

# 國立交通大學

資訊科學與工程研究所

## 博士論文

運動影片內容分析、理解與註釋之研究

Sports Video Content Analysis, Understanding and Annotation



研究生：陳華總

指導教授：李素瑛 教授

中華民國九十八年八月

運動影片內容分析、理解與註釋之研究

Sports Video Content Analysis, Understanding and Annotation

研究生：陳華總

Student : Hua-Tsung Chen

指導教授：李素瑛

Advisor : Suh-Yin Lee

國立交通大學  
資訊科學與工程研究所  
博士論文



A Dissertation

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

Doctor of Philosophy

in

Computer Science

August 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年八月

# 運動影片內容分析、理解與註釋之研究

研究生：陳華總

指導教授：李素瑛 教授

國立交通大學資訊工程學系

## 摘要

隨著教育、娛樂、運動以及其它各式各樣多媒體應用的發展，數位化的影音多媒體數位內容與日劇增。因此，許多研究致力於多媒體內容的分析與理解並研發實用的系統，讓使用者可以快速地獲得所要的多媒體資料。運動影片是影音多媒體資料中相當重要的一環，有著相當可觀的商業利益、多樣的娛樂效果以及龐大的觀眾群，所以有越來越多的研究著眼於運動影片分析。目前大多數的運動影片分析以場景分類或精彩片段的擷取為主。然而，有越來越多的觀眾或球員希望能有多媒體系統的輔助來取得更豐富的運動資訊。甚至，裁判也要求利用電腦技術來輔助判決以提高公平性。本論文研究重點在於單一視角之視訊特徵的整合並設計相關演算法以達成運動影片內容理解、索引、註釋與擷取。

在運動影片中，重要的事件主要發生於球跟球員之間的互動。為了得知意義上與戰術上的相關內容，首先我們提出了一個有效且快速的方法來追蹤球路並計算球在各畫面中的位置。球路追蹤是一個相當艱難的問題。球在畫面中的體積小且不明顯，移動速度又快，想在單一畫面中辨別出哪一個物體是球，幾乎是不可能的。因此，我們利用球在畫面中移動的特性來辨識哪一段軌跡是球路，而不是從各畫面中去辨識哪一個物體是球。為了取得更豐富的比賽資訊並對比賽內容有更深刻的理解，我們提出一套創新的方法，能夠從單一視角之影片重建 3D 球路。此 3D 球路重建之演算法可用於籃球、排球、

網球之類擁有特定球場模型且多數場地特徵能被鏡頭所拍攝到之運動影片。在這類的運動影片中，利用攝影機所拍攝到的球場邊線與特徵物體，可以計算出三維空間位置與二維畫面座標之間的轉換關係。要從二維資訊去推論三維的資訊本來就是相當具有挑戰性的一個問題，因為在影像拍攝的過程中已經損失了空間中的深度資訊。在此，我們利用物理特性來設立球在三維空間的移動模型，再加上先前所求計算得之二維軌跡以及三維二維間的轉換關係，我們將可以估算出三維球路模型的參數，進而重建球在三維空間的運動軌跡。所取得之二維球路與重建之三維球路在運動影片中有著多樣化的重要應用，像是籃球的投籃出手點定位、排球事件偵測，以及棒球的投球球路分析。從三維球路所產生之三百六十度虛擬重播更可以讓觀眾隨己意變換不同視角來觀看球的動向。

在棒球比賽中，投球的進壘點（球經過打者時，與好球帶的相對位置）是影響球被打擊出去後移動方向的一個重要因素。好球帶是決定投球進壘點的一個參考指標，因此我們提出了一演算法分析打者姿勢與輪廓來設定好球帶，不論左打或右打姿勢都可適用。除了投打之間的對決外，球被打擊出去後的守備過程亦是吸引觀眾注目的焦點。經由辨識畫面中的特徵物體與線段，我們分類目前攝影機所拍攝的球場區域。因為攝影機所拍攝之區域即為事件發生之區域，所以我們可以利用影片中不同球場區域之轉換來推論球的移動路線與防守過程，並提供相似防守片段之比較，以分析守備策略。

我們以籃球、排球與棒球影片為測試資料，進行了多樣的實驗來評估所提出各種方法的效能。在我們的實驗中，其結果驗證所提各方法之可行性與優越性，並顯示從單一視角之運動影片即可取得相當多的比賽內容資訊供球員、教練做戰術分析與資料統計之用，並讓觀眾對比賽有更深入的了解。我們亦相信，本論文所提之運動資訊擷與影片內容理解諸多方法將可以應用於更多種類之運動影片。

# **Sports Video Content Analysis, Understanding and Annotation**

Student: Hua-Tsung Chen

Advisor: Prof. Suh-Yin Lee

Department of Computer Science,  
National Chiao Tung University

## **Abstract**

The explosive proliferation of multimedia data in education, entertainment, sport and various applications necessitates the development of multimedia application systems and tools. As important multimedia content, sports video has been attracting considerable research efforts due to the commercial benefits, entertainment functionalities and a large audience base. The majority of existing work on sports video analysis focuses on shot classification and highlight extraction. However, more keenly than ever, increasing sports fans and professionals desire computer-assisted sports information retrieval. Even more, the umpires demand assistance in judgment with computer technologies. In this thesis, we concentrate on the feature integration and semantic analysis for sports video content understanding, indexing, annotation and retrieval from single camera video.

