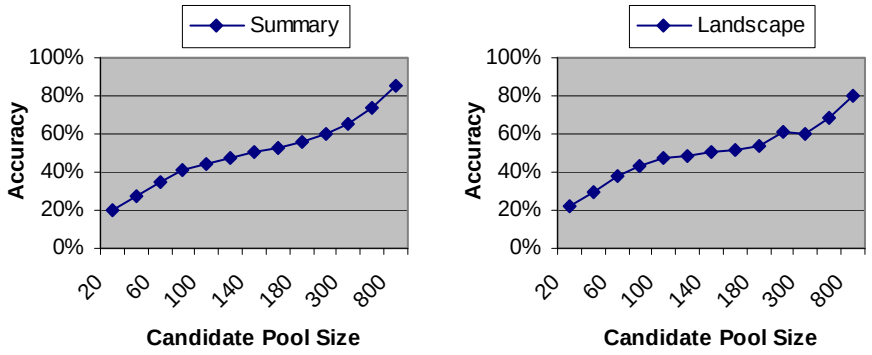


(e) Vehicle



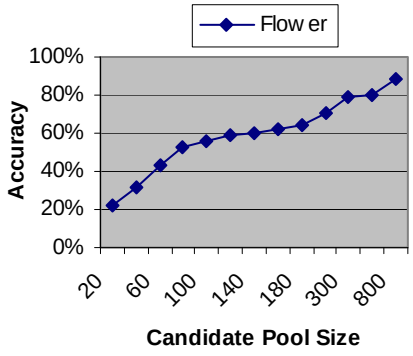
(f) Human

Figure 4.11 Accuracy Comparison of two methods. ‘PCA 400’ denotes the proposed method using the candidate pool of 400 images, and ‘Brute-Force’ denotes the full search method without using the access pattern and access frequency information.

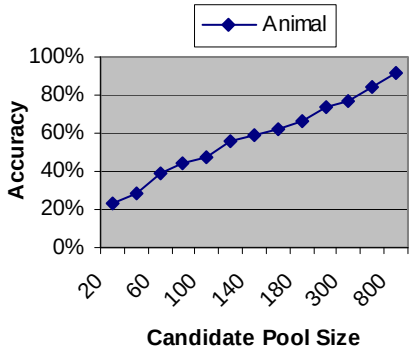


(a) Summary

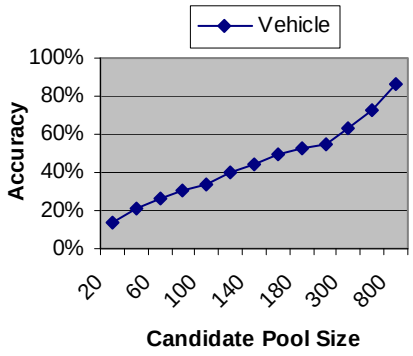
(b) Landscape



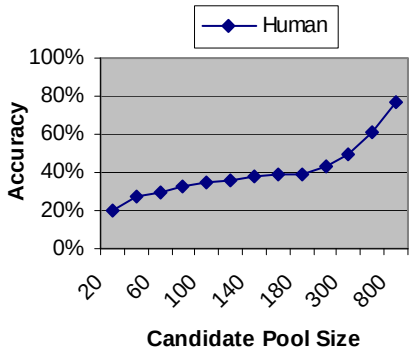
(c) Flower



(d) Animal



(e) Vehicle



(f) Human

Figure 4.12 The retrieval accuracy in the top 20 images versus the PCA candidate pool sizes.

The results shown in Figure 4.12(a)-(f) reveal the following observations: 1) The accuracy increases with the increase of candidate pool size; 2) The accuracy can be maintained above 80% when the retained PCA candidate pool has 800 or more candidate images which only accounts for 8% of the 10,000 images in our database, and the advantage brought by this reduction is that the retrieval speed is significantly increased and that the memory use is greatly reduced; and 3) If less accuracy is allowed (e.g., requiring only 70% accuracy), then the candidate pool of 300~400 images should be enough, so that the system efficiency can be further boosted in terms of time and space. Another observation is that the performance for 'Human' category is worse than the other categories. The reason is that the randomly selected query images for 'Human' category include not only the portraits which are relatively simple in perception, but also some images containing human objects with very complex scenes, which increases the difficulty for retrieval.

To demonstrate the effectiveness of our mechanism, two query-by-image example queries where one is a 'Landscape' image and the other belongs to the 'Car' category are shown in Figure 4.13 and Figure 4.14, respectively.

1. Query I:

As shown in Figure 4.13, the query image is located in the top-left of the retrieval interface, marked by a red box. The returned top twelve images are ranked and displayed in the decreasing order of their similarity scores from left to right and top to bottom.

In this case, the query image is Img3140 that belongs to the 'Landscape' category and contains complicated scenes. However, as can be seen from this figure, besides the low-level features, the perceptions contained in these returned images are quite similar. In addition, the ranking is reasonably good.

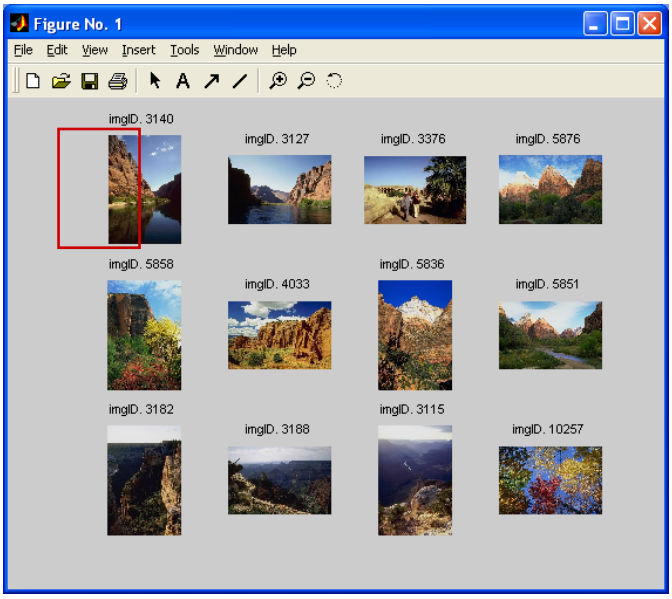


Figure 4.13 Results for Query I.

2. Query II:

In this query, query image Img5169 mainly contains ‘car’ and ‘lawn.’ Figure 4.14 exhibits the top twelve images retrieved from the image database. As can be easily seen from this figure, except the last one, almost all of them have the similar perceptions with the query image.

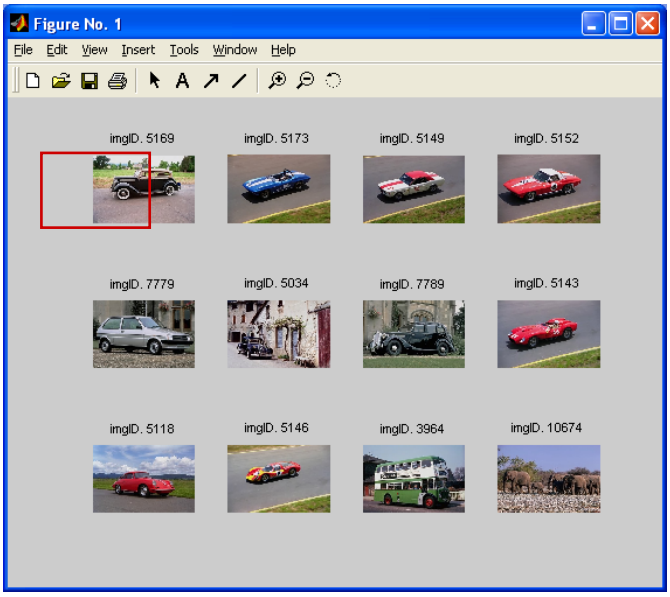


Figure 4.14 Results for Query II.

4.2.5 Conclusions

The current problems in the CBIR systems come from the concerns of both effectiveness and efficiency. Though Relevance Feedback has been proposed to overcome the semantic gap between the low-level features and high-level concepts, it requires the concepts to be trained in real-time and the users are required to take heavy responsibilities during the retrieval process for only one single query image. In addition, search efficiency is another critical issue besides the semantic gap for large-scale image databases. In response to these issues, the Markov Model Mediator (MMM) mechanism is applied to the image databases for content-based image retrieval. First, the pre-filtering step is conducted before the exact similarity matching and retrieval process. A dynamic process based on the MMM mechanism is then applied to deal with the candidate images output by the pre-filtering step and to find the similar images with respect to the query image. Our proposed mechanism provides the capability to learn the concepts and affinities among the images off-line based on the training data set, such as access patterns and access frequencies, without any user interactions. This off-line learning is in fact an affinity-mining process which can reveal both the inner-query and inter-query image affinities. Then it performs a similarity comparison between the query image and the candidate image based on not only the low-level features, such as color and location features, but also the concepts obtained by training in the previous steps among all the images in the image database. Several experiments were conducted and the result analysis was reported. The fact that the proposed content-based image retrieval system utilizes the MMM mechanism and supports both spatial and color information offers more flexible and accurate results for the user queries. The experimental results exemplify this point, and the overall retrieval performance of the presented system is promising. It is also shown by experiments that the pre-filtering step enabled by the principal component analysis (PCA) can dramatically reduce the search space to 4%~8% of its original size without sacrificing much accuracy, which makes our proposed mechanism efficient and scalable.

4.3 Multi-Object Based Image Retrieval

In addition to general-purpose image retrieval, we also propose a method to effectively discover users' concept patterns when multiple objects of interests (e.g., foreground and background objects) are involved in content-based image retrieval. The proposed method incorporates Multiple Instance Learning into the user relevance feedback in a seamless way to discover where a user's objects/regions of most interest and how to map the local features of that region(s) to a user's high-level concepts. A three-layer neural network is used to model the underlying mapping progressively through the feedback and learning procedure.

4.3.1 Overview

The subjectivity of human perception of visual content plays an important role in content-based image retrieval (CBIR) systems. A fixed image similarity measure cannot meet the need to adapt to different focuses of attention of different users. As mentioned before, relevance feedback and region-based image retrieval are two techniques used to deal with this issue. The region-based retrieval systems segment an image into several homogenous regions, and then the features for each region can be extracted and compared. Relevance feedback (RF) [Rui98] is an interactive process in which the user judges the quality of the retrieval results returned by the system. The user feedback information is then used to refine the original query. Recently, the research in integrating these two major techniques has gained much attention. The representative is the RF-based Multiple Instance Learning (MIL) mechanism proposed in [Zhang02, Huang03] which integrates RF and single object-based retrieval seamlessly.

In this study, we propose a method that can dynamically discover the visual concept of a specific user from the user's relevance feedback when multiple objects of interest are involved in that user's focus of attention. Especially, it can simultaneously find the multiple objects/regions of the user's interests and learn the mapping between the local image features of those objects and the user's concept. This method has the following distinctive features.

First, Multiple Instance Learning (MIL) is integrated into the query refining process to learn the region of interest from user relevance feedback and to tell the system to shift its focus of attention to that region. Our method extends the existing MIL system [Huang03] in a way that the user can provide feedback information to multiple objects instead of one, and the multiple objects of interest can be discovered simultaneously by feedback information fusion. In the scenario of MIL, each image is viewed as a bag of image regions (instances). The labels (relevant/irrelevant) of the individual regions in the training data are not available; instead the labeled unit is a set of instances (images). In other words, a training example is a labeled image. The goal of learning is to obtain a hypothesis from the training examples that generates labels to the unseen images. The original MIL technique has the assumption that the user's concept can be represented as a single "best" object. However, the discovery of multiple objects of interest is also very common and it is more natural to have one visual concept corresponding to more than one significant object. For example, one user may look for those images with red cars parked on the grassland; while another user may be more interested in red cars running on the highway. Second, the neural network technology is applied to map the low-level image features to the user's concepts. The parameters in the neural network are dynamically updated according to the user feedback to best represent the user's concepts. Third, the WavSeg method (see Section 4.1.3) is used in this study to automate the process of segmenting the image into multiple regions. The color and texture features are collected for each image region to form a feature vector for each instance (region) in a bag (image).

4.3.2 Multiple Instance Learning (MIL)

In Multiple Instance Learning, the label of each bag is either *Positive* or *Negative*. A bag is labeled *Positive* if it has at least one positive instance, and *Negative* if and only if all its instances are negative. The goal of learning is to generate a mapping function from the training data set to predict the labels of the unseen bags.

Definition 6: Given the instance space μ, the bag space v, the label space K = [0,1], a set of training examples $T = \langle B, L \rangle$ where $B = \{ B_i \mid B_i \in v,\ i = 1...n \}$ is a set of n bags and $L = \{ L_i \mid L_i \in K, i = 1...n \}$ is

the set of their associated labels with L_i being the label of B_i, the problem of Multiple Instance Learning is to generate a hypothesis $h_B : \nu \to K = [0,1]$ which can predict the labels of unknown bags accurately.

Actually, each instance in a particular bag has a label in the closed interval [0,1], which represents the degree of that instance being Positive (Label 0 means *Negative.*), although it is unknown. Given the labels of all the instances in a bag, the label of the bag (i.e., the degree of the bag being Positive) can be represented by the maximum of the labels of all its instances. In other words, $L_i = \underset{j}{MAX}\{l_{ij}\}$ where the label L_i is the label of bag B_i and l_{ij} is the label of the j^{th} instance I_{ij} in B_i. Let $h_I : \mu \to K = [0,1]$ denote the hypothesis that predicts the label of an instance. The relationship between hypotheses h_B and h_I can be depicted in Equation (24):

$$L_i = h_B(B_i) = \underset{j}{MAX}\{l_{ij}\} = \underset{j}{MAX}\{h_I(I_{ij})\} \tag{24}$$

In our proposed Multiple Instance Learning framework, the Minimum Square Error (MSE) criterion is adopted. That is, we try to learn the hypotheses $\hat{h}_B$ and $\hat{h}_I$ to minimize the function shown in Equation (25).

$$E = \sum_{i=1}^{n}\left(L_i - \hat{h}_B(B_i)\right)^2 = \sum_{i=1}^{n}\left(L_i - \underset{j}{MAX}\{\hat{h}_I(I_{ij})\}\right)^2 \tag{25}$$

In addition, in our algorithm, the Multilayer Feed-Forward Neural Network is used as the hypothesis $\hat{h}_I$ and the Back-propagation (BP) learning method is used to train the neural network to minimize E.

4.3.3 Image Segmentation and Feature Extraction

As mentioned earlier, the WavSeg algorithm is used to segment each image into multiple regions/segments. Both the local color and local texture features are extracted for each image region.

Color Features:

HSV color space and its variants are proven to be particularly amenable to color image analysis. Therefore, we quantize the color space using color categorization based on H S V value ranges. Twelve representative colors are identified: black, white, red, red-yellow, yellow, yellow-green, green, green-blue, blue, blue-purple, purple, and purple-red. The hue is divided into five main color slices and five transition color slices. Each transition color slice, such as yellow-green, is considered in both adjacent main color slices. We disregard the difference between the bright chromatic colors and the chromatic colors. Each transition color slice is treated as a separate category instead of being combined into both adjacent main color slices. A new category "gray" is added so that there are totally thirteen color features for each region in our method.

Texture Features:

One-level wavelet transformation using Daubechies wavelets is used to generate four subbands of the original image. They include the horizontal detail sub-image, the vertical detail sub-image, and the diagonal detail sub-image. For the wavelet coefficients in each of the above three subbands, the mean and variance values are collected, respectively. Therefore, totally six texture features are generated for each image region in our method.

4.3.4 Learning and Retrieval Process

In the content-based image retrieval process, the user submits a query example (image) and the CBIR system retrieves the images that are most similar to the query image from the image database according to some similarity measures. However, in many cases, when a user submits a query image, what the user is really interested in is just one or two region(s) of the image. For example, "Find all the images that contain a brown horse object and a white horse object." In this study, we target the retrieval of multiple objects of interest by integrating multiple instance learning into the user relevance feedback. We also realized that the number of user interested objects is usually about 2~3 (If more than that, the whole image query is more appropriate.), and therefore, the two-object retrieval scenario is used in this study to

illustrate the basic idea. Our proposed method first segments the image into multiple regions by using WavSeg and then uses the user's relevance feedback and Multiple Instance Learning to automatically capture the user-interested regions during the query refining process. Another advantage of our method is that the underlying mapping between the local visual feature vectors of the regions of interest and the user's high-level concept can be progressively discovered through the feedback and learning procedure.

Taking the two-object retrieval scenario as an example, there exist two mapping functions for objects 1 and 2 between a region of an image and the user's concept. Our system uses the Multilayer Feed-Forward Neural Network to map a low-level feature vector to a real value in [0,1], which represents how much the region meets the user's concept. The extent to which an image belongs to the user's concept is the maximum one of all its regions. Therefore, an image can be viewed as a bag and its regions are the instances of the bag in Multiple Instance Learning. During the image retrieval procedure, the users are asked to provide relevance feedback (relevant/irrelevant) to the whole image for each object of interest (objects 1 and 2). For each such object, there is a set of positive relevant images as well as a set of negative images. Since the labels are assigned to the individual images, not on the individual regions, the image retrieval task can be viewed as an MIL task. In the two-object retrieval case, two neural networks are learned, which can identify the user's two most interested regions and capture the user's high-level concepts from the low-level features.

At the beginning of the retrieval, the learning method is not available since there are no training examples. Hence, we use a simple distance-based metric to measure the similarity of two images. Assume Image Q is the query image and consists of nq regions and Image I consists of ni regions, where $Q=\{Q_i\}(i=1, \ldots, nq)$ and $I=\{I_j\}(j=1, \ldots, ni)$. The difference between Images Q and I is defined as:

$$Dist(Q, I) = \sum_{1 \le i \le nq} \underset{1 \le j \le ni}{Min} \left\{ \left\| Q_i - I_j \right\| \right\} \tag{26}$$

Upon the first round of retrieving those "most similar" images according to Equation (26), users can give their feedbacks by labeling each retrieved image, and a set of training examples can be

constructed for each object of interest based on the user feedbacks. Then the MIL is applied to train the neural networks for the two objects of interest. Each image in the database will be passed as an input to the two trained neural networks respectively, and the outputs for each image are two similarity scores, one for each object of interest. The retrieval system will rank the images according to the similarity function which is a combination of the two scores, and present the most similar images to the user. Currently, we use the sum of the two scores as the similarity function. However, other methods of combination and fusion can also be tested. The feedback and learning are executed iteratively, and the capturing of user's high-level concept is refined until the user is satisfied.

4.3.5 Experimental Results

We select 2,100 images of various categories from the Corel image library to build our testing image database. In our experiments, a three-layer Feed-Forward Neural Network is used. Specifically, the input layer has nineteen neurons with each corresponding to one of the nineteen image features. The output layer has only one neuron and its output indicates the extent to which an image region meets the user's concept. The number of neurons at the hidden layer is experimentally set to nineteen.

Figure 4.15 shows the two-object retrieval interface of this system and the initial retrieval results using the similarity function defined in Equation (26). The query image is at the top-left corner. The query results are listed from top left to bottom right in decreasing order of their similarities to the query image. The user can also use the two pull-down menus under each image to input his/her feedback on that image and carry out the next round of retrieval. The first pull-down menu contains the feedback for object 1, while the second pull-down menu collects the feedback for object 2. The user's concept is then learned in a progressively way through the user feedback, and the refined query will return a new collection of the matching images to the user. It needs to be noted that the user's two most interested regions can be discovered within four iterations in most cases.

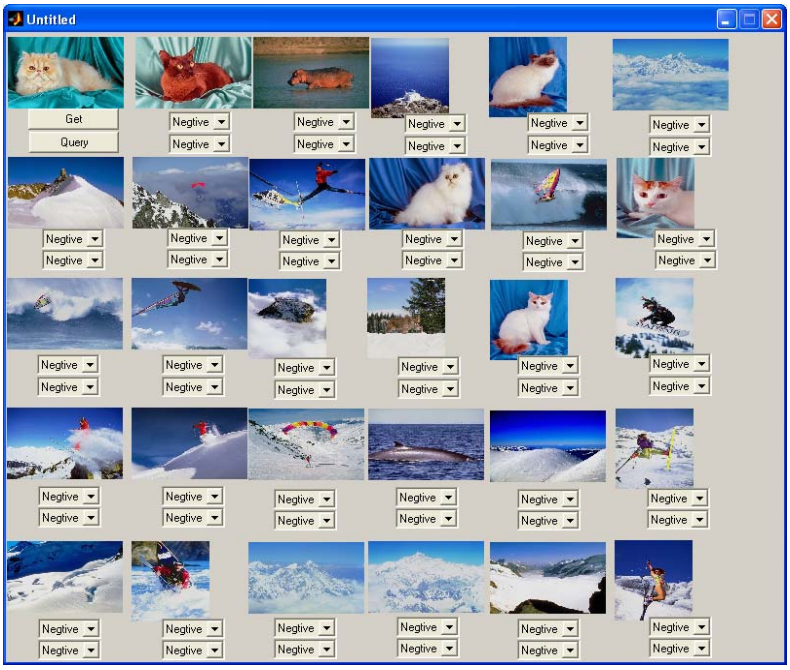


Figure 4.15 The CBIR retrieval interface and the initial query results.

As can be seen from Figure 4.15, assume the two objects of interest are "white-yellow cat" and "blue-tone silken background." In the initial retrieved images, many of them contain snow scenes with blue skies without any of the two required objects in them. Figure 4.16 shows the retrieved images after four iterations of user feedback, and 24 out of 30 retrieved images contain both of the two query objects. We also conducted a number of other experiments on different image categories such as horses, mountain scenes, snow scenes, leopards, apes, owls, and race cars. The average accuracy within the top 30 retrieved images is around 70%, which demonstrates the effectiveness of our multiple object retrieval method using MIL and RF.

Figure 4.16 The query results after 4 iterations of user feedback.

4.3.6 Conclusions

In this study, we propose a method to discover user's high-level concepts from the low-level image features using Relevance Feedback and Multiple Instance Learning. Compared with other MIL-based CBIR systems, our system has the following advantages: 1) Instead of manually dividing each image into many overlapping regions, we proposed the WavSeg image segmentation method to partition the images in a more natural way; 2) Instead of discovering one single object of interest, the proposed method can deal with the multiple object retrieval scenarios in which the users may have different focuses of attention. By putting negative feedback on those images that do not meet a user's specific concepts despite of the similar image features, the system can better distinguish the user's real needs from the "noisy" or unrelated information via MIL; and 3) In our system, the neural network is used to map the low-level image features to the user's concepts. The parameters of the neural network are adaptively updated during the feedback process.

Chapter 5. Content-Based Video Indexing for Video Databases

The methods used for video indexing and retrieval include metadata-based method (video title, producer, video type, etc.), text-based method (transcripts, subtitles, etc.), audio-based method, and content-based method. In the content-based method, instead of treating the video as a huge amount of independent frames (images) and applying the image indexing and retrieval methods on all the frames, a better way is to divide video clips into video shots and scenes. Video shot detection and scene detection are two major tasks for content-based video indexing. Many advanced video applications such as video on demand (VOD) and digital library also require the shot/scene change detection to organize the video content. In this chapter, we first present an effective shot change detection method using the unsupervised object segmentation algorithm and the technique of object tracking based on the *segmentation mask maps*. Compared with other state-of-the-art techniques such as the method of DC images and some global feature based techniques (Histograms, etc.), our results have shown that not only can the proposed method perform more accurate shot change detection, but also obtain object level information of the video frames, which is very useful for video content indexing and analysis in multimedia databases. Secondly, a scene change detection method is presented based on the proposed shot detection method and audio clues. Then using the soccer video as an example, a soccer video event detection framework using data mining techniques is presented in detail, which is also based on the proposed shot detection method. This event detection framework is the focus of this chapter because it provides the high-level semantic indexing (interesting/significant events) in addition to the basic video indexing such as shots/scenes and key frames.

5.1 Video Shot Detection

Video shot change detection is a fundamental operation used in many multimedia applications involving content-based access to videos such as digital libraries and video on demand, and it must be performed prior to all other processes [Shahraray95, Zhang94]. Video data can be divided into different shots. A *shot* is a video sequence that consists of continuous video frames for one camera action. In addition to the

natural shot cuts, there are also shot boundaries caused by special edit effects (dissolve, zoom in/out, etc.) and camera motions (panning, tilting, etc.) as well as object motions. Shot change detection is an operation that divides video data into physical shots. There are a large number of methods for video shot change detection in the literature. The matching process between two consecutive frames is the essential part of it. Many of them use low-level global features such as the luminance pixel-wise difference [Zhang93], luminance or color histogram difference [Swanberg93, Zhang93] and edge difference [Yeo97] to compare two consecutive frames. Other recent work related to low level features includes the orientation histogram [Ngo00]. The edge image based method proposed in [Zabih95] works especially well in many cases when it is difficult to detect shots with intensity histograms. However, since luminance or color is sensitive to small changes, these low-level features cannot give a satisfactory answer to the problem of shot change detection. For example, the method proposed in [Yeo95] used the luminance histogram difference of DC images, which is very sensitive to luminance changes. Recently, there has been much research work done on the compressed video data domain such as fast shot change detection [Lee00] and directional information retrieving [Hwang98] by using the discrete cosine transform (DCT) coefficients in MPEG video data. In addition to all the above techniques, some research work has been done on the dynamic and adaptive threshold determination problem [Alattar97, Truong00, Gunsel98], which can be used to enhance the accuracy and robustness of the existing techniques in shot cut detection.

While parsing the video data for analysis is time consuming, it is expected to produce as much information as possible in one pass for efficiency purposes. For example, object extraction and key frame selection can be performed together with video shot segmentation. However, to our best knowledge, there is little work in the literature trying to automate the process of obtaining object level information in video scenes while doing video shot segmentation. Such object level information is very useful in identifying the semantic meaning of a specific scene. Existing approaches to shot analysis primarily employ low level image features. The semantic primitives such as interesting objects and events are ignored or

accommodated manually.

As stated above, while most of the work focused on the low level features of the video frames, our research will be devoted to extracting the semantic meaning of the video data. While automatic extraction of complete semantics may not be feasible, the extraction of regions/objects of interest is regarded as the basis for extracting high level semantic meaning. For example, the presence/absence, movement patterns, relative positions, etc. of those objects of interest denote significant events in the shots. The frames denoting such events could serve as key frames and the movement patterns could provide a basis for building concise visual representations. In this section, focusing on the uncompressed video data domain, we propose an innovative shot change detection method using the unsupervised image segmentation algorithm SPCPE and the object tracking technique. By using the unsupervised image segmentation algorithm, the significant objects or regions of interests as well as the *segmentation mask map* of each video frame can be automatically extracted. The *segmentation mask map*, in other words, can be deemed as the clustering feature map of each frame. In such a way, the pixels in each frame have been grouped into different classes (for example, 2 classes). Then two frames can be compared by checking the difference between their segmentation mask maps. In addition, in order to better handle the situation of camera panning and tilting, the object tracking technique based on the segmentation results is used as an enhancement to the basic matching process. Since the segmentation results are already available, the computation cost for object tracking is almost trivial compared to manual template-based object tracking methods. Moreover, our key frame representation uses the information of the segmentation results such as the bounding boxes and the positions of the segments within that frame. For efficiency purposes, in addition to segmentation and object tracking, we also apply traditional pixel-level comparison combined with histogram comparison for pre-screening. The advantages of using unsupervised segmentation and object tracking are summarized below:

- The extraction of regions/objects is fully unsupervised, i.e., without any user interventions or domain-dependent knowledge.

- The algorithm for comparing two frames is simple and fast based on the segmentation results.
- It is robust to small changes in luminance or contrast, and has high accuracy even when the video quality is pretty poor.
- The unsupervised object level segmentation results can be further used for video indexing (e.g., key frame selection) and content analysis (e.g., video retrieval by the key object, event identification, etc.).

Our proposed shot change detection method combines three main techniques, namely the pixel-histogram comparison method, segmentation, and object tracking. In the following sections, we first explain the pixel-histogram comparison method, the unsupervised segmentation algorithm and object tracking, and then present the steps of the shot change detection method based on the discussion. The unsupervised segmentation algorithm (SPCPE) is ignored here since it has been explained in detail in Chapter 4. Experimental results are analyzed and compared with other methods to show the effectiveness of the proposed method. Finally, conclusions are given.

5.1.1 Pixel-Histogram Comparison

In the traditional pixel-level comparison approach, the gray-scale values of the pixels at the corresponding locations in two successive frames are subtracted and the absolute value is used as a measure of dissimilarity between the pixel values. If this value exceeds a certain threshold, then the pixel gray scale is said to have changed. The percentage of the pixels that have changed is the measure of dissimilarity between the frames. This approach is computationally simple but sensitive to digitalization noises, illumination changes and the object movement. Since histogram comparison should be less sensitive to object motion than the pixel comparison, in our framework we use the pixel comparison as the first filter and the histogram comparison as the second filter to verify the possible shot boundaries. It should be noted that the introduction of the histogram comparison as the second-level filter would not bring much extra computation, because it can be done in one pass with pixel comparison for each video frame.

However, the combined two methods still suffer from object motions. On the other hand, our proposed segmentation and object tracking techniques (to be discussed in Sections 3.2 and 3.3) are much less sensitive to the above factors. In fact, tracking of objects is a very strong criterion for determining the shot boundaries. Besides, the objects can serve as fundamental units for any video indexing.

In our method, we use the pixel-histogram comparison for pre-screening. By applying a strict threshold for the percentage of changed pixels, we want to make sure that we will not introduce any incorrect shot boundaries that are identified by pixel-level comparison by fault. For those 'skeptical' shot boundaries, histogram comparison is then used to exclude some false detections due to the sensitivity of pixel comparison. The advantage to combining the pixel-histogram comparison is its simplicity. In other words, we apply the segmentation and object tracking techniques only when it is necessary.

5.1.2 Extend SPCPE for Video Frame Segmentation

Since the successive video frames do not differ much due to the high temporal sampling rate, the partitions of the adjacent frames do not differ significantly. The key idea is then to use the unsupervised image segmentation method (SPCPE) successively on each frame of the video, incorporating the partition of the previous frame as the initial condition while partitioning the current frame, which can greatly reduce up to 90% of the computation cost because the number of iterations needed is much less than that of randomly initial partition. In our experiment, we just use two classes in segmentation since two classes are efficient and good enough for our purpose in this application domain. Another practical issue is that it is not necessary to wait until there is no further change in the class partition. Instead, when the percent of pixels that change their class labels is less than a threshold, the class partition can be deemed stable so that the iteration stops. In brief, we adopt the following strategies to achieve computation efficiency for extracting objects from a video sequence [Zhang03]:

- In order to achieve computation efficiency, we use the incremental computation together with parallel computation to speed up the clustering process. The basic idea of incremental computation is to compute the class parameters at the $(k+1)^{th}$ iteration using the intermediate

results at the k^{th} iteration rather than calculate it from scratch, thus to reduce the computation significantly. To further improve the speed, the parallel computation is also applied on sub-images by using MPI (Message Passing Interface) and SPMD (Single Processor/Multiple Data) on Cluster Computing.

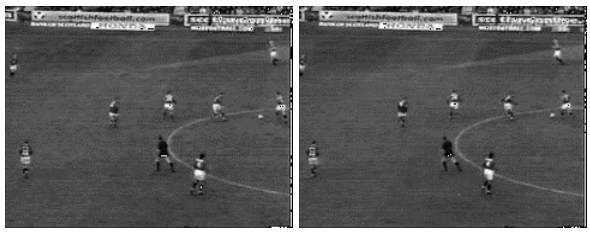
- Another strategy is that, it is not necessary to wait until there is no further change in the class partition. Instead, when the percent of pixels that change their class labels is less than a threshold (say 5%), the class partition can be deemed stable so that the iteration stops.
- Further, since the consecutive frames in video sequences are closely related in contents, incorporating the partition of the previous frame as the initial condition while partitioning the current frame can greatly reduce the computation cost up to 90%.

As a result, the combined speed-up factor can achieve 100~200. The time for segmenting one video frame ranges from 0.03~0.12 sec. Since a pre-screening step based on pixel comparison is used to filter out most of the video frames, the number of frames that need to do segmentation is small.

