

國立交通大學

資訊科學與工程研究所

碩 士 論 文

籃球影片之場景偵測及其在戰術分析之應用

Scene Change Detection of Basketball Video and
Its Application in Tactics Analysis



研 究 生：田敏君

指 導 教 授：李素瑛 教授

中 華 民 國 九 十 五 年 六 月

籃球影片之場景偵測及其在戰術分析之應用

Scene Change Detection of Basketball Video and
Its Application in Tactics Analysis

研究生：田敏君

Student：Min-Chun Tien

指導教授：李素瑛 教授

Advisor：Prof. Suh-Yin Lee

國立交通大學
資訊科學與工程研究所
碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

國立交通大學

研究所碩士班

論文口試委員會審定書

本校 資訊科學與工程 研究所 田敏君 君

所提論文：籃球影片之場景偵測及其在戰術分析之應用

Scene Change Detection of Basketball Video and
Its Application in Tactics Analysis

合於碩士資格水準、業經本委員會評審認可。

口試委員：

李素瑛

杭學鳴

吳坤榮

指導教授：

李素瑛

所長：

曾文貴

中華民國九十五年六月十二日

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the thesis prepared by Min-Chun Tien entitled Scene Change Detection of Basketball Video and Its Application in Tactics Analysis and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Master of Science.

Committee Members:

_____ Suh-Yin Lee

Hsueh-Hong Hwang _____
Quen-Zong Wu

Thesis Advisor: Suh-Yin Lee

Director: Wu-Gung Jung

Date: _____

國立交通大學

博碩士紙本論文著作權授權書

(提供授權人裝訂於全文電子檔授權書之次頁用)

本授權書所授權之學位論文，為本人於國立交通大學資訊科學與工程研究所 _____ 組，94 學年度第 7 學期取得碩士學位之論文。

論文題目：籃球影片之場景偵測及其在戰術分析之應用
指導教授：李素瑛

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：_____，請將論文延至____年____月____日再公開。

授權人：田敏君

親筆簽名： 田敏君

中華民國 95 年 7 月 19 日

國立交通大學

博碩士論文全文電子檔著作權授權書

(提供授權人裝訂於紙本論文書名頁之次頁用)

本授權書所授權之學位論文，為本人於國立交通大學資訊科學與工程研究所 _____ 組，94 學年度第 2 學期取得碩士學位之論文。

論文題目：籃球影片之場景偵測及其在戰術分析之應用
指導教授：李素瑛

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：

本校及台灣聯合大學系統區域網路	■ 立即公開
校外網際網路	■ 立即公開

■ 全文電子檔送交國家圖書館

授權人：田敏君

親筆簽名： 田敏君

中華民國 95 年 7 月 19 日

國家圖書館
博碩士論文電子檔案上網授權書

(提供授權人裝訂於紙本論文本校授權書之後)

ID:GT009317519

本授權書所授權之論文為授權人在國立交通大學資訊科學與工程研究所
94 學年度第__學期取得碩士學位之論文。

論文題目：籃球影片之場景偵測及其在戰術分析之應用
指導教授：李素瑛

茲同意將授權人擁有著作權之上列論文全文(含摘要)，非專屬、無償
授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數
位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上
載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或
列印。

※ 讀者基於非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關
規定辦理。

授權人：田敏君

親筆簽名：田敏君

民國 95 年 7 月 19 日

籃球影片之場景偵測及其在戰術分析之應用

研究生: 田敏君

指導教授: 李素瑛 教授

國立交通大學資訊科學與工程研究所

摘要

運動影片之分析是近年來多媒體影像處理領域中一項重大議題，其中籃球影片由於場景與場地的複雜度較高，成為最富挑戰性的研究。目前已有相關論文利用事件偵測技術進而找尋比賽精華片段，然而對於專業籃球教練與球員，觀看籃球影片的目的則須提升至戰術分析。因此我們運用以GOP為基礎之場景變換偵測方法找出關鍵畫面，將影片切割成多個片段，並以球場主要顏色分布及片段長度作為場景分類的依據，分出近景、中景與遠景三類片段。取出含有較多比賽資訊的遠景片段作進一步分析，利用顏色、形狀等資訊找出可能是籃球的區塊並追蹤球的軌跡，最後利用相機參數估算以及球軌跡之物理特性將二維軌跡對應到三維真實球場，並推論可能的出手點位置。

檢索詞： 場景變換偵測、場景分類、軌跡追蹤、相機參數估

Scene Change Detection of Basketball Video and Its Application in Tactics Analysis

Students: Min-Chun Tien

Advisor: Prof.Suh-Yin Lee

Institute of Computer Science and Information Engineering

National Chiao Tung University

Abstract

Sports video analysis has been a major issue of multimedia in recent years. For basketball videos, most researches only put emphasis on searching highlights of the game since the content of basketball video is too complicated. In order to look for more information of tactics from basketball videos, we propose a system that can automatically segment a basketball video into several clips by GOP-based scene change detection method. The length of each clip and the number of dominant color pixels of each frame could be used to classify shots into close-up view, medium view, and full-court view. We choose full-court view shots to do advanced analysis such as tracking the ball, and finding the transformation parameters from 3D real-world court to 2D image by camera calibration techniques. After that, we match the 2D ball trajectory to the corresponding coordinate in a real-world court and compute the statistics of shooting positions. Eventually we obtain information of the most possible shooting positions.

Index Terms: scene change detection, shot classification, tracking, camera calibration

Acknowledgment

I greatly appreciate the kind guidance of my advisor, Prof, Suh-Yin Lee. Without her graceful suggestions and encouragement, I would not complete this thesis. Besides, thanks are extended to all the members in the Information System Laboratory for their suggestion and instruction, especially Mr. Ming-Ho Hsiao, Mr. Hua-Tsung Chen and Mr. Yi-Wen Chen. Finally, I would like to express my appreciation to my family and my friends for their supports and consideration. This thesis is dedicated to them.



Table of Contents

摘要.....	i
Abstract.....	ii
Acknowledgment.....	iii
Table of Contents	iv
Lists of Figures	vi
Lists of Tables	viii
Chapter 1 Introduction.....	1
1.1 Motivation and Overview	1
1.2 Organization.....	1
Chapter 2 Background and Related Work.....	3
2.1 Overview of MPEG Standard.....	3
2.2 Related Work in Sports Video Analysis	5
2.3 Related Work in Tracking	8
2.4 Related Work in Camera Calibration	8
Chapter 3 Scene Change Detection of Basketball Video and Its Application in Tactic Analysis	15
3.1 Scene Change Detection Using GOP-Based Method	16
3.2 Shot Classification	21
3.3 Ball Candidate Search.....	25
3.4 Ball Tracking	29
3.5 Camera Calibration	31
Chapter 4 Experiment	45
4.1 Experimental Result of Scene Change Detection and Shot Classification	45
4.2 Experimental Result of Tracking the Ball.....	46

4.3 Experimental Result of Camera Calibration and Shooting Position.....47

Chapter 5 Conclusion and Future Work49

Bibliography50



Lists of Figures

Fig. 2-1 An example of GOP structure in MPEG coding.	4
Fig. 2-2 Image geometry showing relationship between 3D points and 2D image plane pixels.	9
Fig. 3-2 Structure of GOP.	16
Fig. 3-3 The workflow of the scene change detection method.	17
Fig. 3-4 Scene change occurs on I-frame or P-frame.	19
Fig. 3-5 Scene change occurs on the first B-frame.	19
Fig. 3-6 Scene change occurs on the second or later B-frame.	20
Fig. 3-7 Flowchart of dominant color region detection algorithm.	21
Fig. 3-8 The histograms of dominant color (court color).	22
Fig. 3-9 Three kinds of view shots.	24
Fig. 3-10 The process of ball candidate identification.	25
Fig. 3-11 Observation of the color of basketball.	26
Fig. 3-12 Background subtraction of the image.	27
Fig. 3-13 Result of ball candidate search after color and shape filtering.	28
Fig. 3-14 Ball Candidate Reduction.	29
Fig. 3-15 Result of ball candidate reduction.	29
Fig. 3-16 Tracking process.	30
Fig. 3-17 Result of tracking ball.	31
Fig. 3-18 Correspondence between 2D court image and 3D court model.	32
Fig. 3-19 Line correspondences between image and basketball court model.	33
Fig. 3-20 The flow chart of camera calibration.	34
Fig. 3-21 Part of the image containing a white line pixel.	35
Fig. 3-22 White line pixel detection.	36
Fig. 3-23 Applying line-structure constraint.	37
Fig. 3-24 Hough transform for straight lines.	38
Fig. 3-25 Line detection by Hough Transform.	38
Fig. 3-26 Six intersections of the court line candidates.	39
Fig. 3-27 Detection of line-segment boundaries.	40
Fig. 3-28 Boundaries of the backboard top-line.	40
Fig. 3-29 Match the eight points to the court model and calculate the camera parameters.	41
Fig. 3-30 Camera parameter prediction.	42
Fig. 3-31 Extract possible 2D shooting trajectory.	43
Fig. 3-32 Choose three points on the 2D trajectory.	44
Fig. 4-1 The tracking result of a shot without camera motion.	46

Fig. 4-2 The tracking result of a shot with camera motion.....47
Fig. 4-3 The 2D location of the points for camera calibration and the backboard
position.....47
Fig. 4-4 The real 2D ball trajectory.....48
Fig. 4-5 The obtained shooting position in 3D court model.....48



Lists of Tables

Table. 1 Shot classification results of two testing sequences.46



Chapter 1

Introduction

1.1 Motivation and Overview

Before basketball games, the coach and players have to watch the basketball videos of the opponent and look for their defense rank, offense strategies, the offense habitual behavior of the top players and the most possible shot positions of that team. For human eyes, it is not difficult to observe above information. However, it is obviously time-consuming and exhausting to watch a 40 minutes long basketball video. Therefore, we propose an approach which could automatically segment the video into clips by detecting the scene change, and classify all clips into three kinds of view shots: close-up view, medium view, and full-court view. Since full-court view shot contains more information, we use such kind of clips to do advanced analysis. The system then tracks the ball in each clip and finds the transformation parameters from 3D real-world court to 2D image by camera calibration techniques. With the calibration parameters and the physical property of the ball in 3D real-world coordinate, we can extract the 3D trajectory of basketball and gather statistics to conclude the most possible shooting positions in the games, which provide useful information for the coach.

1.2 Organization

The rest of this thesis is organized as follows. In Chapter 2, we introduce some

background knowledge required for video technology. We also survey some previous related works in sports video analysis, event detection, shot classification, image enhancement, object extraction, object tracking and camera calibration. Chapter 3 introduces an algorithm to detect scene changes of videos and constructs a shot classification model to identify clips into close-up view, medium view, and full-court view shots. Chapter 3 also shows the tracking process of the ball and players, describes how the points in the 2D video image correspond to the determined 3D court model, and infers the shot position. Chapter 4 presents the experimental result and discussion. Finally, we conclude the thesis and describe the future work in Chapter 5.

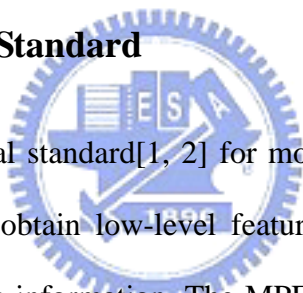


Chapter 2

Background and Related Work

In Chapter 2, we introduce the background knowledge for video technology and some previous related works in sports video analysis. In section 2.1, we present an overview of MPEG standard. In the following sections, some related works in shot classification, event detection, tracking and camera calibration for sports video are described.

2.1 Overview of MPEG Standard



MPEG is the international standard [1, 2] for moving picture video compression. In compressed domain, we can obtain low-level features such as DC values and motion vectors to infer more semantic information. The MPEG video syntax support three types of coded frames or pictures, intra (I-) pictures, coded separately by themselves; predictive (P-) pictures, coded with respect to the immediately previous I- or P-pictures; and bi-directionally predictive (B-) pictures, coded with respect to the immediately previous I- or P-pictures as well as the immediately next I- or P-pictures. **Fig. 2-1** shows an example picture structure in MPEG video coding that uses three B-pictures between two reference (I- or P-) pictures. In MPEG video coding, an input video sequence is divided into units of groups of pictures (GOPs). Each GOP typically starts with an I-picture and the rest of the GOP is made up of P-pictures and B-pictures in a certain arrangement. A GOP serves as a basic access unit, and the start picture, an I-picture, is the entry point to facilitate random access.

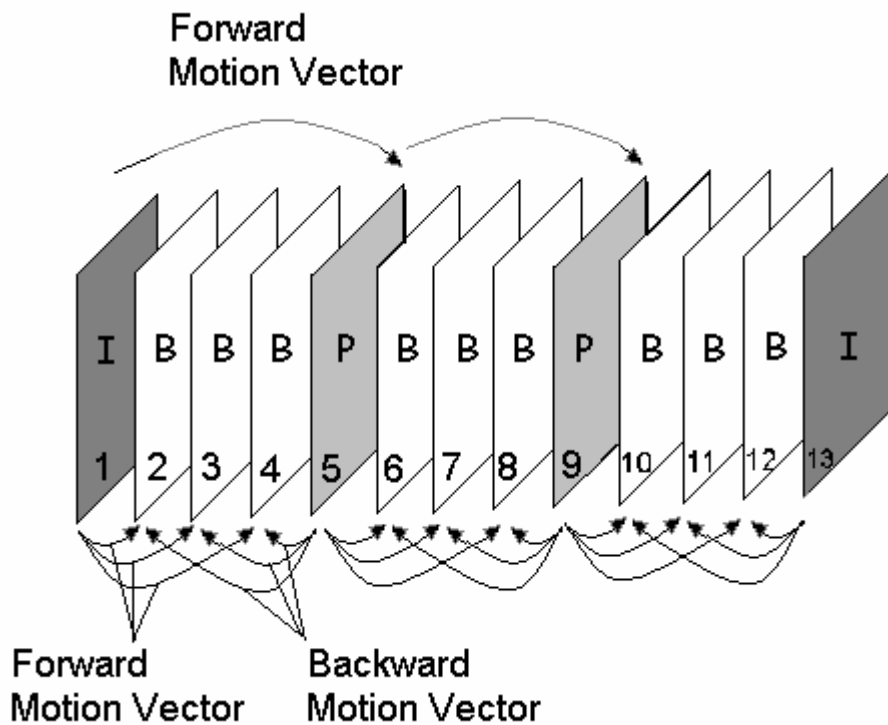
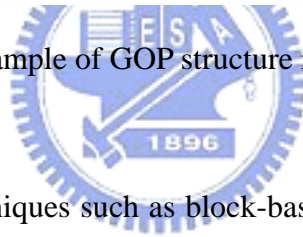


Fig. 2-1 An example of GOP structure in MPEG coding.