In sports games, important events are mainly caused by the ball-player interaction and the ball trajectory contains significant information and semantics. To infer the semantic and tactical content, we first propose an efficient and effective scheme to track the ball and compute the ball positions over frames. Ball tracking is arduous task due to the fast speed and small size. It is almost impossible to distinguish the ball within a single frame. Hence, we utilize the ball motion characteristic over frames to identify the true ball trajectory, instead of recognizing which object is the ball in each frame. To retrieve more information about the

games and have a further insight, we design an innovative approach of 3D ball trajectory reconstruction in single camera video for court sports, where the court lines and feature objects captured in the frames can be used for camera calibration to compute the transformation between the 3D real world and the 2D frame. The problem of 2D-to-3D inference is intrinsically challenging due to the loss of the depth information in picture capturing. Incorporating the 3D-2D transformation and the physical characteristic of ball motion, we are able to approximate the depth information and accomplish the 2D-to-3D trajectory reconstruction. Manifold applications of sports video understanding and sports information retrieval can be achieved on the basis of the obtained 2D trajectory and the reconstructed 3D trajectory, such as shooting location estimation in basketball, event detection in volleyball, pitch analysis in baseball, etc. The 3D virtual replay generated from the 3D trajectory makes game watching a whole new experience that the audience are allowed to switch between different viewpoints for watching the ball motion.

In baseball, the pitch location (the relative location of the ball in/around the strike zone when the ball passes by the batter) is an important factor affecting the motion of the ball hit into the field. Strike zone provides the reference for determining the pitch location. Hence, we design a contour-based strike zone shaping and visualization method. No matter the batter is right- or left-handed, we are able to shape the strike zone adaptively to the batter's stance. Computer-assisted strike/ball judgment can also be achieved via the shaped strike zone. In addition to the pitcher/batter confrontation, the defense process after the ball is batted also attracts much attention. Therefore, we design algorithms to recognize spatial patterns in frames for classifying the active regions of event occurrence in the field. The ball routing patterns and defense process can be inferred from the transitions of the active regions captured in the video. Furthermore, the sequences with similar ball routing and defense patterns can be retrieved for defense strategy analysis.

Comprehensive experiments on basketball, volleyball and baseball videos have been

conducted to evaluate the performance of the proposed methods. The experimental results show that the proposed methods perform well in retrieving game information and even reconstructing 3D information from single camera video for different kinds of sports. It is our belief that the preliminary work in this thesis will lead to satisfactory solution for sports information retrieval, content understanding, tactics analysis and computer-assisted game study in more kinds of sports videos.



## 誌 謝

首先，最感謝的是指導教授李素瑛老師。六年的博士班期間，李老師在研究上的諄諄善誘與耐心教誨，才讓我得以成就此篇博士論文。李老師在生活態度、待人處世，以及各方面應對進退上給我的教導，都讓我終身受益無窮。此外，在我難過的時候，老師給了我關心；在我沮喪的時候，老師給了我鼓勵；在我茫然的時候，老師給了我指引。這一切都讓我銘感於心。

感謝蔡文錦老師在論文研究方面的指導。也感謝蔡老師在生活上對我的幫助與鼓勵，讓我知道人生還是可以有新的目標。感謝杭學鳴教授在計畫書口試、校內口試以及校外口試時提供寶貴的建議與鼓勵。感謝所有口試委員：吳家麟教授、廖弘源教授、范國清教授、張隆紋教授在口試過程中不吝提供多年的珍貴研究經驗，充實了本論文的深度與廣度，使本論文更趨完善。諸位口試委員都是我在學術研究上最佳學習典範。

感謝學長陳敦裕博士在研究上的熱心指導與心得分享，並在我遭遇困挫時給我鼓勵與幫助。資訊系統實驗室的學長與學弟妹都是我博士班研究生涯的好夥伴，感謝並祝福大家早日收穫豐富的研究成果。

感謝交大排球校隊、資工系排、應數女排的成員們，有了你們的陪伴，讓我的博士生涯中增添了不少歡笑與感動。感謝在我最低潮的時候陪我一起走過的朋友們，因為有你們的關心、鼓勵與陪伴，才讓我有繼續走下去的勇氣。

家人給予我的關懷與支持讓我在博士求學過程中無後顧之憂。一路走來，父母從不給我壓力，總是給予我溫暖的關懷。妹妹與弟弟的關心，以及對於父母的照顧，更是讓我能夠勇往直前的動力。有了他們的辛苦與支持，才有今日的我。感恩家人與其它親友對我的祝福與勉勵。