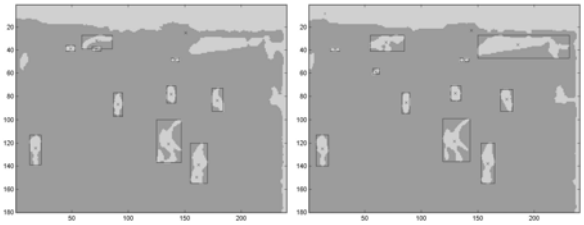
5.1.3 Object Tracking

The first step for object tracking is to identify the segments in each class in each frame. Then the bounding box and the centroid point for that segment are obtained. For example, Figure 5.1(b) shows the segmentation mask maps of the video sequence in Figure 5.1(a). In this figure, the player, soccer ball and the signboard belong to class 2 while the ground belongs to class 1. As shown in Figure 5.1(b), the segments corresponding to the ball, player and signboard are bounded by their minimal bounding boxes and represented by their centroid points.

After SPCPE is applied to obtain the segmentation mask map, the following *Line Merging Algorithm* is used to extract the bounding boxes and centroids for all segments. In this case, suppose we need to extract such information for all segments belonging to class 2 (as shown in Figure 5.1(b)):



(a): Example video sequence



(b): Segmentation mask maps and bounding boxes for (a).

Figure 5.1 Object tracking.

5.1.4 Shot Change Detection Method

Figure 5.2 shows the flowchart of the proposed shot change detection method. The steps are given in the following:

1. Do the pixel-level comparison between the currently processed video frame and the immediate preceding frame (see chart boxes 1 and 2 in Figure 5.2).

   Let the percentage of the changes be *change_percent* and the variance of the pixel-level differences be *change_variance*. Check these two variables (chart box 3).

   If the current frame is not identified as a shot boundary, which means that *change_percent*$<\delta_{ph}$ or *change_variance*$<\delta_{v}$, then go on to process the next video frame (chart box 1). Otherwise go to step 2 (chart box 4).

   (The purpose of checking *change_variance* is to pre-screen the fade in and fade out situations because they usually result in high *change_percent* and low *change_variance.* Although object tracking can deal well with both of the situations as will be shown later, conducting this will reduce the number of frames that need segmentation.)

2. Compare the histogram difference between the current frame and the immediate preceding frame.

Let *Histo_diff* denote the histogram difference between two frames (frame *m* and frame *m+1*):

$$Histo\_diff = \frac{\sum_{i=1}^{H} \left| B_m(i) - B_{m+1}(i) \right|}{N} \tag{27}$$

where H is the number of histogram bins, $B_m(i)$ denotes the i^{th} histogram bin of frame m, and N is the total number of pixels in the video frame.

If $Histo\_diff < \delta_{hh}$ (chart box 5), the current frame is not identified as a shot boundary, and then continue to process the next video frame (chart box 1). Otherwise go to step 3 (chart box 6).

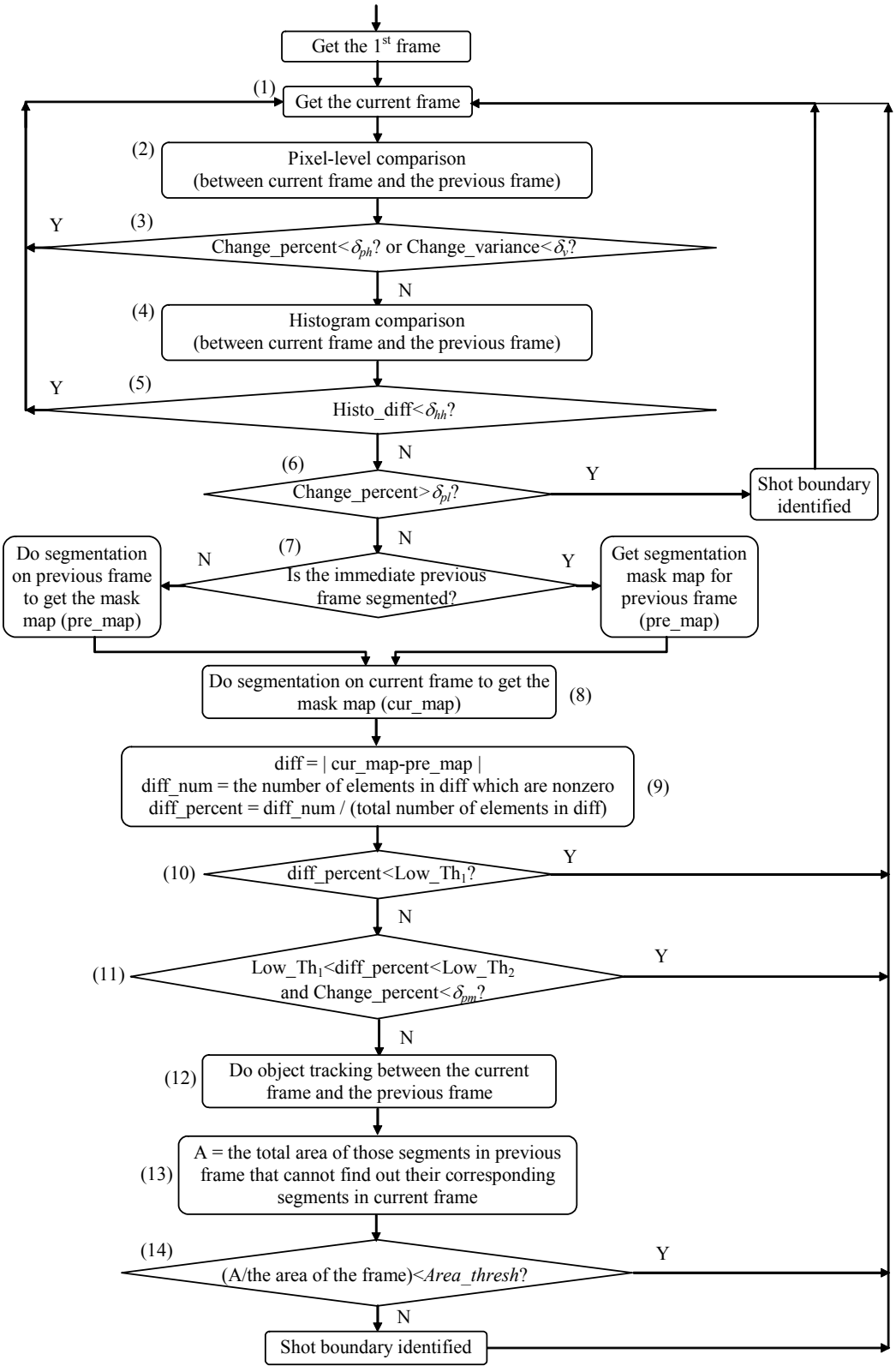


Figure 5.2 The flowchart of the proposed shot change detection method.

3. If *change_percent*> δ_{pl} (chart box 6), the current frame is identified as a shot boundary. Go to step 1 and process the next frame (chart box 1).

   If *change_percent*<=$\delta_{pl,}$ go to step 4 (chart box 7).

4. Do the segmentation on the previous frame only if it has never been segmented (chart box 7).

   If the previous frame has been segmented before, we only need to obtain its segmentation mask map directly.

   Then do segmentation on the current frame (chart box 8). Get the current and the previous segmentation mask maps for these two frames. Let the variable *cur_map* represent the current segmentation mask map's value and variable *pre_map* represent the value of the previous segmentation mask map. Note that the variables *cur_map* and *pre_map* can be deemed as two matrices. Go to step 5 (chart box 9).

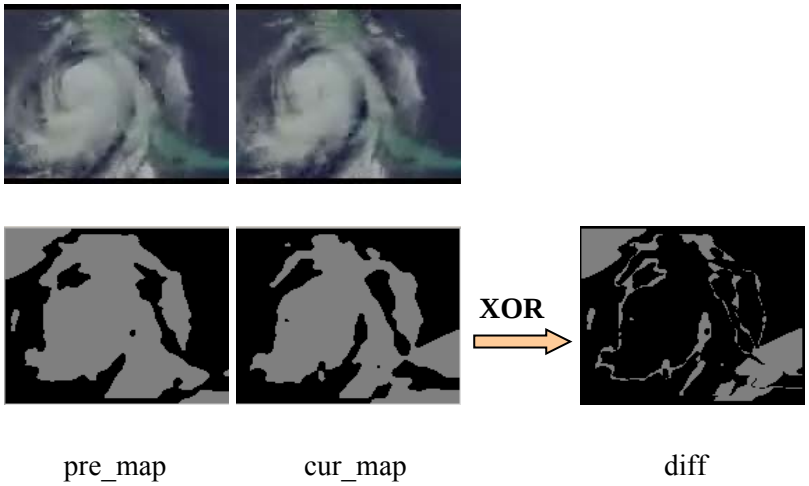


Figure 5.3 Subtraction of segmentation mask maps

5. *diff* = | *cur_map-pre_map* |, where the variable *diff* is the point-to-point subtraction between two successive segmentation mask maps. In fact, since the pixel value in segmentation mask map is binary data (either *1* or *2*), using the XOR operation is very efficient. An example of segmentation mask map subtraction is shown in Figure 5.3.

diff_num = the number of nonzero elements in *diff*;

diff_percent = *diff_num* / (total number of elements in *diff*), where the variable *diff_percent* is the percentage of the changes between the two successive segmentation mask maps.

Go to step 6 (chart box 10).

6. Check the variable *diff_percent* (chart box 10).

If *diff_percent* < Low_Th_1

Not shot change. Go to step 1 and process the next frame (chart box 1).

Else

If $Low\_Th_1 < diff\_percent < Low\_Th_2$ and $change\_percent < \delta_{pm}$ (chart box 11)

Not shot change. Go to step 1 and process the next frame (chart box 1).

Else

Do object tracking between the current frame and the previous frame. Let variable *A* be the total area of those segments in the previous frame that cannot find out their corresponding segments in the current frame (chart boxes 12 and 13).

If (*A*/the area of the frame)<*Area_thresh* (chart box 14)

Not shot change. Go to step 1 and process the next frame (chart box 1).

Else

The current frame is identified as shot boundary.

Go to step 1 and process the next frame (chart box 1).

End if;

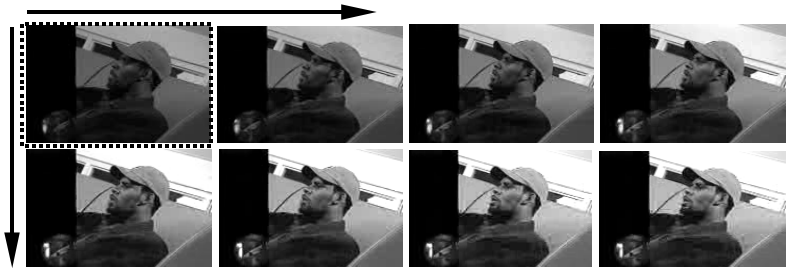
End if;

End if.

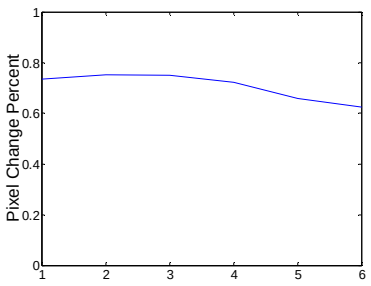
(Here, δ_{ph}, δ_{pl}, $\delta_{v,}$ δ_{pm}, δ_{hh}, Low_Th_1 and Low_Th_2 are threshold values for variables *change_percent, diff_percent* and *Histo_diff*, and they are derived from the experiential values.)

5.1.5 More Sophistication in Shot Detection

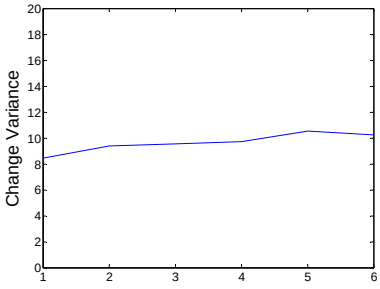
In the case of fade in and fade out, both luminance and histogram change dramatically along the video sequence, which results in many false positive shot boundaries. However, the variance values of frame to frame differences (The *change_variance* defined in previous subsection) remain relatively low during fade in and fade out especially when the transition is moderate. Figure 5.4(c) gives an example curve of variance during a mild fade in transition. As one can see in Figure 5.4(b), the values of pixel *change_percent* during fade in keep going high (above *50%*), while the variance values of the frame to frame difference (Figure 5.4(c)) keep relatively low (around 10). When both of the above two conditions are satisfied, a fade in/out is said to occur.



(a)



(b)



(c)

Figure 5.4 (a) An example video sequence of fade in. The temporal order of the sequence is from the top-left to the bottom-right; (b) the values of pixel change percent during fade in; (c) the variance of the frame to frame difference during fade in.

However, if the luminance changes during fade in/out are not relatively even (for example, the logarithmic increase in brightness), or the rate of change is too fast, using the above two

conditions alone cannot guarantee to exclude the false positives because in this case the *change_variance* may jump to a higher value due to the unevenness of the luminance change. Figure 5.5(a-c) gives such an example of a fade in sequence. Let the luminance difference between frames i and $i+1$ be $L_{i,i+1}$, then

$$Dev_{i,i+1}^{j} = \left| L_{i,i+1}^{j} - mean(L_{i,i+1}) \right| \tag{28}$$

where $1 \leq j \leq N$ (N is the total number of pixels).

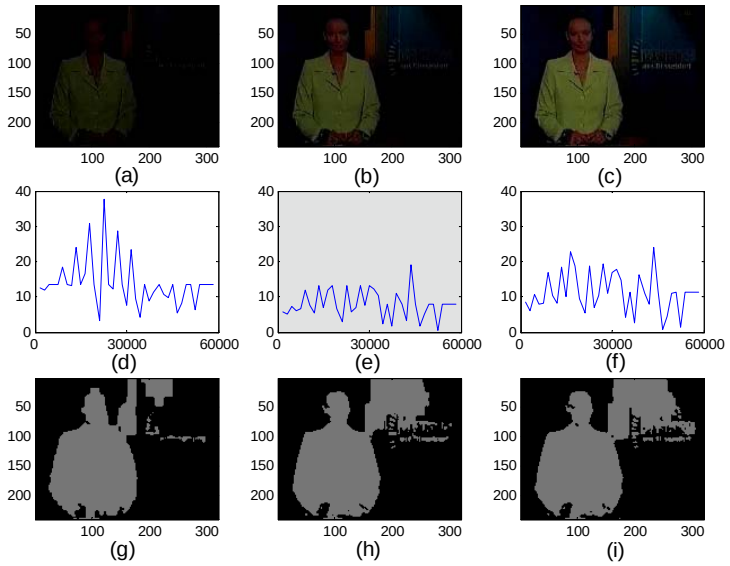


Figure 5.5 (a-c) The example fade in sequence with the logarithmic luminance change between frames; (d) the distribution of $Dev_{a,b}^{j}$; (c) the distribution of $Dev_{i,i+1}^{j}$ for a typical fade in sequence where the luminance change is more even; (f) the distribution of $Dev_{b,c}^{j}$; (e) a typical distribution of $Dev_{i,i+1}^{j}$ when the luminance change is relatively smooth; (g-i) the corresponding segmentation mask maps for the video sequence shown in (a-c).

Figure 5.5(d) shows the distribution of $Dev_{a,b}^{j}$ for frames a and b, while Figure 5.5(f) gives the distribution of $Dev_{b,c}^{j}$ for frames b and c. The plot shown in Figure 5.5(e) shows what a typical distribution of $Dev_{i,i+1}^{j}$ looks like when the luminance change is more even. It is not

difficult to see that the more evenly the luminance changes, the more even the distribution of $Dev^{j}_{i,i+1}$. Comparing $Dev^{j}_{a,b}$ and $Dev^{j}_{b,c}$ with the distribution in Figure 5.5(e), the video sequence shown in Figure 5.5(a-c) indicates a high level of non-linear changes in luminance during fade in. In this case, using pixel-histogram comparison alone will misidentify all these three frames as shot boundaries. However, we can avoid this situation by the assistance of segmentation mask map comparison. As illustrated in Figure 5.5(g-i), the segmentation mask maps of the three video frames look very similar to each other so that the false positives can be eliminated. In addition, it is very difficult to see the actual object inside Frame *a* by looking at the original image although the object is indeed emerging. However, our proposed segmentation method can still identify the object, which shows that the proposed segmentation together with object tracking technique is not sensitive to luminance changes.

In addition to dealing with the fade in/out situation, we also propose a method to handle the flash light effect, which is one of the common sources of errors. Assume the flash light effect will not last more than one frame (as shown in Figure 5.6). By comparing the preceding frame and the succeeding frame using pixel-histogram comparison, most of the false positives can be eliminated.

Figure 5.6 An example sequence of flash lights effect.

5.1.6 Implementation and Experiments

We have performed a series of experiments on a variety of video types such as TV news videos, commercial video, documentary video, and one movie video. This video collection provides representative material for testing purpose. The formats of the collected videos include MPEG-2

and RealVideoPlayer formats. We use RealVideoPlayer format for its low video quality due to the high compression rate (usually its compression rate is about four times of that in MPEG-2), which provides an ideal data set for testing the robustness and limitations of the proposed method. The size of the video frame in the sample video clips ranges from 112×160 to 288×352. All the video clips are downloaded from the URLs listed in [Web1, Web2, Web3, Web4]. Table 5.1 gives the statistics of all the video clips used. The performance was evaluated based on the shot boundaries identified manually.

Table 5.1 Video data used for experiments.

Name	Type	Duration (Min:Sec)	Number of Frames	Number of Shots	Size (row*col)
V1	Documentary	8:58	16117	107	240*352
V2	Documentary	23:08	41596	246	240*352
V3	Documentary	43:37	65441	360	240*320
V4	Documentary	1:29	2145	12	120*160
V5	Documentary	1:45	2615	15	120*160
V6	Documentary	1:36	1962	18	120*160
V7	Documentary	1:25	1774	12	120*160
V8	Documentary	0:20	495	6	240*320
V9	Documentary	1:31	1798	18	120*160
V10	Documentary	1:33	1992	18	120*160
V11	Documentary	1:29	2225	16	120*160
V12	Documentary	1:31	2282	16	120*160
V13	Movie	37:02	55572	362	288*352
V14	News	2:28	3703	32	120*160
V15	News	2:21	4225	55	112*160
V16	News	0:42	999	6	112*160
V17	News	3:13	4634	21	240*320
V18	Commercial	0:51	1294	27	120*160
V19	Commercial	1:30	2691	25	240*352
V20	Commercial	1:34	2805	21	240*352
V21	Commercial	0:29	727	24	288*352
Total		**138:32**	**217092**	**1417**	

5.1.6.1 Experimental Results

The performance is given in terms of the *precision* and *recall* parameters. N_C means the number of correct shot change detections, N_E means the number of incorrect shot change detections, and N_M means the number of missed shot detections.

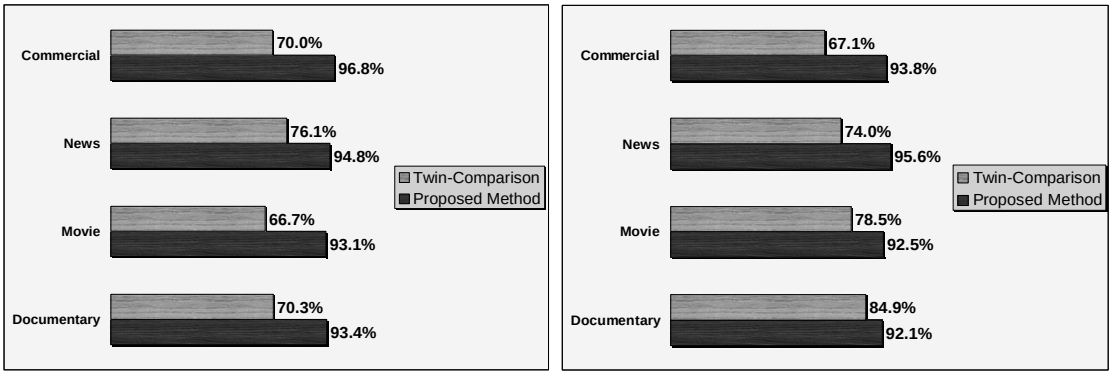
$$precision = \frac{N_C}{N_C + N_E} \tag{29}$$

$$recall = \frac{N_C}{N_C + N_M} \tag{30}$$

A summary of the proposed method compared with the standard histogram method [Zhang93] is shown in Table 5.2 and Figure 5.7via the *precision* and *recall* parameters.

Table 5.2 The Precision and Recall Parameters.

Name	Number of Shots	Proposed Method					Twin-Comparison Histogram				
		Detected	Correct	Missed	Precision	Recall	Detected	Correct	Missed	Precision	Recall
V1	107	107	107	0	*100.0%*	*100.0%*	112	103	4	*92.1%*	*96.3%*
V2	246	234	227	19	*97.0%*	*92.3%*	320	225	21	*70.1%*	*91.3%*
V3	360	364	317	43	*87.0%*	*88.0%*	453	272	88	*60.1%*	*75.6%*
V4	12	12	12	0	*100.0%*	*100.0%*	13	12	0	*92.3%*	*100.0%*
V5	15	15	15	0	*100.0%*	*100.0%*	13	12	3	*92.8%*	*81.2%*
V6	18	15	15	3	*100.0%*	*83.3%*	14	14	4	*100.0%*	*78.9%*
V7	12	12	12	0	*100.0%*	*100.0%*	15	12	0	*80.0%*	*100.0%*
V8	6	6	5	1	*85.7%*	*85.7%*	5	4	2	*80.0%*	*66.7%*
V9	18	18	18	0	*100.0%*	*100.0%*	19	17	1	*89.4%*	*94.4%*
V10	18	17	17	1	*100.0%*	*94.4%*	17	15	3	*88.2%*	*83.3%*
V11	16	16	16	0	*100.0%*	*100.0%*	22	15	1	*66.7%*	*93.3%*
V12	16	16	16	0	*100.0%*	*100.0%*	15	15	1	*100.0%*	*94.1%*
SubTotal for Documentary	**844**	**832**	**777**	**67**	***93.4%***	***92.1%***	**1019**	**716**	**128**	***70.3%***	***84.9%***
V13	362	360	335	27	*93.1%*	*92.5%*	426	284	78	*66.7%*	*78.5%*
SubTotal for Movie	**362**	**360**	**335**	**27**	***93.1%***	***92.5%***	**426**	**284**	**78**	***66.7%***	***78.5%***
V14	32	34	31	1	*91.2%*	*96.8%*	30	21	11	*70.6%*	*66.7%*
V15	55	56	53	2	*94.6%*	*96.4%*	51	39	16	*76.2%*	*71.1%*
V16	6	6	6	0	*100.0%*	*100.0%*	6	6	0	*100.0%*	*100.0%*
V17	21	19	19	2	*100.0%*	*90.5%*	23	18	3	*76.7%*	*85.2%*
SubTotal for News	**114**	**115**	**109**	**5**	***94.8%***	***95.6%***	**111**	**84**	**30**	***76.1%***	***74.0%***
V18	27	27	27	0	*100.0%*	*100.0%*	24	18	9	*75.0%*	*66.7%*
V19	25	25	23	2	*92.0%*	*92.0%*	31	19	6	*63.0%*	*77.3%*
V20	21	19	19	2	*100.0%*	*90.5%*	26	17	4	*66.7%*	*81.8%*
V21	24	23	22	2	*95.7%*	*91.7%*	12	11	13	*84.6%*	*44.0%*
SubTotal for Commercial	**97**	**94**	**91**	**6**	***96.8%***	***93.8%***	**93**	**65**	**32**	***70.0%***	***67.1%***
Total	**1417**	**1401**	**1312**	**105**	***93.6%***	***92.6%***	**1649**	**1150**	**267**	***69.7%***	***81.1%***



(a) Precision (b) Recall

Figure 5.7 The comparison results of *Precision* and *Recall* for different types of video clips (News, MTV, Documentary, Commercial and Sports).

In Table 5.2, the detailed comparison results with the twin-comparison histogram method are presented. As shown in Table 5.2, the performance of the proposed method is much more stable than the twin-comparison histogram method. The major sources of false positives for the twin-comparison histogram method are object motion, fade in/out, and camera panning. Also, it tends to miss more shot cuts because of the following reasons: (1) two video frames with similar histograms but totally different content, (2) improper threshold selection, and (3) low image quality of the video frames. In our implementation, we use the recommended threshold selection proposed in [Zhang93]. Moreover, as can be seen from Figure 5.7, the overall values of *recall* and *precision* for the various types of video clips are all above ninety percent in our proposed method.

Figure 5.8 gives an example of the missed shot boundaries by using the twin-comparison histogram method. In this sequence, a dissolve occurs, which is partly due to the low quality of the video frames (RealVideoPlayer format), and neither the frame to frame pixel change percent nor the histogram difference is high enough to detect this shot boundary. However, as one can see from the second row in Figure 5.8, which shows the corresponding segmentation mask maps for those video frames in the first row, the frame to frame difference in the segmentation mask maps

is significant enough to identify the shot boundary. This is another example demonstrating the robustness of the proposed framework.

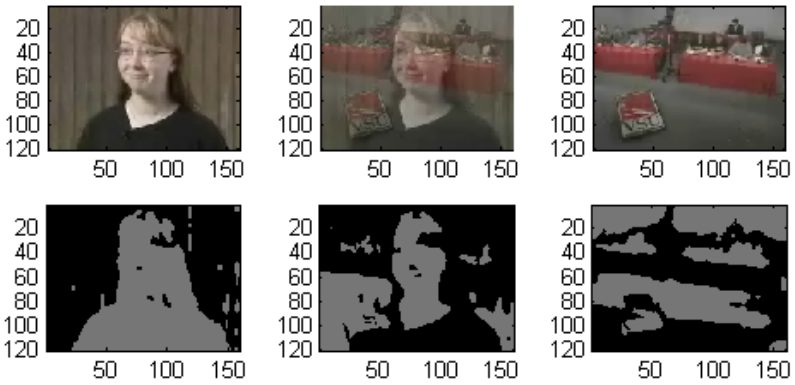


Figure 5.8 Missed shot boundaries by using the twin-comparison histogram method.

By applying the proposed method, the missed camera shots mainly result from the slow dissolve situations. In such cases, both the pixel-histogram difference and segmentation mask map difference are similar. The major source of false positives for the proposed method is the large object motion. Figure 5.9 gives a false detection due to the large object motion.

Figure 5.9 False identified shot boundary by the proposed method due to large object motion.

5.1.6.2 Performance Analysis

As mentioned before, the method of using low-level features is very sensitive to luminance and color change, but our segmentation-based method is not. One thing that should be mentioned here is that even if it is efficient to simply compare the segmentation mask maps, the employment of the object tracking technique is very useful in case of camera panning and tilting. It helps to reduce the number of incorrectly identified shot cuts. And by combining the pixel-histogram

comparison, only a small number of the video frames need to do segmentation and object tracking. Based on our experiments, usually the percentage of the frames that need to do segmentation ranges from *0.9%* to *4%*. The processing rates for each video are listed in Table 5.3. The implementation platform is WindowsXP with 1.3GHz CPU.

Table 5.3 Processing rate of proposed method.

Name	Type	Number of Frames	Number of Shots	Average Number of Frames/Shot	Frame Size (row*col)	Number of Frames Processed per Second	Processing Time (Sec.)
V1	Documentary	16117	107	151	240*352	7.71	2091
V2	Documentary	41596	246	169	240*352	8.82	4718
V3	Documentary	65441	360	182	240*320	9.41	6952
V4	Documentary	2145	12	179	120*160	36.17	59
V5	Documentary	2615	15	174	120*160	33.19	79
V6	Documentary	1962	18	109	120*160	35.40	55
V7	Documentary	1774	12	148	120*160	32.18	55
V8	Documentary	495	6	83	240*320	7.72	64
V9	Documentary	1798	18	100	120*160	39.41	46
V10	Documentary	1992	18	111	120*160	37.99	52
V11	Documentary	2225	16	139	120*160	42.37	53
V12	Documentary	2282	16	143	120*160	37.26	61
V13	Movie	55572	362	154	288*352	7.50	7412
V14	News	3703	32	116	120*160	36.35	102
V15	News	4225	55	77	112*160	29.02	146
V16	News	999	6	167	112*160	42.51	23
V17	News	4634	21	221	240*320	9.97	465
V18	Commercial	1294	27	48	120*160	33.04	39
V19	Commercial	2691	25	108	240*352	9.15	294
V20	Commercial	2805	21	134	240*352	7.35	381
V21	Commercial	727	24	30	288*352	4.59	158
Average						**9.32**	**23305**

As can be seen from Table 5.3, the average processing rate is around *10* frames per second. Figure 5.10 also includes a column chart roughly illustrating the relationship between the processing rate (normalized) and the different video categories. One can observe that commercial videos have the lowest processing rate because there are many fancy transitions and object motion in their content.

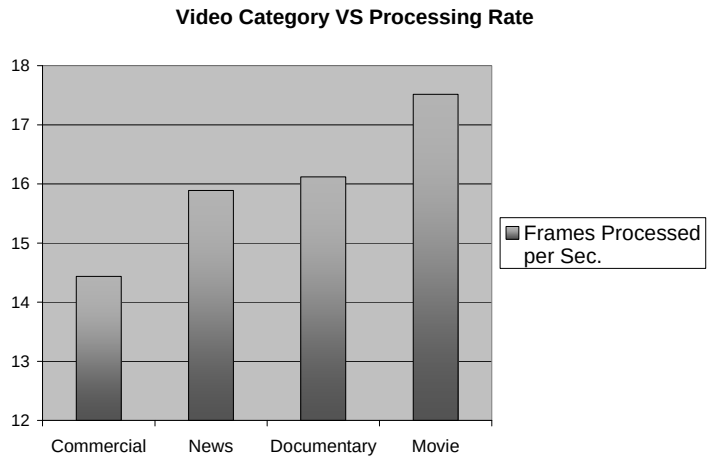


Figure 5.10 Video processing rate (frames/sec) against different video categories by using proposed method.

It should be noted that comparing each pair of consecutive frames' difference is the worst case in shot change detection, and any effort to reduce the spatial or temporal resolution of comparison will definitely give rise to the processing speed. As a matter of fact, many related works adopt multi-pass architecture as described in [Zhang93] to compare fewer frames by setting a 'skip factor' (for example, only checking two frames out of a *10* frames' segment). Since the multi-pass approach is not the focus of this study, we would like to do future research on that based on current promising results in the near future. The second way to reduce computation cost is to lower the spatial resolution. As one can see from Table 5.3, the processing rate is proportional to the video frame size. As for the proposed framework, it is highly possibly to lower the image resolution for all processing because of the robustness of the image segmentation method as shown earlier. Moreover, although the proposed framework is focusing on the video segmentation in the uncompressed data domain, it can be easily applied to the compressed data domain. For example, the proposed algorithm can operate directly on a DC image that is a small fraction of the compressed data and can be easily extracted without full frame decompression [Yeo95], which also avoids the extra processing time for re-sampling of the original image. The

key point is that such data size reduction still captures the core information of videos, thus allowing the operations to be preformed more efficiently. Our future work will put major effort on applying and evaluating the proposed framework on compressed data domain as well as to testing the idea of spatial-temporal resolution reductions in order to achieve high performance.

5.1.7 Conclusions

In this section, we proposed an innovative shot change detection method using the unsupervised segmentation algorithm and object tracking technique, and showed the precision and recall performance by using different types of sample video clips (in MPEG-2 and RealVideoPlayer formats). What distinguishes the proposed method from the previous methods is the adoption of fully unsupervised object segmentation and tracking method in shot change detection without any training or user intervention. This can serve as an important indication in shot detection and complement the shortcomings of the low-level feature based methods. The key idea of the matching process in our shot change detection method is to compare the segmentation mask maps between two successive video frames, which is simple and fast. In addition, the object tracking technique is employed as a complement to handle the situations of camera panning and tilting with little overhead. Unlike many methods using the low-level features of the video frames, the proposed method is not sensitive to small changes in luminance or color, and is robust to low quality videos. Moreover, it has high precision and recall values as shown in our experiment results.

5.2 Video Scene Detection

More and more research groups use both audio and video information to detect a scene boundary [Yoshitaka01, Sundaram00]. Despite several initial successes, it is still challenging to find a good way to combine audio and video information. In [Yoshitaka01], the candidate scene boundaries are extracted from the video data based on the extraction of visual effects such as dissolve or fade in/out. In addition, for the audio data, they only analyze the starting and ending audio frames of

each shot using the average power of subbands. Since the audio features within a short time duration often cannot represent the characteristics of the whole shot, monitoring the audio changes within a short time around the video shot boundaries cannot guarantee to identify the audio changes between the neighboring video shots. For example, when an audio change occurs before the video shot change, the method in [Yoshitaka01] cannot work well. In [Sundaram00], the authors used a finite-memory model to separately segment the audio and video data into scenes, and then applied two ambiguity windows to merge the audio and video scenes. In [Muramoto00], the authors did not combine the audio and video segmentation results. In [Jiang00], audio breaks were first detected in the one-second intervals. If a video shot boundary is within the one-second interval, this boundary is marked as a scene candidate that is then analyzed by a color correlation algorithm. The problem is that the audio data with one-second duration is too sensitive to represent the characteristics of the audio data.