MPEG coding uses techniques such as block-based transform coding, predictive coding, entropy coding, motion-compensated interpolation, etc. Among the above techniques, block-based transform coding and motion compensation are the most important ones.

Block-based transform coding reduces the spatial redundancy in digital video. By 8x8-block discrete cosine transform (DCT), pixels in spatial domain are transformed to frequency coefficients, and the substantial correlation between neighbor pixels is greatly reduced. Coefficients in frequency domain need not be coded with full accuracy and can be entropy-coded for compression. The first coefficient of each block is called DC value, which contains most information of that block.

Motion compensation reduced the temporal redundancy in digital videos. In the current frame, a best match for each block in previous frame will be found, and the difference between the block and the match will be coded. MPEG-1 and MPEG-2 apply backward and bi-directional motion compensations which provide higher coding efficiency.

In the proposed framework, we will use the DC values, Rb ratio (the ratio of number of backward motion vectors over the number of forward motion vectors), and Rf ratio (the ratio of number of forward motion vectors over the number of backward motion vectors), to detect scene change of basketball video.

2.2 Related Work in Sports Video Analysis

Due to tremendous commercial potentials, sports video has been widely studied. Y.H. Gong *et al.*[3] proposed a system that can automatically parse soccer video programs using domain knowledge. The parsing process was mainly built upon line mark recognition and motion detection. They categorized the position of a play into several predefined classes by recognizing the compound line pattern with signature method. The motion vectors field is used to infer the play positions for those scenes without line marks. Despite the strong semantic indexes from the categorization of play positions, they have yet to address the following two problems: 1) how to identify different camera angle and shooting scale, otherwise the line mark recognition cannot be robust; 2) how to determine reasonable segments for processing. Frame-by-frame processing is improper for large amounts of video data, and moreover, the customized algorithms have to undergo much noise from unrelated segments. As discussed above, video shots have furnished us with natural segments.

Y.P. Tan *et al.*[4] introduced camera motion estimation into the analysis and

annotation of MPEG basketball video. They estimated camera motion directly from MPEG motion vectors fields. By measuring the variation of the estimated pan rate and the persistence of accumulated directional pan, a semantic annotation was generated, such as Fast breaks (FB), Full court advances (FCA), Close-up shots, etc. No doubt camera motion is an important clue to annotate video and select interesting video segments. However, if the projective transformation parameter is utilized to recover camera motion, the camera motion does not necessarily dominate the change in image intensity between frames. Although we can exploit robust statistics technique to deal with noisy motion vectors, the global motion estimation may be poor if the underlying motion vectors field is totally unreliable, for instance with unstructured scenes or the loss of focus caused by fast camera movement. Hence we must evaluate the quality of motion vectors fields before applying regression procedure.

D. Zhong *et al.*[5] proposed a general framework to analyze the temporal structure of live broadcasted sports videos. They formulated structure analysis as the problem of detecting the fundamental views by using supervised learning and domain-specific rules. For instance, in tennis, we can determine the serve scene by detecting the court view. They utilized the techniques of color-filtering, object segmentation and edge verification. This kind of view-based approach depends on two assumptions: 1) the fundamental views consist of unique visual cues, such as color, motion, and object layout; 2) the basic units start with a special scene. Since sports videos usually feature a fixed number of camera views, it is useful to perform frame-level view analysis. However, the complete view analysis does not make full use of motion vectors information. Despite the combination of view analysis and individual motion field analysis [6], it is difficult to capture the distinguished dynamic characteristics from an individual motion vector field, which could be contaminated. An alternative is to perform motion analysis at the shot level and capture the dominant

motion pattern within one shot.

We have discussed several representative works in sports video analysis with an emphasis on motion information usage. Now let us briefly review some other related works. G. Sudhir *et al.* [7] developed an approach for automatic classification of tennis video. They used the automatically extracted tennis court lines and the players' position for highlevel reasoning where the relative positions of the two players are mapped to high-level events such as baseline-rallies, passing-shot, etc. W. Hua *et al.* [8] introduced the maximum entropy scheme to integrate multimedia clues for baseball scene classification. J. Assfalg *et al.* [9] tried to use HMM for modeling the transitions between the states of camera motion patterns or players locations for each soccer highlights. Once all the HMMs are trained, the maximum likelihood function is computed to recognize an unknown video shot.

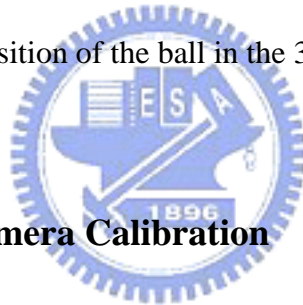
Below, we will review related works about shot classification. C.W. Ngo *et al.* [10] proposed a hierarchical clustering approach by aggregating shots with similar motion and color features. By coupling clustering issues with retrieval problems, the clustering structure inherently provides an indexing scheme for retrieval. Through manual investigation of the clustering results, they have tried to explain the semantic meanings for each cluster. However, this kind of clustering procedure did not establish direct relationships between resulting shot clusters and clear semantic meaning. Moreover, the clustering-based approach did not provide a feasible solution for classifying unknown video shots into known shot classes with strong semantic meanings.

J. Assfalg *et al.* [11] proposed an approach for semantic annotation of sports video according to elements of visual content at different layers of semantic significance. They used neural network classifiers to perform the classification of visual shot features (e.g. edge, segment, and color features, etc.) in terms of playing

field, player, and audience classes. Such classification scheme is based on the key frames. Motion information has never been used.

2.3 Related Work in Tracking

Most researches track players by using template matching[12-14], however, users often have to specify the position of players manually during occlusion. Moreover, most methods do not track a ball or only track a ball in easy cases [15, 16]. [17] proposed a system which could automatically track players and a ball in soccer games in the images taken by fixed camera. The method proposed in [17] can also cope with occlusion and the posture change and can calculate the position of the players on the field and the position of the ball in the 3D space.



2.4 Related Work in Camera Calibration

The mapping between the observed image and the real-world coordinates can be taken to be a projective transform. With a set of positions well-defined in an image, we can obtain the transformation parameters. Lines provide a good feature for calibration when the sport has specific line structure on the playfield. In early work[7], a method to detect four predefined points on a tennis court for calibration is proposed. However, the algorithm has to be initialized manually and it is not robust against occlusions of the court lines connecting these four points. In[18, 19], more detection of court (for soccer videos) is described, but it requires computationally complex initialization because of using an exhaustive search through the parameter space. [20] applies a Hough transformation to detect court lines for calibration, but the use of heuristics to assign the detected lines in the court model is not suitable for general

case. [21] uses a combinatorial search to establish correspondences between the lines that were detected with a Hough transform and the court model. This provides a high robustness even for bad lightening conditions or large occlusions.

2.4.1 Transformation from 3D to 2D

We typically use a pinhole camera model that maps points in a 3D camera frame to a 2D projected image frame. Using similar triangles, we can relate 2D image plane and 3D real world space coordinates by a transformation matrix. As **Fig. 2-2** shows, \bar{X}_c , \bar{Y}_c and \bar{Z}_c are three axes in 3D camera coordinates; \bar{x} and \bar{y} are the axes in 2D image plane. We have 3D points $P = (0, Y_c, Z_c)$ and $Q = (X_c, 0, Z_c)$ which project onto the image plane at $p = (0, y)$ and $q = (x, 0)$. O_c is the origin of camera coordinate system, known as the center of projection (COP) of the camera. The origin of the image plane is O . The camera focal length is denoted by f_c .

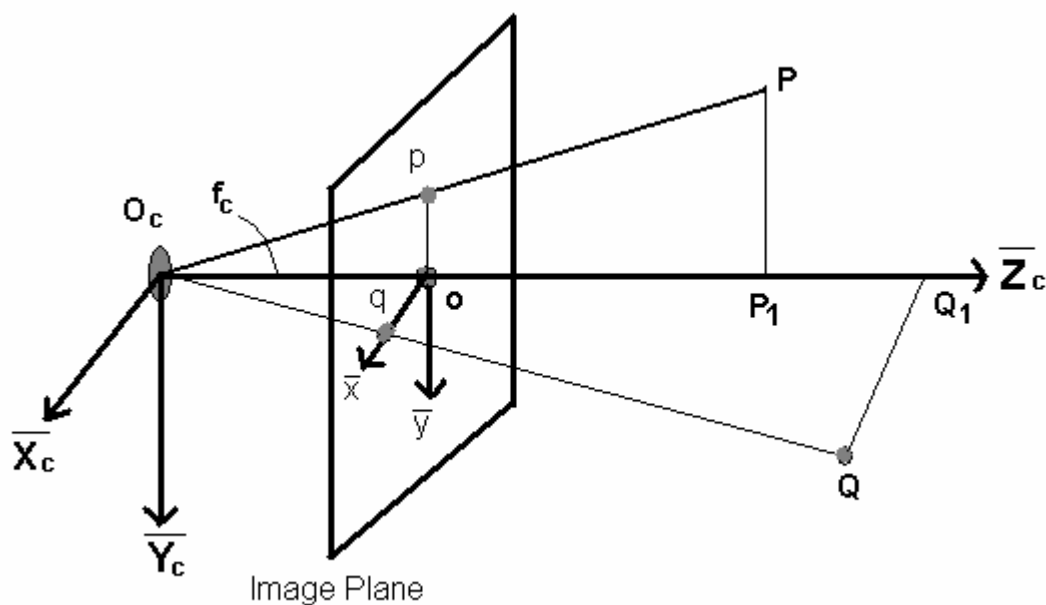


Fig. 2-2 Image geometry showing relationship between 3D points and 2D image plane pixels.

From similar triangles PP_1O_c and poO_c and also similar triangles QQ_1O_c

and qoO_c , we can write down the relationships:

$$\frac{X_C}{x} = \frac{Z_C}{f_c} ; \frac{Y_C}{y} = \frac{Z_C}{f_c}$$

$$\rightarrow x = f_c \frac{X_C}{Z_C} ; y = f_c \frac{Y_C}{Z_C}$$

If $f_c=1$, note that perspective projection is just scaling a world coordinate by its Z value. All 3D points along a line from the COP through a position (x,y) will have the same image plane coordinates. We can also describe perspective projection by the matrix equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} s \cdot x \\ s \cdot y \\ s \end{bmatrix} = \begin{bmatrix} f_c & 0 & 0 & 0 \\ 0 & f_c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \quad (1)$$

where s is a scaling factor and $[x,y,1]^T$ are the projected coordinates in the image plane.

We can generate image space coordinates from projected camera space coordinates. However, in image processing, we use the actual pixel values. Hence we have to transform the 2D image coordinates (x,y) to pixel values (u,v) by scaling the camera image plane coordinate in the x and y directions, and adding a translation to the origin of the image space plane. We can call these scale factors D_x and D_y , and the translation to the origin of the image plane as (u_0, v_0) . If the pixel coordinates of the projected point (x,y) are (u,v) , then we can write:

$$\frac{x}{D_x} = u - u_0 ; \frac{y}{D_y} = v - v_0$$

$$\rightarrow u = u_0 + \frac{x}{D_x} ; v = v_0 + \frac{y}{D_y}$$

where D_x and D_y are the physical dimensions of a pixel and (u_0, v_0) is the origin of the pixel coordinate system. $\frac{x}{D_x}$ and $\frac{y}{D_y}$ are simply the number of pixels, and we center them at the pixel coordinate origin. We can also put this into matrix form as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

Combine (1) and (2), we obtain:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = \begin{bmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_c & 0 & 0 & 0 \\ 0 & f_c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (3)$$

Camera calibration is used to find the mapping from 3D to 2D image space coordinates. There are 2 approaches:

- Method 1: Find both extrinsic and intrinsic parameters of the camera system. However, this can be difficult to do.
- Method 2: An easier method is the “Lumped” transform. Rather than finding individual parameters, we find a composite matrix that relates 3D to 2D. Given Eq.(3), we can derive a 3x4 calibration matrix C :

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{D_x} & 0 & u_0 \\ 0 & \frac{1}{D_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_c & 0 & 0 & 0 \\ 0 & f_c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4)$$

We apply method 2 which finds the 11 parameters to transform an arbitrary 3D world point to a pixel in a computer image:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \cdot \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \quad (5)$$

C is a single 3x4 transform that we can calculate empirically.

$$\underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\substack{3D \text{ coordinates} \\ 4 \times 1}} = \underbrace{\begin{bmatrix} u \\ v \\ w \end{bmatrix}}_{\substack{2D \text{ image} \\ \text{coordinates} \\ 3 \times 1}} \stackrel{\Delta}{=} \underbrace{\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}}_{\substack{2D \text{ pixel} \\ \text{coordinates} \\ 3 \times 1}} \quad \text{where} \quad \begin{aligned} u' &= \frac{u}{w} \\ v' &= \frac{v}{w} \end{aligned} \quad (6)$$

Multiplying out the equations, we get:

$$\begin{cases} c_{11}x + c_{12}y + c_{13}z + c_{14} = u \\ c_{21}x + c_{22}y + c_{23}z + c_{24} = v \\ c_{31}x + c_{32}y + c_{33}z + c_{34} = w \end{cases} \quad (7)$$

Substituting $u = u'w$ and $v = v'w$, we get:

$$c_{11}x + c_{12}y + c_{13}z + c_{14} = u'(c_{31}x + c_{32}y + c_{33}z + c_{34}) \quad (8)$$

$$c_{21}x + c_{22}y + c_{23}z + c_{24} = v'(c_{31}x + c_{32}y + c_{33}z + c_{34}) \quad (9)$$

➤ If we know all the c_{ij} and x, y, z , we can find u', v' . This means that if we know calibration matrix C and a 3D point, we can predict its image space coordinates.

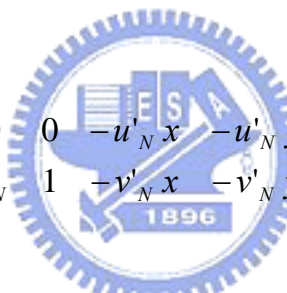
➤ If we know x, y, z, u', v' , we can find c_{ij} . Each 5-tuple gives 2 equations in c_{ij} .

This is the basis for empirically finding the calibration matrix C (more on this later).

- If we know c_{ij}, u', v' , we have 2 equations in x, y and z . The two equations represent two planes in 3-D and form an intersection which is a line. These are the equations of the line emanating from the center of projection of the camera, through the image pixel location (u', v') and containing point (x, y, z) .