要感謝的人很多，在此向所有曾經幫助關心過我的人，致上最真切的謝意。

僅以此論文，獻給關心與幫助過我的大家。



# Table of Content

摘要 .....	i
Abstract .....	iii
誌謝 .....	vi
Table of Content .....	vii
List of Figures .....	x
List of Tables.....	xvi
Chapter 1. Introduction.....	1
Chapter 2. Physics-based Ball Tracking and 3D Trajectory Reconstruction with Applications to Shooting Location Estimation in Basketball Video .....	6
2.1 Introduction.....	6
2.2 Overview of the Proposed Physics-Based Ball Tracking and 3D Trajectory Reconstruction System in Basketball Video.....	9
2.3 Court Shot Retrieval.....	11
2.4 Camera Calibration.....	13
2.4.1 White line pixel detection.....	15
2.4.2 Line Extraction.....	17
2.4.3 Computation of camera calibration parameters.....	18
2.5 2D Shooting Trajectory Extraction.....	19
2.5.1 Ball candidate detection.....	19
2.5.2. Ball tracking.....	23
2.6 3D Trajectory Reconstruction and Shooting Location Estimation.....	27
2.7 Experimental Results and Discussions in Basketball Video.....	29
2.7.1 Parameter setting.....	29
2.7.2 Performance of basketball shot boundary detection and court shot retrieval..	30

2.7.3 Results of court line and backboard top-border detection.....	31
2.7.4 Performance of basketball tracking and shooting location estimation.....	32
2.7.5 Comparison and discussion.....	35
2.8 Summary.....	37
<b>Chapter 3. Ball Tracking and 3D Trajectory Approximation with Applications to Tactics</b>	
<b>Analysis in Volleyball Video.....</b>	<b>39</b>
3.1 Introduction.....	40
3.2 Related Work.....	41
3.2.1 Related work on camera calibration.....	41
3.2.2 Related work on ball/player tracking.....	42
3.3 Overview of the Proposed Ball Tracking and 3D Trajectory Approximation in Volleyball Video.....	44
3.4 Audio Event Detection.....	45
3.4.1 Whistle detection.....	46
3.4.2 Attack detection.....	46
3.5 Camera Calibration.....	47
3.6 2D Volleyball Trajectory Extraction.....	48
3.6.1 Ball candidate detection.....	48
3.6.2 Potential trajectory exploration.....	49
3.6.3. Trajectory identification and integration.....	51
3.7 3D Volleyball Trajectory Approximation.....	52
3.8 Trajectory-Based Applications in Volleyball Games.....	57
3.8.1 Action detection and set type recognition using 2D trajectory.....	57
3.8.2 3D virtual replays and serve placement estimation using 3D trajectory .....	58
3.9 Experimental Results and Discussion in Volleyball Video.....	59
3.9.1 Results of audio event detection.....	59

3.9.2 Results of 2D volleyball trajectory extraction.....	60
3.9.3 Simulation results of 3D volleyball trajectory approximation.....	63
3.9.4 Comparison with Kalman filter-based algorithm.....	67
3.10 Summary.....	68
<b>Chapter 4. Sports Information Retrieval in Baseball Video.....</b>	<b>69</b>
4.1 Introduction.....	70
4.2 Trajectory-Based Baseball Tracking Framework.....	72
4.2.1 Moving object segmentation.....	73
4.2.2 Ball candidate detection.....	74
4.2.3 Candidate distribution analysis and potential trajectory exploration.....	75
4.2.4 Trajectory identification.....	76
4.2.5 Baseball trajectory extraction.....	77
4.2.6 Trajectory-based pitching evaluation and visual enrichment.....	79
4.3 Automatic Strike Zone Determination.....	82
4.3.1 Overview of the proposed strike zone determination algorithm.....	84
4.3.2 Home plate detection.....	87
4.3.3 Batter region (BR) outlining.....	88
4.3.4 Batter contouring.....	88
4.3.5 Dominant point locating.....	90
4.4 Baseball Exploration using Spatial Pattern Recognition.....	93
4.4.1 Visual feature extraction.....	94
4.4.2 Spatial pattern recognition.....	96
4.4.3 Play region type classification.....	97
4.5 Experimental Results.....	100
4.5.1 Results of trajectory-based baseball tracking.....	101
4.5.2 Results of strike zone determination.....	106