In this section, an integrated audio-visual framework for video scene change detection is presented. Our framework consists of two parts. The first part considers the shot detection method as discussed in previous subsection. The second part analyzes the audio features based on the detected video shots. According to the distance between two shots with respect to the audio features, scene changes can be detected. Due to hard cuts, fades and dissolves in a video sequence, some of the video shots may be very short. In order to reduce these types of effects, if a shot consists of less than 5 frames, it is merged with its preceding shot before the audio processing. The main advantage of this method is that we use a very natural way to combine the audio and video data. Our audio processing is based on the video shots, which can avoid the conflicts between the audio and video data. The conflicts often occur when they are processed independently. One the other hand, in the audio processing part, there is no need to focus on the content (music, conversation, etc.) of the audio data. Instead, the focus is on their differences between the video shots. In this way, our method can be simplified and at the same time it can

still be very effective. Experimental results show that the proposed method is better than those that separately segment the audio and video data into scenes and then integrate them.

5.2.1 Audio Feature Extraction

In our experiments, the audio data is sampled by 16bits, 22.05KHZ. We regard 1024 samples as a frame, and each frame is shifted by 256 samples from the previous frame. Nine different features (mentioned in [Wang00, Lu01, Sundaram00a]) are used to analyze the audio data: 1) volume (*V*); 2) energy (*P*); 3) sub-band energy (*Sub-P*); 4) low shot-time energy ratio (*LSTER*); 5) zero crossing rate (*ZCR*); 6) frequency centroid (*FC*); 7) frequency bandwidth (*FB*); 8) spectral flux (*SF*); and 9) cepstral flux (*CF*). Features are extracted from each frame first. In addition, we divide the frequency domain into four sub-bands dynamically: 0 – 1/16 *fs*, 1/16 *fs* – 1/8 fs, 1/8 *fs* – 1/4 *fs*, and 1/4 *fs* – 1/2 *fs* respectively, where *fs* is the sample rate. From our observations, we found that the first and the third sub-bands are more suitable for our framework, so these two are selected to obtain the sub-band energy.

5.2.2 Shot-Level Processing

We observe that the audio changes that happen within a short time (such as one second) often cannot indicate the existence of a scene boundary. Therefore, instead of only monitoring the audio track changes around the video shot boundaries, the proposed method analyzes the audio features within the whole range of a video shot since these features may contain more useful information.

After dividing the audio data into different shots based on the video shot boundaries, we need to extract the audio features for each shot. Since it is not necessary to extract features for a silent shot, we first identify those silent shots using the following criteria. For each frame, if its volume < 0.003 and $ZCR > 50$, we consider it is a silent frame. Within each shot, if the percentage of the silence frames is larger than 0.7, this shot is considered as a silent shot, which will be ignored in the later processing. Moreover, for any non-silent shot, if it consists of

consecutive "silent frames" that last more than 0.33 second, these silent frames will also be skipped from the processing, which can prevent them from significantly affecting the audio feature value of that shot. Another observation is that editing effects often generate very short shots, which do not have enough audio information. Hence, we regard such a shot as a part of its neighboring shots (either the preceding shot or the succeeding one). In our framework, we select the preceding shot.

According to the properties of the nine features, we divide them (except *ZCR*) into three groups, namely Volume, Power and Spectrum. In each group, we calculate different values (such as mean, standard deviation, volume dynamic range, etc.) for the features in this group and add these values together. The idea is formalized as follows:

Volume group
$Vec(shot_i) = mean(V_i) + dev(V_i) + vdr(V_i) + diff(V_i)$
Power group
$Pvec(shot_i) = dev(P_i) + dev(Sub\text{-}P_i) + lster(P_i) + lster(Sub\text{-}P_i) + diff(P_i) + diff(Sub\text{-}P_i)$
Spectrum group
$Svec(shot_i) = dev(FB_i) + dev(FC_i) + diff(SF_i) + diff(CF_i)$

Where:

mean:the mean value of a feature in the *ith* shot;

dev: the standard deviation of a feature in the *ith* shot;

vdr: the volume dynamic range in the *ith* shot;

diff: the standard deviation of the frame-to-frame difference of a feature in the *ith* shot.

Because different features may have big differences in their values, the values are normalized by dividing them by their maximal value, and the normalized values are used in the

above equations to obtain the three values (*Vec*, *Pvec*, and *Svec*) for the three groups in each shot. These three values will be used to detect scene changes.

5.2.3 Scene Change Detection

First, we determine the distances between two neighboring shots with respect to the *Vec*, *Pvec*, and *Svec* values. For each value, a threshold can be obtained from the following equation.

$$T_{sv}=(mean(dist(sv_i, sv_{i+1})) + dev(dist(sv_i, sv_{i+1}))) / \sqrt{2} \quad (31)$$

Where *sv* can be either *Vec*, *Pvec*, or *Svec*, sv_i and sv_{i+1} are their values in two neighboring shots (the *ith* shot and the *(i+1)th* shot), and *dist* is the Euclidean distance.

Based on these threshold values, the following rule is used to determine whether there is a scene change.

```
If dist (Vec(shot_i), Vec(shot_i+1))>T_Vec or
    dist (Pvec(shot_i),Pvec(shot_i+1))>T_Pvec or
    dist (Svec(shot_i),Svec(shot_i+1))>T_Svec
then a scene boundary is recorded
```

If the distance of any one of the three values is larger than its corresponding threshold value *T*, we consider there is a scene boundary. We do not combine the three distance values into one distance measure because of the following two reasons. First, each value contains different audio semantic meanings, and simply combining them will destroy these meanings. Second, it is difficult to give them appropriate weights if we combine them. Different audio features may have different levels of importance for different audio data. A static combination of them will reduce the flexibility. Hence, we consider the above three values are complementary to each other.

5.2.4 Experimental Results

A series of experiments on a long movie video and several TV news videos were conducted. The performance was measured in terms of the precision (Pre.) and recall (Rec.) parameters. Table 5.4 lists the video features and the experiment results.

Table 5.4 Scene detection performance using joint audio and video clues.

	shot	scene	correct	miss	fault	pre.	rec.
V1	68	14	13	1	1	0.93	0.93
V2	29	13	12	1	1	0.92	0.92
V3	27	9	7	2	0	1	0.78
V4	18	11	10	1	1	0.91	0.91
V5	514	144	129	15	21	0.86	0.90
Average						**0.92**	**0.89**

In this table, we use V1 to V5 to denote the video sequences that we used in our experiments. The second and third columns indicate the numbers of shots and scenes in each video sequence. The fourth and fifth column give the numbers of scenes that our method correctly identifies and misses. The sixth column indicates the number of scenes that our method misidentifies. The last two columns give the precision and recall values for each video sequence using our method. For example, the number of scenes in V1 should be 14. Our method correctly identifies 13 out of 14 of them (i.e., one miss), and misidentifies one scene, which results in 0.93 in precision and 0.93 in recall values. As can be seen from the last row in this table, our method can achieve 0.92 in precision and 0.89 in recall on average, which demonstrates that our method works very well. Particularly, it works very well for the following two situations, where most of the existing approaches have difficulty in dealing with them.

1. At a shot boundary that happens to be a scene boundary, the visual feature changed but the audio feature did not change (as shown in Figure 5.11).

   In Figure 5.11, the audio data after point C corresponds to Shot B. For example, the scene changes between Shot A and Shot B. Near the end point of Shot A, the audio belonging to Shot A has been muffled while the audio belonging to shot B appears (for example, the voice of a speaker who belongs to Shot B). In other words, the audio changes corresponding to

scene changes occur before the video changes. In such a situation, most of the existing approaches cannot detect it correctly because there are no audio changes in point D. On the other hand, our method can detect it correctly since the extraction of the audio features is based on the whole shot.

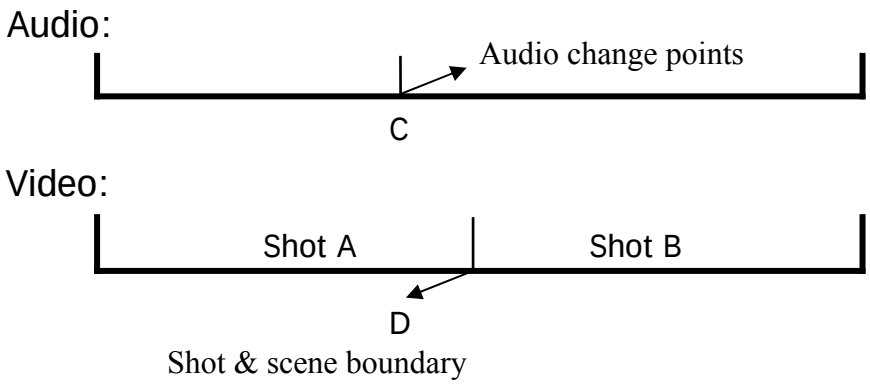


Figure 5.11 A scene boundary where audio does not change around shot boundary.

2. Audio feature changed within a short time around a shot boundary that is not a scene boundary (as shown in Figure 5.12).

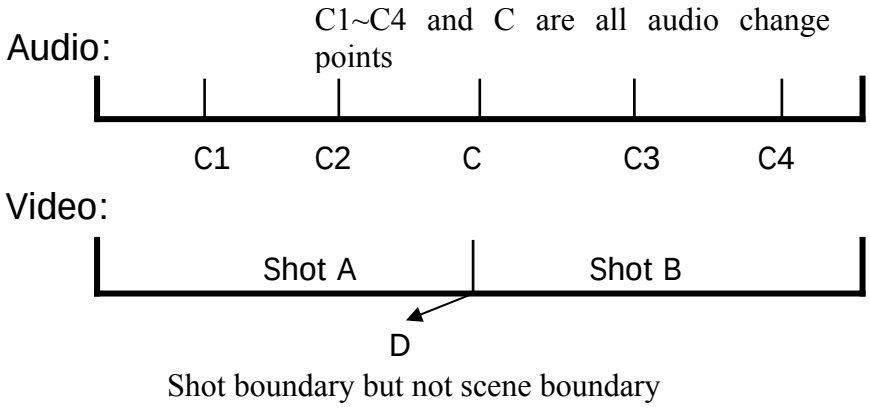


Figure 5.12 A scene boundary where audio does not change around shot boundary.

In Figure 5.12, Shot A and Shot B are in the same scene. Several speakers exist in both of the two shots. However, at the shot boundary, the speaker changes. In most of the existing

approaches, such situation is misidentified as a scene change since both audio and video features change. The experiments show that our method does not misidentify this as a scene change because the distance between the overall audio features of Shot A and Shot B is not that significant.

Moreover, our method of shot detection tends to detect a few more (still reasonable) shots instead of missing the correct hits [Chen01b], which is appropriate for the later audio feature extraction. Also, the extracted audio feature in turn can reduce the false hits in shot detection.

5.2.5 Conclusions

In this section, we presented an innovative scene change detection method using joint audio and video clues. Unlike the traditional methods that first analyze audio and video data separately and then combine them, we analyze them at different phases. The audio feature extraction is based on the detected video shots, which tends to be more stable and more reliable in charactering the audio data. The experimental results demonstrate that our method performs very well in terms of precision and recall values.

5.3 Soccer Event Detection Using Joint Multimedia Features and Data Mining Techniques

With the increasing amount of sports video, the need for automatic summarization and annotation has posed a great challenge to the multimedia database research community. In this section, we propose a new multimedia data mining framework for automatic soccer event detection from soccer videos by using multimodal feature analysis. The proposed framework consists of three major components: visual and audio feature extraction, data cleaning, and data mining. The video feature extraction component adopts a robust shot boundary detection method (See Section 5.1) and extracts the multimodal features (visual and audio features) at different granularities. A set of novel algorithms has been developed for efficient extraction of low-level features and object-level features. Then with the aid of domain knowledge, the data cleaning step is performed to eliminate the large amount of irrelevant data, and to prepare a cleaned data set for the decision-tree based

data mining process. This framework can detect two types of major events: soccer goal events and corner kick events. As a result, it can further tell whether a goal is a direct result of a corner kick, which is very useful in high-level semantic indexing and selective browsing of soccer videos. The effectiveness and efficiency of the proposed framework are demonstrated over a large collection of soccer video data with different styles as produced by different broadcasters.

5.3.1 Introduction

In the literature, the related work can be categorized into two groups of methods: unimodal and multimodal approaches. Unimodal approaches utilize the features of a single modality, such as visual [Gong95, Xu01], audio [Xu03], or textual, in soccer highlights detection. However, because the content of a video is intrinsically multimodal, in which the semantic meanings are conveyed via visual, auditory, and textual channels, such unimodal approaches have their limitations. Currently, the integrated use of multimodal features has become an emerging trend in this area. In [Dagtas01], a multimodal framework using combined audio, visual, and textual features was proposed. Though multimodal analysis shows promise in capturing more complete information from video data, it is still a major challenge to fully capture the high-level contents of the video events using only the low-level features [Kang03]. On the other hand, the middle-level features, e.g., the camera view types (global or close-up), replays, and grass areas/audience areas, provide a good base for high-level analysis of video semantics. The high-level features, such as object motion trajectories, offer an effective way to enable the direct reasoning of certain soccer events like goal attempts, kick-off, etc. However, the extraction of the object-level features is usually time-consuming and computationally expensive. Therefore, the existing work [Intille01, Tovinkere01] that is based on object features often relies on the assumption that the object trajectories and object interactions are already available. For example, in [Tovinkere01], the authors claimed that their method could detect a wide range of player actions and game events by using a set of heuristic rules which are derived from a hierarchical entity-relationship model

representing the prior knowledge of soccer events. However, the method works only with the assumption that the locations of the players and ball are available as the input. Therefore, their application to soccer event detection is limited. In contrast, in this paper, we propose to use a selective mixture of low-level features, middle-level features, and object-level features for the detection of certain soccer events without any manual effort.

In addition to the feature-based approaches, there are other approaches proposed in the literature using semantic rules defined based on domain knowledge. In [Kang03], an event detection grammar was built to detect the "Corner Kick" and "Goal" soccer events based on the detection rules. However, these rules need to be completely studied and pre-defined for each target event prior to generating the grammar trees that are used to detect the events. For example, there were totally 16 semantic rules defined for the corner kick events in [Kang03], which were derived by carefully studying the co-occurrence and temporal relationships of the sub-events (represented by semantic video segments) in soccer videos. However, there are several disadvantages to this approach: (1) The derived rules are based on the limited observation of a small set of soccer videos (4 FIFA2002 videos), which may not hold true when applied to other soccer videos produced by different broadcasters. For example, the <PR> in [Kang03] refers to the sub-event that one player runs to the corner just before the corner kick events. However, it is not a necessary pre-condition for corner kick events. (2) The classification performance of such rules largely depends upon the detection of sub-events. However, the detection of such sub-events is of the same difficulty level as or sometimes even more difficult than the target event. (3) The derivation of such a large set of rules requires considerable manual effort, which limits its generality. In contrast, the framework proposed in this paper does not rely on deriving the complete set of domain-specific rules. Instead, a set of simple and more general rules are used just for data cleaning, and the task of actual event detection is left to the data mining phase.

In this study, we propose a new multimedia data mining framework for automatic soccer event detection from soccer videos by using multimodal feature analysis. Three major components, namely the *visual and audio feature extraction*, *data cleaning*, and *data mining* are integrated into a single framework. The main contributions are summarized as follows:

(1) A robust shot boundary detection method based on our previous work [Zhang03, Chen03c] is used to segment the soccer videos, and the multimodal features (visual and audio features) are extracted for each video shot at different granularities. The video shot is the basic indexing unit for video analysis. Although the soccer event boundaries do not have to be aligned with the shot boundaries, the values of the color and audio features are usually more consistent within one shot than across different shots. Thus, it can help to better capture the semantic contents of, and to locate, the target events. One unique feature that distinguishes our shot detection method from many others is the production of some object-level features while parsing and segmenting the videos. In fact, an effective object segmentation algorithm is systematically integrated into the process of shot boundary detection [Zhang03], which can be further used in this framework to roughly distinguish the play fields from the players and audience areas. It is also efficient in the sense that there is no need to do object segmentation when the low-level features (e.g., pixel-wise difference, histogram difference) are sufficient for the detection of shot boundaries.

(2) We introduce data mining techniques to automatically detect certain soccer events. Data mining techniques have been actively studied due to the great capabilities in mining the interesting patterns from a huge data set. Intuitively, they can be applied to the discovery of the event patterns from a large set of soccer video data. In our previous study [Chen03b, Chen04a], we proposed a set of novel features for soccer goal event detection, and tested its performance by using decision trees and classification rules techniques,

which shows great promise in terms of high precision and recall values. In this study, we further improve our previous work by providing the capability to detect more soccer events like corner kicks and those goal events that are the direct results of the corner kicks [Chen04b]. The detection of such semantic events is very useful for high-level semantic indexing and selective browsing of soccer videos.

(3) A set of simple, intuitive, yet general enough rules are used in the data cleaning step to eliminate the large amount of irrelevant data, and to prepare a cleaned data set for the decision-tree based data mining process. Unlike the work proposed in [Kang03] in which the 'complete' set of detection rules needs to be identified manually and turns out to be too restrictive, the data cleaning rules in our framework are used only for the purpose of increasing the ratio of the positive samples to the negative samples. Thus the small amount of target events will not be discarded as noises during the data mining process.

It is worth mentioning that the proposed framework combines all these three components systematically and efficiently. The low-level and middle-level features produced during shot boundary detection can be used to derive high-level semantic features such as the audience areas and player areas, and these features are further utilized in the data cleaning rules for data cleaning. Part of the cleaned data set with their corresponding class labels are fed into the data mining process as the training data. Here, the label 'YES' ('NO') is used to denote that the event is (is not) contained in that video shot. The event patterns are explored and a decision tree model is built for each target event accordingly. In particular, in this framework, the mining process for the corner kick events and corner-goal events is carried out in a hierarchical way. In other words, the candidate pool is narrowed down at each data mining level. The more specific the target event is, the lower the data mining level it resides in. The derived data mining model serves as a bridge between the low-level features and the high-level video contents in detecting soccer events. The

effectiveness and efficiency of the proposed framework are demonstrated over a large collection of soccer video data with different styles produced by different broadcasters.

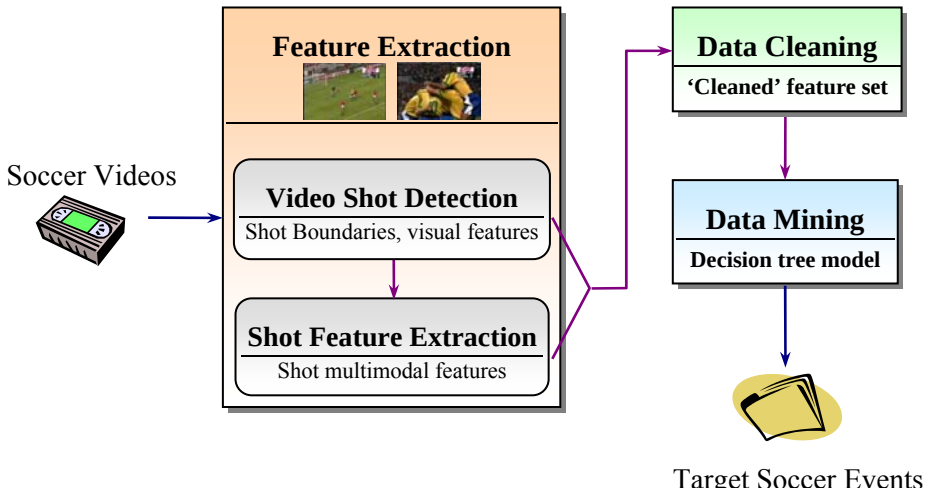


Figure 5.13 Architecture of the proposed framework.

5.3.2 Architecture of the Proposed Framework

The proposed framework is divided into three major components based on their functionalities, namely *visual and audio feature extraction*, *data cleaning*, and *data mining*, as shown in Figure 5.13. In the *visual and audio feature extraction* component, the shot boundaries and shot-level multimodal features (visual and audio features) are extracted. Since the ratio of positive video shots (corner kick, goal shots, etc.) over the entire set of shots is very small in soccer videos, a data cleaning process is applied before the actual data mining process for the sake of accuracy and efficiency. Then during the model-training step, 2/3rds of the 'cleaned' data set is fed into the data mining component as the training data set to build the decision tree model(s) for soccer event detection. Thereafter, the performance of the decision tree model is tested over the testing data set which consists of 1/3rd of the 'cleaned' data set, as will be shown later.

5.3.3 Visual and Audio Feature Extraction

The visual and audio feature extraction component includes two subcomponents: 1) an unsupervised video shot boundary detection subcomponent is used to temporally segment the raw

soccer video sequences a set of consecutive video shots; and 2) the detected shot boundaries are passed to the feature extraction subcomponent, where the shot-level multimodal features (visual and audio features) are extracted. The details of shot-detection method can be found in Section 5.1 and thus are ignored here. The following subsections will focus on the multi-modal feature extraction component.

5.3.3.1 Visual Feature Extraction

In this framework, multimodal features (visual and audio) are extracted for each shot based on the shot boundary information obtained in shot detection step. As shown below, totally five visual features are extracted for each shot, namely *pixel_change*, *histo_change*, *background_mean*, *background_var*, and *grass_ratio*.

Feature Name	Description
pixel_change_pencent	The average percent of the changed pixels between frames within a shot
histo_change	The mean value of the histogram difference between frames within a shot
grass_ratio	The average percent of grass areas in a video shot
background_var	The mean value of the variance of background pixels
background_mean	The mean value of the background pixels

Here, *pixel_change* denotes the average percentage of the changed pixels between the consecutive frames within a shot and can be obtained by averaging all the *pixel_change_frames* (the percentage of changed pixels between consecutive frames) within the shot. Similarly, *histo_change* represents the mean value of the frame-to-frame histogram differences in a shot and is the average of all the *histo_change_frames* (histogram difference between consecutive frames) within that shot. Obviously, *pixel_change* and *histo_change* can be obtained simultaneously and at low cost during the video shot detection process. It should be pointed out that both the two features are important indications of camera motion and object motion. For example, as shown in Figure 5.14, we have four consecutive video shots. The first shot is a close-up shot with a high

pixel-change value and a low histogram-change value, which indicates the object motion but with a slow camera motion. Shot 2 is a global shot or a play field shot, and it has low values for both *pixel-change* and *histo-change*. Usually in global shots, the visual effects of object motion or camera motion are not that significant. Shot 3 is a medium to close-up shot with high values for both *histo-change* and *pix-change*, which indicates the motion of large objects or a significant camera motion. Shot 4 is another global shot with low *pix-change* value and *histo-change* value. One observation here is that the feature values for Shot 4 are a bit less than of Shot 2, although both of them are global shots. The reason is that Shot 2 involves a more significant camera motion than Shot 4 does.

Shot 1 (close-up) Shot 2 (global) Shot 3 (close up) Shot 4 (global)

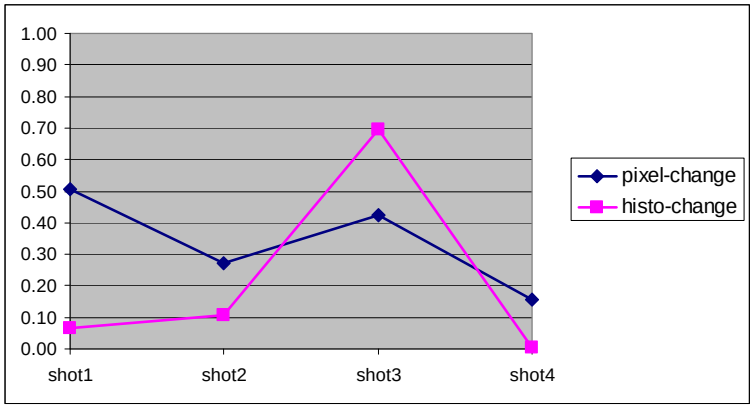


Figure 5.14 The two visual features *pixel_change* and *histo_change* as well as their indications for object motion and camera motion.

Grass Area Detection

Grass_ratio is a very important indication for classifying shot types (global, close-up, etc.) according to the video shooting scale. As we can see from Figure 5.15(a)-(b), a large amount of

grass areas are present in global shots (including **goal** shots and **corner-kick** shots), while there is less or hardly any grass area in the mid- or the close-up shots (including the cheering shots following the goal shots). Another computable observation is that the global shots usually have a much longer duration than the close-up shots.

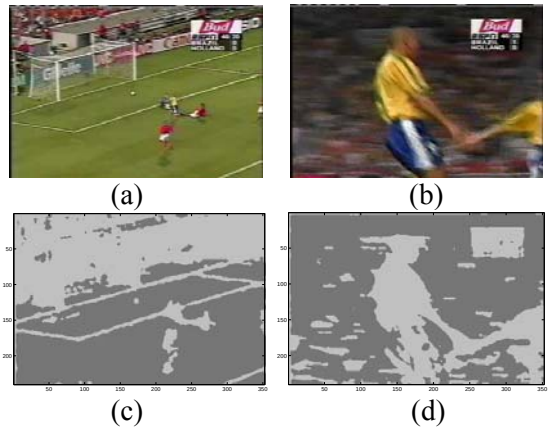


Figure 5.15 (a) a sample frame from a goal shot (global view); (b) a sample frame from the cheering shot following the goal shot for (a); (c) object segmentation result for (a); (d) object segmentation result for (b).

It is a challenge to distinguish the grass colors from others because the color values may change under different lighting conditions, different play fields, or different shooting scales, etc. The method proposed in [Gong95] relies on the assumption that the play field is always green in order to extract the grass areas. However, this is not always true for the reasons mentioned above. In [Sun03], a method is proposed to solve this problem by building a table containing candidate grass color values. However, this method is not fully automatic because it still needs manually picked training data in order to build such a table. As a more robust solution, the methods based on the dominant color for grass area detection have been proposed in the literature recently. The work in [Ekin03] proposed to use the dominant color based method to detect grass areas, while our previous work [Chen03b] came up with the similar idea independently around the same time. Both of them do not assume any specific value for the play field color. The major theoretical

difference between them is the learning process. In [Ekin03], it assumed there is a single dominant color indicating the play field. The initial value for the field color is obtained by using only a few seconds of a video, and this field color will be adjusted by using the local temporal color features when necessary as it proceeds. In our method, we assumed the existence of multiple dominant colors that indicate the grass areas.

The first step in our method is to distinguish the possible grass areas from the player/audience areas, which is achieved by examining the segmentation mask maps of video frames. As shown in Figure 5.15, the dark-gray areas (class 1) roughly correspond to the grass areas, while the light-gray areas (class 2) include almost all of the players, audience, and backboard areas. The feature *background_var* represents the standard deviation of the pixels values that belong to a certain class. The class with a smaller *background_var* value will be selected as the possible grass area. The feature *background_mean* represents the mean color of each possible grass area, which indicates the possible play field colors. The color histogram is then calculated over the pool of all the possible field colors collected for a single video clip. The actual play field colors are computed around the histogram peaks, where the threshold for selecting histogram peaks can be adjusted. Once the play field colors are identified, the segments corresponding to play fields can also be identified. Therefore, the derivation of the *grass-ratio* value for a video frame is quite straightforward. In this work, we use the mean value of *grass-ratio* within a shot to indicate the shot type (global, close-up, etc.). Figure 5.16 shows the values of the *grass-ratio* and the corresponding *background-var* for shots 1-4 (see Figure 5.14). As can be seen from this figure, Shot 1 is a close-up shot and it has the highest *background-var* value and the lowest *grass-ratio* value (0). On the other hand, in Shots 2, 3, and 4, the values of *background-var* are relatively low, while the values of *grass-ratio* are high (in this example, *grass-ratio* ranges from 50% to 90%). It is worth mentioning that sometimes medium shots may

also have high grass-ratios. For example, Shot 3 is a medium view shot, but it contains a high ratio of the grass areas.

Figure 5.16 The two visual features *grass-ratio* and *background-var* as well as their indications for shot type classification.

In brief, the *grass_ratio* is obtained via the following three steps:

1. Within each shot, draw a set of video frames at 50-frame interval and do object segmentation for them. By object segmentation, the background (grass, crowd, etc.) areas and foreground areas (player, ball, etc.) are detected as shown in Figure 5.15(c)-(d), where foreground areas are marked in gray and background areas are marked in black. As can be seen from this figure, in global view shots, the grass areas tend to be detected as the background, while in close-up shots, the background is very complex and may contain crowd, sign board, etc.

2. Check the *background_var* of the background areas for each shot; if the *background_var* < *threshold*, indicating the possible grass area, then put its corresponding *background_mean* into a candidate pool containing possible grass values.

3. Once all the possible grass values are collected, filter off the outliers in the candidate pool by taking out those shots that are too short and those shots whose *background_mean* values are out of a reasonable scope of the average *background_mean*. Generate the histogram for the

grass values in the 'purified' candidate pool. Based on our observations on a large set of video data, there are two possible situations in the histogram: (1) there is only one peak in the histogram, indicating good video quality and stable lighting conditions, and (2) there are multiple peaks in the histogram, which correspond to the difference in grass colors between the global shots and the close-up shots caused by camera shooting scale and lighting condition. In situation (1), the only one peak is selected as the grass pixel detector to calculate *grass_ratio*, while in situation (2), multiple peaks within a reasonable range are all selected as grass detectors. Based on our experiments, most of the multi-peak cases are caused by two different shooting scales (global, close–up) as shown in Figure 5.17. Figure 5.18 shows the detected grass areas for three sample images from different types of shots (close-up, global, etc.), and the results are very good.

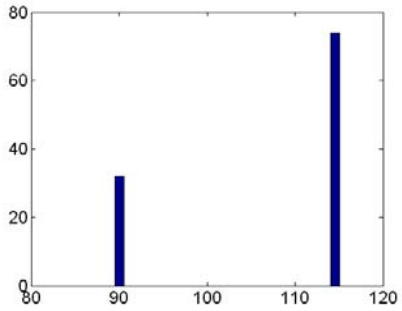


Figure 5.17 The histogram of the candidate grass values for a 20-minute long soccer video. Two peaks correspond to two major types of shooting scales in the video data – global and close-up.

It should be pointed out that this grass area detection method is unsupervised and the grass values are learned through unsupervised learning within each video sequence, which is invariant to different videos.

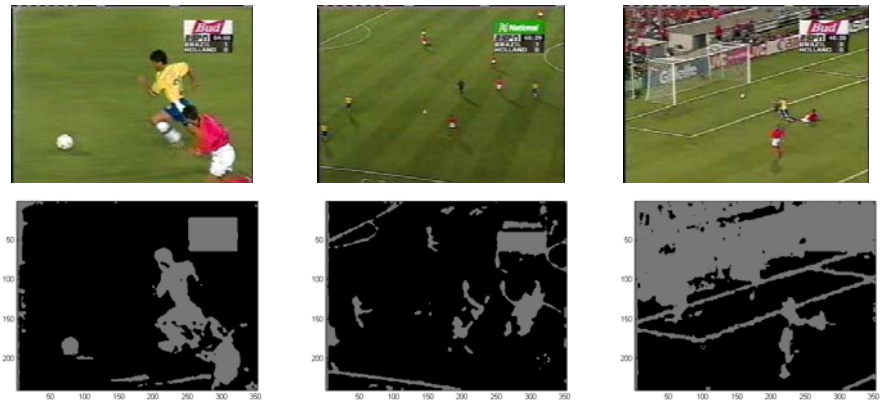


Figure 5.18 Detected grass areas (black areas) for 3 sample video frames from different types of shots.

As a brief summary to the visual feature extraction:

1. First, it takes advantages of the advanced video shot detection process proposed in Section 5.1, such that only limited extra effort is needed in order to extract visual features for each shot.
2. Some high-level semantic information, such as the grass area ratio, can also be obtained automatically by using the object segmentation component in video shot detection.
3. This method has great potential to provide more high-level semantic information such as the locations of players and balls, and the spatio-temporal characteristics among objects as well.
4. Necessary data normalization is done within each video sequence. By doing this, the values of each visual feature are normalized to a [0, 1] range.