Set up a linear system to solve for $c_{ij} : AC = B$

$$\begin{bmatrix}
 x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u'_1 x & -u'_1 y & -u'_1 z \\
 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v'_1 x & -v'_1 y & -v'_1 z \\
 x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u'_2 x & -u'_2 y & -u'_2 z \\
 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v'_2 x & -v'_2 y & -v'_2 z \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & -u'_N x & -u'_N y & -u'_N z \\
 0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & -v'_N x & -v'_N y & -v'_N z
 \end{bmatrix}_{2N \times 11}
 \begin{bmatrix}
 c_{11} \\
 c_{12} \\
 c_{13} \\
 c_{14} \\
 c_{21} \\
 c_{22} \\
 c_{23} \\
 c_{24} \\
 c_{31} \\
 c_{32} \\
 c_{33}
 \end{bmatrix}_{11 \times 1}
 =
 \begin{bmatrix}
 u'_1 \\
 v'_1 \\
 u'_2 \\
 v'_2 \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 u'_N \\
 v'_N
 \end{bmatrix}_{2N \times 1}$$



We can assume $c_{34}=1$

N is the number of points whose 2D and 3D coordinates are known and used to solve for c_{ij} . Each set of points x, y, z, u' , and v' yields 2 equations in 11 unknowns (the c_{ij} 's). To solve for C , A needs to be invertible (square). We can over determine A and find a Least-Squares fit for C by using a pseudo-inverse solution.

If A is $2N \times 11$, where $2N > 11$:

$$\begin{aligned}
 AC &= B \\
 A^T AC &= A^T B \\
 C &= \underbrace{(A^T A)^{-1}}_{\text{pseudo inverse}} A^T B
 \end{aligned}$$

For basketball video, most of the previous work emphasizes on shot classification and event detection[22-25]. In this paper, we want to stress the analysis of tactics.



Chapter 3

Scene Change Detection of Basketball Video and Its Application in Tactic Analysis

In this chapter, we will present the framework of our system as depicted in **Fig. 3-1**. The system architecture has three main parts: Full Court Shot Retrieval, 2D Ball Trajectory Extraction, and 3D Shooting Location Positioning. Full Court Shot Retrieval utilizes scene change detection to cut a video into clips and classifies each clip as close-up view, medium view, or full court view shot. 2D Ball Trajectory Extraction uses all the full court view shots to search the ball candidates and to track the 2D ball trajectory. 3D Shooting Location Positioning applies camera calibration to find the relationship between 2D and 3D points. Therefore, we can extract the 3D trajectory of the basketball. Finally the shooting position could be found.

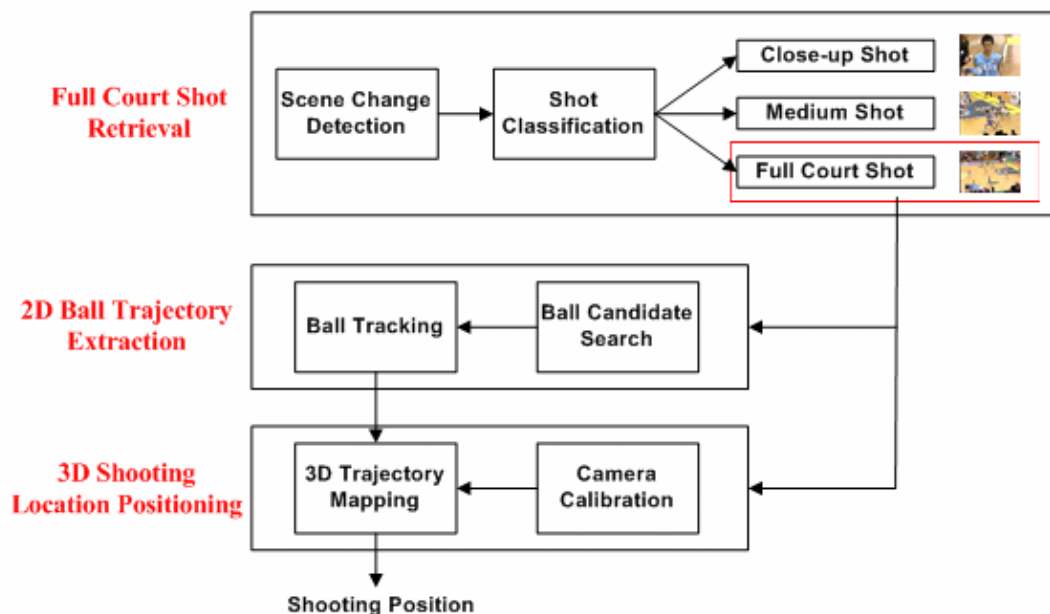


Fig. 3-1 The framework of the system.

Section 3.1 introduces a GOP-based approach to detect scene changes in videos. Section 3.2 constructs a shot classification model to find “full-court view shots”. Section 3.3 shows how to find ball candidates. Section 3.4 represents the tracking process of the ball. In section 3.5, we describe a camera calibration model to establish correspondence between points in the video image and the determined court model.

3.1 Scene Change Detection Using GOP-Based Method

In order to analyze tactics in the basketball video, we have to detect scene change and cut the video into clips. After that, we classify clips into three kinds of shot and choose the full-court view shots to do further processing.

Most of existed approaches detect scene change frame by frame. However, the scene change does not occur on each frame, hence it is not necessary to do frame-wise scene change detection. We use a GOP-based method to improve the efficiency of scene change detection. The format of MPEG- II includes a GOP layer. As **Fig. 3-2** shows, a GOP structure contains the header and an intra-frame coding frame (I-frame) accompanies series of frames in two types including predictive coding frame (P-frame), and bi-directionally predictive coding frame (B-frame).

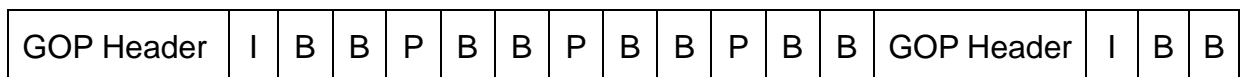


Fig. 3-2 Structure of GOP.

The GOP-based scene change detection approach has two steps[26]. The workflow of this approach is shown in **Fig. 3-3**. In the first step (Inter-GOP scene

change detection), the possible occurrence of scene change is checked GOP by GOP instead of frame by frame. If a GOP is detected having possible scene change, go to the second step. In the second step (Intra-GOP scene change detection), we check whether the scene change exists and find the actual frame where the scene change occurs within the GOP. The detailed process of the two steps is described in Section 3.1.1 and 3.1.2, respectively.

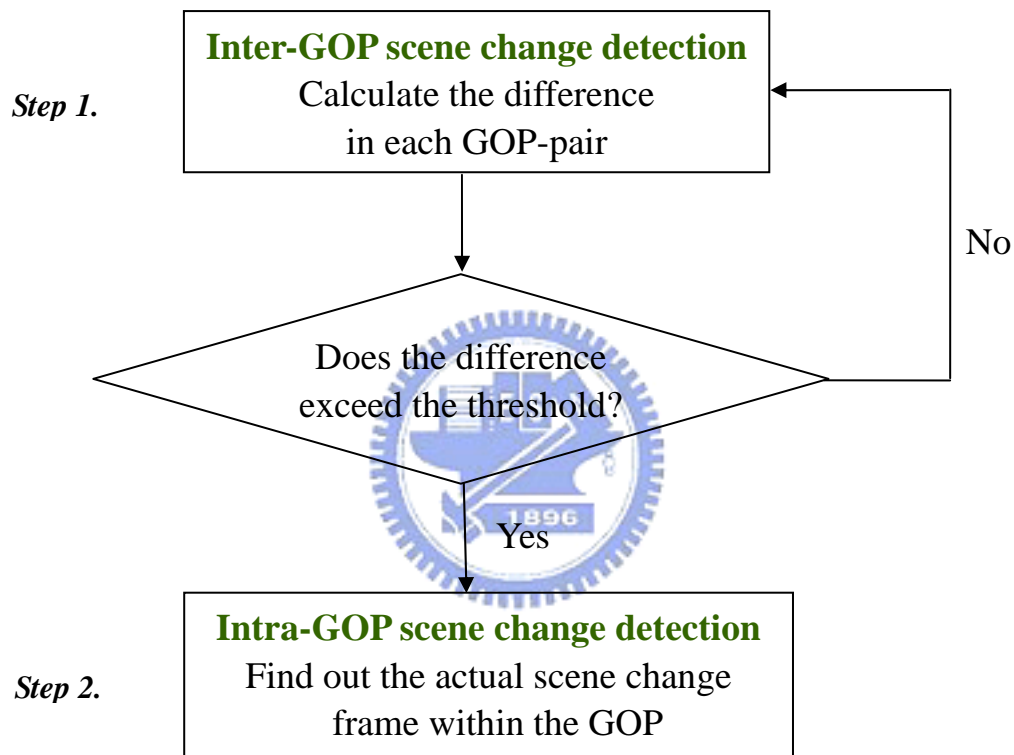


Fig. 3-3 The workflow of the scene change detection method.

3.1.1 Inter-GOP scene change detection

For each I frame, divide it into k sub-regions. Sum the DC values in each sub-region. The image feature of a GOP $g = \{SumDC_{g,i} = \sum_{j=1}^{N_i} DC_{i,j} \mid i = 1, \dots, k\}$, where i is the index of sub-region in I-frame, N_i is the total number of DC values in the i th sub-region, and $DC_{i,j}$ is the j th DC value of sub-region i . The Distance

between two GOPs g and $g+1$ is represented as $D(g, g + 1)$, and the value of $D(g, g+1)$ is computed as follows:

$$mark_i = 1 \quad \text{if } |SumDC_{g,i} - SumDC_{g+1,i}| > threshold_subregion$$

$$mark_i = 0 \quad \text{otherwise}$$

$$D(g, g + 1) = \sum_{i=1}^k mark_i$$

When $D(g, g + 1) \leq threshold_GOP$, which means the successive GOPs are similar, we say no scene change occurs. When $D(g, g + 1) > threshold_GOP$ which means GOP g and GOP $g+1$ are dissimilar, we assume that a possible scene change occurs in GOP $g+1$. However, large difference may be caused by the camera motion and object moving rather than the real scene change. To solve this problem, Intra-GOP scene change detection is proposed.

3.1.2 Intra-GOP scene change detection

The Fast Pure Motion Vector Approach[27] is used for efficient scene change detection within a GOP. This approach only uses motion vectors of B-frames to detect scene change since B-frames are motion-compensated with respect to referential frames. If a B-frame is most similar to previous referential frame, most of the motion vector will refer to forward direction. If a B-frame is most similar to back referential frame, most of the motion vector will refer to backward direction. Two notations are defined below.

Rb : The ratio of number of backward motion vectors over the number of forward motion vectors.

Rf : The ratio of number of forward motion vectors over the number of backward motion vectors.

In a GOP, there are three cases where scene change may occur:

Case1: Scene change occurs on I-frame or P-frame.

Case2: Scene change occurs on the first B-frame between two successive reference frames.

Case3: Scene change occurs on the second or later B-frame between two successive reference frames.

We discuss the three cases and infer the rule to find the actual scene change frame.

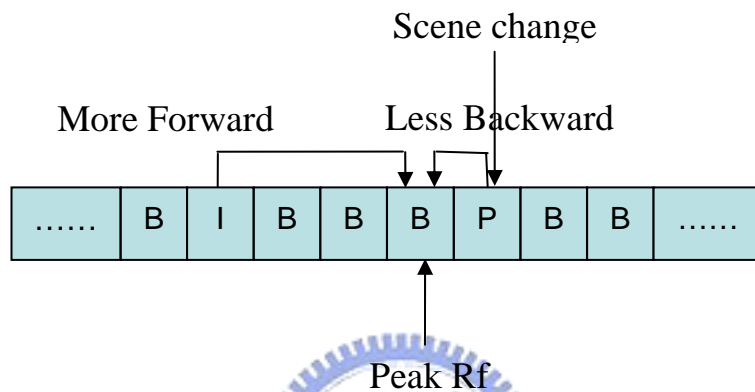


Fig. 3-4 Scene change occurs on I-frame or P-frame.

Case 1 is shown in **Fig. 3-4**. If scene change occurs on I-frame or P-frame, the first previous B-frame will be similar to its previous referential frame. Therefore, most of the motion vectors of the first previous B-frame refer to the forward referential frame, and Rf of the first previous B-frame will be very large and exceed the $threshold_Rf$.

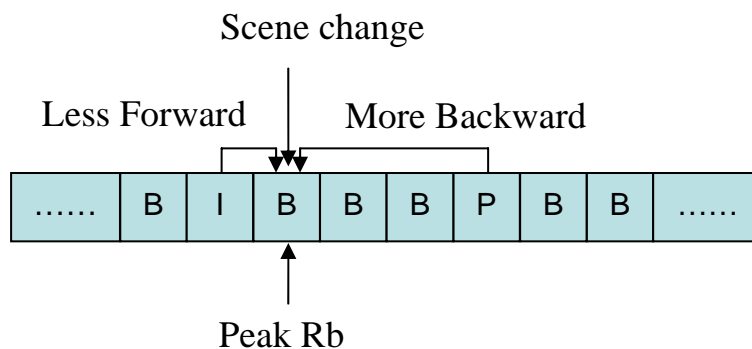


Fig. 3-5 Scene change occurs on the first B-frame.

Case 2 is shown in **Fig. 3-5**. If Scene change occurs on the first B-frame between two successive reference frames, the B-frame itself will be similar to its back referential frame. Therefore, most of the motion vectors of this B-frame refer to the backward referential frame, and Rb of the first B-frame will be very large and exceed the threshold Rb .

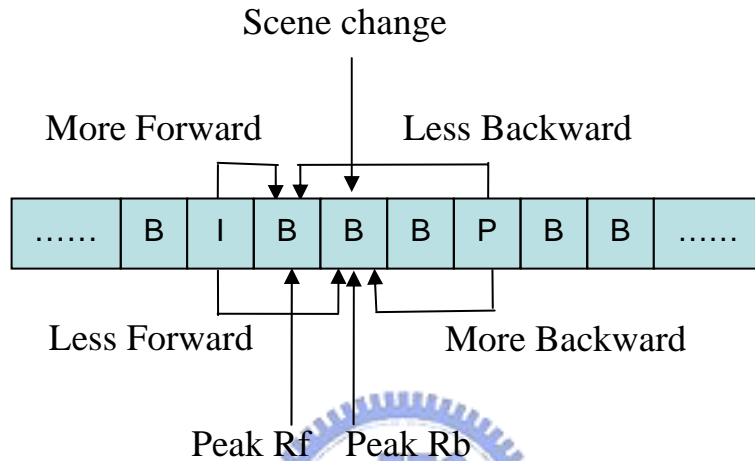


Fig. 3-6 Scene change occurs on the second or later B-frame.