4.5.3 Results of play region classification.....110

4.6 Summary.....111

**Chapter 5. Conclusions and Future Work.....113**

**Reference.....116**



## Lists of Figures

<b>Fig. 1-1.</b> Overview of the proposed framework.....	2
<b>Fig. 2-1.</b> Statistical graph of shooting locations: <b>O</b> (score) or <b>X</b> (miss).....	9
<b>Fig. 2-2.</b> Flowchart of the proposed system for ball tracking and 3D trajectory reconstruction in basketball video.....	11
<b>Fig. 2-3.</b> Examples of shot types in a basketball game: (a) Court shot; (b) Court shot; (c) Medium shot; (d) Medium shot; (e) Close-up shot; (f) Out-of-court shot.....	12
<b>Fig. 2-4.</b> Examples of Golden Section spatial composition: (a) Frame regions; (b) Court view; (c) Medium view.....	13
<b>Fig. 2-5.</b> Flowchart of camera calibration.....	15
<b>Fig. 2-6.</b> Point-correspondence between the 2D frame and the basketball court model: (a) Court image; (b) Court model.....	15
<b>Fig. 2-7.</b> Illustration of part of an image containing a white line.....	16
<b>Fig. 2-8.</b> Sample results of white line pixel detection: (a) Original frame; (b) Without line-structure constraint; (c) With line-structure constraint.....	16
<b>Fig. 2-9.</b> Detection of backboard top-border: (a) Detected court lines; (b) Computing vanishing point; (c) Searching backboard top-border.....	18
<b>Fig. 2-10.</b> Color histograms of 30 manually segmented basketball images.....	20
<b>Fig. 2-11.</b> Illustration of ball pixel detection: (a) Source frame; (b) Moving pixels; (c) Extracted ball pixels.....	21
<b>Fig. 2-12.</b> Left: detected ball candidates, marked as yellow circles. Right: motion history image to present the camera motion: (a) Fewer ball candidates produced if the camera motion is small; (b) More ball candidates would be produced if there is large camera motion.....	23
<b>Fig. 2-13.</b> Diagram of a long shoot.....	25
<b>Fig. 2-14.</b> Relation between $Vb$ and $H$ .....	25
<b>Fig. 2-15.</b> Illustration of ball tracking process( $X$ and $Y$ represent ball coordinates).....	26
<b>Fig. 2-16.</b> Illustration of the best-fitting function.....	27

<b>Fig. 2-17.</b> Statistical data of the shot lengths for different basketball shot classes.....	30
<b>Fig. 2-18.</b> Example results of detecting court lines and corresponding points (marked with yellow circles).....	32
<b>Fig. 2-19.</b> Example of a shooting trajectory being separated.....	33
<b>Fig. 2-20.</b> Demonstration of shooting location estimation. In each image, the blue circles are the ball positions over frames, the green circle represents the estimated shooting location and the red squares show the movement of corresponding points due to the camera motion.....	34
<b>Fig. 2-21.</b> Example of directional deviation in 3D trajectory reconstruction: (a) Shooting location estimation; (b) 3D virtual replay.....	35
<b>Fig. 2-22.</b> Error case of shooting location estimation caused by the misdetection of backboard top-border: (a) Corresponding points; (b) shooting location estimation.....	35
<b>Fig. 2-23.</b> Comparison of shooting location estimation with/without vertical (height) information: (a) Original shooting location in the frame; (b) Estimated shooting location with vertical information; (c) Estimated shooting location without vertical information.....	37
<b>Fig. 3-1.</b> Framework overview of the proposed VIA system.....	44
<b>Fig. 3-2.</b> Illustration of the non-coplanar feature points.....	47
<b>Fig. 3-3.</b> Illustration of the compactness filter: (a) Compactness degree $D_{compact}$ is defined as the ratio of the object size to the area of the bounding box; (b) Objects with low $D_{compact}$ would be removed while objects with high $D_{compact}$ would be retained.....	48
<b>Fig. 3-4.</b> Illustration of the Y- and X-distribution images for different process stages of a volleyball game: (a) Plotting the y-and x- coordinates of the ball candidates over time (indexed by the frame serial number $n$ ). Black dots represent isolated candidates and green crosses represent contacted candidates; (b) Potential trajectories: the sequences of the linked ball candidates in YDI and XDI; (c) Integrated trajectory.....	50
<b>Fig. 3-5.</b> Procedure of potential trajectory exploration.....	51
<b>Fig. 3-6.</b> Procedure of 3D trajectory approximation.....	55
<b>Fig. 3-7.</b> Illustration of set type diagram.....	58
<b>Fig. 3-8.</b> Illustration of volleyball detection and ball tracking: (a) Original frame; (b) Ball	