5.3.3.2 Audio Feature Extraction

The soundtrack of a soccer video consists of speech and vocal crowd reactions, along with other environmental sounds such as whistles, clapping, etc. Because of their strong indications for the occurrences of certain soccer events, e.g., fouls, referees, commentators and audience, audio patterns are critical in identifying some interesting events, such as soccer goal events.

In this framework, the representations of the audio features are exploited in both time-domain and frequency-domain, which are divided into three distinct groups: volume related features, energy related features, and Spectrum Flux related features. Totally, 14 generic audio features are utilized (4 volume features, 7 energy features and 3 Spectrum Flux features). The audio data is sampled by 16bits and 16,000HZ, and divided into audio clips of one-second long. We regard 512 samples as a frame, and each frame is shifted by 128 samples from the previous frame. Thus an audio clip contains more than 30 frames. For sub-band energy, the frequency domain is divided into four sub-bands dynamically, which are 0 – 1/16 *fs*, 1/16 *fs* – 1/8 *fs*, 1/8 *fs* – 1/4 *fs*, and 1/4 *fs* – 1/2 *fs* respectively, where *fs* is the sample rate. A complete list of all the audio features as well as their feature descriptions is presented as follows.

Volume Related Features

Volume is one the most frequently used and simplest audio features. As an indication of the loudness of sound, volume is very useful for soccer video analysis. Four volume-based features used are:

Feature Name	Description
volumn_mean	The mean value of the volume
volumn_std	The standard deviation of the volume, normalized by the maximum volume
volume_stdd	The standard deviation of the difference of the volume
volume_range	The dynamic range of the volume, defined as $(\max(v) - \min(v)) / \max(v)$

Energy Related Features

Short time energy means the average waveform amplitude defined over a specific time window. To model the energy properties more accurately, energy characteristics of sub-bands are explored as well. Four energy sub-bands are identified, which covers, respectively, the

frequency interval of 1HZ-(fs/16)HZ, (fs/16)HZ-(fs/8)HZ, (fs/8)HZ-(fs/4)HZ and (fs/4)HZ-(fs/2)HZ, where *fs* is the sample rate.

Feature Name	Description
energy_mean	The mean RMS energy
sub1_mean	The average RMS energy of the first sub-band
sub3_mean	The average RMS energy of the third sub-band
energy_lowrate	The percentage of samples with RMS power less than 0.5 times the mean RMS power
sub1_lowrate	The percentage of samples with RMS power less than 0.5 times the mean RMS power of the first sub-band
sub3_lowrate	The percentage of samples with RMS power less than 0.5 times the mean RMS power of the third sub-band
sub1_std	The standard deviation of the mean RMS power of the first sub-band energy

Spectrum Flux Related Features

Spectral Flux (Delta Spectrum Magnitude) is defined as the 2-norm of the frame-to-frame spectral amplitude difference vector.

Feature Name	Description
sf_mean	The mean value of the Spectrum Flux
sf_std	The standard deviation of the Spectrum Flux, normalized by the maximum Spectrum Flux
sf_stdd	The standard deviation of the difference of the Spectrum Flux, which is normalized too

In addition, audio features are captured at different granularities: frame-level, clip-level, and shot-level, to explore the semantic meanings of the audio track. Audio features are extracted for each frame first. At the audio clip-level, a set of statistical features, e.g., *mean*, *standard deviation*, *dynamic range*, and *kurtosis value*, are calculated. Then the shot-level audio feature

vector can be obtained by calculating the statistics of all those audio clips. In addition, we also compute the clip features for the first three-second (clip) audio track and the last three-second (clip) audio track of each video shot, which are good indications of the goal events as will be shown in next section.

5.3.4 Data Cleaning

Data cleaning is conducted to remove the noise data and thus to improve the mining accuracy. The data cleaning process is extremely important in this study for the following two reasons: 1) First, production noise may be introduced into the videos (consequently into the shot features) and cause biases in the feature set. 2) Second, only a small portion of the shots in a long soccer game contains certain interesting soccer events like goals and corner kicks in this study. For instance, in our case, the number of corner kick shots only accounts for 1.6% of all the shots and the ratio of goal shots versus non-goal shots is even smaller (less than 1:100). Such a small ratio may cause difficulties for the data mining process to discover the event patterns from such a huge amount of irrelevant data. This section will present the details of data cleaning strategies for the detection of corner kicks and soccer goals, respectively.

Data Cleaning for Soccer Goal Events:

In the soccer videos, the sound track mainly includes the foreground commentary and the background crowd noise. Based on observation and prior knowledge, the commentator and crowd become excited at the end of a goal shot. In addition, different from other sparse happenings of excited sound or noise, normally this kind of excitement will last to the following shot(s). Thus the duration and intensity of sound can be used to capture the candidate goal shots as defined in the following rule:

- **Audio Rule 1**: As a candidate goal shot, the last three (or less) seconds of its audio track and the first three (or less) seconds of its following shot should both contain at least one exciting point.

Here the exciting point is defined as a one-second period whose volume is larger than 60% of the highest one-second volume in this video. It is worth mentioning that actually this volume threshold can be assigned to an even greater value for most of the videos. However, based on our experiments, 60% is a reasonable threshold since the number of the candidate goal shots can be reduced to 17% of the whole search space while including all the goal shots. In addition, this rule performs as a data cleaning step to remove some of the noise data, because though normally the noise data has high volume, it will not last for long.

As mentioned earlier, we have two basic types of shots, close-up shots and global shots, for soccer videos based on the ratio of the green grass area. We observe that the goal shots belong to the global shots with a high grass ratio and are always closely followed by the close-up shots which include cutaways, crowd scenes and other shots irrelevant to the game without grass pixels, as shown in Figure 5.19. Figure 5.19(a)-(c) capture three consecutive shots starting from the goal shot (Figure 5.19(a)), and Figure 5.19(d)-(f) show another three consecutive shots where Figure 5.19(d) is the goal shot. As can be seen from this figure, within two consecutive shots that follow the goal shot, usually there is a close-up shot (Figure 5.19(b) and (f), respectively).

According to these observations, two rules are defined as follows:

- **Visual Rule 2**: A goal shot should have a grass ratio larger than 40%.
- **Visual Rule 3**: Within two succeeding shots that follow the goal shot, at least one shot should belong to the close-up shots.

(a) (b) (c)

(d) (e) (f)

Figure 5.19 Goal shots followed by close shots: (a)-(c) three consecutive shots in a goal event. (b) is the close shot that follows (a) the goal shot; (d)-(f) another goal event and its three consecutive shots, (f) is the close shot that follows (d) the goal shot.

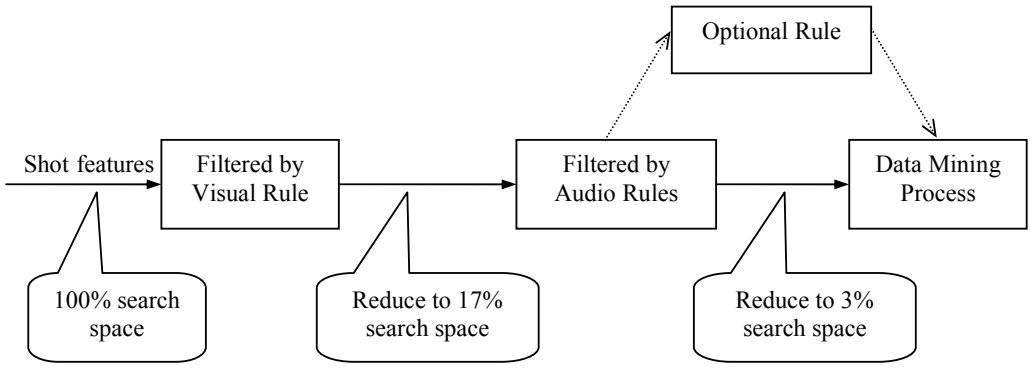


Figure 5.20 Pre-filtering processes.

Note that the threshold defined in Rules 2 can be altered to a higher value for most of the videos. However, our experiments show that 82% of the candidate pool obtained after applying Rule 1 can be reduced by using Rule 2 and Rule 3, which means that only 3% of the whole search space is remains as the input for data mining process. In addition, according to the prior knowledge, a goal shot normally lasts more than three seconds, which can be used as an optional

filter called Optional Rule. In our case, since the search space has been dramatically reduced, this rule has small effects. In summary, the workflow as well as the performance of the pre-filtering process is illustrated in Figure 5.20.

Data Cleaning for Corner Kick Events:

In recording soccer games, usually there are a fixed number of cameras in the play field. Based on our observation over a diversified set of soccer videos, during the corner kick events, there are four main types of camera views, namely *right/left-corner facing view* and *right/left-corner side view*, as shown in Figure 5.21. These four kinds of views differ in their camera shooting angles and camera positions, and the difference is usually reflected as the relative layout and shapes of the audience areas and play fields. Motivated by this observation, we can filter out most of the non-corner kick video shots and generate a much cleaner data set for the data mining component.

Identify right/left-corner facing views:

In the right/left-corner facing views, we have the following observations:

1) The right/left corners are usually visible and located in the upper part of a video frame.
2) The upper one-third of the video frame is occupied by the audience areas which expand across the frames horizontally, but not large enough to occupy the lower part of the frame. In addition, there is a length difference between the left edge and the right edge of the audience area as shown in Figure 5.22(b). Figure 5.22(a) shows a video frame belonging to a corner kick shot (left-corner facing view). Figure 5.22(b) shows its corresponding segmentation mask map in which the black areas are identified as the play field according to the grass area detection method described above, while the gray areas correspond to the players, balls, and audience areas, etc.

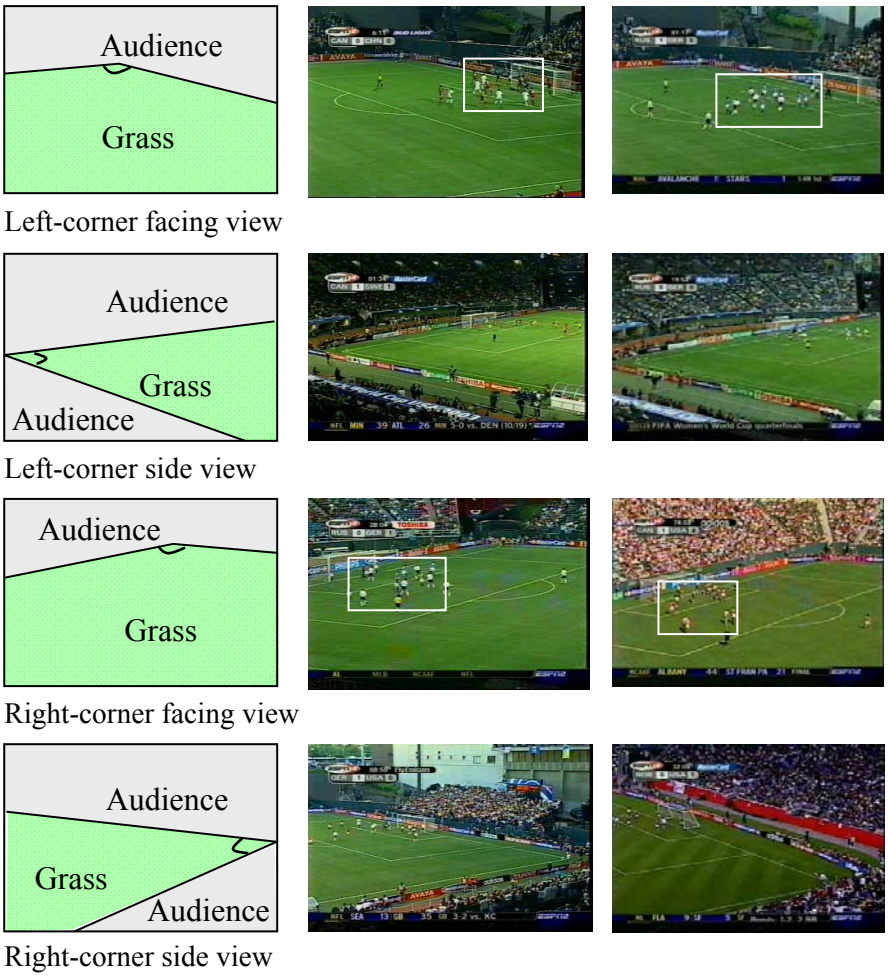


Figure 5.21 The four main camera views for corner kicks.

3) During corner kicks, the majority of the players are likely to stand close to each other within a small area in front of the goal post, which forms a 'player block' with high concentration of the player objects, as shown in both Figure 5.21 (areas marked by white rectangle boxes) and Figure 5.22(a) and (c). Another observation is that the corner point (as shown in Figure 5.22(c)) actually serves as a dividing point, which means the player areas on the right-hand/left-hand side of the corner point are much larger than those on the left-hand/right-hand side in left/right-corner facing view. We use an effective method to obtain the corner point, in which a horizontal sliding window (see Figure 5.22(b)) is used to monitor the changes of non grass areas inside the window. For example, in case

of left-corner facing view as shown in Figure 5.22(b), the window slides from left to right with a fixed window size (e.g., 14% of the frame length). Let the horizontal starting position of a window n represented by wh_n (wh_1 equals 0). As shown in Figure 5.22(b), when there is a significant increase of audience areas in the current window (window **3** in Figure 5.22(b)), the corner point can be roughly located with coordinates (wh_3, |*left edge*|), given the top-left point of the frame as the origin point. Here |*left edge*| denotes the length of '*left edge*' in Figure 5.22(b). If the window size is 40 pixels long, then wh_3=40×(3-1)=80. In order to locate the 'player block', an approximate centroid of the player block is estimated first by using the centroid of all the non-grass areas excluding the audience areas. Then by using another sliding window whose size is dynamically determined by the video frame size, we move it along 8 directions within a limited area around the initial centroid, trying to locate a dense player block.

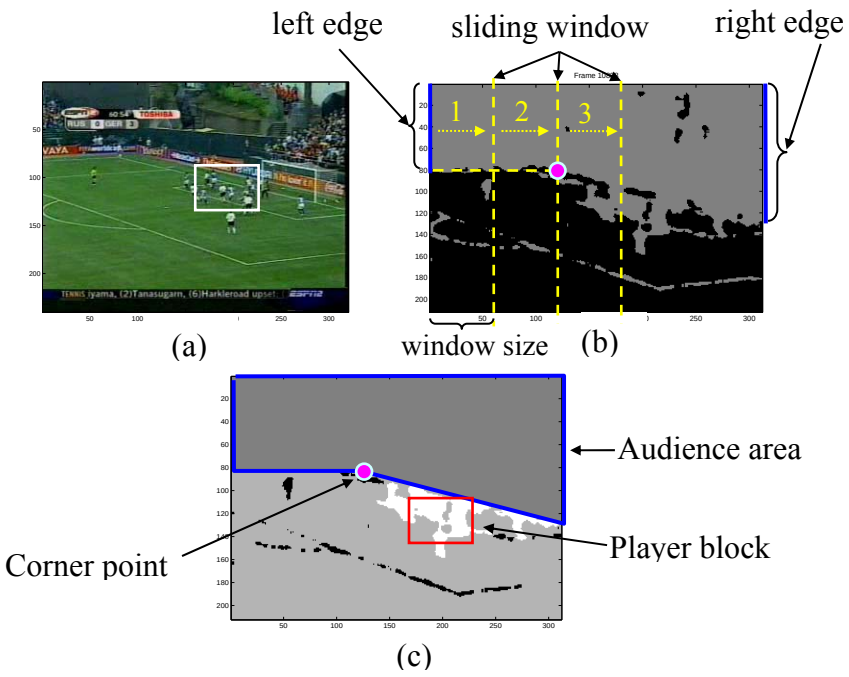


Figure 5.22 (a) The corner kick frame; (b) the segmentation mask map for (a); and (c) the identified audience area, corner point, and player block for (b).

Identify right/left-corner side views:

In the right/left-corner side views, we also have some observations:

1) The right/left corners are visible and located close to the right/left side of a video frame.

2) The audience areas exist in both the upper part and the lower part of the video frames.

3) The shape of the playing field is close to a full triangle or a 'chopped' triangle. Figure 5.23 shows an example left-corner side view with the shape of the playing field close to a 'chopped' triangle. As shown in Figure 5.23(b), the three vertex points of that playing field can be estimated by analyzing its minimum bounding box. Then a segmentation template is created based on the three vertex points as shown in Figure 5.23 (c). If the difference between the segmentation map and its corresponding segmentation template is not significant, then this shot is regarded as a possible side view corner kick event.

Using the strategies described above, we can locate the candidate corner kick shots effectively and efficiently, which will be fed into the data mining process as both training data set and test data set.

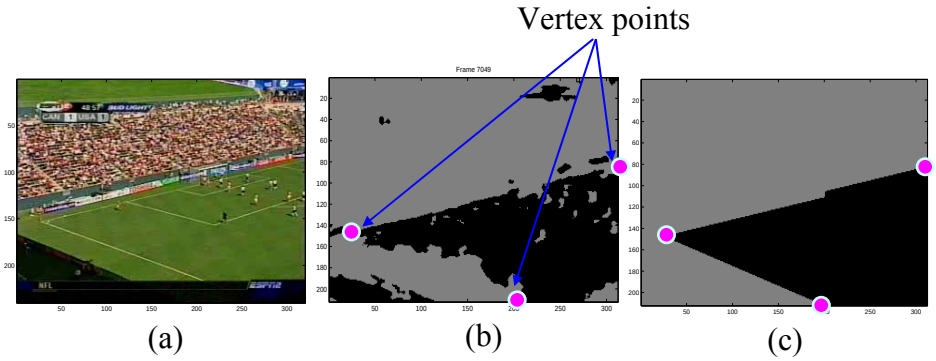


Figure 5.23 (a) The corner kick frame; (b) the segmentation mask map for (a) with 3 vertex points identified; and (c) the segmentation template for (b)

5.3.5 Hierarchical Data Mining

In this study, the decision tree logic is used to mine goal shots in soccer videos adopted to classify three target events: soccer goal events, corner kick events and corner-goal events, where the corner-goal events refer to those goal events which are the direct result of the corner kick events. Since goal events and corner kick events are more general events than corner-goal events, we first build up two decision tree models for these two events, respectively. Then, a hierarchical data mining framework based on the decision tree techniques is proposed to deal with corner-goal events. As shown in Figure 5.24, a two-level hierarchical data mining framework is proposed for the detection of the corner kicks and corner-goal events in soccer videos. At the first level (Level 1), the cleaned data are used to train a decision tree model for corner kicks detection. At the second level (Level 2), the input data are only those video shots that are identified as corner kicks. At this level, the decision tree model previously developed for soccer goal is used to test the input data and generate the class labels (yes/non) for the corner-goal events.

The algorithm exploited in this study is adopted from C4.5 decision tree [Quinlan93]. A decision tree is constructed by recursively partitioning the training set with respect to certain criteria until all the instances in a partition have the same class label, or no more attributes can be used for further partitioning. An interior node in a decision tree involves testing a particular attribute, and the branches that fork from that node correspond to all possible outcomes of a test. Eventually, a leaf node is formed which carries a class label that indicates the majority class within the final partition. The classification phase works like traversing a path in the tree. Starting from the root, the instance's value of a certain attribute decides which branch to go at each internal node. Whenever a leaf node is reached, its associated class label is assigned to the instance. In the decision tree generation process, the information gain ratio criterion is used to determine the most appropriate attribute for partitioning due to its efficiency and simplicity. Numeric attributes are accommodated by a two-way split, which means one single breakpoint is

located and serves as a threshold to separate the instances into two groups. The voting of the best breakpoint is based on the information gain value.

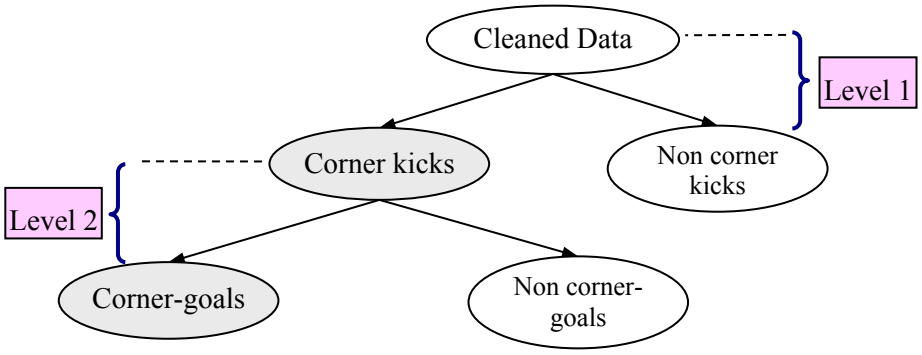


Figure 5.24 Hierarchical data mining framework

The advantage of using the hierarchical data mining for corner-goal events detection is obvious. According to our observation over the 14 video clips with nearly a 6-hour duration, the number of corner-goal events is only about 20% of that of the soccer goals, and around 10% of the total corner kicks. In order to build an independent decision tree model for this kind of events, it is required that the data cleaning phase has to be very strict in selecting the candidates. In that case, the data cleaning rules may become too complicated to be easily understood. Since the corner-goal events can be derived from the other two major events – corner kicks and soccer goals, it is natural to use their data cleaning rules and decision tree models in the detection of corner-goal events. The performance of the proposed data mining framework will be presented in the following sections.

5.3.6 Experimental Results

5.3.6.1 Results for Soccer Goal Detection

In our previous study [Chen04a], we collected 27 soccer video files from a wide range of sources via the Internet, with different styles and produced by different broadcasters. The total duration is 9 hours and 28 minutes. Among the total 4,885 video shots, only 41 are goal shots, which constitute only 0.8% of the total shots. By data cleaning, the candidate shots contain much less

noise and outliers compared to the original data set. The resulting pool size after pre-filtering is **886**. These 886 candidate shots are randomly selected to serve as either the training data (666 shots, about 75% of the total data) or the testing data (the remaining 220 shots). The training data set contains 28 goal shots; while the other 13 goal shots are included in the testing data set.

Construct Decision Tree:

The decision tree is induced by the C4.5 approach based on the training data set. Both visual features (*histo_change*, etc.) and audio features (*volume_mean*, etc.) are used in constructing the decision tree. In addition, we explore another two effective features based on **Audio Rule 1** (specified in Section 5.3.4). First, for each shot, the peak volumes of its last three-second audio track and its following shot's first three-second track (for short, *nextfirst3*) are summated as the feature *volume_sum*. Second, the mean volume of its *nextfirst3* acts as another audio feature *volume_nextfirst3*. Total **25** goal shots and **637** non-goal shots are correctly identified (i.e., labeled as "*yes*" and "*non*", respectively). In other words, only three "*yes*" and one "*non*" instances are misclassified.

Table 5.5 Testing result of goal shot detection

Total	Identified	Missed	Misidentified	Recall	Precision
13	12	1	1	92.3%	92.3%

Table 5.6 Overall performance

Total	Identified	Missed	Misidentified	Recall	Precision
41	37	4	2	90.2%	94.9%

The generated decision tree model is applied against the testing data set which contains 13 goal shots and 207 non-goal shots. The classification result of the goal shot classification on the testing data set using the proposed framework is shown in Table 5.5, where both the precision and recall are 92.3% (12/13). Table 5.6 gives the overall performance when both the training and

testing data sets are used. The recall is 90.2% (37/41), and the precision is 94.9% (37/39). As we can see, the result is rather satisfactory and encouraging.

5.3.6.2 Results for Corner Kick Detection

Table 5.7 shows the detailed information (duration, number of frames, shots etc.) of each video clip used for corner kick detection. In addition, the number of corner kick shots and the goal shots resulting from the corner kicks (namely corner-goal events) is also presented.

As mentioned earlier, video shots are the basic units in our event detection framework. Therefore, it is important to detect the shot boundaries accurately and automatically. For this purpose, the three-level shot detection approach proposed in our previous work [Zhang03] is applied to the soccer video data set. In Table 5.8, the recall and the precision values of the shot boundary detector are given for the whole set. Overall, the algorithm achieves 95.2% precision and 85.2% recall. Some shot boundaries are missed due to the long gradual transition effects in some videos. The major source of the false alarms comes from the special editing effects which combine wipes and spins. The missed shot boundaries do not affect the overall performance in our experiments.

Table 5.7 Information of video clips

Files	Frame#	Shot#	Duration(sec)	Corner Kick#	Corner-Goal#
File1	48,153	346	1,926.07	6	0
File2	51,977	418	2,060.96	5	0
File3	27,624	149	1,104.74	1	0
File4	33,943	137	1,132.58	1	1
File5	48,735	197	1,626.17	3	0
File6	65,677	294	2,191.44	4	1
File7	32,788	125	1,094.04	2	0
File8	63,396	249	2,115.33	8	2
File9	30,440	130	1,015.69	4	0
File10	28,679	141	956.93	4	0
File11	73,138	371	2,440.40	4	0
File12	42,335	197	1,412.59	2	0
File13	20,509	83	820.36	1	0
File14	33,465	142	1,338.17	1	0
Total	***600,859***	***2,979***	***21,235.47***	***46***	***4***

Table 5.8 Shot boundary detection results

Shots	Correct	False Alarm	Recall (%)	Precision (%)
2,979	2,539	128	85.2	95.2

In order to avoid the overfitting problem for training the decision tree classifier, in our experiments, the cross-validation method is adopted for performance evaluation, which allows better estimations of the framework's capability in applying the learned event models to other unseen data.

After data cleaning for corner kicks, the cleaned data set is randomly divided into 3 disjoint subsets with almost equal size. As shown in Table 5.7, there are totally 46 corner kicks in the data set. Therefore, they are divided into subsets 1, 2, and 3, each containing 16, 15, and 15 corner kicks, respectively. During the data mining process, two subsets are randomly selected as the training data set, while each shot record is tagged with label '*yes*' or '*non*' to indicate whether this shot contain the corner kick. The remaining subset is then used as the testing data set. After the decision tree model is constructed based on the training data set, it is then tested by the testing data set. Table 5.9 shows the different combinations of the training and testing data sets. For example, '*Group1*' has subsets 1 and 2 in '*Train1*' (training data set) and subset 3 in '*Test1*' (testing data set). Since there are totally 3 different combinations in constructing the training data set and testing data set (as shown in Table 5.9), the cross-validation is carried out by repeating the model-building and testing processes for all these combinations.

Table 5.9 Combinations of training and testing data sets

	Group1		***Group2***		***Group3***	
	Train1	*Test1*	*Train2*	*Test2*	*Train3*	*Test3*
Subset No.	1,2	3	1,3	2	2,3	1
Corner kick #	31	15	31	15	30	16

Table 5.10 Performance of corner kicks detection

	Corner Kick #	Identified	Missed	Misidentified	Recall	Precision
Train1	31	30	1	1	96.8%	96.8%
Test1	15	14	1	1	93.3%	93.3%
Group1	46	44	2	2	95.7%	95.7%
Train2	31	30	1	1	96.8%	96.8%
Test2	15	13	2	2	86.7%	86.7%
Group2	46	43	3	3	93.5%	93.5%
Train3	30	30	0	1	100%	96.8%
Test3	16	14	2	3	87.5%	82.4%
Group3	46	44	2	4	95.7%	91.7%

In Table 5.10, the recall and precision values of the corner kick detection are given for each combination. As can be seen from this table, in most of the cases, the recall and precision values are very high (>90%). For instance, the decision tree model built on the '*Train1*' data set can correctly identify 30 corner kicks from the training data set and 14 corner kicks from the testing data set. In both cases, only one corner kick is missed and one is misidentified. Here 'Missed' means that the '*yes*' instance is classified as '*non*', and 'Misidentified' represents the reverse case. In another words, only one '*yes*' and one '*non*' instances are misclassified. The overall recall and precision values reach 95.7% in this case. Through the cross-validation, we can conclude that the corner kick event model mined by our framework is robust.

5.3.6.3 Results for Corner-Goal Detection

As discussed above, the corner kicks can be effectively detected by using the proposed framework. Moreover, in our previous work [Chen04a] (See Section 5.3.6.1), the performance of goal shots detection has very high recall (94.9%) and precision values (92.3%). In this section, we will show that our framework can also identify the corner-goal events with reasonably good performance. The idea is intuitive and two different approaches can be adopted. 1) The corner kick detection and goal shot detection can be conducted simultaneously and the corner-goals can be obtained via the intersection operation. 2) On the other hand, a hierarchical mining structure

can be adopted, where the data set is filtered by the corner kick mining (soccer goal mining) and the soccer goal mining (corner kick mining) sequentially. We choose the latter approach because it is more computationally efficient.

In corner-goal detection, for each group of video shots in the corner kick detection process, the shots classified as 'yes' instances (including correctly identified and misidentified corner kick shots in both training and testing data sets) are taken as the testing data set for goal events detection. The videos used in our previous work [Chen04a] are used as the training data set to build the decision tree model for goal detection. We make sure that all the testing videos used are excluded from the training videos.

As shown in Table 5.7, there are totally 4 corner-goals in the data set. Table 5.11 shows the results of corner-goal detection for each of the testing data sets. Note that for '*Group3*', 2 corner kicks are missed in the corner kick detection process, where one of them is also a goal shot. Therefore, the '*Group3*' testing data set only contains 3 corner-goal shots. Since the numbers of corner kicks, goals and corner-goals within a soccer video clip only account for a small percent of the total number of shots, it is not desired to miss any shots which contain the target events. Therefore, at this stage, we intend to include all the shots that contain the target events, even at the cost of introducing a certain amount of false alarms. As shown in Table 5.11, the recall values are 100% in all the cases and the overall results are quite reasonable. The same principle is also applied to the goal shot detection [Chen04a, Chen03b] and the corner kick detection (see Table 5.10).

Table 5.11 Performance of corner-goal detection

Group	Corner-Goal #	Identified	Missed	Misidentified	Recall	Precision
Group1	4	4	0	4	100%	50%
Group2	4	4	0	5	100%	44.4%
Group3	3	3	0	5	100%	37.5%

5.3.7 Conclusions

In this section, we introduced a new hierarchical multimedia data mining framework for soccer event detection. The proposed framework allows effective and efficient mining of corner kicks and corner-goals by using a selective mixture of low-level features, middle-level features, and object-level features. Based on the object-segmentation results (segmentation mask maps) produced during shot detection, some high-level features such as the grass-ratio and audience/player area can be derived and located at a low cost, which are further used in the detection of the target events. The proposed framework mines the target events in a way that the more generic events such as corner kicks are mined at the higher level, while the more specific events like corner-goals are mined at the lower level. This 'narrow-down' event mining strategy is both effective and efficient according to our experimental results. The proposed framework has many implications in video indexing and summarization, video database retrieval, and semantic video browsing, etc.

Chapter 6. Application: Learning-Based Spatio-Temporal Vehicle Tracking and Indexing for Transportation Multimedia Database Systems

In this chapter, we present our experience of applying the multimedia technologies on Intelligent Transportation Systems (ITS) applications. In recent years, Intelligent Transportation Systems (ITS), which integrate advances in telecommunications, information systems, automation, and electronics to enhance the efficiency of existing road networks, have been identified as the new paradigm to address the growing mobility problems, and to alleviate congestion and augment the quality of vehicular flow. One example of ITS technologies is the use of advanced sensor systems for on-line surveillance to gather detailed information on traffic conditions. Traffic video analysis can provide a wide range of useful information to traffic planners. In this context, the object-level indexing of video data can enable vehicle classification, traffic flow analysis, incident detection and analysis at intersections, vehicle tracking for traffic operations, and update of design warrants.

In this chapter, a learning-based automatic framework is proposed to support the multimedia data indexing and querying of spatio-temporal relationships of vehicle objects in a traffic video sequence. The spatio-temporal relationships of vehicle objects are captured via the proposed unsupervised image/video segmentation method and object tracking algorithm, and modeled by using a multimedia augmented transition network (MATN) model and multimedia input strings [Chen01]. An efficient and effective background learning and subtraction technique is employed to eliminate the complex background details in the traffic video frames. It substantially enhances the efficiency of the segmentation process and the accuracy of the segmentation results to enable more accurate video indexing and annotation. In this study, we use four real-life traffic video sequences from several road intersections under different weather conditions in the study experiments. The results show that the proposed framework is effective in automating data collection and access for complex traffic situations.