Case 3 is shown in **Fig. 3-6**. If scene change occurs on the second or later B-frame between two successive reference frames, the B-frame itself will be similar to its back referential frame and the first preceding B-frame will be similar to previous referential frame. Therefore, most of the motion vectors of the first preceding B-frame refer to its forward referential frame, and the second B-frame mostly refer to the backward referential frame; i.e. Rf of the first B-frame and Rb of the second B-frame will be very large.

After examining values of Rb and Rf on B-frames, scene changes could be detected in GOP while Rb or Rf on B-frame exceeds the predefined threshold. Some noise such as camera or object moving which leads to possible scene change in the first step can be removed because such kinds of frame are usually not similar to both its previous and back referential frame, and its values of Rb and Rf on B-frames will not exceed the threshold.

3.2 Shot Classification

To analyze tactics in basketball video, we must have enough information to support the inference of possible shot positions. Three kinds of basketball shots such as close-up view, medium view and full court view are predefined. We will use the full court view shots which contain more information of the game to do better analysis. Some related works in shot classification are described in Chapter 2, and we apply the main idea of dominant color ratio [28].

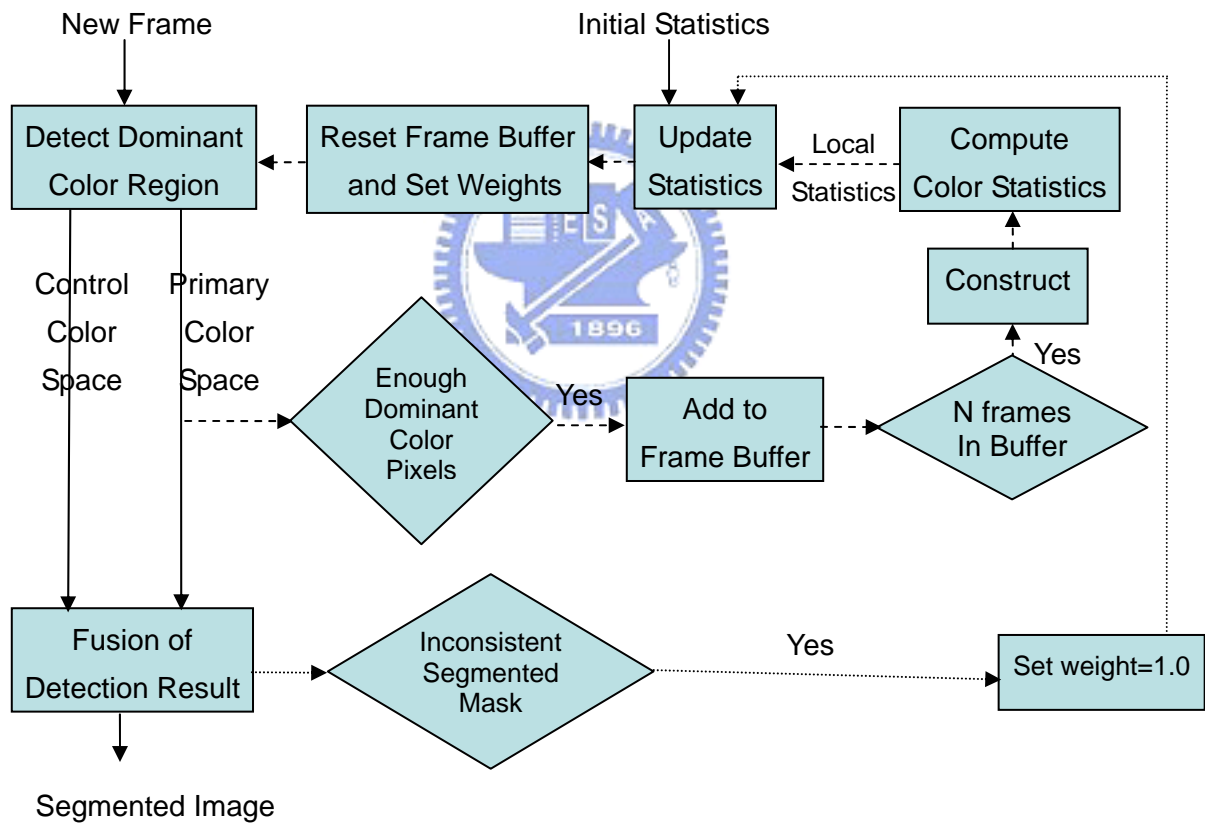


Fig. 3-7 Flowchart of dominant color region detection algorithm.

The flowchart of dominant color region detection algorithm is shown in **Fig. 3-7**. At start-up, the system computes initial statistics and the values of several parameters for each color space from the frames in the training set. After the initialization of

parameters, dominant color region for each new frame is detected in both control and primary color spaces. Segmentation results in these spaces are used by the fusion algorithm to obtain more accurate final segmentation mask. The rest of the blocks in the flowchart are utilized for adaptation of primary color space statistics by two feedback loops. The inner feedback loop, connected with the dashed lines, computes local statistics in primary color space and captures local variations, whereas the other feedback loop, connected with the dotted lines, becomes active when segmentation results conflict with each other, which indicates drifting of local statistics from true statistics in primary color space. The activation of this outer feedback loop resets primary color statistics to their initial values.

The RGB and HSI histograms of dominant color (the color of the court) are illustrated in **Fig. 3-8**, where the x-axis represents the quantized bins for each color component, and the y-axis is the number of pixels in corresponding bin. The ratio of dominant color pixels can be exploited to identify which kind of shot the current frame belongs to.

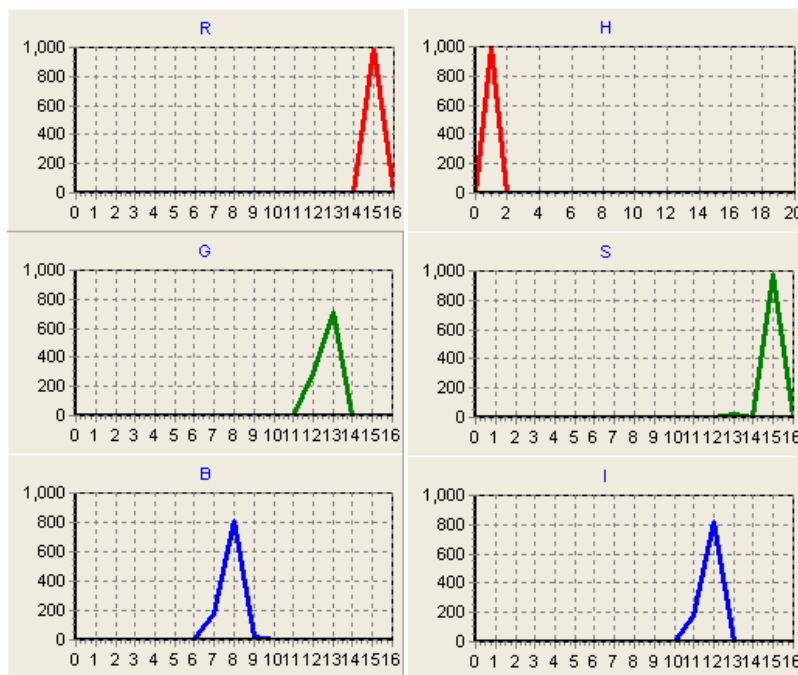
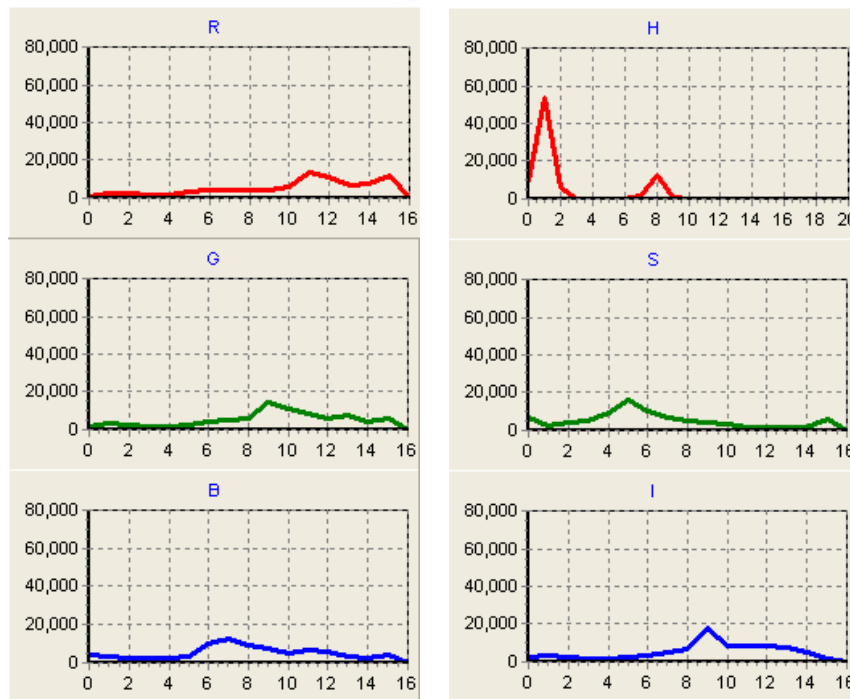
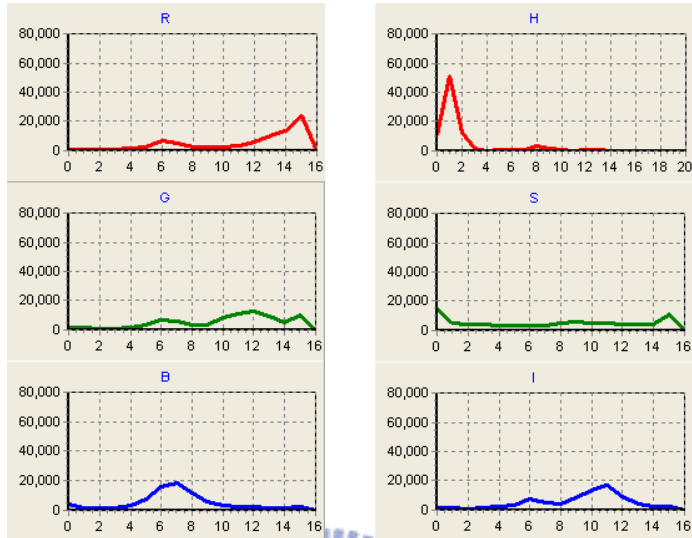


Fig. 3-8 The histograms of dominant color (court color).

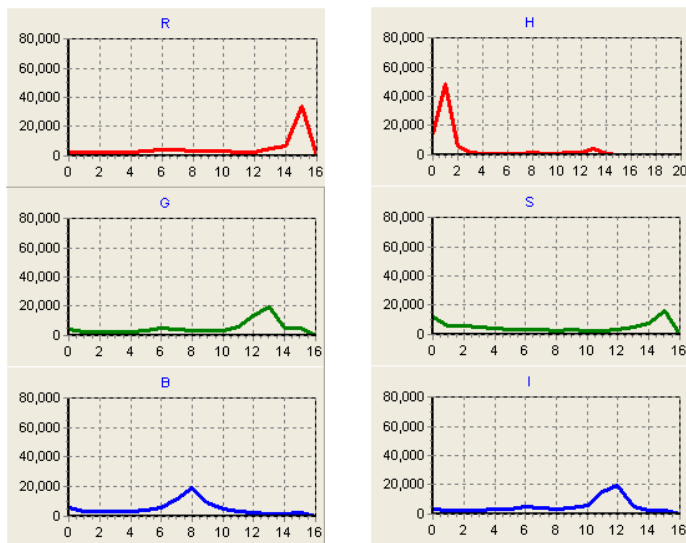
Fig. 3-9 (a) shows an instance of close-up view and its histograms in RGB and HSI color space. Since a close-up view shot contains less part of court, the color distribution of a close-up view image is much different from the color distribution of the dominant color. (b) shows an instance of medium view and its histograms in RGB and HIS color space. A medium view shot have a moderate amount of court pixels, hence the distribution of a medium view image is a little similar to the color distribution of the dominant color. (c) shows an instance of full court view and its histograms in RGB and HIS color space. A full court view shot usually implies a large number of court pixels, and consequently the distribution of a full court view image is much similar to the color distribution of the dominant color.



(a) Close-up View



(b) Medium View



(c) Full Court View

Fig. 3-9 Three kinds of view shots.

After obtaining the scene change frames of basketball video, we identify the full court view shot since most information about tactics is involved in this kind of shots. For full-court view shots, the ratio of dominant color pixels should be large. Therefore, with a threshold T_{ratio} , we can filter out Close-up view or Medium view.

Since clips with longer length comprise more information of tactics, we select long clips having length bigger than L_{min} , and use these clips to achieve better analysis.

3.3 Ball Candidate Search

Identifying a ball in the image is difficult because the ball is usually small and sometimes moves very fast. The process of ball candidate identification is described in **Fig. 3-10**. For each frame in a full court view clip, we use color filtering, background subtraction, morphological operation, shape and size filtering to find possible ball candidates. The ball candidate reduction step is applied to simplify the tracking process by avoid too much ball candidates.

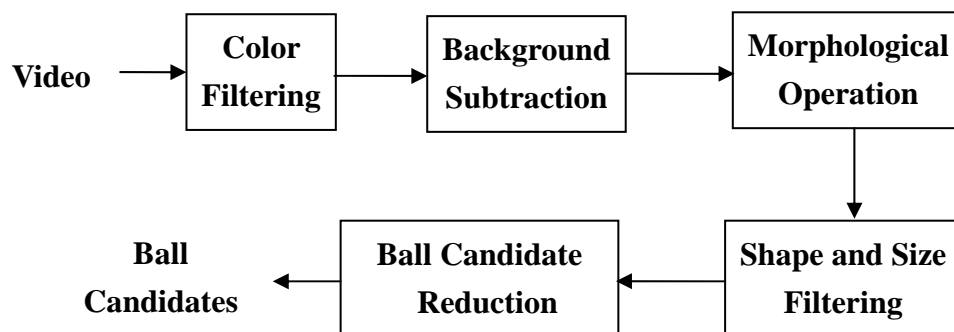


Fig. 3-10 The process of ball candidate identification.

In the color filtering step, color feature is utilized for ball pixel identification. For each frame, the image is divided into overlapping blocks of size $M \times N$. The

overlapping is achieved by moving the center of the first block by $m \times \frac{M}{2}$ and $n \times \frac{N}{2}$ to span the whole image, where m and n are arbitrary integers. Calculate the averages of R and G values in each block, and identify whether this block contains ball color.