detection; (c) Ball tracking.....	61
<b>Fig. 3-9.</b> Demonstration of ball trajectory extraction and action detection in volleyball video: (a) Serve; (b) Reception; (c) Set and (d) Attack.....	63
<b>Fig. 3-10.</b> Demonstration of 3D trajectory approximation: (a)~(d) The enriched frames for serve, reception, set and attack, respectively; (e) Ball trajectory projected on the court model; (f) Serve placement estimation; (g)~(h) 3D virtual replays from different viewpoints.....	64
<b>Fig. 3-11.</b> Demonstration of 3D trajectory approximation: (a)~(d) The enriched frames for serve, reception, set and attack, respectively; (e) Ball trajectory projected on the court model; (f) Serve placement estimation; (g)~(h) 3D virtual replays from different viewpoints.....	64
<b>Fig. 3-12.</b> Error case in 3D trajectory approximation: (a) Detected ball candidates of serve; (b) The 3D approximated trajectory of serve; (c) Detect ball candidates in the video sequence; (d) The 3D trajectory approximated from the ball candidates in (c).....	66
<b>Fig. 3-13.</b> Error case in 3D trajectory approximation: (a) Ground truth 2D ball positions; (b) Ball candidates detected; (c) and (d) 3D virtual replays from different viewpoints .....	66
<b>Fig. 4-1.</b> Block diagram of the proposed baseball tracking framework.....	73
<b>Fig. 4-2.</b> Illustration of segmenting the moving objects where the ball is included: (a) Original frame; (b) FDI; (c) PFDI.....	74
<b>Fig. 4-3.</b> Illustration of the Y- and X-distribution images for different processing stages of trajectory extraction, where $n$ is the frame serial number, $y_c$ in YDI and $x_c$ in XDI are the y- and x-coordinates of each candidate in the original frame, respectively. (a) Ball candidate distribution analysis. Black dots represent isolated candidates and green crosses represent contacted ones. (b) Trajectory exploration. Potential trajectories are shown as the linking of ball candidates. (c) Trajectory identification. The ball trajectory identified is shown as the parabolic curve in YDI and the straight line in XDI.....	75
<b>Fig. 4-4.</b> Summarized demonstration of baseball trajectory extraction.....	78
<b>Fig. 4-5.</b> Demonstration of trajectory-based pitching evaluation and visual enrichment. Left: the superimposed ball trajectory and pitching evaluation. Right: the final ball location spotlighted with a crosshair. (a) Example of a MLB (Major League Baseball) pitch shot with a left-handed pitcher. (b) Example of a JPB (Japan	

Professional Baseball) pitch shot with a right-handed pitcher.....	81
<b>Fig. 4-6.</b> Illustration of strike zone definition.....	82
<b>Fig. 4-7.</b> Applications for strike zone determination: (a) Pitch location image; (b) Count of pitches in each region.....	84
<b>Fig. 4-8.</b> Sample results of combining ball trajectory extraction with strike zone.....	84
<b>Fig. 4-9.</b> Specifications of the baseball, home plate and batter's boxes.....	85
<b>Fig. 4-10.</b> Block diagram of the proposed strike zone determination system.....	85
<b>Fig. 4-11.</b> Layout of home plate, batter's boxes and batter regions.....	86
<b>Fig. 4-12.</b> Procedure of home plate detection: (a) Original frame of a pitch scene; (b) Pixels with high intensity around the frame center; (c) Detected home plate.....	87
<b>Fig. 4-13.</b> Example of moving edge extraction (within the OBR): (a) Gray level image $I_n$ ; (b) Edge map $E_n$ ; (c) Luminance difference image $ I_n - I_{n-1} $ ; (d) Difference edge map $DE_n$ ; (e) Moving edge map $ME_n$ .....	89
<b>Fig. 4-14.</b> Procedure of batter contouring.....	89
<b>Fig. 4-15.</b> Back and front contours for right- and left-handed batters.....	91
<b>Fig. 4-16.</b> Dominant point locating using the points of curvature extremes.....	91
<b>Fig. 4-17.</b> Sample results of dominant point locating.....	92
<b>Fig. 4-18.</b> Strike zone shaping and visualization: (a) A right-handed batter; (b) A left-handed batter.....	93
<b>Fig. 4-19.</b> Overview of the proposed baseball exploration framework.....	93
<b>Fig. 4-20.</b> Prototypical baseball field: (a) Full view of a real baseball field; (b) Illustration of field objects and lines.....	94
<b>Fig. 4-21.</b> Spatial distribution of dominant colors and white pixels: (a) First field frame; (b) Hue histogram; (c) Segmented regions; (d) Extracted white pixels.....	95
<b>Fig. 4-22.</b> Detection of field lines and field objects.....	97
<b>Fig. 4-23.</b> Twelve typical play region types.....	98
<b>Fig. 4-24.</b> Scheme of play region type classification within a field shot.....	100
<b>Fig. 4-25.</b> Illustration of ball detection and ball tracking in baseball video: (a) Ball detection.	



Two ball positions are missed when passing over the white uniform. (b) Ball tracking. Positions of missed balls can be recovered.....	103
<b>Fig. 4-26.</b> Examples of ball tracking and visual enrichment for various baseball clips: (a) MLB pitch shot; (b) JPB pitch shot; (c) CPBL pitch shot.....	103
<b>Fig. 4-27.</b> Error cases of home plate detection.....	107
<b>Fig. 4-28.</b> Illustration of $A_{sz}$ , $A_{gt}$ and $A_{ov}$ .....	107
<b>Fig. 4-29.</b> P-R distributions of MLB, JPB and CPBL sequences.....	108
<b>Fig. 4-30.</b> Example results of strike zone determination.....	109
<b>Fig. 4-31.</b> Examples of the improper strike zone determination: (a) Improper batter contouring due to the dynamic advertising board. (b) Improper batter contouring due to the movement of the plate umpire.....	110