6.1 Introduction

Over the past decade, Intelligent Transportation Systems (ITS) have been identified as the new paradigm to address the growing mobility problems. ITS integrate advances in telecommunications, information systems, automation, computers, and electronics to enhance the efficiency and safety of existing transportation systems and foster seamless intermodal transportation. The advent of ITS has significantly enhanced the ability to provide timely and relevant information to road users through advanced information systems. With the exponential growth in computational capability and information technology, traffic monitoring and large-scale data collection have been enabled through the use of new sensor technologies. One ITS technology, Advanced Traffic Management Systems (ATMS) [Mahmassani94, Peeta94, Peeta95a, Peeta95b], aims at using advanced sensor systems for on-line surveillance and detailed information gathering on traffic conditions. Another, Advanced Traveler Information Systems (ATIS), provides network-wide routing information to road users. In addition, Advanced Public Transportation Systems (APTS) target mass transportation systems to enable greater operational efficiency and travel convenience. While a whole new generation of methodological and algorithmic constructs are being developed to exploit the powerful capabilities afforded by the ITS technologies, concurrent efforts needed to enable practical implementation are lacking in some crucial aspects. One such aspect with sparse focus is the ability to collect, analyze, and store large-scale multimedia traffic flow data for real-time usage. It implies capabilities to (i) store and catalogue data in an organized manner for easy access, (ii) reconstruct traffic situations through off-line analysis for addressing traffic safety and control, and (iii) automate the process of data indexing and retrieval by obviating the need for human intervention and supervision. While each of these capabilities significantly enhances operational feasibility, the last capability has critical implications for real-time implementation in terms of substantially reducing computational time for the associated control procedures. One key application domain that addresses these three

capabilities is the ability to track video sequences both in time and space for easy and unsupervised access.

Image processing and object tracking techniques have previously been applied to traffic video analysis to address queue detection, vehicle classification, and vehicle counting. In particular, vehicle classification and vehicle tracking have been extensively investigated [Stauffer99, URL2, URL3, URL4, Smith95, Cohen99, Kamijo00]. [Stauffer99, Smith95] employed optical-flow analysis, while [Kamijo00] used spatio-temporal Markov random field (MRF) for vehicle tracking with the occlusion effect among vehicles. Though several approaches have been proposed for vehicle identification and tracking, to the best of our knowledge, none of them connect to databases or have limited capabilities to index and store the collected data. Therefore, they cannot provide organized, unsupervised, easily accessible, and easy-to-use multimedia information. Hence, there is a critical need to index the data efficiently in traffic multimedia databases for transportation operations.

In this chapter, our emphasis is on automatic traffic video indexing for capturing the spatio-temporal relationships of vehicle objects so that they can be catalogued for efficient access by using a multimedia database system. Issues associated with extracting traffic movement and accident information from real-time video sequences are discussed in [Cucchiara00, Dailey00, Huang98a, Kamijo99, Koller94]. Two common themes exist in these studies. First, the moving objects (vehicles) are extracted from the video sequences. Next, the behavior of these objects is tracked for immediate decision-making purposes. However, these efforts do not have capabilities to (i) index the data for on-line analysis, storage or off-line pattern querying, and (ii) automate data processing. To enable the indexing of information at the object-level for video data, an intelligent framework should have the ability to segment the area of video frames into different regions, where each of them or a group of them represents a semantic object (such as a vehicle object in traffic video) that is meaningful to users [Courtney97, Fan00, Ferman97]. While most

previous studies are based on some low-level global features such as color histograms and texture features, the unsupervised video segmentation method (SPCPE) used in our framework focuses on obtaining object-level segmentation, objects in each frame, and their traces across frames. Thus, the spatio-temporal relationships of objects can be further elicited by applying object tracking techniques. Then, the multimedia augmented transition network (MATN) model and multimedia input strings [Chen99, Chen01] are used to capture and model the temporal and spatial relations of vehicle objects. We have addressed unsupervised image segmentation and object tracking techniques, and applied these techniques to some application domains such as traffic monitoring [Chen01c, Chen01a, Chen00]. In this chapter, a learning-based object tracking and indexing framework is proposed to improve the vehicle identification process for object tracking and indexing for transportation multimedia database systems.

For traffic intersection monitoring, digital cameras are fixed and installed above the area of the intersection. A classic technique to resolve the moving objects (vehicles) is background subtraction [Gonzalez93]. This involves the creation of a background model that is subtracted from the input images to create a difference image. Ideally, the difference image contains only the moving objects (vehicles). Various approaches to background subtraction and modeling techniques have been discussed in the literature [Dailey00, Grimson98, Haritaoglu98, Stauffer99]. They range from modeling the intensity variations of a pixel by a mixture of Gaussian distributions, to simple differencing of successive images. [Toyama99] provides some simple guidelines for the evaluation of various background modeling techniques. There are two key problems in this context: 1) a complex learning model is highly time-consuming, and 2) a simple differencing technique cannot guarantee good segmentation performance.

To overcome these problems, an effective background learning algorithm is proposed in our learning-based object tracking and indexing framework. By incorporating the background learning algorithm with the unsupervised segmentation method, the initial inaccurate background

information can be refined and adjusted in a self-adaptive way throughout the segmentation. That is, the background learning process and the segmentation process will benefit each other symbiotically in an iterative way as the processes go further. Experiments are conducted using a real-life traffic video sequence from a road intersection. The results indicate that almost all moving vehicle objects can be successfully identified at a very early stage of the processing; thereby ensuring that accurate spatio-temporal information of objects can be obtained through object tracking.

After the vehicle objects are successfully identified, the MATN model and multimedia input strings are proposed to represent and model their spatio-temporal relations [Chen99, Chen01]. The capability to represent the spatio-temporal relations of the objects is critical from several perspectives.

- First, the ability to store multimedia data efficiently provides significant insights on traffic data collection and monitoring vis-à-vis exploiting recent advances in sensor systems. This is especially important in the context of real-time data access for large-scale traffic system operation and control.
- Second, an ability to obtain and store spatio-temporal relationships provides powerful capabilities to analyze and/or address problems characterized by time-dependency. For example, such a capability can significantly aid the off-line analysis of traffic accidents to isolate their causes and identify potential design issues or operational conflicts.
- Third, it can significantly reduce manual effort and intervention by automating the data collection and processing. For example, it can aid in revising traffic warrants without the need for supervised analysis, which is a significant improvement over current labor-intensive approaches involving the painstaking manual examination of the video data collected. This is especially critical when huge amounts of time-dependent traffic data are collected.

- Fourth, the ability to store data from different media on the same traffic situation in an automatic and efficient manner simplifies data access and fusion. This has significant real-time implications as data storage and processing can constitute a substantial part of the real-time implementation of traffic control strategies.

The next section introduces the proposed learning-based object tracking and indexing framework. The experiments, results, and the analysis of the proposed multimedia traffic video data indexing framework are also discussed. A real-life traffic video sequence is used for the experiments. Conclusions and future work are presented as well.

6.2 Learning-Based Object Tracking and Indexing for Traffic Video Sequences

Traffic video analysis at intersections can provide a rich array of useful information such as vehicle identification, queue detection, vehicle classification, traffic volume, and incident detection. To the best of our knowledge, traffic operations currently either do not connect to databases or have limited capabilities to index and store the collected data (such as traffic videos) in their databases. Therefore, they cannot provide organized, unsupervised, conveniently accessible and easy-to-use multimedia information to the end users. The proposed learning-based object tracking and indexing framework includes background learning and subtraction, vehicle object identification and tracking, the MATN model, and multimedia input strings. The additional level of sophistication enabled by the proposed framework in terms of spatio-temporal tracking generates a capability for automation. This capability alone can significantly influence and enhance current data processing and implementation strategies for several problems vis-à-vis traffic operations.

In the proposed framework, the unsupervised segmentation method called the Simultaneous Partition and Class Parameter Estimation (SPCPE) algorithm is applied to identify the vehicle objects in the video sequence [Chen00a, Sista00]. In addition, the technique of background subtraction is introduced to enhance the basic SPCPE algorithm to help get better segmentation

results, so that more accurate spatio-temporal relationships of objects can be obtained. After the spatio-temporal relationships of the vehicle objects are captured, the MATNs and multimedia input strings are used to represent and model their temporal and relative spatial relations. In the following subsections, we will first discuss the motivation for the proposed framework. Then, the object tracking techniques will be discussed, followed by an introduction to the background subtraction technique. Then, we will briefly describe how to use the MATNs and multimedia input strings to model traffic video frames. Two example video frames are used to demonstrate how video indexing is modeled through the MATNs and multimedia input strings.

6.2.1 Motivation

Image segmentation techniques have been used previously to extract the semantic objects from images or video frames, but in most cases the non-semantic content (or background) in the images or video frames is very complex. For example, at a traffic intersection, there are non-semantic objects such as the road pavement, trees, and pavement markings/signage in addition to the semantic objects (vehicles), which introduces complications for the segmentation methods. Therefore, an effective way to obtain background information can enable better segmentation results. This leads to the idea of background subtraction. Background subtraction is the technique for removing non-moving components from a video sequence. The main assumption for its application is that the camera remains stationary, and the basic principle is to create a reference frame of the stationary components in the image. Once created, the reference frame is subtracted from any subsequent images. The pixels resulting from new (moving) objects will generate a difference not equal to zero.

The traditional way to eliminate background details is to manually select video sequences containing no moving objects and then average them together. This is done through the construction of a reference background frame by accumulating and averaging images of the target area (e.g., a road intersection) for some time interval [Haritaoglu00, Kamijo99]. However, the

determination of the time interval is subjective, and is based on experience or estimation from experimental results. For the traditional method to work well, one key condition is that the video sequence should have approximately constant lighting conditions. Hence, it is not a robust technique as it is sensitive to intensity variations, as lighting conditions are not controlled [Haritaoglu98]. That is, it can generate false positives by incorrectly detecting moving objects solely due to lighting changes. It can also generate false negatives by adding static objects to the scene that are not part of the reference background frame. In the proposed framework, instead of manually selecting the suitable frames to generate one reference background image at a time, an adaptive background learning method is used to achieve this goal. The idea is to first use the unsupervised segmentation method together with the object tracking technique to distinguish the static objects from the mobile objects. Then, these static objects are grouped with the already identified background area to form a new estimation of the background.

The basic workflow of the proposed framework is shown in Figure 6.1. In the first step, a background learning method is applied on the first few video frames (for example, the first four frames) to obtain the initial reference background image. By applying the unsupervised segmentation method, the static and mobile objects are roughly determined, and then the static objects are grouped with the already identified background to form the initial reference background image. The second step involves the subtraction of the initial reference background image from the current frame to generate the difference image whose background is much cleaner compared to the original frame. Then, the unsupervised segmentation method is applied on the difference image to get the segmentation results. Using them, a new reference background image is generated in a self-adaptive way. The details are described in the following subsections. After the vehicle objects (such as cars and buses) are successfully identified, their relative spatio-temporal relationships are further indexed and modeled by the MATN model together with multimedia input strings. The proposed segmentation method can identify vehicle objects but

cannot differentiate between them (into cars, buses, etc.). Therefore, *a priori* knowledge (size, length, etc.) of different vehicle classes should be provided to enable such classification. In addition, since the vehicle objects of interest are the moving ones, stopped vehicles will be considered as static objects and will not be identified as mobile objects until they start moving again. However, the object tracking technique ensures that such vehicles are seamlessly tracked though they "disappear" for some duration due to the background subtraction. This aspect is especially critical under congested or queued traffic conditions.

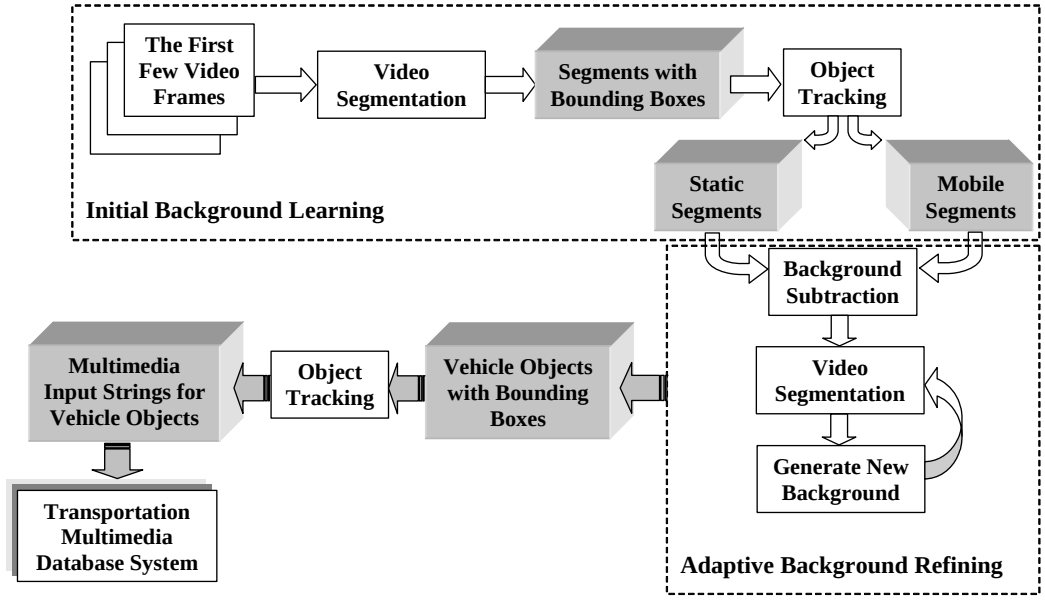


Figure 6.1 The basic workflow of the proposed framework.

In a traffic video monitoring sequence, when a vehicle object moves out of the monitor area (intersection) or stops in the intersection area (including the approaches to the intersection), our framework may deem it as part of the background information. In the former case, tracking is not necessary as the vehicle is out of the monitoring area. Usually, in such a situation, the centroid of its bounding box will be very close to the boundary of the monitoring area. In the latter case, since the vehicle objects move into the intersection area before stopping, they are identified as

moving vehicles before their stop due to the characteristics of our framework. In this situation, their centroids identified before they stop will be in the intersection area. For these vehicles, the tracking process is frozen until they start moving again and they are identified as "waiting" rather than "disappearing" objects. That is, the tracking process will follow the same procedure as before unless one or more new objects abruptly appear in the intersection area. Then, the matching and tracking of the previous "waiting" objects will be triggered to continue tracking the trails of these vehicles.

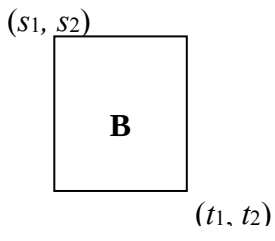
6.2.2 Object Tracking

In order to index the vehicle objects, the proposed framework must have the ability to track the moving vehicle objects (segments) within successive video frames [Chen01c]. Since the tracking trail information can be obtained for each segment, it is possible to distinguish the static objects from the mobile objects in the frame, enabling the estimation of the background information. Furthermore, this tracking technique can be used to determine the trace tubes (trails) for vehicle objects, which enable the proposed framework to provide useful and accurate traffic information for ATIS and ATMS.

6.2.2.1 Identifying Static and Mobile Objects Using Object Tracking

After video segmentation, the segments (objects) with their bounding boxes and centroids are extracted from each frame. Intuitively, two segments that are spatially the closest in adjacent frames are connected. Euclidean distance is used to measure the distance between their centroids.

Definition 7: A bounding box B (of dimension 2) is defined by the two endpoints S and T of its major diagonal [Gonzalez93]:



B = (S, T), where S = (s_1, s_2) and T = (t_1, t_2) and $s_i \leq t_i$ for i = 1, 2.

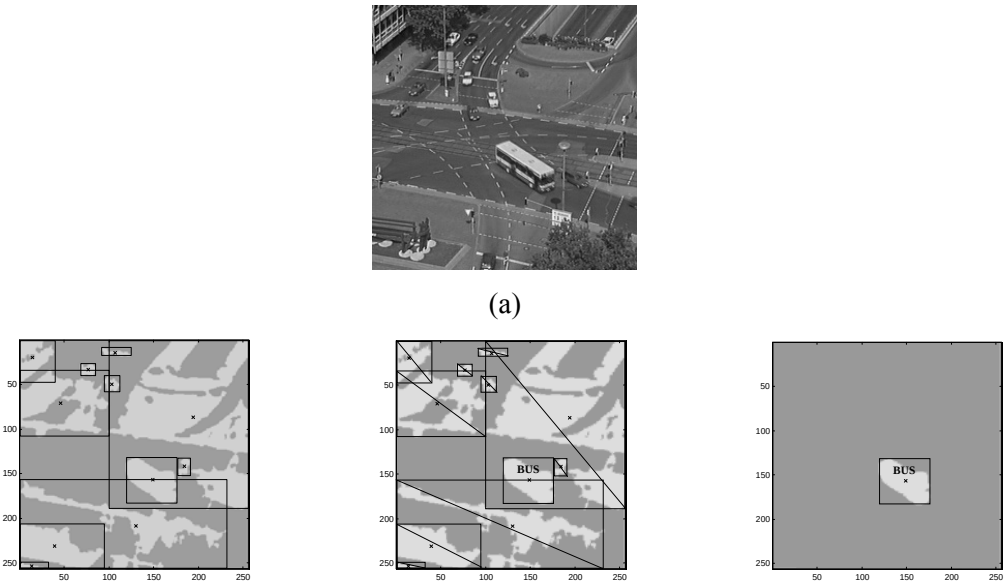
Due to Definition 7, the area of B*:* $Area_B = (t_1 - s_1) \times (t_2 - s_2)$.

Definition 8: The centroid ctd_O of a bounding box B corresponding to an object O is defined as follows:

$$ctd_O = [ctd_{O1}, ctd_{O2}], \text{ where } ctd_{O1} = (\sum_{i=1}^{No} O_{xi}) / No \; ; \; ctd_{O2} = (\sum_{i=1}^{No} O_{yi}) / No \; ;$$

where No is the number of pixels belonging to object O within bounding box B, O_{xi} represents the x-coordinate of the *ith* pixel in object O, and O_{yi} represents the y-coordinate of the *ith* pixel in object O.

Let ctd_M and ctd_N, respectively, be the centroids of segments M and N that exist in consecutive frames, and δ be a threshold. The Euclidean distance between them should not exceed the threshold δ if M and N represent the same object in consecutive frames:



(a) (b) (c) (d)

Figure 6.2 (a) the original video frame 3; (b) the segmentation result along with the bounding boxes and centroids for (a); (c) the segments with diagonals are identified as 'static segments'; and (d) the final segmentation result for frame 3 after filtering the 'static segments'.

In addition to the use of the Euclidean distance, some size restriction is applied to the process of object tracking. If two segments in successive frames represent the same object, the difference between their sizes should not be large. The details of object tracking can be found in

[Chen00]. Figure 6.2 illustrates the segmentation result for frame 3, where the dark gray area represents the background and the light gray segments (class 2) are supposed to correspond to the vehicle objects we want to extract. However, there are several segments that do not correspond to moving vehicles. For example, part of the road pavement, road lamps, and trees are also identified as objects though they are not vehicle objects of interest.

Since the location of the camera monitoring the intersection area is fixed above the ground, the centroids of static segments (road pavement, trees, etc.) should remain fixed throughout the video sequence. This is how 'static segments' are differentiated from 'mobile segments' (moving vehicles) in this application domain. Figure 6.2(c) shows the 'static segments' identified in frame 3. However, there is problem related to vehicles that move very slowly; they may be identified as static segments and thus become part of the background information learned till the current frame. For example, as shown in Figure 6.2(d), except the bus object in the middle of the intersection area, the other 10 cars (eight of them are located in the upper left part, one white car is located in the upper right part, and a gray car is in front of the bus, as seen in Figure 6.2(a)) are identified as static segments and are merged with the already identified background area (the dark gray area) even though they are moving slowly. As mentioned earlier, based on the object tracking results, an initial reference background image can be generated. However, the initial background area information obtained here is not very accurate because many slow moving vehicles are identified as part of the background. Later in this section we will show that it is not necessary to obtain very accurate initial background information in order to achieve good segmentation results. By applying a self-adaptive background adjusting and subtraction method, the proposed framework can robustly capture the spatio-temporal relationships of vehicle objects in real-life traffic video sequences.

6.2.3 *Handling Occlusion Situations in Object Tracking*

As mentioned earlier, in most cases, the complexities associated with the background preclude good segmentation results and complicate the solution for object occlusion situations. However, by applying the background subtraction method which will be discussed later, substantially simpler difference images are obtained. This enables fast and satisfactory segmentation results, greatly benefiting the handling of object occlusion situations. A more sophisticated object tracking algorithm integrated in the proposed framework is given in [Chen00, Chen01d], which can handle the situation of two objects overlapping under certain assumptions (e.g., the two overlapped objects should have similar sizes). It considers the situation when overlapping happens between two objects that separate from each other in a later/earlier frame. In this case, it can find the split object and use the information in the current frame to update the previous frames in a backtrack-chain manner. As will be shown in next subsection, this algorithm is effective in handling two-object occlusions.

However, there are cases where a large object overlaps with a small one. For example, as shown in Figure 6.3, the large bus merges with the small gray car to form a new big segment in frame 20 though they are two separate segments in frame 19. In this scenario, the car object and the bus object that were separate in frame 19 cannot find their corresponding segments in frame 20 by centroid-matching and size restriction. However, from the new big segment in frame 20, we can reason that this is an "*overlapping*" segment that actually includes more than one vehicle object. For this purpose, a difference binary map reasoning method is proposed in this paper to identify which objects the "*overlapping*" segment may include. The idea is to obtain the difference binary map by subtracting the segment result of frame 19 from that of frame 20 and check the amount of difference between the segmentation results of the consecutive frames. As shown in the difference binary map in Figure 6.3, the white areas in it indicate the amount of difference between the segmentation results of the two consecutive frames. The car and bus

objects in frame 19 can now be roughly mapped into the area of the big segment in frame 20 with relatively small differences. Hence, the vehicle objects in the big segment in frame 20 can be obtained by reasoning that this segment is most probably related to the car and bus objects in frame 19. For the big segment (the "*overlapping*" segment) in frame 20, therefore, the corresponding links to the car and bus objects in frame 19 can be created, which means that the relative motion vectors of that big segment in the following frames will be automatically appended to the trace tubes of the bus and car objects in frame 19.

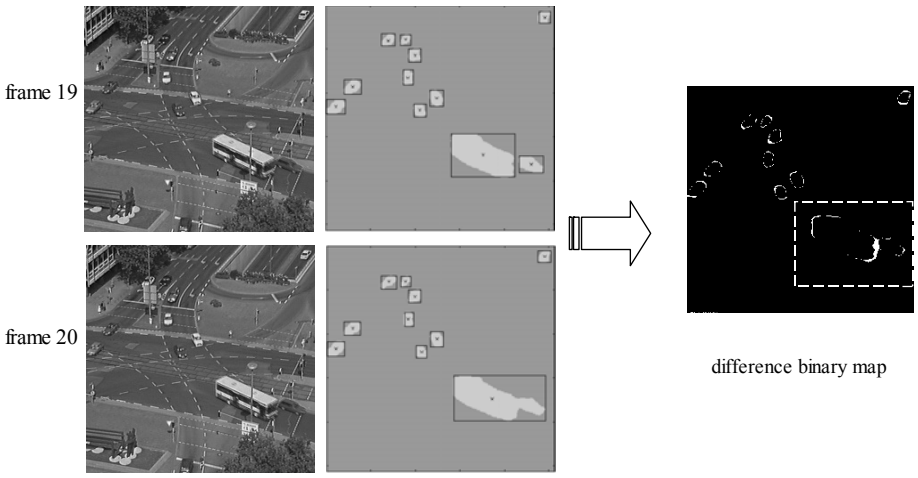


Figure 6.3 Handling object occlusion in object tracking.

6.2.3.1 Enhanced Object Tracking: *Backtrack-Chain-Updation Split Algorithm*

In this subsection, an enhanced object tracking method called *backtrack-chain-updation split algorithm* [Chen00, Chen01d] is introduced to help handle the object occlusion situation. The proposed *backtrack-chain-updation split algorithm* can find the split segment (object) and use the information in the current frame to update the previous frames in a backtrack-chain manner. This enhanced object tracking method is able to distinguish two separate objects (with similar sizes) that were overlapped previously.

Let us think about what happens when overlapped segments separate from each other in successive frames. When the split happens, some segment with overlapping in the previous frame

may not find its corresponding part in the current frame since either the centroid or the size changes considerably. As a result, there may be some segments in the current frame that cannot be tracked back to the segments in the previous frame. We call these segments the *unidentified segments*. Then, we try to build up the relationship between those in the previous frame and those in the current frame based on the information (i.e., the size and position information of those *unidentified segments*) we get. In our algorithm, the concept of MINDIST in [Roussopoulos95] is adopted.

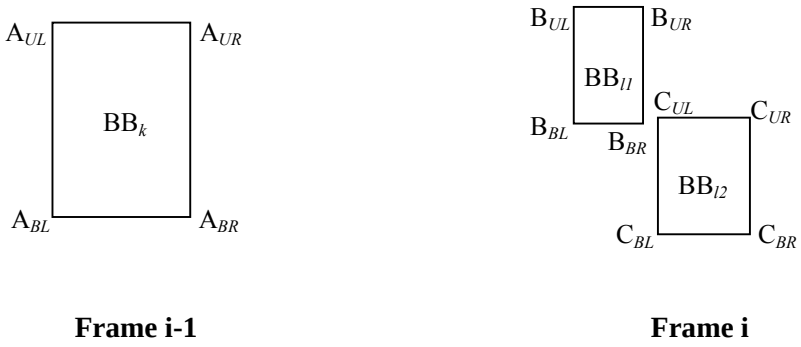
Definition 9: The distance of a point P = [p_1, p_2] from a bounding box B (see Definition 7) in the same space, denoted *MINDIST*(P, B), is defined as follows.

$$MINDIST(P,B) = \sum_{i=1}^{2} \left| p_i - r_i \right|^2 \text{ , where}$$

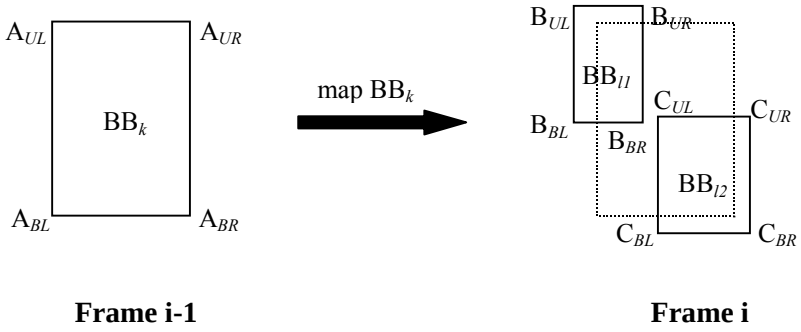
$$r_i = \begin{cases} s_i & \text{if } p_i < s_i \\ t_i & \text{if } p_i > t_i \\ p_i & \text{otherwise} \end{cases}$$

The *MINDIST* is a variation of the classic Euclidean distance applied to a point and a rectangle. When the point is inside the rectangle, the distance between them is zero. Whereas, when the point is outside the rectangle, the square of the Euclidean distance between the point and the nearest edge of the rectangle is used. The square of the Euclidean distance is used since fewer and less costly computations are involved.

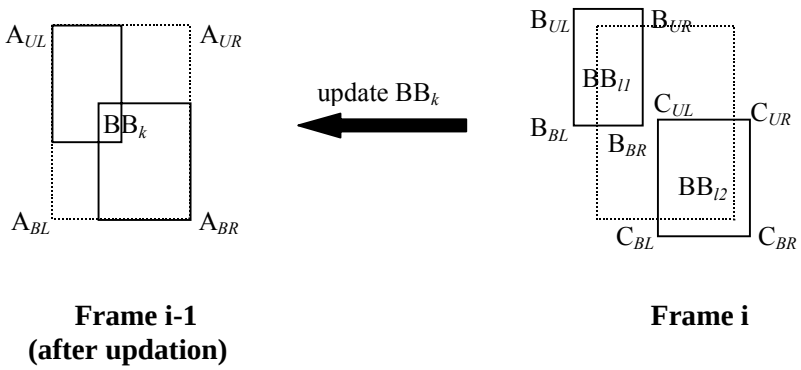
Let ctr_k and BB_k be the centroid and bounding box of segment k in frame i-1, and ctr_l and BB_l be the centroid and bounding box of segment l in current frame i. The backtrack-chain-updation split algorithm is given as follows. Figure 6.4(a)-(c) are used to illustrate the algorithm.



(a): Bounding boxes of segment *k* in frame *i* –1 and segments *l1, l2* in frame *i.*



(b): Find the split segments in frame *i.*



(c): Find the recovery vertices for BB_{l1} and BB_{l2} and update BB_k in frame *i-1*.

Figure 6.4 The basic workflow of the backtrack-chain-updation algorithm.

Step 1: *Identify as many as possible of the related segments.*

If $dist(ctr_k - ctr_l) = \| ctr_k - ctr_l \|_2 \leq \delta$ **AND** $|size\ (BB_k) - size\ (BB_l)| \leq \beta$, where δ and β are threshold values for distances and sizes, then segment *l* in frame *i* is related to segment *k* in frame *i*-1. Mark segments *l* and *k* as "Identified." Let segment *k* be the "parent" of segment *l*, and let segment *l* be the "child" of segment *k*.

Step 2: *Find the split segments in the current frame.*

Select one segment that is not identified in frame *i*-1. Let its centroid and bounding box be ctr_k and BB_k. Based on the size of BB_k, find all the unidentified segments in frame *i* whose bounding box BB_l overlaps with BB_k and $\beta_{Min} < (size(BB_k)/size(BB_l)) < \beta_{Max}$. Note that we apply a size restriction to avoid the interference of some small segments ("noise"). Select the first two segments (say BB_{l1} and BB_{l2}) whose $MINDIST(ctr_k, BB_{l1})$ and $MINDIST(ctr_k, BB_{l2})$ are the smallest and the second smallest, and mark them as "Identified." Then build up a *parent-child* relationship between BB_k and BB_{l1}, and between BB_k and BB_{l2}.

Figure 6.4(b) shows how to map BB_k into frame *i* and to find all the unidentified segments in frame *i* whose bounding boxes overlap with BB_k. BB_{l1} and BB_{l2} in frame *i* overlap with the boundary of BB_k; they are selected as the children segments of segment *k* in frame *i*-1. Now there is a parent segment with bounding box BB_k in frame *i*-1, and there are two children segments with bounding boxes BB_{l1} and BB_{l2} in frame *i* (as shown in Figure 6.4(a)). The vertices of each of the bounding boxes are given with the subscripts "UR", "UL", "BR", "BL" which represent the upper-right, upper-left, bottom-right, and bottom-left, respectively.

Step 3: *Do segmentation on the next frame and get the parameter for size adjustment in Sep 4.*

Once the split segments are identified, we can use the information we have obtained so far to update the parent segment *k*'s bounding box in frame *i*-1. The main idea is to find the *recovery vertex* in the parent segment's bounding box, then "paste" the children's bounding boxes into the

previous frame without changing their sizes and shapes. Remember, we assume that the sizes and shapes of the same object (segment) do not change much in the consecutive frames, but we do allow some small changes in the length or width of its bounding box. In such cases, sometimes the changes may exceed the updated bounding boxes, which results in unsatisfactory recovery results if no adjustment is applied.

Let the current frame be frame *i*, and the previous one be frame *i*-1. In this step, we do segmentation on frame *i*+1 and build up the parent-child relationship between frame *i* and frame *i*+1. For the split segments we just identified in frame *i*, their children segments may exist in frame *i*+1. If that is true, then the ratios of size changes on length and width for each split segment in frame *i* are calculated. For example, suppose the parent segment in frame *i* is segment *l* (BB_l), and the corresponding child segment in frame *i*+1 is segment *p* (BB_p). *T*hen for segment *l,* its parameters are

$$Width\_Sensitivity_l = \frac{\left| Width_{BB_p} - Width_{BB_l} \right|}{Width_{BB_l}}$$

$$Length\_Sensitivity_l = \frac{\left| Length_{BB_p} - Length_{BB_l} \right|}{Length_{BB_l}}$$

If $Width\_Sensitivity_l > Length\_Sensitivity_l$, we say that segment *l* is more *width-sensitive*. This means the ratio of width changes is more than that of length changes, or the width changes of that segment/object may be more frequent and significant than that of length changes. Otherwise, it is said to be more *length-sensitive*. Another possibility is that we cannot find the corresponding child segment in frame *i*+1 due to object merging or disappearing. In this case,

$$Width\_Sensitivity_l = \frac{Width_{BB_l}}{Length_{BB_l}}$$

$$Length\_Sensitivity_l = \frac{Length_{BB_l}}{Width_{BB_l}}$$

In step 4, we will show how to use this sensitivity parameter for size adjustment during the backtrack-chain-updating process.