However, the color of a basketball is not steady owing to the light condition and the angle of view. After choosing ball blocks from different video source manually and calculating their mean values of R, G, B, H, S, and I components, we observe that the R and G values of the basketball are in the range $110 \leq r \leq 175$ and $70 \leq g \leq 135$. Therefore, we identify blocks having average R and G values in the basketball color range to be possible ball blocks. **Fig. 3-11** demonstrates some cases of ball block color. In case (a), the ball is stationary and its color is similar to the real ball color. Case (b), (c), and (d) show the moving ball color. Since the ball moves fast, its color is influence by the background.

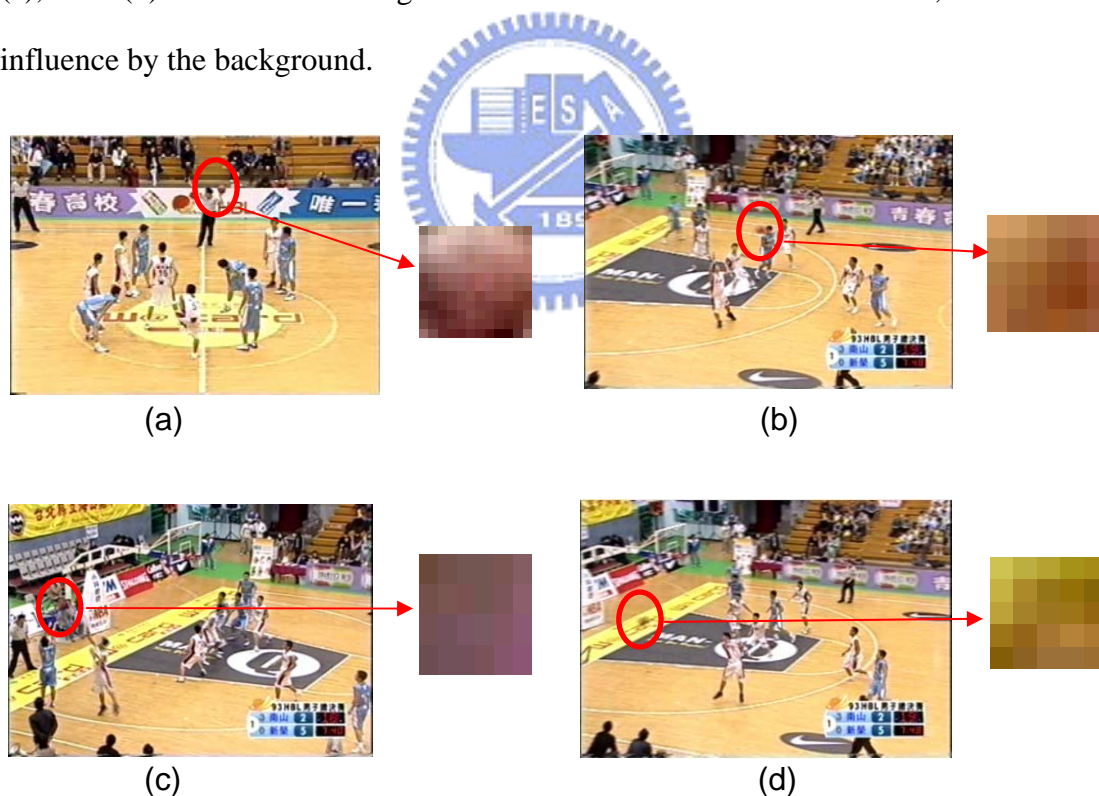


Fig. 3-11 Observation of the color of basketball.

Only using the values of R and G is not enough to find out correct ball candidates because of complex background and noise. Background subtraction is also

used to select the correct ball candidates. Each possible ball block is compared to the corresponding position in the previous frame. Since the basketball is moving in high speed, the ball blocks must have large luminance difference between the two frames. As shown in **Fig. 3-12**, (a) is a source image containing a moving ball, and (b) shows the pixels having large luminance difference between (a) and its previous frame. If the luminance difference is large enough, the pixel is dotted as white; otherwise, the pixel is dotted as black. The red circles indicate the ball positions. Most of the possible ball blocks that are not the ball will be filtered by background subtraction.



(a) Source image.

(b) Frame difference.

Fig. 3-12 Background subtraction of the image.

The region with largest number of connected ball blocks is found after applying a region generation algorithm [31]. The minimum bounding rectangle (MBR) around the region is defined for two purposes: 1) Filter out noise having the same color feature such as the audience. 2) Obtain the center of the ball region.

Many noisy regions rather than the ball region might be detected. Therefore, the area and aspect ratio of the minimum bounding rectangle (MBK) are used as characteristics to identify the possible ball region. Moreover, we define the ball center

coordinate($centerX, centerY$) = $(\frac{1}{n} \sum_{i=1}^n Px_i, \frac{1}{n} \sum_{i=1}^n Py_i)$, where n is the total number of pixel in the minimum bounding rectangle and (Px_i, Py_i) is the coordinate of pixel i .

Fig. 3-13 shows the result of ball candidate search after color and shape filtering. (a) is the case without camera motion and (b) is the case with camera motion. When the camera is fixed, there are fewer ball candidates. However, when there is camera motion, there will be too many ball candidates in a frame. To reduce the number of ball candidates, we perform the Ball-Candidate-Reduction step.

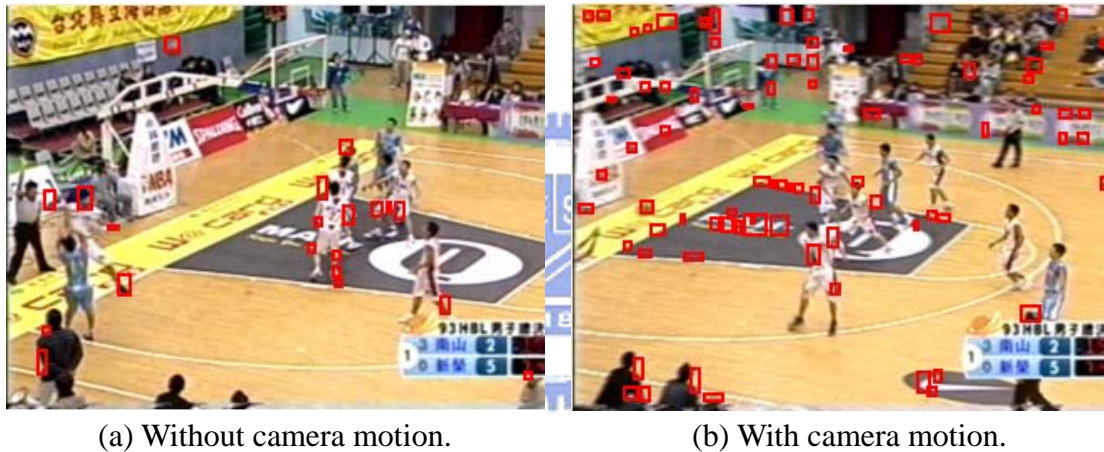
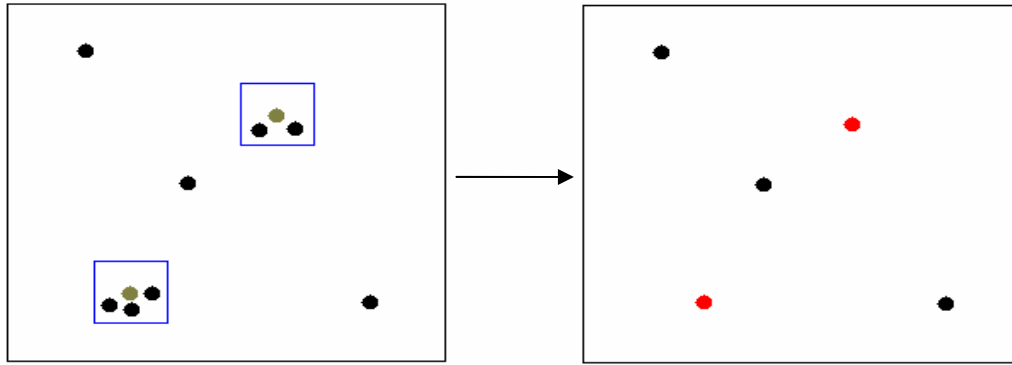


Fig. 3-13 Result of ball candidate search after color and shape filtering.

Ball-Candidate-Reduction is implemented by examining each ball candidate to see whether the search range around it has any other candidate. Take the average coordinate of all candidates in the search range as the new candidate position. Thus we can delete many noisy candidates. As shown in **Fig. 3-14**, (a) represents ball candidates before reduction, and (b) depicts ball candidates after reduction. **Fig. 3-15** is the result of applying Ball-Candidate-Reduction step to the real image, where (a) shows the candidate positions before reduction with blue circles and (b) displays the new ball candidates after reduction with red circles.



(a) Before Reduction of ball candidates.

(b) After Reduction of ball candidates.

Fig. 3-14 Ball Candidate Reduction.



(a) Before Reduction of ball candidates.

(b) After Reduction of ball candidates.

Fig. 3-15 Result of ball candidate reduction.

3.4 Ball Tracking

An array of basketball positions in different frames is found in the foregoing processing. However, there are still chances that some of the data are not the ball. To track the ball, route detection based on dynamic programming is used to find out the correct trajectory among such data.

Suppose there are two frames, frame i and frame $j (i < j)$. The 2D velocity of the ball can be calculated by

$$Velocity_{i \rightarrow j} = \frac{\sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2}}{T_{i \rightarrow j}}$$

where (X_i, Y_i) and (X_j, Y_j) are the positions of the ball candidates in frame i and j , and $T_{i \rightarrow j}$ is the time duration between frame i and j . For two near frames in a shot, the velocity of the ball will be in a certain range. The tracking conception is described in **Fig. 3-16**. The X and Y axes represent 2D coordinates of the ball candidates, and the horizontal axis shows frame number of the current candidate. Assume the candidates are nodes of **Fig. 3-16**. When the velocity of the ball calculated by candidates in frame i and j satisfy the velocity constraint, the nodes corresponding to these candidates will be connected by an edge.

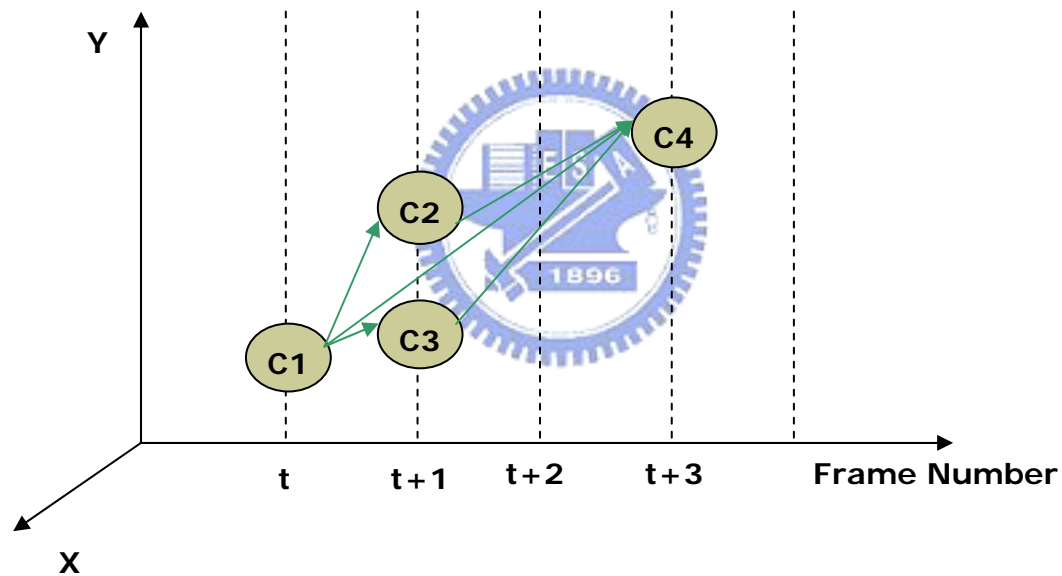


Fig. 3-16 Tracking process.

After connecting the candidates by edges, a complete route that represents the trajectory of the ball is searched. If a candidate is not connected within T_{frame} frames, we check its current connected route to see whether it is a ball trajectory by calculating the variance and distortion of all candidates in the route. Since the ball is usually passed or shot by players, its position will not stay in a small range and its route will be a parabola. We estimate the parabola and determine the distortion as the sum of

difference between the parabola and each candidate position. For each route, if the length is long enough, the variance is large, and the distortion is small, we determine it as a possible ball trajectory. As we can see in **Fig. 3-17**, the tracking process may result in several 2D trajectories. We will introduce a method to find out the real shooting ball trajectory in section 3.5.

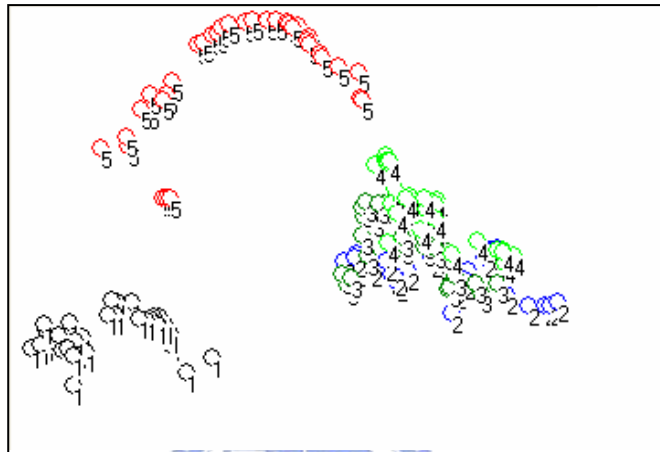


Fig. 3-17 Result of tracking ball.

More tracking algorithm are described in [29, 30]. Robust tracking requires multiple levels of representation. A robust, integrated system needs less specific models for tracking. Tracking players is much more difficult than tracking the ball since the number of people determines the complexity of tracking process and accuracy. Moreover, object occlusion is also a problem in tracking process.

3.5 Camera Calibration

For semantic analysis of sport videos, camera calibration parameters are required to convert the positions of a ball and players in the video frame to 3D space in the real-world coordinates or vice versa. **Fig. 3-18** shows the correspondence between a 2D court image and a 3D court model.