## Lists of Tables

<b>Table 2-1.</b> Performance of basketball shot boundary detection.....	31
<b>Table 2-2.</b> Performance of basketball court shot retrieval.....	31
<b>Table 2-3.</b> Performance of basketball tracking.....	33
<b>Table 2-4.</b> Performance of basketball shooting location estimation.....	33
<b>Table 2-5.</b> Comparison between the proposed physics-based method and the Kalman filter-based method in basketball video.....	37
<b>Table 3-1.</b> Discriminants of ten common set types.....	58
<b>Table 3-2.</b> Performance of whistle detection.....	60
<b>Table 3-3.</b> Performance of attack detection.....	60
<b>Table 3-4.</b> Performance of volleyball detection and tracking.....	61
<b>Table 3-5.</b> Performance of action detection.....	62
<b>Table 3-6.</b> Comparison between the proposed physics-based method and the Kalman filter-based method in volleyball video.....	67
<b>Table 4-1.</b> Ball speed estimation with five-star evaluation using the ball trajectory.....	80
<b>Table 4-2.</b> Trajectory curvature measurement with five-star evaluation.....	80
<b>Table 4-3.</b> Rules of play region type classification.....	99
<b>Table 4-4.</b> Testing data used in the experiments.....	100
<b>Table 4-5.</b> Performance of baseball detection and tracking.....	102
<b>Table 4-6.</b> Comparison between the Kalman filter-based algorithm and the proposed physics-based algorithm in baseball video.....	105
<b>Table 4-7.</b> Performance of home plate detection.....	106
<b>Table 4-8.</b> Performance of strike zone determination.....	108
<b>Table 4-9.</b> Performance of play region classification.....	111