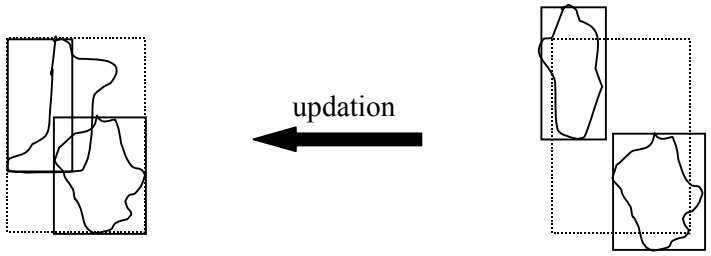
Step 4: *Backtrack and update the previous frames plus size adjustment*

After we find the split segments (i.e., the children segments BB_{l1} and BB_{l2}) in frame i, we can use this information to update the previous frame i-1. The goal is to distinguish the separate bounding boxes on the parent segment k (BB_k) with overlapping. The first step is to check the *MINDIST* from the four vertices of BB_k's to the children's bounding boxes, respectively. For BB_{l1}, the vertex P on BB_k with the minimum $MINDIST(P, BB_{l1})$ is selected as the *recovery vertex* for BB_{l1}. According to this *recovery vertex*, there is a *corresponding vertex* in BB_{l1}. So the next step is to move the *corresponding vertex* to the *recovery vertex*, and to copy the bounding box BB_{l1} within the boundary of BB_k in frame i-1. The same procedures are applied to BB_{l2}.

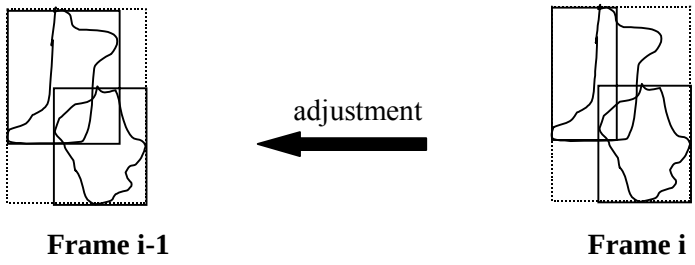
As shown in Figure 6.4(b), BB_k is mapped to frame i and used to compute *MINDIST*. For example, for BB_{l1}, compute $MINDIST(A_{vex}, BB_{l1})$, where *vex* is one of "UR", "UL", "BR", and "BL". Choose the one with minimum *MINDIST* as the *recovery vertex* for BB_{l1}. Here, A_{UL} is chosen as the recovery vertex for BB_{l1}, and B_{UL} is chosen as the *corresponding vertex*. Similarly, for BB_{l2}, A_{BR} and C_{BR} are selected as the recovery vertex and the corresponding vertex. To update the bounding box BB_k in frame i-1, the bounding box BB_{l1} is copied into BB_k with B_{UL} overlapping with A_{UL} and BB_{l2} is copied into BB_k with C_{BR} overlapping with A_{BR}. Notice that all the "copies" should be within the boundary of BB_k. By doing so, the updated version of frame i with separate bounding boxes can be obtained (as shown in Figure 6.4(c)).

In many cases, it seems satisfactory to just copy the children's bounding boxes to their parent's bounding box without any size adjustment. But there are also many situations that require necessary size adjustments to reduce the recovery error and achieve better results. For example, as shown in Figure 6.5(a), we can see separate bounding boxes in frame i-1, but the upper bounding box seems a little narrow due to the length change of that bounding box. This

seems somewhat unsatisfactory for human eyes and for the purpose of video indexing. The parameters obtained in Step 3 will be helpful for size adjustment. In this case, we can decide the upper segment in frame *i*-1 is *length-sensitive*, so what we do is to adjust the length of that bounding box to best fit its shape in frame *i*-1. Figure 6.5(b) shows the result after size adjustment. It looks fairly good.



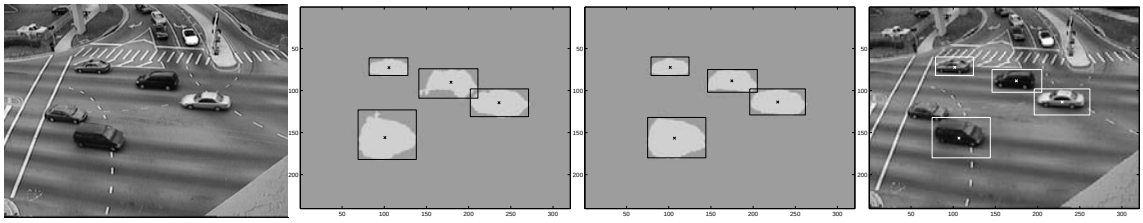
(a) Update frame *i*-1 without size adjustment



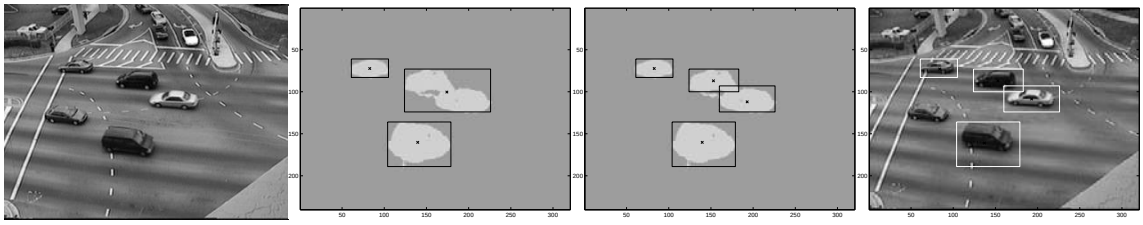
(b) Bounding boxes of frame *i*-1 after size adjustment

Figure 6.5 Size adjustment after updation for frame *i*-1.

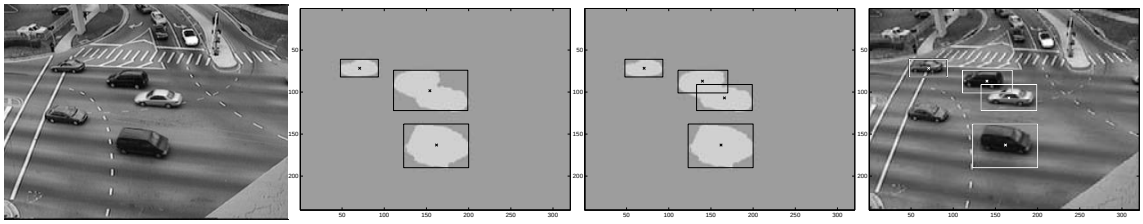
This algorithm, which can be applied to update more previous frames by utilizing the information obtained so far, is called *backtrack-chain-updation*. The experimental results are demonstrated in Figure 6.6(a)-(d), where two vehicles have some overlapping in frames 138 and 142, but are identified as two separate objects in frame 132. Figure 6.6(d) demonstrates the final results by applying the occlusion handling method proposed in [Chen00, Chen01d]. The segmentation results accurately identify all the vehicles objects' bounding boxes and centroids.



frame 132



frame 138



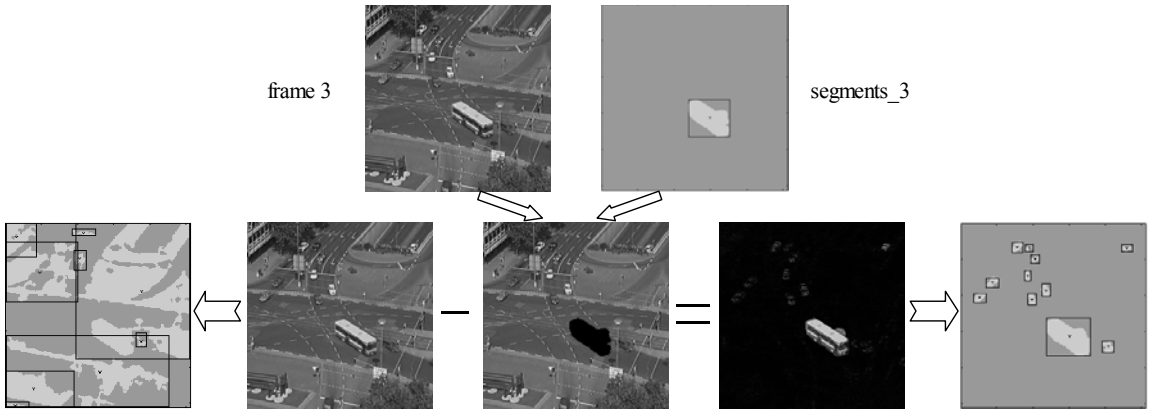
frame 142

(a) **(b)** **(c)** **(d)**

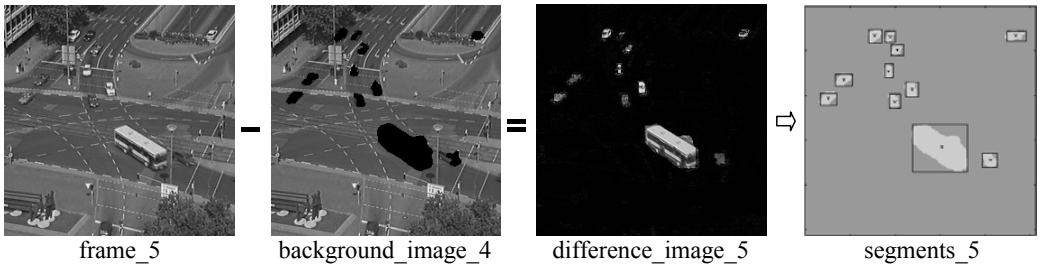
Figure 6.6 Handling two object occlusion in object tracking. (a) Video frames 132, 138, and 142; (b) Segmentation maps for frames in (a) without occlusion handling; (c) Results by applying occlusion handling; (d) The final results by overlaying the bounding boxes in (c) to frames in (a).

6.2.4 Self-Adaptive Background Subtraction

After background learning using the first few video frames, most of the static segments can be identified and subtracted from the set of segments in the current frame (as shown in Figure 6.2(d)). In our experiment, we use the first three frames for initial background learning.



(a) Generating the initial reference background image (background_image_3) and subtracting it from the next frame (frame 4), then applying the segmentation on the obtained difference image (differenc_image_4).



(b) Self-adaptive adjustment in generating new background images.

Figure 6.7 Self-adaptive background learning and subtraction in the traffic video sequence.

As shown in Figure 6.7(a), an initial reference background image (background_image_3) is obtained by extracting the final segmentation result of frame 3 (segments_3) from the original frame 3. Then, this initial background image is subtracted from the next frame (frame 4) to obtain the difference image (difference_image_4). As illustrated in Figure 6.7(a), the difference_image_4 not only stores the visual information for the bus object identified in frame 3, but also shows the motion difference information for each car object that has been identified as part of the background in frame 3. That is, from the segmentation result for difference_image_4, all 11 vehicle objects are successfully identified as separate segments in frame 4 no matter whether they moved fast or slow. Though, in difference_image_4, the visual information

representing the motion difference of the slow moving vehicles is very obscure and much darker when compared with that of the bus object, the SPCPE method can successfully identify all the vehicles in frame 4, which provides a better estimation for a new background image. For comparison purposes, we also show the original segmentation result (original_segments_4) for frame 4 without any background learning and subtraction. There, the bus object merged with the road pavement to form a big segment, and most of other vehicles cannot be identified as separate segments.

A key point here is that if a new background image is always constructed based on the current frame's segmentation result, the construction error will accumulate and finally become unacceptable. This means that the trail of a moving vehicle will also be identified as part of the object, which causes inaccurate or even unacceptable segmentation results after processing a number of frames. When an object moves, a small part of the background area will appear in the current frame though this area was identified as part of the vehicle object in the preceding frame. Without any adjustments, the accumulative construction error will lead to unacceptable segmentation results. In our framework, a simple but effective self-adaptive adjustment is applied in creating a new background image. The adjustment is done by shrinking the size of the bounding box of each segment before constructing a new background image for use in the next frame's segmentation based on the current segmentation results. This adjustment can be thought as the prediction of the changes in the background area. The key aspect of this self-adaptive process is the strong support derived from the robustness of the SPCPE segmentation method. Although the background area is enlarged a little as the result of the shrinking of the bounding boxes, the resulting difference image still includes the new information of the motion difference, if any, that can be identified as part of the moving object by the SPCPE segmentation method. In other words, due to the shrinking of bounding box, the object size will decrease in the newly created background image. However, this will not affect the segmentation result of the next frame

because the motion difference area will still appear in the difference image, and can be identified as part of the vehicle object to compensate the size loss. As a vehicle object moves from the current frame to the next frame, part of the area of the vehicle object identified in the current frame may become part of the background area in the next frame. Without shrinking, this part of the background area may be misidentified as part of the current object. That is, the shrinking of the bounding box is used to achieve the motion prediction without losing any information in segmenting and identifying the moving vehicle objects. Figure 6.7(b) shows the segmentation result for frame 5 by applying the proposed background adjustment method. The segmentation result accurately identifies the bounding boxes of all vehicle objects.

6.2.5 Using MATNs and Multimedia Input Strings

The spatio-temporal relations of the vehicle objects in the video sequence must be captured in an efficient way. In the proposed framework, the spatio-temporal relations are indexed and modeled by using a multimedia augmented transition network (MATN) model and multimedia input strings [Chen99, Chen01]. An MATN can be represented diagrammatically as a labeled directed graph, called a *transition graph*. Multimedia input strings are used as inputs to an MATN, and are proposed to represent the spatio-temporal relations of the vehicle objects in the video sequences. Multimedia input strings adopt the notations from regular expressions [Kleene56]. A multimedia input string is accepted by the grammar if there is a path of transitions which corresponds to the sequence of symbols in the string and which leads from a specified initial state to one of a set of specified final states.

Figure 6.8 illustrates the use of MATNs and multimedia input strings to model the spatio-temporal relations of the vehicle objects in traffic video frames. Assume three objects, the *ground*, *car*, and *bus* represented by ***G***, ***C***, and ***B***, respectively. As introduced in [Chen99, Chen01], one semantic object is chosen as the target semantic object in each video frame. The minimal bounding rectangle (MBR) concept in R-trees [Guttman84] is also used so that each

semantic object is covered by a rectangle. Here, we choose the *ground* as the target object. In order to distinguish the 3-D relative spatial positions, twenty-seven numbers are used [Chen01]. In this example, each frame is divided into nine 2-D sub-regions with the corresponding subscript numbers shown in Figure 6.8(a).

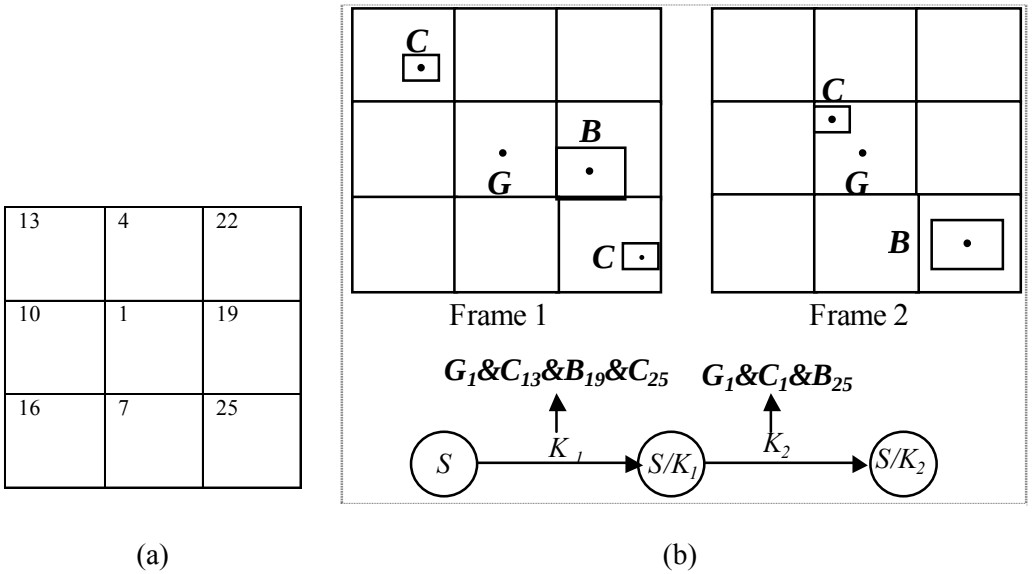


Figure 6.8 MATN and multimedia input strings for modeling the key frames of traffic video shot *S*. (a) the nine sub-regions and their corresponding subscript numbers; (b) an example MATN model.

Each video frame is represented by an input symbol in a multimedia input string. The "&" symbol between two vehicle objects denotes that the vehicle objects appear in the same frame. The subscripted numbers are used to distinguish the relative spatial positions of the vehicle objects relative to the target object "*ground*" (as shown in Figure 6.8(a)). For simplicity, two example frames (frames 1 and 2) are used to explain how to construct the multimedia input strings and the MATN (as shown in Figure 6.8(b)). The multimedia input string to represent these two frames is as follows:

$$\underbrace{(G_1 \,\&\, C_{13} \,\&\, B_{19} \,\&\, C_{25})}_{K_1}\ \underbrace{(G_1 \,\&\, C_1 \,\&\, B_{25})}_{K_2}$$

Two input symbols, K_1 and K_2, are used for this purpose. The appearance sequence of the vehicle objects in an input symbol is based on the relative spatial locations of the vehicle objects in the traffic video frame from left to right and top to bottom. For example, frame 1 is represented by input symbol K_1. $\boldsymbol{G_1}$ indicates that $\boldsymbol{G}$ is the target object. $\boldsymbol{C_{13}}$ implies that the first car object is to the left of and above $\boldsymbol{G}$, $\boldsymbol{B_{19}}$ denotes that the bus object is to the right of $\boldsymbol{G}$, and $\boldsymbol{C_{25}}$ means that the second car is to the right of and below $\boldsymbol{G}$. The multimedia input string for frame 2 is different from that of frame 1 in that the car $\boldsymbol{C_{25}}$ that appeared in frame 1 has already left the road intersection in frame 2, the car $\boldsymbol{C_{13}}$ in frame 1 moved into the same sub-region as $\boldsymbol{G}$ in frame 2 and thus becomes $\boldsymbol{C_1}$, and the bus object $\boldsymbol{B_{19}}$ in frame 1 moved into the lower right corner in frame 2 and becomes $\boldsymbol{B_{25}}$. Hence, the spatial locations of vehicle objects change, and the number of vehicle objects decreases from three to two. This example illustrates the ability of a multimedia input string to represent the spatial relations of objects.

Figure 6.8(b) is the MATN for the two frames in this example. The starting state name for this MATN is *S*/. As shown in Figure 6.8(b), there are two arcs with labels K_1 and K_2. The different state nodes and the order of their appearance in the MATN are based on the temporal relations of the selected video frames. The multimedia input strings model the relative spatial relations of the vehicle objects.

6.3 Experimental Analysis

Four real-life traffic video sequences are used to analyze spatio-temporal vehicle tracking using the proposed learning-based vehicle tracking and indexing framework. These video sequences are obtained from different sources, showing four different road intersections with different qualities and under different weather conditions. Videos #1 to #3 are grayscale videos downloaded from the research website of University Karlsruhe [URL1]. Video #3 was taken in the winter where there was snow on the lane and light snow was falling with a strong wind. The qualities of video

#2 and video #3 are significantly worse than that of video #1. Video #4 is also a grayscale video captured with a common digital camera. The proposed new framework is fully unsupervised in that it can enable the automatic background learning process that greatly facilitates the unsupervised video segmentation process without any human intervention. Based on our experiments, by applying the background learning process, the computation time savings for the segmentation process are eighty percent of the original time cost. As shown in Table 6.1, the overall performance of vehicle object identification over the four video sequences is robust. The precision and recall values are approximately 95 and 90, respectively.

Table 6.1 Overall performance of vehicle object identification.

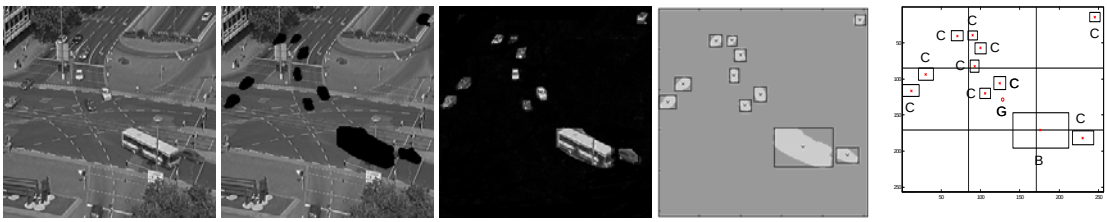
Video #	Number of Frames	Frame Size	Quality	Correct	Missed	False	Precision	Recall
Video #1	50	512×512	Good	64	0	0	100%	100%
Video #2	300	268×251	Medium	83	12	1	99%	87%
Video #3	1733	353×473	Poor	621	66	14	98%	90%
Video #4	3000	240×320	Medium	1611	195	103	94%	89%
Overall	**5083**			**2379**	**273**	**118**	**95%**	**90%**

A portion of the traffic video sequence #1 is used to illustrate how the proposed framework can be applied to address spatio-temporal queries such as "estimate the traffic flow at this road intersection approach from 5:00 PM to 5:30 PM." This requires the proposed framework to elicit information on the number of vehicles passing through that intersection approach in the stated time duration. Further, the collected information must be indexed and stored into a multimedia database in real-time or off-line. These aspects are addressed hereafter.

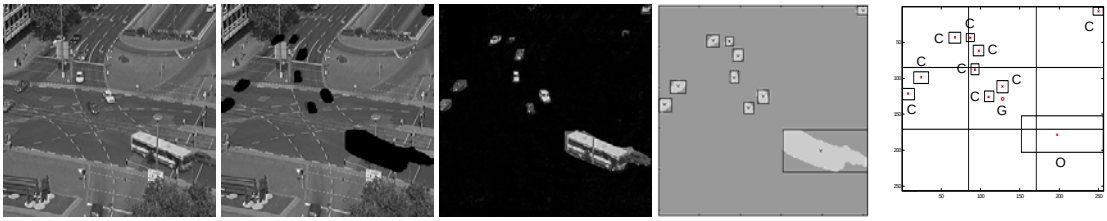
The enhanced video segmentation method is applied to the video frames by considering two classes. Consider a video frame like the one in Figure 6.2(a). There could be several variations in the background such as road pavement, trees, pavement markings/signage, and ground. Since the interest is only in the vehicle objects, it is a two-class problem. The first frame

is partitioned with two classes using random initial partitions. After the partition of the first frame is obtained, the partitions of the subsequent frames are computed by using the previous partitions as the initial partitions since there is no significant difference between consecutive frames. By doing so, the segmentation process will converge fast, thereby providing support for real-time processing. The most time-consuming part at the beginning is the background learning process because enough background information does not exist at that time. Hence, the segmentation process has to deal with the original video frames which include very complex backgrounds. The effectiveness of the proposed background learning process ensures that a long run is not necessary to fully determine the accurate background information. In our experiments, the preliminary background information can be obtained usually within five consecutive frames, and it is good enough for the future segmentation process. In fact, by combining the background learning process with the unsupervised segmentation method, our framework can enable the adaptive learning of background information.

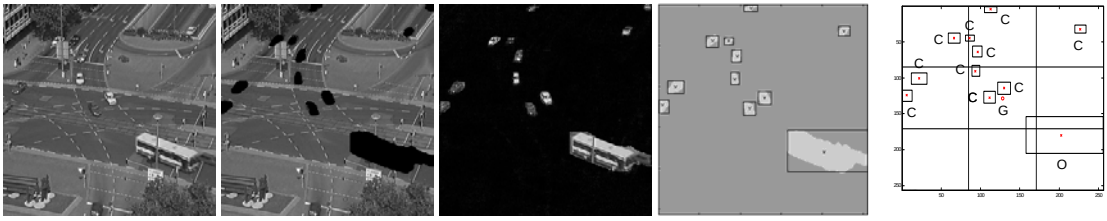
Figure 6.9 shows the segmentation results and the corresponding multimedia input strings for a few frames (19, 25, 28 and 34) along with the original frames, background images, and the difference images. As illustrated by the figure, the background of this traffic video sequence is very complex. Some vehicle objects (for example, the small gray vehicles in the upper left part of the video frames) can easily be ignored or confused with the road surface and surrounding environment. While there is an existing body of literature [Friedman97, Huang98a] that addresses relatively simple backgrounds, our framework can address far more complex situations, as illustrated here.



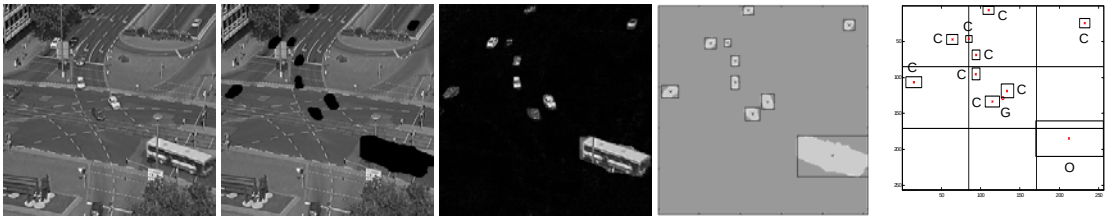
frame 19 background_18 difference_image_19 segments_19 bounding_boxes_19
frame 19 *Multimedia Input String:* ***G1&C10&C10&C13&C4&C4&C4&C1&C1&B25&C25&C22***



frame 25 background_24 difference_image_25 segments_25 bounding_boxes_25
frame 25 *Multimedia Input String:* ***G1&C10&C10&C13&C4&C1&C4&C1&C1&O25&C22***



frame 28 background_27 difference_image_28 segments_28 bounding_boxes_28
frame 28 *Multimedia Input String:* ***G1&C10&C10&C13&C4&C1&C4&C4&C1&C1&O25&C22***



frame 34 background_33 difference_image_34 segments_34 bounding_boxes_34
frame 34 *Multimedia Input String:* ***G1&C10&C13&C13&C1&C4&C4&C1&C1&O25&C22***

(a) (b) (c) (d) (e)

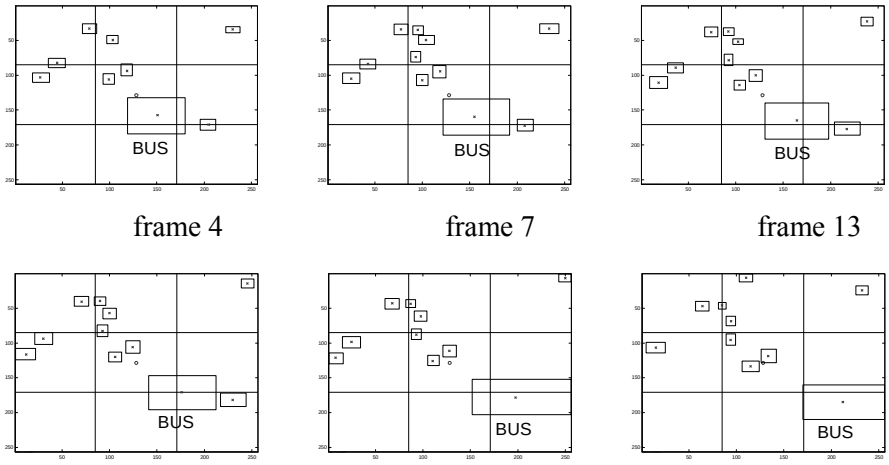
Figure 6.9 Segmentation results and multimedia input strings for frames 19, 25, 28 and 34. (a) the original video frames; (b) the background reference images derived from the immediate preceding frames; (c) the difference images obtained by subtracting the background reference images from the original frames; (d) the vehicle segments extracted from the video frames; (e) the bounding box and centroid for each segment in the current frame.

In Figure 6.9, the video frames in the leftmost column (Figure 6.9(a)) represent the original frames. The second column (Figure 6.9(b)) shows the background images derived from the immediate preceding frames. The third column (Figure 6.9(c)) shows the difference images after background subtraction. The segmentation results are illustrated in the fourth column (Figure 6.9(d)), and the rightmost column (Figure 6.9(e)) shows the bounding box and centroid for each segment in the current frame. As illustrated by Figure 6.9(d), a single class can capture almost all vehicle objects, even those vehicles that look small and obscure in the upper left area of the video frames. Another class captures most part of the ground. Also, Figure 6.9(d) shows that almost all vehicle objects are captured as separate segments. However, the bus and the gray car (in the lower right part of the intersection area) are identified as one big segment in frames 25, 28, and 34; while they are identified as separate segments in frame 19. As discussed earlier, this occlusion situation can be detected by the proposed difference binary map method. In our indexing schema for a multimedia database, we use a special symbol to denote such an "*overlapping*" segment that has the corresponding links to the related vehicle segments in the preceding frame. Excluding this overlapping segment, other vehicle objects are successfully identified in all video frames.

Figure 6.9 also lists the multimedia input strings for the selected frames. As discussed earlier, we use symbolic representations (multimedia input strings) to represent the spatial relationships of the objects in each frame. In Figure 6.9(e), the ground (***G***) is selected as the target object, the segments are denoted by ***C*** for cars or ***B*** for buses, and the "*overlapping*" objects are denoted by symbol ***O*** which has the corresponding links to the related segments in the preceding frame. As shown in Figure 6.9, there are 11 vehicle objects visible in frame 19 -- two gray cars ($\boldsymbol{C_{10}}$&$\boldsymbol{C_{10}}$) are in the left middle area, one white car is located in the upper left area ($\boldsymbol{C_{13}}$), three cars are in the upper middle area ($\boldsymbol{C_4}$&$\boldsymbol{C_4}$&$\boldsymbol{C_4}$), two cars are located in the middle area ($\boldsymbol{C_1}$&$\boldsymbol{C_1}$), one bus ($\boldsymbol{B_{25}}$) and one dark gray car ($\boldsymbol{C_{25}}$) are in the lower right corner, and another white car

driving towards northeast is located in the upper right area ($\boldsymbol{C_{22}}$). Frame 25 indicates that the white car (the second $\boldsymbol{C_4}$ in frame 19) is moving slowly into the middle area so that its symbol changes to $\boldsymbol{C_1}$, and the bus and dark gray car ($\boldsymbol{B_{25}}$ and $\boldsymbol{C_{25}}$ in frame 19) are identified as an "*overlapping*" object $\boldsymbol{O_{25}}$ in frame 25. In frame 28, the white car ($\boldsymbol{C_{22}}$ in frames 19 and 25) in the upper right corner disappeared from the intersection area while another white car appears ($\boldsymbol{C_{22}}$) in the upper right corner from the underneath tunnel. Also in frame 28, we can see part of a new car object entering into the intersection area from the upper bound, which is successfully identified as a new segment (the third $\boldsymbol{C_4}$ in frame 28) even though the small area it occupies in frame 28 is part of the background in the preceding frames. And in frame 34, one gray car (heading southwest, denoted by the first $\boldsymbol{C_{10}}$ in the preceding frames) disappears from the intersection area. A point to note here is that though the white car (located in the upper left part) that is slowly heading southeast has part of its body becoming progressively invisible due to the rectangular poster in front of it in the frames, our framework can successfully identify it as a complete segment (denoted by $\boldsymbol{C_{13}}$ in all the selected frames) throughout the entire video sequence. As illustrated by Figure 6.9, the multimedia input strings can model not only the number of vehicle objects, but also the relative spatial relations of the vehicle objects.

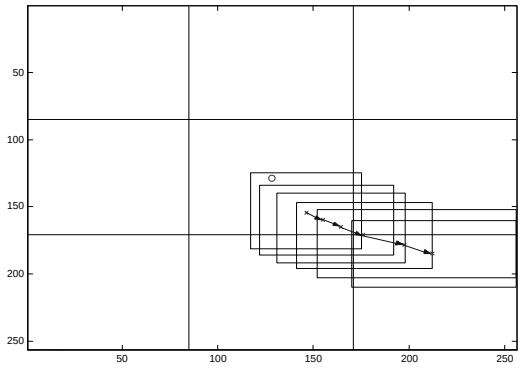
As mentioned in Section 6.2, we apply the object tracking technique to track the trail of each vehicle object to the extent possible. Figure 6.10 shows the tracking of the trail of the bus object in the video sequence. Figure 6.10(a) shows the bounding boxes and centroids of the bus object from frame 4 to frame 34, while Figure 6.10(b) shows the trail information of the bus object by applying the object tracking technique. In fact, in the proposed indexing schema, it is not necessary to record the position of the bus segment in each frame. Instead, it can be done when there is a *major move* in that object or based on a fixed frequency.



frame 4 frame 7 frame 13

frame 19 frame 25 frame 34

(a) bounding boxes and centroids for the bus object in the video sequence



(b) the trail of the bus object from frame 4 to frame 34

Figure 6.10 Tracking the trail of a bus in the traffic video sequence.