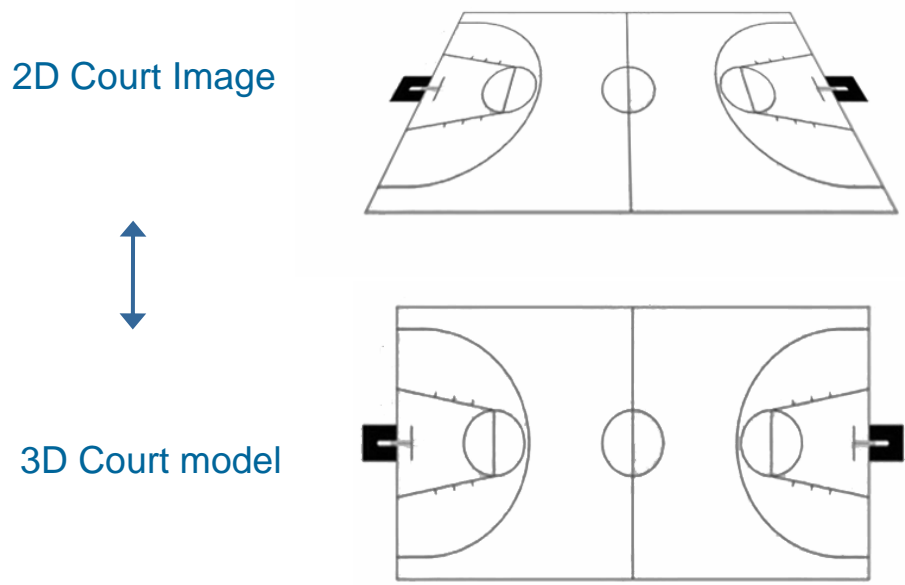


Fig. 3-18 Correspondence between 2D court image and 3D court model.

As mentioned in section 2.4.1:

$$\begin{bmatrix}
 x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u'_1 x & -u'_1 y & -u'_1 z \\
 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v'_1 x & -v'_1 y & -v'_1 z \\
 x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u'_2 x & -u'_2 y & -u'_2 z \\
 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v'_2 x & -v'_2 y & -v'_2 z \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 \cdot & & & & & & & & & & \\
 x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & -u'_N x & -u'_N y & -u'_N z \\
 0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & -v'_N x & -v'_N y & -v'_N z
 \end{bmatrix}_{2N \times 11}
 \underbrace{\begin{bmatrix}
 c_{11} \\
 c_{12} \\
 c_{13} \\
 c_{14} \\
 c_{21} \\
 c_{22} \\
 c_{23} \\
 c_{24} \\
 c_{31} \\
 c_{32} \\
 c_{33}
 \end{bmatrix}}_{11 \times 1}
 =
 \begin{bmatrix}
 u'_1 \\
 v'_1 \\
 u'_2 \\
 v'_2 \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 u'_N \\
 v'_N
 \end{bmatrix}_{2N \times 1}$$

We can assume $c_{34}=1$

To calculate the eleven camera parameters(c_{ij}), we need at least six non co-plane points whose 2D and 3D coordinates are both known. In court sport like basketball, the marker lines on the court and the backboard boundary can be used to determine the calibration parameters since both the color and length of the marker lines and

backboard boundary are determined by the official rules. **Fig. 3-19** shows the line correspondences between image and basketball court model. If we can find white lines in the image, the crossing or boundary points of lines can be used to calculate the transformation between the image and the real court. After that, the positions of the ball and players on the court can be estimated by detecting the center point and footing points respectively.

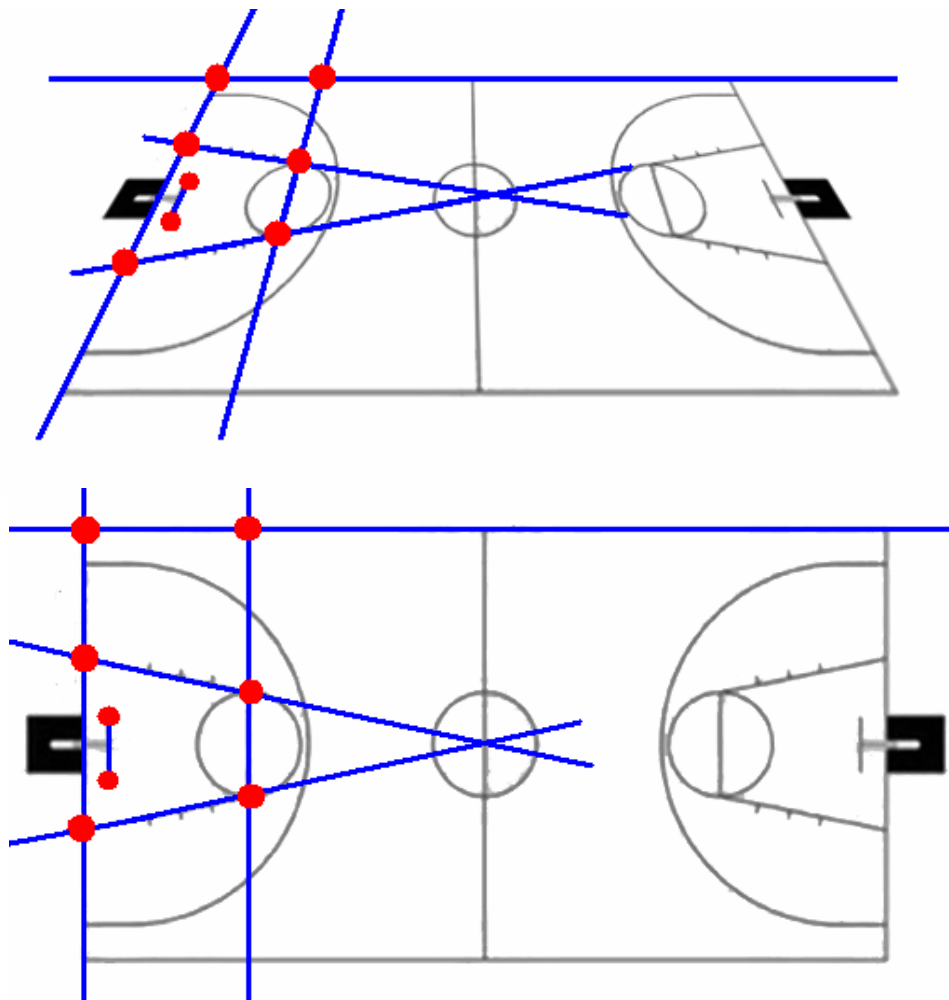


Fig. 3-19 Line correspondences between image and basketball court model.

Fig. 3-20 is the flow chart of camera calibration. For each frame, image pixels are classified as court line and backboard boundary pixels by some color and local texture constraints. Hough Transform and Court Model Fitting are applied at first

frame to extract line candidates and initialize the court and backboard location. In subsequent frames, we make a fast local search for the new camera parameters with the previous approximate court and backboard locations rather than performing Hough Transform and Court Model Fitting again. We will explain each step in following sections.

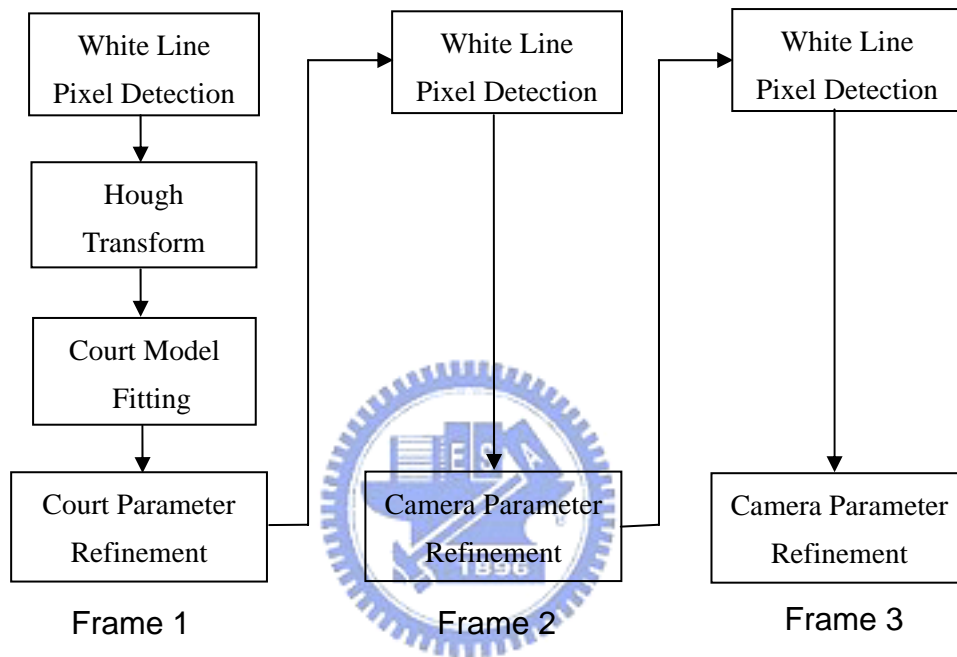


Fig. 3-20 The flow chart of camera calibration.

3.4.1 White Pixel Detection

The color of court lines is white by the official rule. However, there are other white objects in an image such as advertisement logos, part of the stadium, the spectators or the players dressed in white clothes. These not correctly detected white pixels result in too many line candidates after using the subsequent Hough line-detection method, and make the fitting of the court model time consuming and unreliable. We use additional criteria to constrain the set of court line pixels. As illustrated in **Fig. 3-21**, assume that the court line has a width= τ pixels and the candidate pixel is drawn as gray. O and X represent pixels that are T_d pixels

away from the current pixel in the vertical and horizontal directions respectively. We check if the brightness of O or X is darker than the candidate pixel.

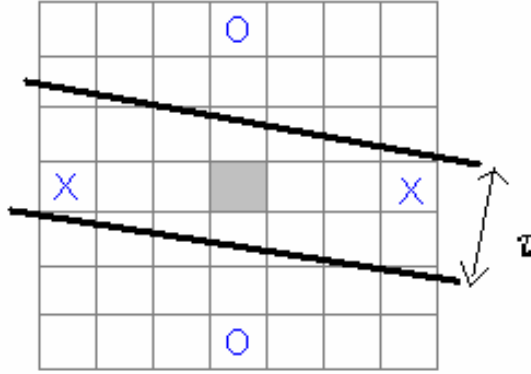


Fig. 3-21 Part of the image containing a white line pixel.

We identify a pixel as a court line pixel or not according to Eq.(10):

$$LinePixel(x, y) = \begin{cases} 1 & Y(x, y) \geq T_a \wedge Y(x, y) - Y(x - \tau, y) > T_b \wedge Y(x, y) - Y(x + \tau, y) > T_b \\ 1 & Y(x, y) \geq T_a \wedge Y(x, y) - Y(x, y - \tau) > T_b \wedge Y(x, y) - Y(x, y + \tau) > T_b \\ 0 & \text{else} \end{cases} \quad (10)$$

where $Y(x, y)$ is the luminance value in YCbCr space. **Fig. 3-22** is an example after applying Eq.(10) to detect possible white line pixels.(a) is the original image, (b) shows detected white line pixels by red points, and (c) extracts white line pixels by black points.

Since pixels in finely textured areas of small white letters in logos, white areas in the stadium, or spectators wearing white clothes will still pass the above white line test, the result will contain many noise pixels. Therefore, we exclude those white pixels that are in textured regions to prevent too much false detection in the line-extraction step.

Textured regions are recognized by observing the two eigenvalues of the structure matrix S , computed over a small window of size $2b + 1$ around each

candidate pixel (p_x, p_y) . The structure matrix is defined in [21]:

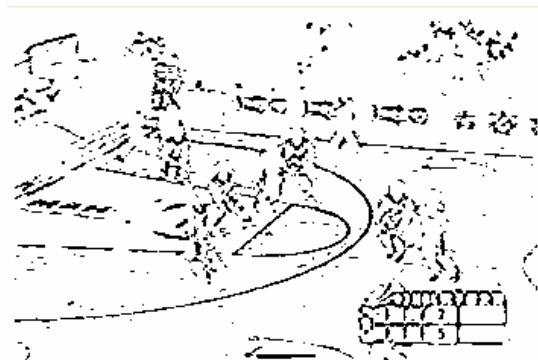
$$S = \sum_{x=p_x-b}^{p_x+b} \sum_{y=p_y-b}^{p_y+b} \nabla Y(x, y) \cdot (\nabla Y(x, y))^T$$

If both eigenvalues of matrix S , called λ_1 and λ_2 ($\lambda_1 \geq \lambda_2$) are large, it indicates a two-dimensional texture area. If one eigenvalue is large and the other is small, image gradients are oriented along a common axis. On the straight court lines, the latter case will be applied to define an additional rule which retains white pixels if $\lambda_1 \geq c \cdot \lambda_2$.



(a) The original image.

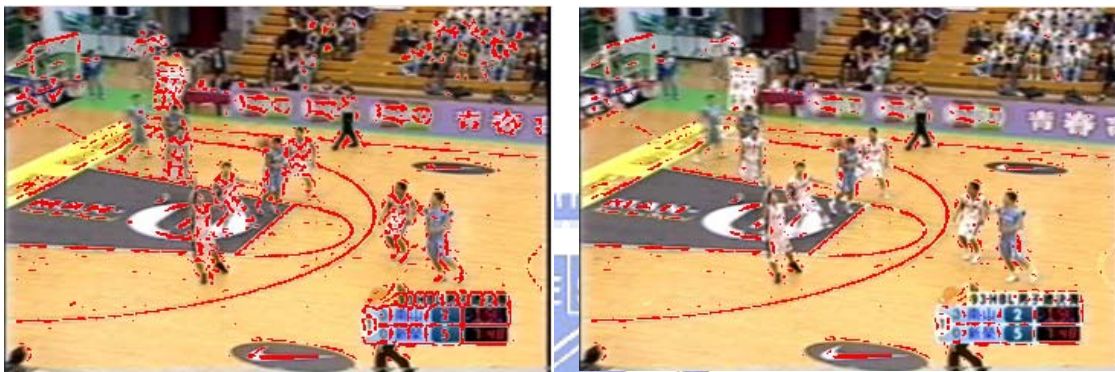
(b) White line pixels shown by red points.



(c) Extracted white line pixels.

Fig. 3-22 White line pixel detection.

Results of the proposed structure constraint can be seen in **Fig. 3-23**. (a) shows the white line pixels without line-structure constraint by red points. (b) shows the white line pixels with line-structure constraint by red points. (c) extracts white line pixels without line-structure constraint by black points. (d) extracts white line pixels with line-structure constraint by black points. We can observe that many noise pixels in the area of small white letters in logos, white areas in the stadium, or spectators wearing white clothes are removed after applying line-structure constraint.



(a) White line pixels without line-structure constraint shown by red points. (b) White line pixels with line-structure constraint shown by red points.



(c) Extracted white line pixels without line-structure constraint. (d) Extracted white line pixels with line-structure constraint.

Fig. 3-23 Applying line-structure constraint.

3.4.2 Court Line and Backboard Line Candidates Detection

After obtaining the white pixels, the system has to identify the court lines and the top boundary of the backboard. A standard Hough transform on the set of the previously detected white pixels is used to detect these line candidates. As depicted in **Fig. 3-24**, the parameter space used to represent the lines is (θ, d) , where θ is the angle between the line normal and the horizontal axis, and d is the distance of the line to the origin. We construct an accumulator matrix for all (θ, d) and sample the accumulator matrix at a resolution of one degree for θ and one pixel for d . As **Fig. 3-25** shows, since a line in (x, y) space corresponds to a point in (θ, d) space, line candidates are determined by extracting the local maxima in the accumulator array.