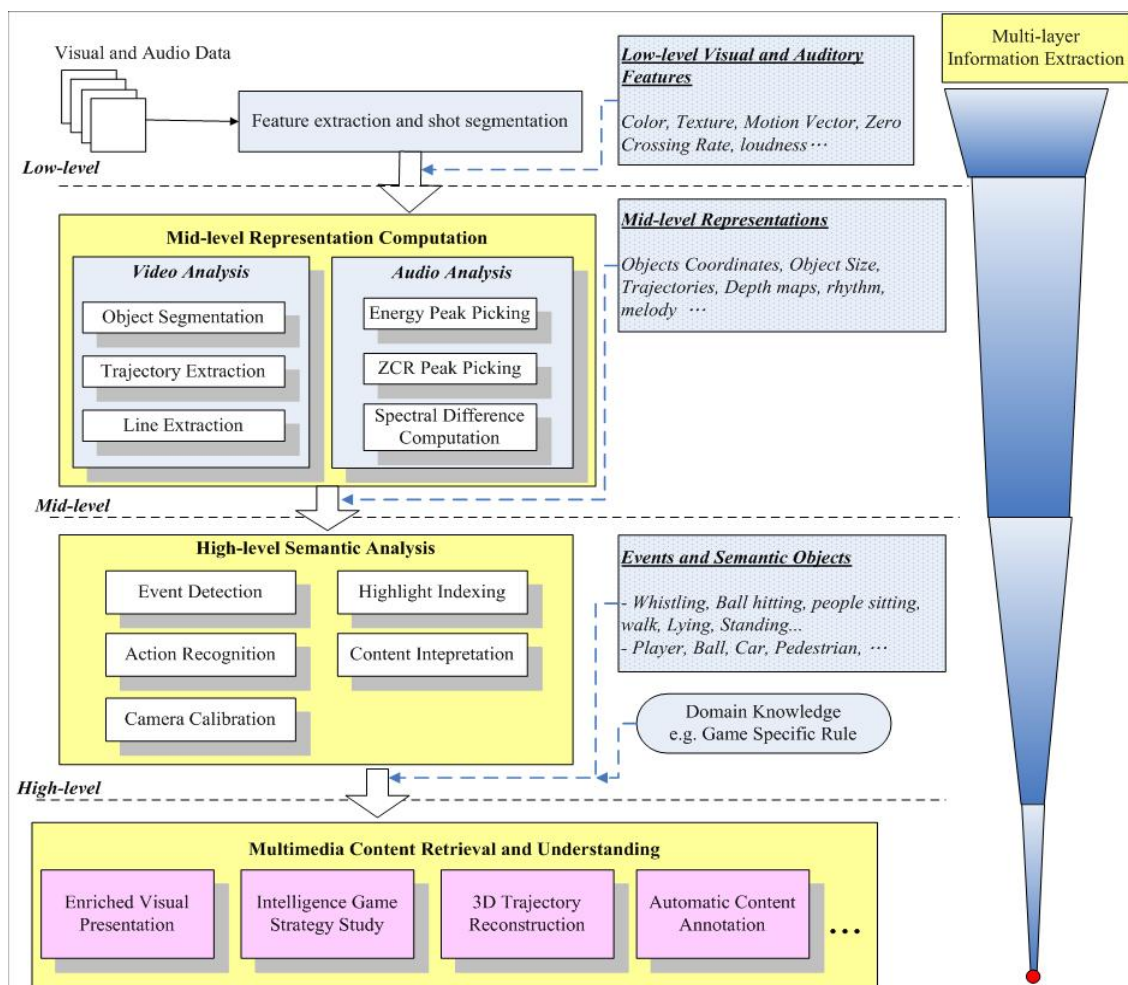
## Chapter 1. Introduction

The advances in video production technology and the consumer demand have led to the ever-increasing volume of multimedia data. The rapid evolution of digital equipments allows general users to archive multimedia data much easily. The explosive proliferation of multimedia data in education, entertainment, sport and various applications makes manual indexing and annotation no more practical. The development of practical systems and tools for multimedia content analysis, understanding, indexing and retrieval is undoubtedly compelling [1, 4, 6, 72, 73, 74].

A large number of content retrieval approaches have been proposed on the basis of low-level features. However, human interpret video in terms of semantics rather than low-level features. The demand for automatic video understanding and interpretation requires the mid-level representations mapping from low-level features to high-level semantics, such as shot class, camera motion pattern, color layout, object shape and object trajectory. Especially, object trajectory is one of the most informative representations which human use to analyze events frequently. Hence, the trajectory-based approaches have been gaining popularity [8, 15, 34, 35].

As important multimedia content, sports video has been attracting considerable research efforts due to the commercial benefits, entertainment functionalities and a large audience base [1, 2, 22, 25, 30, 35, 74]. In this thesis, we take sports video as source material for research. Techniques of event detection, content understanding and sports information retrieval are proposed for automatic annotation and enriched visual presentation. Most viewers prefer retrieving the designated events, scenes and payers to watching a whole game in a sequential way. Therefore, various algorithms have been developed for shot classification, highlight extraction and semantic annotation based on the fusion of audiovisual features and the game-specific rules.

In this thesis, we focus on feature integration and algorithm development for sports video content analysis and understanding. Sports information retrieval, tactics analysis, enriched visual presentation can provide the audience and professionals a further insight into the games. Fig. 1-1 depicts the overview of our research work. We first extract low-level features adaptive to different event detection so as to infer the high-level semantic information. Then, various mid-level representations are computed to bridge the gap between low-level features and semantic content meanings. Since significant events are mainly caused by the interaction of moving objects, object trajectories bring much semantic information contributive to content understanding. Thus, we design several trajectory-based algorithms for sports video content analysis and understanding.



**Fig. 1-1.** Overview of our research work.

For semantic and tactical content analysis in sports video, we first propose an effective and efficient ball tracking algorithm. Object tracking is usually the medium to convert the low-level features into high-level events in video processing. Object tracking has been an arduous problem despite the long research history. Ball tracking is even a more challenging task due to the fast speed and small size. It is almost impossible to distinguish the ball within a single frame, so information over successive frames, e.g. motion information, is required to facilitate the discrimination of the ball from other objects. In several kinds of ball games, the ball moves following the physical characteristic that the ball trajectory forms a (near-) parabolic curve. For example, the ball shot toward the basket in a basketball game, the ball passed between players in a volleyball game, the ball moving between players in a tennis game and the pitched ball in a baseball game. Utilizing the physical characteristics of ball motion, we present a physics-based ball tracking method to compute the 2D ball trajectory in different kinds of single camera sports videos.

To have a further insight into the games and retrieve more detailed sports information, we propose an innovative approach capable of reconstructing 3D ball trajectory from single camera video for court sports. The 2D-to-3D inference is intrinsically challenging due to the loss of 3D information in projection to 2D frames. For court sports, the court lines and feature objects are captured in video frames. We utilize the domain knowledge of the court specifications to compute the transformation between 3D real world positions and 2D frame coordinates for camera calibration. Involving the physical characteristic of ball motion, we are able to recover the 3D information and reconstruct the 3D ball trajectory.

The obtained 2D trajectory and the reconstructed 3D trajectory enable manifold applications to sports information retrieval and computer-assisted game study. In basketball games, *shooting location* (the location of a player shooting the ball) is one of the important game statistics providing abundant information about the shooting tendency of a team. The statistical graph of shooting locations facilitates the coach to view the distribution of shooting

locations at a glance and to quickly comprehend where the players have higher possibility of shooting. Then, the players and the coach can infer the offense tactics of an opponent team and adapt their own defense strategy toward the team. Presently, most of the shooting location logging tasks are achieved manually. It is time-consuming and inefficient to watch a whole long video, take records and gather statistics. Thus, we propose a scheme to extract the shooting trajectory in basketball video, reconstruct the 3D trajectory and estimate the shooting location.