As described earlier, the framework can determine not only the indexes for the number of vehicle objects, but also the index information of relative spatial relations by recording the positions of the centroid of the segment throughout the video sequence. However, to address the traffic flow query mentioned at the beginning of this section, it is necessary to have a "judge line" in the frame so that the traffic flow in a specified direction can be estimated. This judge line can be provided by the end user. For example, it could be a line before vehicles go into or out of the intersection area. By using the centroid position information of each vehicle object, the traffic

flow in a specified direction can be roughly estimated. Also, since vehicle classification may be important, the sizes of the bounding boxes are used to determine the vehicle types (such as "car" and "bus"). For "*overlapping*" segments, their links to specific vehicle segments can be used to correctly account for their contribution to the traffic flow. While the traffic flow query mentioned earlier is used as an example here, the proposed framework has the potential to address other, and more complex, spatio-temporal related database queries. For example, it can be used to reconstruct accidents at intersections in an automated manner to identify causal factors to enhance safety.

6.4 Insights

The experimental results demonstrate the effectiveness of vehicle identification and indexing using the proposed framework. The index information can be used to address spatiotemporal queries for traffic applications. In the study experiments, the backgrounds of the traffic video sequences are complex. Our framework can address such complex scenarios for intersection monitoring.

Based on our experiments, the false positives are mainly caused by camera motion. The temporal tracking over a set of frames allows us to reduce this kind of errors since segments identified due to noise and motion are not temporally coherent. By contrast, the typical reason to have the false negatives is because of the slow motion of the vehicles, in which the motion difference exists but is not significant. In such situations, our segmentation method tends to detect only part of the vehicle object. It is typically a small region and will likely be discarded through a noise-filtering phase. In our framework, the segments which are very small will be identified as noise and hence rejected. The selection of the threshold value for determining whether a segment should be identified as noise depends on prior knowledge such as the average size of the vehicle objects (in terms of pixels) under a specific shooting scale. Also, the thresholds selected for

object tracking depend on the shooting scale and the average vehicle speed. The constancy for the shrinking of the bounding box for background update is selected as 4-6 pixels, and it works well in most scenarios.

6.5 Conclusions and Future Work

In this chapter, a learning-based spatio-temporal vehicle tracking and indexing framework is presented for unsupervised video data storage and access for real-time traffic operations. It incorporates an unsupervised image/video segmentation method, background learning and subtraction techniques, object tracking, multimedia augmented transition network (MATN) model, and multimedia input strings. A self-adaptive background learning and subtraction method is proposed and applied to a real life traffic video sequence to enhance the object segmentation procedure for obtaining more accurate spatio-temporal information of the vehicle objects. The background learning process is relatively simple and very effective based on our experiment results. Almost all vehicle objects and vehicle types are successfully identified through this framework. The spatio-temporal relationships of the vehicle objects are captured by the unsupervised image/video segmentation method and the proposed object tracking algorithm, and modeled by using the MATN model and multimedia input strings. Useful information is indexed and stored into a multimedia database for further information retrieval and query. A fundamental advantage of the proposed background learning algorithm is that it is fully automatic and unsupervised, and performs the adjustments in a self-adaptive way. As illustrated by the experiments, the initial inaccurate background information can be iteratively refined as the procedure proceeds, thereby benefiting the segmentation process in turn. Hence, the proposed framework can deal with very complex situations vis-à-vis intersection monitoring.

The proposed research seeks to bridge the important missing link between transportation management and multimedia information technology. In order to develop a transportation

multimedia database system (MDBS) with adequate capabilities, the following future work will be investigated: (i) storing and organizing the rich semantic multimedia data in a systematic and hierarchical model; (ii) identifying the vehicle objects in video sequences under different conditions; and (iii) fusing different types of media data.

Chapter 7. Conclusions and Future Work

In this chapter, the framework for multimedia indexing and retrieval is summarized. In addition, the results from each proposed component are outlined. Future works are then discussed in the last section.

7.1 Conclusions

In this dissertation, an integrated framework as well as the approaches for multimedia indexing and retrieval are proposed. We put the main focus on image and video data management. One of the major obstacles to the content-based multimedia retrieval is the lack of coincidence between the low level information that one can extract from the multimedia data and the high-level interpretation that the same data have for a user. For example, while we may be able to describe images linguistically, the semantics in the image is difficult to capture for computer applications. The proposed framework aims to bridge the gap between semantic concepts and the automatically extractable low- and intermediate-level features. To achieve this goal, a set of techniques have been developed, including image segmentation, content-based image retrieval, object tracking, shot/scene-based video indexing, and video event detection. The core techniques and algorithms developed in this study considerably facilitate image/video searching at higher semantic levels and are expected to have a significant impact on the future growth of image and video databases.

In the image component, we envision two ways for enhancing content-based retrieval for image databases - active learning and object-based retrieval. First, a stochastic framework for general image retrieval (whole-image retrieval) is proposed, which integrates both the global image features and object spatial features. A stochastic mechanism, called Markov Model Mediator (MMM), is used in this framework to facilitate the learning and retrieval process for content-based image retrieval, which serves as the retrieval engine of the CBIR systems and uses stochastic-based similarity measures. Different from the common methods, this mechanism carries out the searching and similarity computing process dynamically, taking into consideration

not only the image content features but also other characteristics of images such as their access frequencies and access patterns. In addition, a PCA based pre-filtering method is incorporated into this framework in order to reduce the search space. The experimental results on a large image database (10,000 images) showed that, by using the MMM mechanism together with the stochastic process, the accuracy can be maintained above 80% when the retained PCA candidate pool has 800 or more candidate images which only accounts for 8% of the 10,000 images in the testing database, and the advantage brought by this reduction is that the retrieval speed is significantly increased and that the memory use is greatly reduced. In addition to general content-based image retrieval, we also propose a machine learning framework to enable object-based image retrieval, which is based on the image segmentation results and utilizes multiple instance learning techniques and user relevance feedback. This framework can deal with the multiple object retrieval scenarios in which the user may have multiple focuses of attention. Therefore, it can help to discover a user's high-level concepts from low-level image features. The neural network is used in this framework to map the low-level image features to the user's concepts, and the parameters of the neural network are adaptively updated through the user feedback process. By using a medium-sized image database (2,100 images), a number of experiments on different image categories have been conducted, including horses, mountain scenes, snow scenes, leopards, apes, owls, and race cars. The averaged accuracy within the top 30 retrieved images is around 70%, which demonstrates the effectiveness of our multiple object retrieval method using MIL and RF.

However, in order to make object information readily available for object related applications such as query-by-object, the underlying image segmentation method has to be effective and efficient. In this dissertation, we propose a method called WavSeg which is based on the unsupervised segmentation method SPCPE and wavelet analysis. WavSeg enhances SPCPE by making its initial partition more reasonable and more stable. The experiments show a

significant improvement in the segmentation results. In other words, the segments/objects extracted from the images are more in accordance with human eyes. In addition, by applying a set of optimizing strategies, the performance of the segmentation is quite acceptable for practical applications. More importantly, the object information obtained by segmentation are also used in video parsing and indexing, and it plays an important role in deriving high-level semantic content from video data.

In the video component, the shot/scene-based indexing is used and forms the basis for video event detection. The video sequences are first segmented into video shots, and then video shots are grouped into video scenes. In shot boundary detection, it integrates several techniques including the enhanced unsupervised image segmentation method, object tracking, and pixel-histogram comparison, etc. Compared with other methods, the object information obtained through the process can compensate for the shortcomings of global feature based criteria, as well as to provide the spatio-temporal information of the video objects. Based on shot detection, a video scene detection method is further proposed by using the joint visual and audio clues. The experimental results indicate that the performance of the proposed shot-detection method on a large video data set is much more stable than the twin-comparison histogram method. The averaged values of *Precision* and *Recall* are both above 90%. The proposed scene change detection method can achieve 92% in Precision and 89% in Recall on average.

As mentioned earlier, the proposed framework is an integrated framework in the sense that the underlying core techniques can be applied in different components and some subcomponents can be seamlessly integrated with other subcomponents. In this dissertation, by using the results of shot detection, we further propose a new hierarchical multimedia data mining framework for the automatic extraction of soccer events in soccer videos by using combined multimodal analysis and the decision tree logic. This framework is composed of three major components: video parsing, data cleaning, and data mining. First, by applying the proposed video

shot detection method, the shot boundaries can be obtained, with the advantage of producing important visual features and even object-level features during the same process. Based on the object-segmentation results (segmentation mask maps) produced during shot detection, some high-level features such as the grass-ratio and audience/player area can be derived and located at a low cost, which are further used in the detection of the target events. Then a complete set of visual/audio features are extracted for each shot at different granularities. This rich multi-modal feature set is filtered by a heuristic data cleaning step to remove the noise as well as to reduce the irrelevant data. The proposed framework mines the target events in a way that the more generic events such as corner kicks are mined at the higher level, while the more specific events like corner-goals are mined at the lower level. This 'narrow-down' event mining strategy is both effective and efficient according to our experimental results. The performance of shot boundary detection is also reported, with an overall performance of 95.2% in Precision and 85.2% in Recall. The proposed multimedia data mining framework has many implications in video indexing and summarization, video database retrieval, and semantic video browsing, etc.

To demonstrate the potential usage of the proposed multimedia indexing and retrieval framework, the core techniques and approaches are further tailored and applied to other practical application domains such as the video surveillance for intelligent transportation systems, which incorporates an unsupervised image/video segmentation method, background learning and subtraction techniques, object tracking, multimedia augmented transition network (MATN) model, and multimedia input strings. As illustrated by the experiments over four different traffic surveillance videos, the overall performance of vehicle object identification over the four video sequences is robust. The precision and recall values are approximately 95% and 90%, respectively.

7.2 Future Work

The possible extensions to the current proposed framework are considered in this section. These include extension to the current object-based image retrieval by applying integrated model learning and accumulative learning techniques, enhancement of video data mining by applying information fusion and feature selection techniques, and integration of the traffic video surveillance application and the unusual event mining. The future work is listed in the following subsections.

7.2.1 Extension to the Object-Based Image Retrieval

In the proposed object-based image retrieval framework, users are required to specify the positive/negative feedback on each object. Although this approach has several advantages such as the modeling hierarchy is very clear, the learning of individual concept models is relatively simple, and the workload of learning can be distributed to multiple machines in parallel because the model training for each object of interest is independent, it also has a major limitation in that the user needs to take too much responsibility in providing feedback for each object of interest while having to remember the correct orders (first object, second object, etc.) in providing such feedback. There are two possible solutions to this problem, depending on how we model the user concept when multiple objects of interests are involved in content-based image retrieval.

The first alternative approach can be achieved by applying the concept of "conceptual object of interest." Suppose the user is interested in M objects in the query image; the M objects of interest constitute a "conceptual object of interest." Thus the input of the concept model is the representation of the conceptual object instead of the individual objects. Under this modeling strategy, the user only needs to provide feedback for the whole image, and the underlying "conceptual object of interest" as well as the mapping function will be discovered through the user's relevance feedback. In this case, an "instance" is not an individual object; instead it

corresponds to a composite object. For example, in the two-object retrieval scenario, an "instance" in MIL would be a pair of objects, and the representation of this conceptual object is the combination of the region feature vectors belonging to the two objects involved. The underlying techniques used to implement the concept model for the "conceptual object of interest" can be any machine learning techniques. However, it can be expected that the learning of the concept model may take much more time and iterations to converge due to the extra complexity brought by the composite object. Therefore, a more proper solution could be a combination of between the current modeling mechanism (individual feedback) and the conceptual modeling mechanism (integrated feedback), or a trade-off between them. This possibility will be investigated as well in the future work.

The aforementioned approach, while it can clearly identify users' interest in one or multiple objects, limits the flexibility of conducting fuzzy learning in object-based image retrieval. For example, it assumes that the number of objects of interest is known prior to actual retrieval. However, in many cases a general user may not have a clear idea of how many objects he/she is interested in, although his/her high-level perception of the query image is based on semantic objects. In addition, sometimes there is no clear boundary between the objects of interest and the rest of the objects in an image. Furthermore, it also complicates the user task by requiring general users to specify the number of objects they are interested in and to stick with their chosen objects with consistence and coherence. A more general solution to this object-based perception modeling problem can be achieved by using an integrated modeling approach. In this possible approach, each region/object in a query image corresponds to a semantic object with a visual concept associated with it. The user's overall concept of the query image (in terms of semantic objects) is the composition of the region concepts which is modeled by an integrated concept model. For each region/object in a query image, a corresponding object concept model

will be built and associated with it. The similarity measurement based on this model is then calculated as follows:

- At the first level, we measure how well an image in the database matches an object concept model of the query image. This is done by feeding the region feature vectors of that image into an object concept model, which produces a similarity score for each region/object inside that image. The overall similarity score between the image and the object concept model is the maximum of all the region similarity scores.
- After the similarity scores for all the object concept models are collected, they are fed into an Integrated Model to produce the overall similarity score between the image and the query image.

A critical issue associated with this approach is how to conduct model learning. In particular, this is an integrated modeling and learning process in which both the object concept models and the integrated model need to be learned. The user's feedback on the retrieved images can be used as training examples to learn the user-concept model. Given the user's relevance feedback, we can apply the constrained optimization methods to search for the optimal model parameters based on certain similarity measurements and error evaluation functions. However, it is a big challenge to train both the object concept models and the integrated model at the same time by using the limited feedback information provided by users. The performance of this approach largely depends on the accuracy of the object segmentation results and the number of positive examples available to users at the time of supplying feedback. Consequently, the robustness of the trained models cannot be guaranteed unless sufficient training examples can be supplied.

Another problem in the current object-based retrieval framework is that, although the query-by-example and the relevance feedback techniques alleviate the problem of the semantic gap, however, the relevance feedback relies on an individual's ability to provide the correct

feedback. Therefore, the techniques are not suitable for non-experienced users. In addition, the process of concept model learning in object-based retrieval is actually a process of optimizing the representation of semantic objects in terms of global features such as color and texture. Thus, its effectiveness is largely limited by the representation power of low-level features, while the user's relevance feedbacks cannot be fully utilized. Furthermore, in object-based image retrieval, it is critical that the number of training samples (especially the positive samples) exceeds a certain threshold to ensure the convergence of the learning process. This problem is exacerbated under the scenario of multiple object retrieval due to the extremely insufficient number of positive examples provided by the untrained system when lacking a concept model at the very beginning. As a possible solution, we plan to integrate the accumulative learning techniques into the object-based retrieval process. In particular, the proposed MMM mechanism (see Section 4.2) for long-term learning can be combined with the current object-based retrieval system. More specifically, the MMM-based long term learning can be applied in the way that the initial query results when given a query image are provided based on the query history from other users in the system. The feedback information provided by previous users serves as general users' preferences or ratings being shared among all the users in the system. This integrated approach takes advantage of shared efforts among the users into the image retrieval process. Thus, the system is expected to present more positive training samples to the users when performing multiple object retrieval, even without the presence of a particular concept model for a particular user.

7.2.2 Enhancement of Video Data Mining

In the current soccer video data mining framework, there are three tiers: multimodal feature extraction, data cleaning, and data mining. The pre-filtering rules in data cleaning step are heuristic and based on domain knowledge. While these rules are effective in filtering out irrelevant video shots, they are constructed directly based on intensive human observations and

thus require significant manual effort and deep understanding of the domain knowledge. In other words, at the current stage, the discovery of heuristic rules for data pre-filtering is not automated, which may limit the flexibility and applicability of the proposed framework in other types of videos or other application domains. Two possible strategies aimed to alleviate this problem can be investigated in our future work:

1. High-level structural analysis of videos using temporal models such as HMM: For example, as discussed in Section 5.3.4, we use a visual rule (see Visual Rule 3 in Section 5.3.4 - "*Within two succeeding shots that follow the goal shot, at least one shot should belong to the close-up shots*.") to capture the temporal relationship between a goal shot and its subsequent cheering shot(s). This rule captures not only the absolute temporal order of goal shots and cheering shots (goal shots precede cheering shots), but also the temporal constraints between these two types of soccer video shots ("Within *two* succeeding shots that follow the goal shot…"). In our future work, rather than constructing heuristic rules directly, we plan to use formal statistical techniques to discover domain-specific syntactic constraints automatically or semi-automatically. The training data for high-level structural analysis will consist of a set of consecutive video events/sub-events, and any models that are capable of capturing interactions and temporal evolvement can be used, such as HMM or dynamic Bayesian networks.
2. Integrating different modalities using statistical models: In the current video mining framework, the integration of different modalities is conducted by manually analyzing the temporal evolvement of the features within each modality and the temporal relationships between different modalities. For example, as presented in Section 5.3.4, the Audio Rule 1 ("*As a candidate goal shot, the last three (or less) seconds of its audio track and the first three (or less) seconds of its following shot should both contain at least one exciting point.*") for soccer goal event pre-filtering indicates the temporal evolvement of volume

features around the end of a goal shot and its succeeding shot. In our future work, the modeling of each modality will be conducted by using statistical models such as HMM. An advantage of using HMMs is that we can integrate different modalities and the temporal constraints can be accommodated in HMMs. In order to integrate and summarize different modalities, it is necessary to identify correspondence between events in different modalities. One possible way for summarizing and merging different modalities is to train a high-level statistical model on the states of the local statistical models for different modalities.

Another possible enhancement of the current video mining framework is to apply feature selection [Xie03] and feature reduction techniques [Chakrabarti98] to the data mining phase. Feature selection and feature reduction are preprocessing methods used to reduce the dimensionality of the data set in order to improve the performance of the data mining algorithms. Other future work includes applying the proposed framework to other types of video sequences such as TV news and other sports videos, and establishing a GUI for visualizing the data mining results. Information visualization can be applied to represent the outputs from the data mining algorithms, which can enhance the users' perception of the outputs.

7.2.3 Integration of the Traffic Video Surveillance Application and the Event Mining

Another possible long-term plan is to apply the developed techniques and tools to build a traffic video surveillance information system that can track and classify the vehicle objects, model the spatio-temporal data, mine the unusual events, and provide surveillance video summarization and information retrieval facilities to support advanced traffic administration. The future work in this subsection includes the following major tasks:

1. Vehicle classification: Vehicle classification data are extremely important in almost all aspects of transportation planning and engineering. For example, the truck volume

information can be used in designing pavements (thickness of the pavement), in the scheduling of reconstruction of highways according to the predicted pavement lifetime, in providing data for the predicted capacity of highways, etc. In our current framework, the individual vehicles can be identified and tracked, which is only the preliminary step leading to vehicle classification. In our future work, we will target 3~4 vehicle classes (cars, mid-size vehicles, large-size vehicles, etc.) for efficiency purposes. The available vehicle features such as dimensions, shapes, and critical edges will be considered in determining the class label of a vehicle, with the aid of machine learning techniques.

2. Spatio-temporal modeling and indexing of vehicle track data: The goal of this task is to develop an efficient and effective spatio-temporal indexing model for modeling vehicle track data. In the current traffic video surveillance framework, although MATN can model the basic spatio-temporal relationships among salient objects, it needs to be extended to enable higher-level spatio-temporal indexing such as trajectory indexing and temporal relationship indexing. A possible solution is to develop a vehicle trajectory model based on the information contained in MATN, such that both the individual vehicle trajectories and the relative spatio-temporal relationships among vehicles can be modeled in a single framework.
3. Traffic event mining: Recently, some learning and data mining techniques have been applied to the analysis of normal traffic behavior versus unusual events [Roman02]. The unusual event mining is not only useful for traffic video surveillance applications, but is also beneficial in extending the state of the knowledge of data mining. In our future work, the indexed vehicle track data will be further analyzed by data mining techniques such as HMM model for unusual event detection. The target events include sudden stopped vehicles, illegal U-turns, wrong way drivers, etc.

4. System implementation: Another important issue is how to develop an efficient summarization and information retrieval system to fully utilize the huge amount of data and discovered events. There has been very little work conducted in the past on traffic surveillance video summarization and information retrieval. A traffic surveillance video information system should be able to provide users with a quick overview of the essential contents of a huge surveillance video and help users to quickly locate the event(s) of interest. However, the data to be handled include not only the information of individual vehicles but also the higher-level information such as interesting events like illegal U-turns. Hence, there is a need to develop a system for traffic video summarization and information retrieval that can provide a set of more comprehensive functionalities that are demanded in real-life traffic control and administration. This would open up a new venue for the study of information retrieval at a semantic level related to spatio-temporal information.

List of References

[Assfalg02] J. Assfalg, M. Bertini, A. D. Bimbo, W. Nunziati, and P. Pala, "Soccer highlights detection and recognition using HMMs," *IEEE International Conference on Multimedia and Expo* (2002).

[Aksoy00] S. Aksoy and R. M. Haralick, "A Weighted Distance Approach to Relevance Feedback," *Proceedings of the International Conference on Pattern Recognition (ICPR00)*.

[Alattar97] A. M. Alattar, "Detecting Fade Regions in Uncompressed Video Sequences," *Proceedings of 1997 IEEE International Conference on Acoustics Speech and Signal Processing*, 3025-3028, 1997.

[Arman93] F. Arman, A. Hsu, and M.-Y. Chiu, "Image Processing on Compressed Data for Large Video Databases," in *Proc. First ACM Intl. Conference on Multimedia*, 1993, pp. 267-272.

[Beigi98] M. Beigi, A. Benitez, and S.-F. Chang, "Metaseek: A Content-Based Meta Search Engine for Image," *In Storage and Retrieval for Image and Video Databases, SPIE Proceedings Series*, San Jose, CA, 1998.

[Brunelli99] R. Brunelli, O. Mich, and C.M. Modena, "A Survey on the Automatic Indexing of Video Data," *Journal of Visual Communication and Image Representation*, 10:78-112, 1999.

[Buckley95] C. Buckley, A. Singhal, and M. Miltra, "New Retrieval Approaches Using SMART: TREC4," *Text Retrieval Conference, Sponsored by National Institute of Standard and Technology and Advanced Research Projects Agency*. (Nov. 1995).

[Carson97] C. S. Carson, et al., "Region-based image querying," in *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, San Juan, Puerto Rico, 42-49.

[Carson02] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8), pp. 026–1038, 2002.

[Chakrabarti98] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable Feature Selection, Classification and Signature Generation for Organizing Large Text Databases into Hierarchical Topic Taxonomies," *VLDB Journal*, vol. 7, no. 3, pp. 163-178, Aug. 1998.

[Chang98] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "A Fully Automatic Content-Based Video Search Engine Supporting Multi-Object Spatio-temporal Queries," *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image and Video Processing for Interactive Multimedia*, 8(5):602-615, September 1998.

[Chang99] Ch.-H. Chang and C.-C. Hsu, "Enabling Concept-Based Relevance Feedback for Information Retrieval on the WWW," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 4, July/August 1999.

[Chang02] B. Li, E. Chang, and C.T. Wu, "DPF – A Perceptual Distance Function for Image Retrieval," in *IEEE International Conference on Image Processing*, 2002, 2:597-600.

[Chen99] S.-C. Chen, S. Sista, M.-L. Shyu, and R. L. Kashyap, "Augmented Transition Networks as Video Browsing Models for Multimedia Databases and Multimedia Information Systems," *11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'99)*, pp. 175-182, Chicago, IL, USA, November 9-11, 1999.

[Chen00] S.-C. Chen, M.-L. Shyu, C. Zhang, and R. L. Kashyap, "Object Tracking and Augmented Transition Network for Video Indexing and Modeling," *12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2000)*, pp. 428-435, Vancouver, British Columbia, Canada, November 13-15, 2000.

[Chen00a] S.-C. Chen, S. Sista, M.-L. Shyu, and R. L. Kashyap, "An Indexing and Searching Structure for Multimedia Database Systems," *IS&T/SPIE conference on Storage and Retrieval for Media Databases 2000,* pp. 262-270, San Jose, CA, USA, January 23-28, 2000.

[Chen01] S.-C. Chen and R. L. Kashyap, "A Spatio-Temporal Semantic Model for Multimedia Database Systems and Multimedia Information Systems," *IEEE Trans. on Knowledge and Data Engineering*, vol. 13, no. 4, pp. 607-622, July/August, 2001.

[Chen01a] S.-C. Chen, M.-L. Shyu, and C. Zhang, "An Unsupervised Segmentation Framework For Texture Image Queries," *the 25th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC)*, pp. 569-573, October 8-12, 2001, Chicago, Illinois, USA.

[Chen01b] S.-C. Chen, M.-L. Shyu, C. Zhang, and R.L. Kashyap, "Video Scene Change Detection Method Using Unsupervised Segmentation and Object Tracking," *IEEE International Conference on Multimedia and Expo (ICME01)*, pp. 57-60, 2001.

[Chen01c] S.-C.Chen, M.-L. Shyu, and C. Zhang, "An Intelligent Framework for Spatio-Temporal Vehicle Tracking," in *the 4th International IEEE Conference on Intelligent Transportation Systems*, Oakland, California, USA, August 25-29, pp. 213-218, 2001.

[Chen01d] S.-C. Chen, M.-L. Shyu, C. Zhang, and R. L. Kashyap, "Identifying Overlapped Objects for Video Indexing and Modeling in Multimedia Database Systems," *International Journal on Artificial Intelligence Tools*, vol. 10, no. 4, pp. 715-734, December 2001.

[Chen02] L. Chen and M. T. Özsu, "Modeling of Video Objects in a Video Database," In *Proc. IEEE International Conference on Multimedia*, Lausanne, Switzerland, August 2002, 217-221.

[Chen02a] S.-C. Chen, M.-L. Shyu, W. Liao, and C. Zhang, "Scene Change Detection By Audio and Video Clues," *IEEE International Conference on Multimedia and Expo (ICME2002)*, pp. 365-368, August 26-29, 2002, Lausanne, Switzerland.

[Chen03a] S.-C. Chen, M.-L. Shyu, N. Zhao, C. Zhang, "An Affinity-Based Image Retrieval System for Multimedia Authoring and Presentation," *Proceedings of the 11th Annual ACM International Conference on Multimedia*, pp. 446-447, November 2-8, 2003 Berkeley, CA, USA.

[Chen03b] S.-C. Chen, M.-L. Shyu, C. Zhang, L. Luo, M. Chen, "Detection of Soccer Goal Shots Using Joint Multimedia Features and Classification Rules," *Proceedings of the Fourth International Workshop on Multimedia Data Mining (MDM/KDD2003), in conjunction with the*

ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 36-44, August 24-27, 2003, Washington, DC, USA.

[Chen03c] S.-C. Chen, M.-L. Shyu, and C. Zhang, "Innovative Shot Boundary Detection for Video Indexing," Edited by Sagarmay Deb, Video Data Management and Information Retrieval, accepted for publication, Idea Group Publishing, 2003.

[Chen04a] S.-C. Chen, M.-L. Shyu, M. Chen, and C. Zhang, "A Decision Tree-based Multimodal Data Mining Framework for Soccer Goal Detection," accepted for publication, *IEEE International Conference on Multimedia and Expo (ICME 2004)*, June 27 - June 30, 2004, Taipei, Taiwan, R.O.C.

[Chen04b] S.-C. Chen, M.-L. Shyu, C. Zhang, and M. Chen, "Soccer Event Detection via Multimedia Data Mining," submitted to *ACM Multimedia 2004*.

[Cheng01] H.D. Cheng and Y. Sun, "A Hierarchical Approach to Color Image Segmentation Using Homogeneity," *IEEE Transactions on Image Processing*, Dec. 2001.

[Ciaccia97]Ciaccia, M. Patella, and P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," *Proceedings of the 23rd VLDB International Conference*, Athens, Greece, September 1997.

[CNI] http://www.dlib.org/dlib/january97/oclc/01weibel.html, Image Description on the Internet, A Summary of the CNI/OCLC Image Metadata Workshop, Sep. 24-25, 1996, Dublin, Ohio.

[Cohen99] I. Cohen and G. Medioni, "Detecting and tracking objects in video surveillance," in *Proc. IEEE Computer Vision and Pattern Recognition*, Fort Collins, June 1999, pp. 319–325.

[Courtney97] J. D. Courtney, "Automatic Video Indexing via Object Motion Analysis," *Pattern Recognition*, vol. 30, no. 4, pp. 607-625, 1997.

[Cox00] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, and P.N. Yianilos, "The Bayesian Image Retrieval System, Pichunter: Theory, Implementation and Psychological Experiments," *IEEE Transactions on Image Processing*, 9(1):20-37, 2000.

[Cucchiara00] R.Cucchiara, M. Piccardi, and P. Mello, "Image Analysis and Rule-based Reasoning for a Traffic Monitoring System," *IEEE International conference on Intelligent Transportation Systems (IEEJ JSAI),* vol. 1, no. 2, pp. 119-130, Tokyo, Japan, June 2000.

[CueVideo] IBM CueVideo Project. http://www.almaden.ibm.com/cs/cuevideo/.

[Dagtas01] S. Dagtas and M. Abdel-Mottaleb, "Extraction of TV Highlights Using Multimedia Features," In *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, 2001.

[Dailey00] D. J. Dailey, F. Cathey, and S. Pumrin, "An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras," *IEEE Transactions on Intelligent Transportations Systems*, vol. 1, no. 2, pp. 98-107, June 2000.

[Dao96] S. Dao, Q. Yang, and A. Vellaikal, "MB$^+$-tree: An Index Structure for Content-Based Retrieval," in Chapter 11 of *Multimedia Database Systems: Design and Implementation Strategies*, MA: Kluwer, 1996.

[Dimitrova02] N. Dimitrova, H-J Zhang, B. Shahraray, I. Sezan, T.S. Huang, and A. Zakhor, "Applications of Video-Content Analysis and Retrieval," *IEEE Multimedia*, 9(3):42-55, 2002.

[Dimitrova03] N. Dimitrova, "Multimedia Content Analysis: The Next Wave," in *International Conference on Image and Video Retrieval,* pp.9-18, 2003.

[Ekin03] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic Soccer Video Analysis and Summarization," *IEEE Transactions on Image Processing*, 12, 7 (July 2003), 796-807.

[Faloutsos94] C. Faloutsos, et al, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems* 3, 231-262, 1994.

[Fan00] L. Fan and K. K. Sung, "Model-Based Varying Pose Face Detection and Facial Feature Registration in Video Images," *8th ACM International Conference on Multimedia*, pp. 295-302, Los Angeles, CA, USA, Oct. 2000.

[Ferman97] A. M. Ferman, B. Guensel, and A. M. Tekalp, "Object-based Indexing of MPEG-4 Compressed Video," *in Proceedings of SPIE: Visual Communications and Image Processing*, pp. 953-963, San Jose, CA, USA, Feb. 1997.

[Flickner95] M. Flickner et al., "Query by image and video content: The QBIC system," *Computer*, pp. 23-32, Sept. 1995.

[Frank86] O. Frank and D. Strauss, "Markov Graphs," *Journal of the American Statistical Association*, 81, pp. 832-842, 1986.

[Friedman97] N. Friedman and S. Russell, "Image Segmentation in Video Sequences: A Probabilistic Approach," *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence (UAI 97)*.

[Gargi98] U. Gargi, R. Kasturi, and S. Antani, "Performance Characterization and Comparison of Video Indexing Algorithms," in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 559-565, 1998.

[Gong95] Y. Gong, L.T. Sin, C.H. Chuan, H. Zhang, and M. Sakauchi, "Automatic Parsing of TV Soccer Programs," In *Proceedings of IEEE Multimedia Computing and Systems*, Washington D.C., 1995.

[Gonzalez93] R. C. Gonzalez and R. E. Woods, *Digital image processing*, Reading, Mass: Addison-Wesley, 1993.

[Guttman84] A. Guttman, "R-tree: A Dynamic Index Structure for Spatial Search," *Proc. ACM SIGMOD*, pp. 47-57, June 1984.

[Grimson98] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using Adaptive Tracking to Classify and Monitor Activities in a Site," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Preceding,* pp. 22-31, 1998.

[Gunsel98] B. Gunsel, A. M. Ferman, and A. M. Tekalp, "Temporal Video Segmentation Using Unsupervised Clustering and Semantic Object Tracking," *Journal of Electronic Imaging*, 7(3), pp. 592-604, 1998.

[Hafner95] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 729-736, July, 1995.