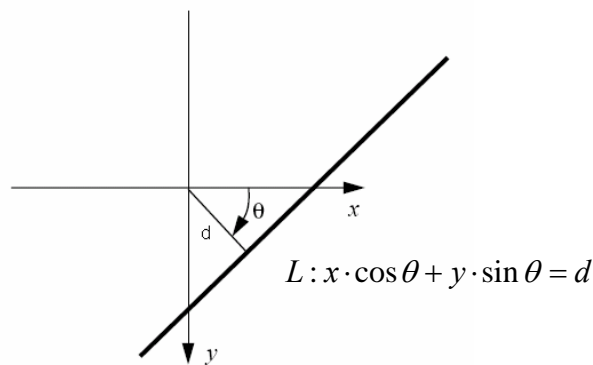


Fig. 3-24 Hough transform for straight lines.

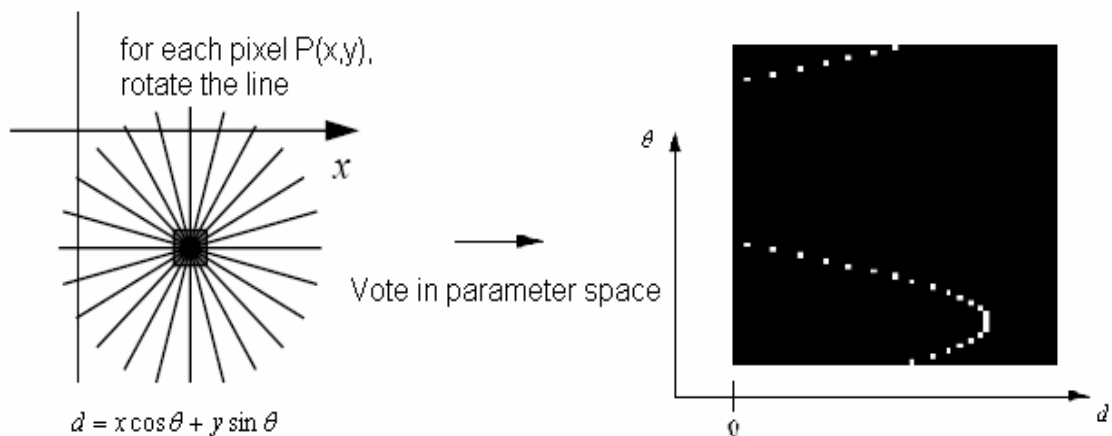


Fig. 3-25 Line detection by Hough Transform.

The Hough transform has the disadvantage that thick lines in the input image usually result in a bundle of detected lines, which all lie close together. Another disadvantage of the Hough transform is that the accuracy of the determined line parameters is depending on the resolution of the accumulator matrix. This problem cannot be easily reduced by increasing resolution of the accumulator matrix, since this also causes that the inexact parameter samples for an input line spread over a larger area in the accumulator matrix. Solve both of the above-mentioned problems by introducing a further step after the Hough transform to improve the accuracy of the detected line parameters by computing the best fit line to the input data. Furthermore, lines whose parameters are nearly equal are considered being duplicates and one of them is removed.

With all line candidates, we can obtain six intersections of the court lines as indicated in **Fig. 3-26**. However, we need two more points of the backboard to calculate the camera parameters.

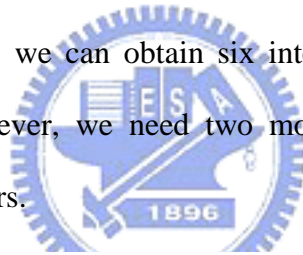


Fig. 3-26 Six intersections of the court line candidates.

As we can see in **Fig. 3-26**, the lighting condition and the material of the backboard usually make the white pixels only distinguishable on the top of the

backboard. If we can obtain the start point and end point of the backboard top-line, we can calculate the camera parameters. Unfortunately, the white top-line of the backboard is too short in comparison with the court lines, which results in the elimination from the line candidates during the Hough Transform step. To solve this problem, we use only the one fourth pixels in the top of the frame to detect the backboard line, and compute the line segment boundaries to know where the line starts and ends. The algorithm of line segment boundaries is described as follows.

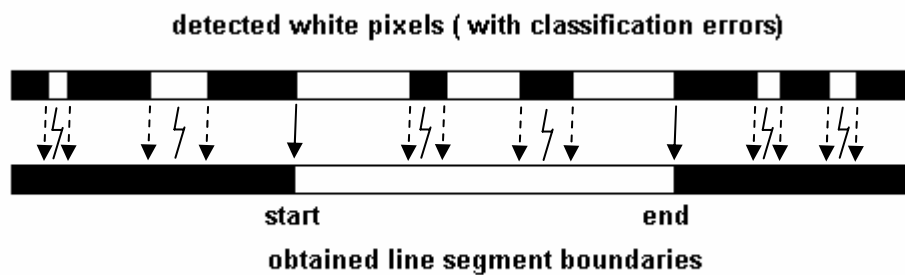


Fig. 3-27 Detection of line-segment boundaries.



Fig. 3-28 Boundaries of the backboard top-line.

Scanning along the detected line, a sequence of white (top-line) pixels and black (non top-line) pixels is obtained. Because of classification errors and occlusions, the data contain noisy data. In **Fig.3-27**, we assume that the line segment starts at position

start and ends at position **end**, and define the number of errors as the number of black pixels in the range **start** – **end** plus the number of white pixels outside the range **start** – **end** (ζ stands for errors). Using this error definition, we place the line segment boundaries such that the error is minimized. This optimization has a linear time complexity, and the result is shown in **Fig. 3-28**.

3.4.3 Model Fitting

With the intersections of court lines and the boundaries of backboard top-line found in the first frame, we can match the eight points to the court model and calculate the camera parameters as **Fig. 3-29** shows.

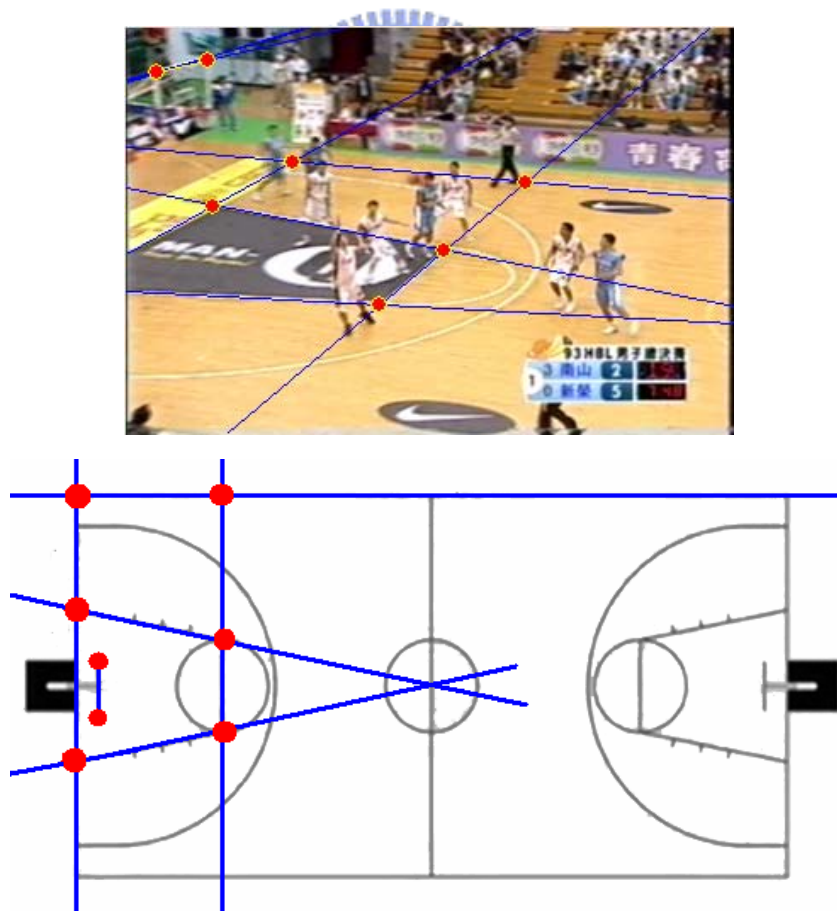


Fig. 3-29 Match the eight points to the court model and calculate the camera parameters.

3.4.4 Court Parameter Refinement

The previous calibration algorithm only need to be applied in the bootstrapping process when the first frame of a new shot is processed. For subsequent frames, we can assume that the acceleration of camera motion is small. This enables the prediction of the camera parameters for the next frame. Since the prediction provides a good first estimate of the camera parameters, a simplified version of the above algorithm can be applied.

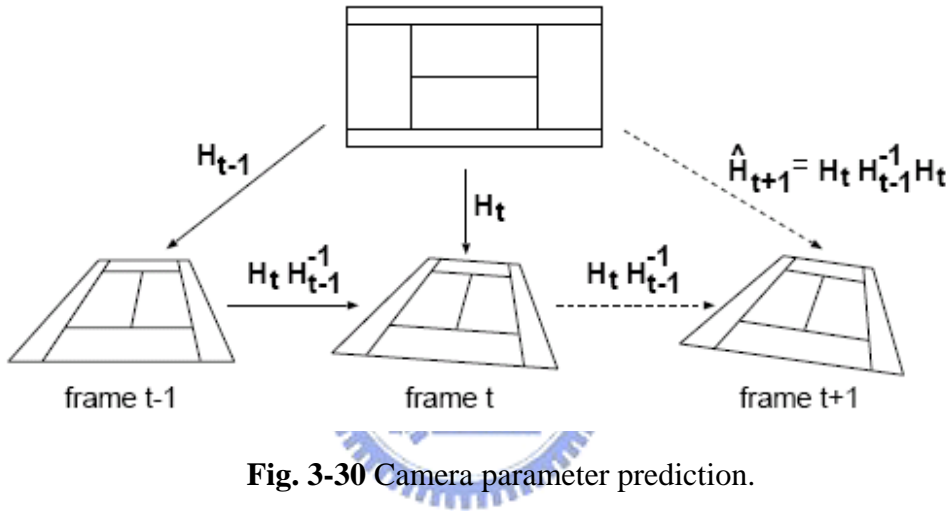


Fig. 3-30 Camera parameter prediction.

As **Fig. 3-30** shows, H_t is the camera parameters for frame t . If we know the camera parameters for frames t and $t - 1$, we can predict the camera parameters \hat{H}_{t+1} for $t + 1$ by $\hat{H}_{t+1} = H_t H_{t-1}^{-1} H_t$. The non-linear Levenberg-Marquardt minimization algorithm can be used to find the new camera parameters [21].

However, the court lines are too complex and varied when the basketball video has camera motion, which cause difficulty in camera parameter refinement. Since tracking the camera parameter is a bottleneck, we only analyze clips without camera motion to estimate the 3D trajectory. From 2D trajectories obtained in the ball tracking step, we can find a real shooting trajectory by examining whether it passes

through the backboard. As represented in **Fig. 3-31**, the four 2D image points of the backboard are marked as A, B, C and D, which can be derived from the 3D real world locations. If the parabola of the 2D trajectory passes through the minimum bounding rectangle of the backboard, it will be a possible shooting trajectory.

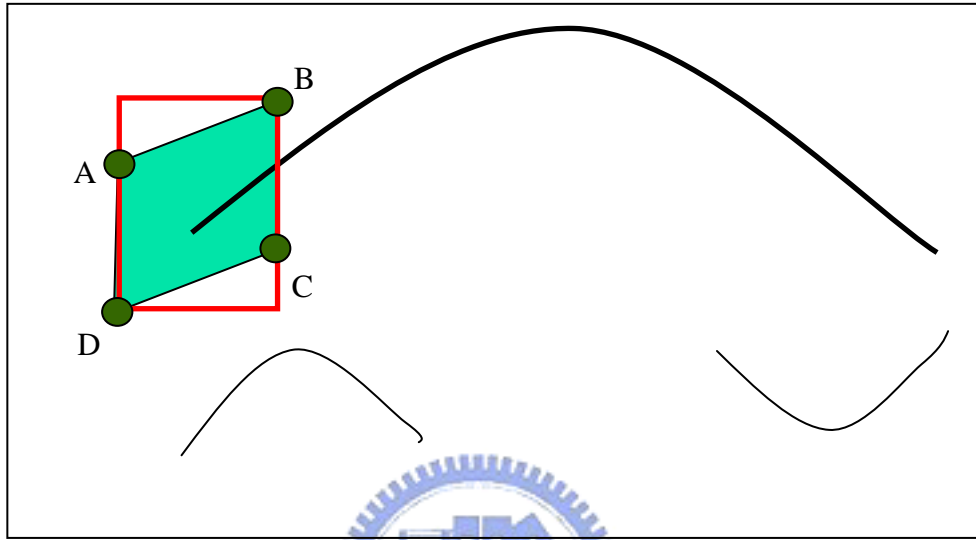


Fig. 3-31 Extract possible 2D shooting trajectory.

The relationship between each pair of corresponding points in the 2D and 3D space is:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \cdot \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (11)$$

where (u, v) is in the 2D image coordinates and (X_C, Y_C, Z_C) is in the 3D real world coordinates. Moreover, the 3D ball trajectory should fit the physical property:

$$\begin{aligned} X_C &= x_0 + V_x t \\ Y_C &= y_0 + V_y t \\ Z_C &= z_0 + V_z t + \frac{1}{2} g t^2 \end{aligned} \quad (12)$$

where (x_0, y_0, z_0) is the initial position of the ball in 3D coordinate, (V_x, V_y, V_z) is the velocity of the ball in 3D coordinate, g is acceleration of gravity, and t is the current time.

Use Eq.(12) to substitute for (X_C, Y_C, Z_C) in Eq.(11):

$$\rightarrow \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 + V_x t \\ y_0 + V_y t \\ z_0 + V_z t + \frac{1}{2} g t^2 \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Since the eleven camera calibration parameters and the time of each point on the trajectory are known, we can calculate the six unknowns $(x_0, y_0, z_0, V_x, V_y, V_z)$ of the parabola with three or more arbitrary points on the 2D trajectory. **Fig. 3-32** indicates the three points that we choose to calculate $(x_0, y_0, z_0, V_x, V_y, V_z)$. With camera parameters matrix C and six physical parameters $(x_0, y_0, z_0, V_x, V_y, V_z)$, we can extract the 3D trajectory and take the starting point of the 3D trajectory as the shot position.