[Hampapur94] A. Hampapur, T. E. Weymouth, and R. Jain, "Digital video segmentation," *In Proc. of ACM Multimedia'94*, pp. 357-364, San Francisco, CA, 1994.

[Han92] J. Han, Y. Cai, and N. Cercone, "Knowledge discovery in databases: An attribute-oriented approach," *Proc. of the 18th Int. Conf. on Very Large Data Bases*, pp. 547-559, Aug. 1992.

[Hanjalic01] A. Hanjalic and J. Biemond (Eds.), Special Issue on Content-Based Image and Video Retrieval, *International Journal of Image and Graphics*, 1(3), 2001.

[Haritaoglu98] I. Haritaoglu, D. Harwood, and L. Davis, "W 4 - Who, Where, When, What: A Real-Time System for Detecting and Tracking People," *IEEE Third International Conference on Face and Gesture Recognition*, pp. 222-227, Nara, Japan, 1998.

[Haritaoglu00] I. Haritaoglu, D. Harwood, and L. Davis, "A Fast Background Scene Modeling and Maintenance for Outdoor Surveillance," *15th IEEE International Conference on Pattern Recognition: Applications, Robotics Systems and Architectures*, pp. 179-183, Barcelona, Spain, Sept. 2000.

[Heisterkamp03] R. Heisterkamp and J. Peng, "Kernel VA-Files for Relevance Feedback Retrieval," *Proceedings of the First ACM International Workshop on Multimedia Databases (ACM MMDB'03)*, pp. 48-54, November 7, 2003, New Orleans, Louisiana, USA.

[Huang98] J. Huang, Z. Liu, and Y. Wang, "Integration of audio and visual information for content-based video segmentation," *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 526-530, Chicago, IL, Oct. 4-7, 1998.

[Huang98a] T. Huang and S. Russell, "Object identification: A bayesien analysis with application to traffic surveillance," *Artificial Intelligence*, vol. 103, pp. 1–17, 1998.

[Huang02] X. Huang, S.-C. Chen, M.-L. Shyu, and C. Zhang, "User Concept Pattern Discovery Using Relevance Feedback and Multiple Instance Learning for Content-Based Image Retrieval," *MDM/KDD'2002*, pp. 100-108, July 23, 2002.

[Huang03] X. Huang, S.-C. Chen, and M.-L. Shyu, "Incorporating Real-Valued Multiple Instance Learning Into Relevance Feedback For Image Retrieval," *Proc. of the IEEE Intl. Conf. on Multimedia & Expo (ICME),* vol. I, pp. 321-324, 2003.

[Hwang98] T.-H. Hwang and D.-S. Jeong, "Detection of Video Scene Breaks Using Directional Information in DCT Domain," *Proceedings of the 10th International Conference on Image Analysis and Processing*, 1998.

[Intille01] S. Intille and A. Bobick, "Recognizing Planned, Multi-person Action," *Computer Vision and Image Understanding*, 81, 3 (Mar. 2001), 414-445.

[Jiang00] H. Jiang, T. Lin, and H.J. Zhang, "Video Segmentation with the assistance of audio content analysis," *IEEE International Conference on Mulatimedia and Expo (ICME00),* pp. 1507-1510, 2000.

[Jin98] J. S. Jin, et al, "Using browsing to improve content-based image retrieval," in *Multimedia Storage and Archiving Systems III, Proc SPIE* 3527, 101-109, 1998.

[Jing02] F. Jing, M. Li, H. Zhang, and B. Zhang, "An Effective Region-based Image Retrieval Framework," *Proceedings of the 2002 ACM workshops on Multimedia***,** Pages: 456 – 465, 2002.

[Jobson92] J.D. Jobson. Applied Multivariate Data Analysis Volumn II: Categorical and Multivariate Methods. Springer-Verlag Inc., NY, 1992.

[Johnson98] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. 4th Ed., Prentice-Hall, NJ, 1998.

[Kamijo99] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic Monitoring and Accident Detection at Intersections," *IEEE International Conference on Intelligent Transportation Systems*, pp. 703-708, Tokyo, Japan, Oct. 1999.

[Kamijo00] S. Kamijo, Y. Matsushita, and K. Ikeuchi, "Traffic Monitoring and Accident Detection at Intersections," *IEEE Trans. Intelligent Transportation Systems*, vol. 1, no. 2, 2000.

[Kang03] Kang, Y.-L., Lim, J.-H., et al. Soccer Video Event Detection with Visual Keywords. In *Proceedings of IEEE Pacific-Rim Conference on Multimedia (ICICS-PCM)*, 2003.

[Kaplan98] L. M. Kaplan, et al, "Fast texture database retrieval using extended fractal features," in *Storage and Retrieval for Image and Video Databases VI*, Proc SPIE 3312, 162-173, 1998.

[Kleene56] S. C. Kleene, "Representation of Events in Nerve Nets and Finite Automata, Automata Studies," *Princeton University Press*, Princeton, N.J., pp. 3-41, 1956.

[Koller94] D. Koller, J. Weber, and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning," *3rd European Conference on Computer Vision, Eccv '94*, pp. 189-196, Stockholm, Sweden, May 1994.

[Lee00] S.-W. Lee, Y.-M. Kim, and S.-W. Choi, "Fast Scene Change Detection using Direct Feature Extraction from MPEG compressed Videos," *IEEE Trans. on Multimedia*, vol. 2, No. 4, Dec. 2000.

[Lew00] M. S. Lew, "Next-Generation Web Searches for Visual Content," *IEEE Computer*, vol. 33, pp. 46-53, 2000.

[Liang98] K. C. Liang, and C. C. J. Kuo, "Implementation and performance evaluation of a progressive image retrieval system," in *Storage and Retrieval for Image and Video Databases VI*, Proc SPIE 3312, 37-48

[Lin97] H. C. Lin, L. L. Wang, and S. N. Yang, "Color Image Retrieval Based On Hidden Markov Models," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 332-339, 1997.

[Liu98] F. Liu and R.W. Picard, "Finding Periodicity in Space and Time," *Proceedings of IEEE Intl. Conf. on Computer Vision*, pp. 376-383, Bombay, India, 1998.

[Lu00] G. Lu, Multimedia Database Management Systems. Boulder, Colo. NetLibrary, 2000.

[Lu00a] Y. Lu, C.-H. Hu, X.-Q. Zhu, H.-J. Zhang and Q. Yang, "A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems," *ACM Multimedia*, 2000.

[Lu01] L. Lu, H. Jiang, and H.J. Zhang, "A Robust Audio Classification and Segmentation Method," *ACM Multimedia 2001*.

[Lu02] G. Lu, "Techniques and Data Structures for Efficient Multimedia Retrieval Based on Similarity," *IEEE Trans. on Multimedia*, 4(3), pp. 372-384, Sep. 2002.

[Ma98] W. Y. Ma and B. S. Manjunath, "A Texture Thesaurus for Browsing Large Aerial Photographs," *Journal of the American Society for Information Science*, 49 (7) pp. 633-648, 1998.

[Mahmassani94] H. S. Mahmassani, T. Hu, S. Peeta, and A. Ziliaskopoulos, "Development and Testing of Dynamic Traffic Assignment and Simulation Procedures for ATIS/ATMS Applications," Technical Report DTFH61-90-C-00074-FG, Center for Transportation Research, The University of Texas at Austin, 1994.

[Maxwell01] B. A. Maxwell, "Towards Object-Based Retrieval for Image Libraries", in *Proceedings of IEEE Workshop on Content Based Image Access of Image and Video Libraries (CBAIVL)*, December 2001.

[Moore97] J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher, "Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering", *The 7th Workshop on Information Technologies and Systems,* December 1997.

[Muramoto00] T. Muramoto and M. Sugiyama, "Visual and Audio Segmentation for video streams," *IEEE International Conference on Mulatimedia and Expo (ICME00),* pp 1547-1550, 2000.

[Nagasaka95] A. Nagasaka and Y. Tanaka, "Automatic Video Indexing and Full-video Search for Object Appearances," *in Visual Database Systems II*, pp. 113-127, Elsevier, 1995.

[Naphade01] M. R. Naphade and T. S. Huang, "A Probabilistic Framework for Semantic Indexing and Retrieval in Video," *IEEE Transactions on Multimedia*, vol. 3, no. 1, March 2001.

[Natsev99] A. Natsev, R. Rastogi, and K. Shim, "WALRUS: a similarity retrieval algorithm for image databases," *In Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pp. 395–406, 1999.

[Ngo00] C.-W. Ngo, T.-C. Pong, and R. T. Chin, "Motion-Based Video Representation for Scene Change Detection," *Proceedings of the International Conference on Pattern Recognition (ICPR'00)*, 2000.

[Niblack98] W. Niblack, et al, "Updates to the QBIC system," *in Storage and Retrieval for Image and Video Databases VI, Proc SPIE 3312*, 150-161, 1998.

[NISO] http://www.niso.org/imagerpt.html, NISO/CLIR/FLG Technical Metadata for Images Workshop Report, April 18-19, 1999, Washington DC.

[Pass99] G. Pass and R. Zabih, "Comparing Images Using Joint Histograms," *ACM Multimedia Systems*, 1999.

[Peeta94] S. Peeta, "System Optimal Dynamic Traffic Assignment in Congested Networks with Advanced Information Systems," Ph.D. Dissertation, The University of Texas at Austin, 1994.

[Peeta95a] S. Peeta and H. S. Mahmassani, "System Optimal and User Equilibrium Time-dependent Traffic Assignment in Congested Networks," *Annals of Operations Research,* pp. 81-113, 1995.

[Peeta95b] S. Peeta and H. S. Mahmassani, "Multiple User Classes Real-time Traffic Assignment for Online Operations: A Rolling Horizon Solution Framework," *Transportation Research,* vol. 3, no. 2, pp. 83-98, 1995.

[Pentland94] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *Proc. Storage and Retrieval for Image and Video Databases II*, Vol. 2185, pp.34-47, SPIE, Bellingham, Washington, 1994.

[Quinlan93] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Rabiner86] L. R. Rabiner and B. H. Huang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine,* Vol. 3, No. 1, pp. 4-16, January 1986.

[Ravela98] S. Ravela and R. Manmatha, "Retrieving Images by Appearance," in *Proceedings of IEEE International Conference on Computer Vision (IICV98), Bombay, India*, 608-613.

[Roman02] R. Pflugfelder, "Visual Traffic Surveillance using Real-Time Tracking," Master Thesis, der Technischen Universität Wien.

[Roussopoulos95] N. Roussopoulos, C. Faloutsos, and T. Sellis, "Nearest Neighbor Queries," *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp. 71-79, 1995.

[Rui98] Y. Rui, T.S. Huang, etc., "Relevance feedback: A power tool in interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Segmentation, Description, and Retrieval of Video Content,* 8(5): pp. 644-655, September 1998.

[Santini99] S. Santini and R. Jain, "Similarity Measures," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871-883, 1999.

[Shahraray95] B. Shahraray, "Scene change detection and content-based sampling of video sequences," in *Proc. SPIE'95, Digital Video Compression: Algorithm and Technologies*, vol. 2419, San Jose, CA, 1995.

[Shyu00a] M.-L. Shyu, S.-C. Chen, and C.-M. Shu, "Affinity-Based Probabilistic Reasoning and Document Clustering on the WWW," *the 24th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC)*, Taipei, Taiwan, pp. 149-154, October 25-27, 2000.

[Shyu00b] M.-L. Shyu, S.-C. Chen, and R.L. Kashyap, "A Probabilistic-Based Mechanism For Video Database Management Systems," *IEEE International Conference on Multimedia and Expo (ICME2000)*, pp. 467-470, July 30-August 2, 2000, New York City, USA.

[Shyu00c] M.-L. Shyu, S.-C. Chen, and R.L. Kashyap, "Organizing a Network of Databases Using Probabilistic Reasoning," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1990-1995, October 8-11, 2000, Nashville, Tennessee, USA.

[Shyu01a] M.-L. Shyu, S.-C. Chen, and C. Haruechaiyasak, C.-M. Shu, and S.-T. Li, "Disjoint Web Document Clustering and Management in Electronic Commerce," *the Seventh International Conference on Distributed Multimedia Systems (DMS'2001)*, pp. 494-497, September 26-28, 2001, Tamkang University, Taipei, Taiwan.

[Shyu02] M.-L. Shyu, S.-C. Chen, C. Zhang, "A Stochastic Content-Based Image Retrieval Mechanism," Edited by Sagarmay Deb, Multimedia Systems and Content-based Image Retrieval, pp. 302-320, Idea Group Publishing, 2004, ISBN: 1-59140-265-4.

[Shyu03] M.-L. Shyu, S.-C. Chen, M. Chen, C. Zhang, K. Sarinnapakorn, "Image Database Retrieval Utilizing Affinity Relationships," *Proceedings of the First ACM International Workshop on Multimedia Databases (ACM MMDB'03)*, pp. 78-85, November 7, 2003, New Orleans, Louisiana, USA.

[Sista99] S. Sista and R. L. Kashyap, "Unsupervised Video Segmentation and Object Tracking," *IEEE International Conference on Image Processing*, Japan, 1999.

[Sista00] S. Sista and R. L. Kashyap, "Unsupervised Video Segmentation and Object Tracking," *Computers in Industry*, vol. 42, no. 2-3, pp. 127-146, Jun. 2000.

[Smeaton02] A. F. Smeaton *et al*. "The TREC-2001 Video Track Report," *In Proc. of TREC-2001, NIST Special Publication* (in press), E. M. Voorhees and D. K. Harman (Eds.), 2002.

[Smith95] S. M. Smith and J. M. Brady, "ASSET-2: Real-time Motion Segmentation and Shape Tracking," *IEEE Trans. Pattern Anal. Machine Intell*., vol. 17, pp. 814-820, 1995.

[Smith96] J. R. Smith and S. F. Chang, "VisualSEEK: A Fully Automated Content-based Image Query System," *In Proceedings ACM Intern. Conf. Multimedia*, pp. 87-98, Boston, Nov 1996.

[Smith99] J. R. Smith and S-F. Chang, "Integrated Spatial and Feature Image Query," in *Multimedia Systems*, 7(2):129-140, 1999.

[Stauffer99] C. Stauffer and W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 246-252, 1999.

[Su01] Z. Su, S. Li, and H. Zhang, "Extraction of Feature Subspaces for Content-Based Retrieval Using Relevance Feedback," *Proceedings of the 9th ACM International Conference on Multimedia 2001* (*MM'01)*, pp. 98-106, Ottawa, Canada, September 30 - October 5, 2001.

[Sun03] H. Sun, J.-H. Lim, Q. Tian, and M. S. Kankanhalli, "Semantic Labeling of Soccer Video," *Proceedings of IEEE ICICS-PCM 2003*, 2003, 1 - 5.

[Smeulders98] A. W. M. Smeulders, M. L. Kersten, and T. Gevers, "Crossing the divide between computer vision and data bases in search of image databases," *in Porc. Of 4th IFIP 2.6 Working Conference on Visual Database Systems-VDB 4*, pp. 223-239, Italy, 1998.

[Smeulders00] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349-1380, 2000.

[Stricker96] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," in *Storage and Retrieval for Image and Video Databases IV,* (Sethi, I K and Jain, R C, eds), Proc SPIE 2670, 29-40.

[Sundaram00] H. Sundaram and S.-F. Chang, "Video Scene Segmentation Using Video and Audio Features," *IEEE International Conference on Mulatimedia and Expo (ICME00),* pp. 1145-1148, 2000.

[Sundaram00a] H. Sundaram and S.-F. Chang, "Audio Scene Segmentation using Multiple Models, Features and Time Scales," *ICASSP*, pp. 2441-2444, 2000.

[Swain93] M. J. Swain, "Interactive Indexing into Image Databases," in *Proc. SPIE Conference Storage and Retrieval in Image and Video Da*tabases, pp. 173-187, 1993.

[Swanberg93] D. Swanberg, C. F. Shu, and R. Jain, "Knowledge guided parsing in video database," in *Proc. SPIE'93, Storage and Retrieval for Image and video Databases, vol. 1908, San Jose, CA, 1993.*

[Theodoridis98] Y. Theodoridis, T. Sellis, et al., "Specifications for efficient indexing in spatiotemporal database," *in Proc. of IEEE International Conference on SSDBM*, pp. 242-245, 1998.

[Pfoser00] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel Approaches to the Indexing of Moving Object Trajectories," *Proc. 26th Int'l Conference on Very Large Databases, VLDB'00,* Cairo, Egypt, September 2000. Morgan Kaufmann.

[Tirthapura98] S. Tirthapura, et al. "Indexing Based on Edit-Distance Matching of Shape Graphs," in *Multimedia Storage and Archiving Systems III* (Kuo, C C J et al, eds), *Proc SPIE 3527*, 25-36.

[Tong01] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," *Proceedings of the ninth ACM international conference on Multimedia*, pp. 107-118, Ottawa, Canada, 2001.

[Tovinkere01] V. Tovinkere and R. J. Qian, "Detecting Semantic Events in Soccer Games: Toward a Complete Solution," In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2001, 1040-1043.

[Toyama99] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," *7th International Conference on Computer Vision (ICCV'99)*, pp. 255-261, held on the Island of Crete, Sept. 1999.

[Truong00] B. T. Truong, C. Dorai, and S. Venkatesh, "New Enhancements to Cut, Fade, and Dissolve Detection Processes in Video Segmentation," in *Proceedings of the 8th ACM International Conference on Multimedia*, 2000.

[URL1] http://i21www.ira.uka.de/cgi-bin/download?bad.

[URL2] http://www.imagesensing.com/.

[URL3] http://www.cts.umn.edu/research/index.html.

[URL4] http://www-2.cs.cmu.edu/ vsam/research.html#COMPUS

[VADS] http://vads.ahds.ac.uk/index.html, The Visual Arts Data Service.

[Vassilakopoulos95] M. Vassilakopoulos and Y. Manolopoulos, "Dynamic inverted quadtree: A structure for pistorial databases," *Information Systems*, 20(6): 483-500, 1995.

[VDBMS] VDBMS: http://www.cs.purdue.edu/vdbms/index.html

[Vellaikal98] Vellaikal, A and Kuo, C C J, "Hierarchical clustering techniques for image database organization and summarization," in *Multimedia Storage and Archiving Systems III, Proc SPIE* 3527, 68-79, 1998.

[Venters] C. C. Venters and M. Cooper, "A Review of Content-Based Image Retrieval Systems," http://www.jtap.ac.uk/reports/htm/jtap-054.html.

[Vinod97] V.V. Vinod, H. Murase, "Video Shot Analysis using Efficient Multiple Object Tracking," *Proceedings of the 1997 International Conference on Multimedia Computing and Systems (ICMCS '97)*.

[Virage] http://www.virage.com

[Vistex] http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[VRA] http://www.oberlin.edu/~art/vra/wc1.html VRA Core Categories Version 2.0

[Wactlar99] H. Wactlar, M. Christel, A. Hauptmann, and Y. Gong, "Informedia Experience on Demand: Capturing, Integrating and Communicating Experiences across People, Time and Space," *ACM Computing Surveys*, Vol. 31, No. 9, June, 1999.

[Wang00] R. Wang, Z. Liu, and J.-C. Huang, "Multimedia Content Analysis," *IEEE Signal Processing Magazine*, pp.12-36, Nov. 2000.

[Wang01] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Maching for Picture Libraries," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9), pp. 947-963, 2001.

[Wasfi99] A.K. Wasfi and G. Arif, "An Approach for Video Meta-Data Modeling and Query Processing," *In Proc. of ACM Multimedia*, pp. 215-224, 1999.

[Web1] http://www.ibroxfc.co.uk

[Web2] http://hsb.baylor.edu/courses/Kayworth/fun_stuff/

[Web3] http://www.cincinnatidockers.com

[Web4] http://www.mormino.net/videos/index.php3

[Web5] http://www.is.informatik.uni-duisburg.de/teaching/lectures/ir_ss03/folien/mm-ind.pdf

[Weber98] R. Weber, H.-J. Schek, and S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces," *Proceedings of the International Conference on Very Large Databases (VLDB),* pp. 194-205, New York City, New York, USA, Aug. 1998.

[Wiederhold92] G. Wiederhold, "Mediators in the Architecture of Future Information Systems," *IEEE Computers*, pp. 38-49, March 1992.

[White96] D. A. White and R. Jain, "Similarity Indexing: Algorithms and Performance," *Proc. SPIE Vol.2670*, San Diego, USA (Jan. 1996) pp.62-73.

[Wolf97] W. Wolf, "Hidden Markov Model Parsing of Video Programs," presented at the *International Conference of Acoustics, Speech and Signal Processing*, 1997.

[Xie03] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Feature Selection for Unsupervised Discovery of Statistical Temporal Structures in Video," *IEEE International Conference on Image Processing (ICIP 2003)*, Barcelona, Spain, September 2003.

[Xiong98] W. Xiong and J. C.-M. Lee, "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification," *Computer Vision and Image Understanding*, 71(2), 1998.

[Xu01] P. Xu, L. Xie, S.-F. Chang, et al. "Algorithms and Systems for Segmentation and Structure Analysis in Soccer Video," *Proceedings of ICME 2001*, 928-931.

[Yeo95] B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Systems Video Technol*., vol. 5, no. 6, pp. 533-544, 1995.

[Yeo97] B. Yeo and M. Yeung, "Retrieving and visualization video," *Comm. of the ACM*, vol. 40, no. 12, pp. 43-52, December 1997.

[Yoshitaka01] A. Yoshitaka, and M. Miyake, "Scene Detection by Audio-Visul Features," *IEEE International Conference on Multimedia and Expo (ICME01)*, pp. 49-52, 2001.

[Zabih95] R. Zabih, J. Miller, and K. Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," *in Proc. ACM Multimedia '95*, 1995, pp. 189-200.

[Zhang93] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia System*, vol. 1, 1993.

[Zhang94] H. Zhang and S. W. Smoliar, "Developing power tools for video indexing and retrieval," in *Proc. SPIE'94, Storage and Retrieval for Image and video Databases II*, vol. 2185, San Jose, CA, 1994.

[Zhang95] H. Zhang and D. Zhong, "A scheme for visual feature based image retrieval," in *Proc. SPIE Storage and Retrieval for Image and Video Databases III,* v(2420), 36-46, 1995.

[Zhang02] Q. Zhang, S. A. Goldman, W. Yu and J. Fritts "Content-Based Image Retrieval Using Multiple-Instance Learning," *Proc. of the 19th Intl. Conf. on Machine Learning*, 2002.

[Zhang03] C. Zhang, S.-C. Chen, M.-L. Shyu, "PixSO: A System for Video Shot Detection," *Proceedings of the Fourth IEEE Pacific-Rim Conference On Multimedia*, pp. 1-5, December 15-18, 2003, Singapore.

[Zhang04] C. Zhang, S.-C. Chen, and M.-L. Shyu, "Multiple Object Retrieval for Image Databases Using Multiple Instance Learning and Relevance Feedback," accepted for publication, *IEEE International Conference on Multimedia and Expo (ICME 2004)*, June 27 - June 30, 2004, Taipei, Taiwan, R.O.C.

[Zhong00] Y. Zhong, A.K. Jain, and M.-P. Dubuisson-Jolly, "Object Tracking Using Deformable Templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(5):544--549, May 2000.

[Zhou02] W. Zhou and C.C.J. Kuo, "Intelligent Systems for Video Understanding," Upper Saddle River, NJ:Prentice-Hall PTR, 2002.

VITA

CHENGCUI ZHANG

January, 1974	Born, MianYang, SiChuan, P. R. China
1996	B.E., Computer Science and Engineering ZheJiang University, P. R. China
1999	M.E., Computer Science and Engineering ZheJiang University, P. R. China
1999-2004	Doctorate in Computer Science Florida International University, Miami, Florida

PUBLICATIONS AND PRESENTATIONS

Chen, S.-C., Shyu, M.-L., Zhang, C., and Kashyap, R.L. (2000). "Object Tracking and Multimedia Augmented Transition Network for Video Indexing and Modeling," in *Proc. of* the *12th IEEE Intl. Conf. on Tools with Artificial Intelligence (ICTAI'00)*, pp. 428-435, Vancouver, British Columbia, Canada.

Chen, S.-C., Shyu, M.-L., and Zhang, C. (2001). "An Intelligent Framework for Spatio-Temporal Vehicle Tracking," *Proc. of the 4th Intl. IEEE Conf. on Intelligent Transportation Systems*, pp. 213-218, 2001.

Chen, S.-C., Shyu, M.-L., and Zhang, C. (2001). "An Unsupervised Segmentation Framework for Texture Image Queries," *Proc. of the 25th IEEE Computer Society Intl. Computer Software and Applications Conf. (COMPSAC)*, pp. 569-573, Chicago, Illinois, USA.

Chen, S.-C., Shyu, M.-L., Jin, X., Chen, Q., Zhang, C., and Strickrott, J. (2001). "A Flexible Image Retrieval and Multimedia Presentation Management System for Multimedia Databases," *Proc. of ACM Multimedia 2001 Conf.*, pp. 601-602, Ottawa, CANADA.

Chen, S.-C., Shyu, M.-L., Zhang, C., and Kashyap, R. L. (2001). "Identifying Overlapped Objects for Video Indexing and Modeling in Multimedia Database Systems," *Intl. Journal on Artificial Intelligence Tools*, vol. 10, no. 4, pp. 715-734.

Chen, S.-C., Shyu, M.-L., Zhang, C., and Kashyap, R.L. (2001). "Video Scene Change Detection Method Using Unsupervised Segmentation and Object Tracking," in *Proc. of IEEE Intl. Conf. on Multimedia and Expo (ICME)*, pp. 57-60, Tokyo, Japan.

Chen, S.-C., Shyu, M.-L., Zhang, C., and Strickrott, J. (2001)., "Multimedia Data Mining for Traffic Video Sequences," *Proc. of the 2nd Intl. Workshop on Multimedia Data Mining (MDM/KDD'2001)*, in conjunction with the *7th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*, pp. 78-85, San Francisco, CA, USA.

Chen, S.-C., Li, S.-T., Chen, H.-C., Shyu, M.-L., and Zhang, C. (2002). "Streaming SMIL Presentations via a Multimedia Semantic Model," in *Proc. of Second Intl. Workshop on Intelligent Multimedia Computing and Networking (IMMCN'2002),* pp. 919-922, Durham, North Carolina, USA.

Chen, S.-C., Li, S.-T., Shyu, M.-L., Zhan, C., and Zhang, C. (2002). "A Multimedia Semantic Model for RTSP-Based Multimedia Presentation Systems," *Proc. of IEEE Fourth Intl. Symposium on Multimedia Software Engineering (MSE2002)*, pp. 124-131, Newport Beach, California, USA.

Chen, S.-C., Shyu, M.-L., Liao, W., and Zhang, C. (2002). "Scene Change Detection by Audio and Video Clues," in *Proc. of IEEE Intl. Conf. on Multimedia and Expo (ICME2002)*, pp. 365-368, Switzerland.

Chen, S.-C., Shyu, M.-L., Peeta, S., and Zhang, C. (2002). "Unsupervised Automated Learning-Based Spatio-Temporal Vehicle Tracking and Indexing for Transportation Multimedia Database Systems," *Transportation Research Board (TRB) 2002 Annual Meeting*, Washington DC., USA.

Chen, S.-C., Shyu, M.-L., Zhang, C., and Strickrott, F. (2002). "A Multimedia Data Mining Framework: Mining Information from Traffic Video Sequences," *Journal of Intelligent Information System, Special Issue on Multimedia Data Mining*, vol. 19, no. 1, pp. 61-77.

Huang, X., Chen, S.-C., Shyu, M.-L., and Zhang, C. (2002). "User Concept Pattern Discovery Using Relevance Feedback and Multiple Instance Learning for Content-Based Image Retrieval," in *Proc. of* the *Third Intl. Workshop on Multimedia Data Mining (MDM/KDD'2002)*, in conjunction with the *8th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*, pp. 100-108, Edmonton, Alberta, Canada.

Chen, S.-C., Shyu, M.-L., and Zhang, C., et al. (2003). "Damage Pattern Mining in Hurricane Image Databases," in *Proc. of the 2003 IEEE Intl. Conf. on Information Reuse and Integration (IRI'2003)*, pp. 227-234, in Las Vegas, Nevada, USA.

Chen, S.-C., Shyu, M.-L., Peeta, S., and Zhang, C. (2003). "Learning-Based Spatio-Temporal Vehicle Tracking and Indexing for Transportation Multimedia Database Systems," *IEEE Trans. on Intelligent Transportation Systems*, vol. 4, No. 3, pp. 154-167.

Chen, S.-C., Shyu, M.-L., Zhang, C., Luo, L., and Chen, M. (2003). "Detection of Soccer Goal Shots Using Joint Multimedia Features and Classification Rules," in *Proc. of the Fourth Intl. Workshop on Multimedia Data Mining (MDM/KDD2003), in conjunction with the ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*, pp. 36-44, Washington, DC, USA.

Chen, S.-C., Shyu, M.-L., Zhao, N. and Zhang, C. (2003). "Component-Based Design and Integration of a Distributed Multimedia Management System," in *Proc. of the 2003 IEEE Intl. Conf. on Information Reuse and Integration (IRI'2003)*, pp. 485-492, in Las Vegas, Nevada, USA.

Chen, S.-C., Shyu, M.-L., Zhao, N., and Zhang, C. (2003). "An Affinity-Based Image Retrieval System for Multimedia Authoring and Presentation," in *Proc. of The 11th Annual ACM Intl. Conf. on Multimedia*, pp. 446-447, Berkeley, CA, USA.

Chen, S.-C., Zhang, C., et al. (2003). "A Three-Tier System Architecture Design and Development for Hurricane Occurrence Simulation," in *Proc. of IEEE Intl. Conf. on Information Technology: Research and Education (ITRE 2003)*, Newark, New Jersey, USA, pp. 113-117.

Chen, S.-C., Zhang, C., et al. (2003). "Information Reuse and System Integration in the Development of a Hurricane Simulation System," in *Proc. of the 2003 IEEE Intl. Conf. on Information Reuse and Integration (IRI'2003)*, pp. 535-542, Las Vegas, Nevada, USA.

Shyu, M.-L., Chen, S.-C., Chen, M., Zhang, C., and Sarinnapakorn, K. (2003). "Image Database Retrieval Utilizing Affinity Relationships," in *Proc. of the First ACM Intl. Workshop on Multimedia Databases (ACM MMDB'03)*, pp. 78-85, New Orleans, Louisiana, USA.

Shyu, M.-L., Chen, S.-C., Chen, M., Zhang, C., and Shu, C.-M. (2003). "MMM: A Stochastic Mechanism for Image Database Queries," in *Proc. of IEEE Fifth Intl. Symposium on Multimedia Software Engineering (MSE2003)*, pp. 36-44, Taichung, Taiwan, ROC.

Chen, S.-C., Zhang, C., et al., (2004). "A Web-Based Distributed System for Hurricane Occurrence Simulation," *Software: Practice and Experience*, Volume 34, Issue 6, pp. 549-571.

Zhang, C. and Yang, C. (1999). "Digital Images Compression Using Sub-band Decomposition With Morphological Filters," *Chinese Computer Application and Research Journal*.

Zhang, C. and Yang, C. (2000). "Adaptive Sub-band Decomposition Using Morphological Filters," *Chinese Journal of IMAGE AND GRAPHICS*, Vol.5, No.3, pp.191-195.

Zhang, C., Luo, L., Chen, S.-C., and Shyu, M.-L. (2001). "Supporting Information Reuse for Video Data Indexing by Using Color Information," in *Proc. of the 3rd Intl. Conf. on Information Reuse and Integration (IRI-2001)*, pp. 92-96, Las Vegas, Nevada, USA.

Zhang, C., Chen, S.-C., and Shyu, M.-L. (2003). "PixSO: A System for Video Shot Detection," in *Proc. of the 4th IEEE Pacific-Rim Conf. on Multimedia*, pp. 1-5, Singapore.

Zhang, C., Chen, S.-C., Shyu, M.-L., Peeta, S. (2003). "Adaptive Background Learning for Vehicle Detection and Spatio-Temporal Tracking," in *Proc. of the 4th IEEE Pacific-Rim Conf. On Multimedia*, pp. 1-5, Singapore.

Zhang, K., Chen, S.-C., Whitman, D., Shyu, M.-L., Yan, J., and Zhang, C. (2003). "A Progressive Morphological Filter for Removing Non-Ground Measurements from Airborne LIDAR Data," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 41, Issue 4, pp. 872-882.