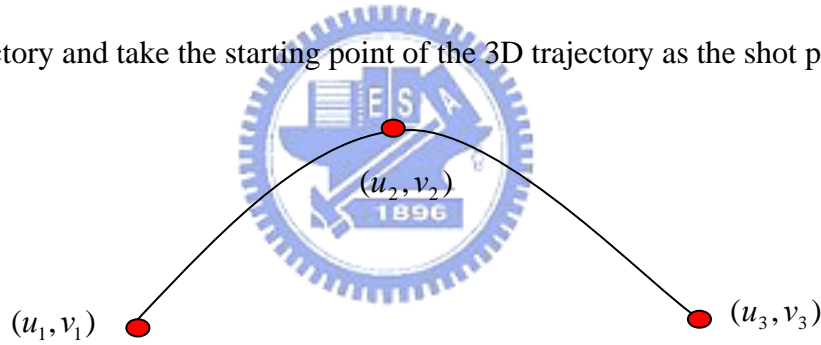
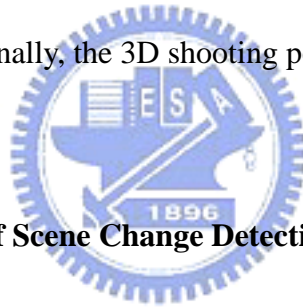


Fig. 3-32 Choose three points on the 2D trajectory.

Chapter 4

Experiment

In this chapter, we present the experimental results of the proposed system. We detect scene changes of MPEG testing sequences in compressed domain. For shot classification and tactic analysis steps, we use AVI sequences and implement the analysis process in pixel domain. The resolution of all sequences is 360×240 . Section 4.1 shows the result of scene change detection and shot classification. In section 4.2 and section 4.3, the outcomes of 2D ball trajectory extraction and camera calibration are illustrated, respectively. Finally, the 3D shooting position is indicated.



4.1 Experimental Result of Scene Change Detection and Shot Classification

We use two basketball videos of HBL (High-school Basketball League) to test the scene change detection and shot classification algorithm. The first video is a 15 minutes long basketball video which contains 96 shots (37 Close-up view shots, 27 Medium view shots, and 32 Full-court view shots), and the other is 10 minutes long and contains 71 shots (26 Close-up view shots, 24 Medium view shots, and 21 Full-court view shots). **Table. 1** shows the classification results.

From **Table. 1**, the accuracy of our shot classification algorithm is about 95.2% (the number of correctly classified shots divided by the number of total shots). The miss and false situation may be caused by the angle of view. For instance, if a real full court view shot contains large portion of spectators, the ratio of the court dominant color will be lower, which results in wrong classification.

	Close up		Medium		Full court	
	Sequence 1	Sequence 2	Sequence 1	Sequence 2	Sequence 1	Sequence 2
Ground Truth	37	26	27	24	32	21
No. of Miss	1	2	2	2	0	1
No. of False	0	1	1	3	2	1

Table. 1 Shot classification results of two testing sequences. Sequence 1 is a 15 minutes basketball video containing 96 shots, and sequence 2 is a 10 minutes basketball video containing 71 shots.

4.2 Experimental Result of Tracking the Ball

Using the proposed ball candidate search and tracking methods, we can obtain the 2D trajectories from the full court view shots. **Fig. 4-1** is the tracking result of a shot without camera motion, and **Fig. 4-2** is the tracking result of a shot with camera motion. No matter the sport video is shot by stationary camera or not, we can obtain its possible 2D trajectories.

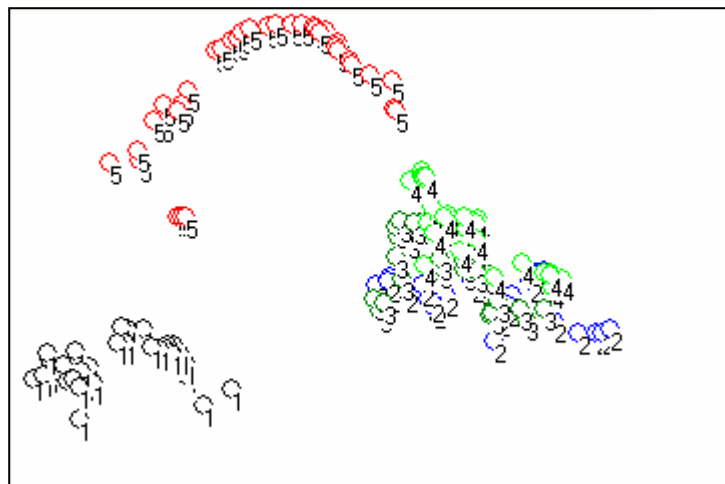


Fig. 4-1 The tracking result of a shot without camera motion.

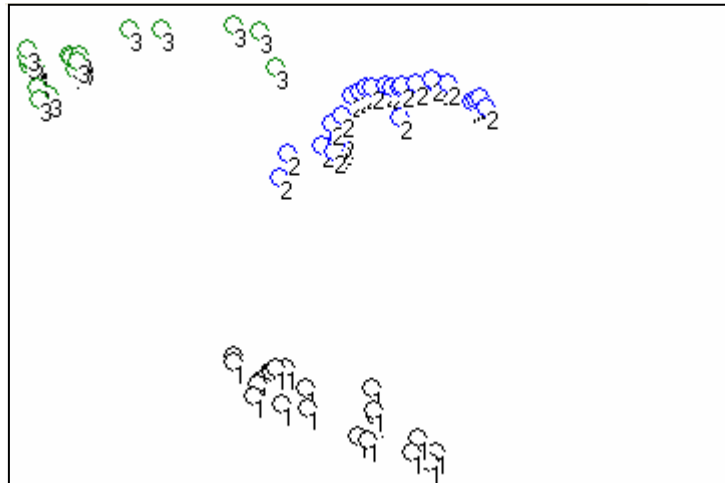


Fig. 4-2 The tracking result of a shot with camera motion.

4.3 Experimental Result of Camera Calibration and Shooting Position

In this section, we only use the clips without camera motion to test the camera calibration algorithm. As **Fig. 4-3** shows, the location of the points for camera calibration and the backboard position can be derived from the image. Therefore, the real shooting trajectory presented by solid circles can be identified as shown in **Fig. 4-4**. Use the transformation relationship from 2D coordinate to 3D coordinate, we can obtain the shot position. **Fig. 4-5** indicates the 3D shooting position by a red point.



Fig. 4-3 The 2D location of the points for camera calibration and the backboard position.



Fig. 4-4 The real 2D ball trajectory.

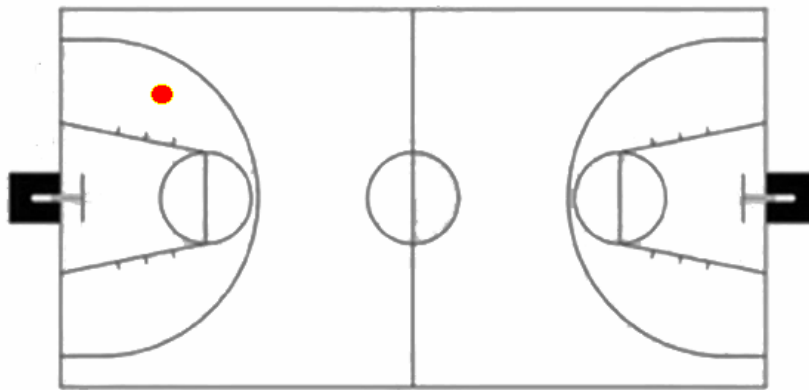


Fig. 4-5 The obtained shooting position in 3D court model.

Chapter 5

Conclusion and Future Work

Sport event detection has been proposed in previous research. However, these events only provide the audience a more efficient way to browse through sport videos. We propose a system that can automatically detect the scene change of the basketball video and classify clips into three kinds of shots. With the full-court-view shots, we can track the ball in the videos, detect the court-line and the backboard positions, and define the transformation relationship from 2D image to 3D real-world court model. After mapping the position of the ball from images to court model, the system concludes the possible shooting positions.

Analyzing tactics in basketball video is difficult due to the variation of view angle, the complexity of background and the intricacy of court lines. Our ball tracking method can be used for any full court view shot no matter whether there is camera motion or not. However, the camera calibration algorithm can only be applied for clips without camera motion.

Since the camera is not fixed, the result of shooting positions might not be accurate enough. The future work can be concentrated on videos shot by stationary camera so that the system will be more reliable. Tracking players in the video is difficult because occlusion occurs when players get close. If we can propose a more effective and efficient tracking algorithm, we could gather more statistics to analyze the behavior of the players in the games. Furthermore, we can conclude useful knowledge such as the defense rank and the offense tactics for professional basketball players and coaches who need more detailed information of the game.

Bibliography

- [1] G. Lu, "Communication and Computing for Distributed Multimedia Systems," *Artech House: Norwood, MA*, 1996.
- [2] A. Puri, R. L. Schmidt, and B. G. Haskell, "Overview of the MPEG Standards," *edit by Atul Puri, and Tsuhan Chen, Maecel Dekker Inc, New York/Basel*, 2000.
- [3] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, and M. Sakauchi, "Automatic Parsing of TV Soccer Programs," *IEEE International Conference on Multimedia Computing and Systems*, pp. 167-174, 1995.
- [4] Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, Issue 1, pp. 133-146, 2000.
- [5] D. Zhong and S. F. Chang, "Structure Analysis of Sports Video Using Domain Models," *IEEE International Conference on Multimedia and Expo*, pp.713-716, 2001.
- [6] L. Xie, S. F. Chang, A. Divakaran, and H. Sun, "Structure Analysis of Soccer Video with Hidden Markov Models," *International Conference on Acoustic, Speech, and Signal Processing*, Vol. 4, pp. 4096-4099, 2002.
- [7] G. Sudhir, J. C. M. Lee, and A.K.Jain, "Automatic Classification of Tennis Video for High-Level Content-Based Retrieval," *IEEE International Workshop on Content-Based Access of Image and Video Databases*, pp. 81-90, 1998.
- [8] W. Hua, M. Han, and Y. Gong, "Baseball Scene Classification Using Multimedia Features," *IEEE International Conference on Multimedia and Expo*, Vol. 1, pp.821-824, 2002.
- [9] 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*, Vol. 1, pp.825-828, 2002.
- [10] C. W. Ngo, T. C. Pong, and H. J. Zhang, "On Clustering and Retrieval of Video Shots," *ACM Multimedia*, pp. 51-60, 2001.
- [11] J. Assfalg, M. Bertini, C. Colombo, and A. D. Bimbo, "Semantic Annotation of Sports Videos," *IEEE Multimedia*, Vol.9, Issue 2, pp. 52-60, 2002.
- [12] Y. Gong, C. Hock-Chuan, and L. T. Sin, "An Automatic Video Parser for TV Soccer Games," *The 2nd Asian Conference on Computer Vision*, Vol. 2, pp. 509-513, 1995.
- [13] D. Yow, B. L. Yeo, M. Yeung, and B. Liu, "Analysis and Presentation of Soccer Highlights from Digital Video," *The 2nd Asian Conference on*

- Computer Vision*, pp. 499-503, 1995.
- [14] T. Tab, J. Hasegawa, and T. Fukumura, "Development of Motion Analysis System for Quantitative Evaluation of Teamwork in Soccer Games," *International Conference on Image Processing*, Vol. 3, pp. 815-818, 1996.
- [15] Y. Seo, S. Choi, H. Kim, and K. S. Hong, "Where Are the Ball and Players? Soccer Game Analysis with Color-Based Tracking and Image Mosaick," *The 9th International Conference on Image Analysis and Processing*, Vol. 2, pp. 196-203, 1997.
- [16] Y. Ohno, J. Miura, and Y. Shirai, "Tracking Players and a Ball in Soccer Games," *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 147-152, 1999.
- [17] Y. Ohno, J. Miura, and Y. Shirai, "Tracking Players and Estimation of the 3D Position of a Ball in Soccer Games," *IEEE International Conference on Pattern Recognition*, Vol. 1, pp. 145-148, 2000.
- [18] H. Kim and K. Hong, "Robust Image Mosaicing of Soccer Videos Using Self-calibration and Line Tracking," *Pattern Analysis & Applications*, Vol. 4, pp. 9-19, 2001.
- [19] T. Watanabe, M. Haseyama, and H. Kitajima, "A Soccer Field Tracking Method With Wire Frame Model From TV Images," *IEEE International Conference on Image Processing*, Vol. 3, pp. 1633-1636, 2004.
- [20] C. Calvo, A. Micarelli, and E. Sangineto, "Automatic Annotation of Tennis Video Sequences," *The 24th DAGM Symposium on Pattern Recognition*, Vol. 2449, pp. 540-547, Springer, 2002.
- [21] D. Farin, S. Keabbe, P. H. N. d. With, and W. Effelsberg, "Robust Camera Calibration for Sport Videos Using Court Models," *In SPIE Storage and Retrieval Methods and Applications for Multimedia*, Vol. 5307, pp. 80-91, 2004.
- [22] M. Xu, L. Y. Duan, C. Xu, M. Kankanhalli, and Q. Tian, "Event Detection in Basketball Video Using Multiple Modalities," *IEEE Joint Conference of the Fourth International Conference on Information, Communications, and Signal Processing*, Vol. 3, pp. 1526-1530, 2003.
- [23] A. Ekin and A. M. Tekalp, "Generic Play-break Event Detection for Summarization and Hierarchical Sports Video Analysis," *IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 169-172, 2003.
- [24] A. Ekin and A. M. Tekalp, "Shot Type Classification by Dominant Color for Sports Video Segmentation and Summarization," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 173-176, 2003.

- [25] G. Xu, Y. F. Ma, H. I. Zhang, and S. Yang, "A HMM Based Semantic Analysis Framework for Sports Game Event Detection," *IEEE International Conference on Image Processing*, Vol. 1, pp. 25-28, 2003.
- [26] C. Y. Wu, "Video Content Representation and Indexing Using Hierarchical Structure," *Master thesis, National Chiao Tung University, Dept. of CSIE*, 2000.
- [27] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compresses Video," *IEEE Transaction on Circuit and System for Video Technology*, Vol. 5, Issue 6, pp. 533-544, 1995.
- [28] A. Ekin and A. M. Tekalp, "Robust Dominant Color Region Detection and Color-based Applications for Sports Video," *IEEE International Conference on Image Processing*, Vol. 1, pp. 21-24, 2003.
- [29] C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8-15, 1998.
- [30] Y. J. Cham and J. M. Rehg, "A Multiple Hypothesis Approach to Figure Tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 239-245, 1999.
- [31] R. C. Gonzalez and R. E. Woods, "Digital Image Processing 2nd Edition," *Prentice Hall*, 2002.