In volleyball games, players are not allowed to hold the ball. Hence, we detect the ball-player interaction events by utilizing the positions and the occurring times of direction changes in the trajectory. Moreover, the reconstructed 3D trajectory can provide the sports information the audience or professionals would like to know, such as set type, attack height, serve speed, serve placement, etc. Most of the informative game statistical data which cannot be directly perceived through human eyes can now be obtained based on the obtained 2D trajectory and the reconstructed 3D trajectory. Furthermore, the 3D virtual replay gives an exciting and practical visualization which enables watching the ball motion from any viewpoint.

In baseball games, the ball speed and the curvature of the ball trajectory are two main factors in determining how difficult the pitched ball can be hit. Hence, we track the pitched ball and extract the ball trajectory. Thus, ball speed and trajectory curvature can be computed for pitch analysis. Due to the capturing viewpoint and the frame rate constraint, the ball speed and trajectory curvature might not be very precise. The proposed pitch analysis is not for grading, but for entertainment effects, enriched visual presentation and sports information retrieval. In addition to ball speed and trajectory curvature, the pitch location (the relative location of the ball in/around the strike zone when the ball passes by the batter) also dominates the direction of the ball batted out. For example, a batter who swings at a lower pitch has a good chance of hitting a ground ball, while a batter who swings at a higher pitch

has a great chance of hitting the ball in the air. Since the strike zone provides reference for determining the pitch location, we propose a contour-based method to shape the strike zone according to the batter's stance. Strike/ball judgment can also be visualized on the video frames by the shaped strike zone. Besides the pitch/batter confrontation, the ball motion and the defense process after the ball is batted into the field is another focus of attention. With the field specifications, we design algorithms to recognize the spatial patterns (field lines and field objects) in frames. Then, the active regions of event occurrence in the field are classified by the spatial patterns. We can infer the ball routing patterns and defense process from the transitions of the active regions captured in the video. Content understanding and annotation can thus be achieved, providing rich information about the games.

Comprehensive experiments on basketball, volleyball and baseball videos show encouraging results. The proposed methods perform well in 2D ball tracking and 3D trajectory reconstruction from single camera video for different kinds of sports. It is our belief that the coach and players will be greatly assisted in game study with the semantic and tactical information derived from our proposed methods. Also, the audience can have a professional insight into the game.

In the following chapters, we give detailed explanation for the proposed methods and techniques. The rest of the thesis is organized as follows. Chapter 2 explains physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video. Chapter 3 describes ball tracking and 3D trajectory approximation with applications to tactics analysis in volleyball video. Chapter 4 elaborates on ball tracking, strike zone shaping and play region classification in baseball video. Finally, Chapter 5 concludes this thesis.

## **Chapter 2. Physics-Based Ball Tracking and 3D Trajectory Reconstruction with Applications to Shooting Location Estimation in Basketball Video**

The demand for computer-assisted game study in sports is growing dramatically. This chapter presents a practical video analysis system to facilitate semantic content understanding. A physics-based algorithm is designed for ball tracking and 3D trajectory reconstruction in basketball video and shooting location statistics can be obtained. The 2D-to-3D inference is intrinsically a challenging problem due to the loss of 3D information in projection to 2D frames. One significant contribution of the proposed system lies in the integrated scheme incorporating domain knowledge and physical characteristics of ball motion into object tracking to overcome the problem of 2D-to-3D inference. With the 2D trajectory extracted and the camera parameters calibrated, physical characteristics of ball motion are involved to reconstruct the 3D trajectories and estimate the shooting locations. Our experiments on broadcast basketball video show promising results. We believe the proposed system will greatly assist intelligence collection and statistics analysis in basketball games.

The rest of this chapter is organized as follows. Section 2.1 gives the introduction. Section 2.2 elaborates the overview of the proposed system. Sections 2.3, 2.4 and 2.5 present the processes of court shot retrieval, camera calibration and 2D shooting trajectory extraction, respectively. Section 2.6 explains on 3D trajectory mapping and shooting location estimation. Experimental results and discussions are presented in section 2.7. Finally, section 2.8 summaries this chapter.

### **2.1 Introduction**

The advances in video production technology and the consumer demand have led to the ever-increasing volume of multimedia information. The rapid evolution of digital equipments



allows the general users to archive multimedia data much easily. The urgent requirements for multimedia applications therefore motivate the researches in various aspects of video analysis. Sports videos, as important multimedia contents, have been extensively studied, and sports video analysis is receiving more and more attention due to the potential commercial benefits and entertaining functionalities. Major research issues of sports video analysis include: *shot classification*, *highlight extraction* and *object tracking*.

In a sports game, the positions of cameras are usually fixed and the rules of presenting the game progress are similar in different channels. Exploiting these properties, many *shot classification* methods are proposed. Duan *et al.* [1] employ a supervised learning scheme to perform a top-down shot classification based on mid-level representations, including motion vector field model, color tracking model and shot pace model. Lu and Tan [2] propose a recursive peer-group filtering scheme to identify prototypical shots for each dominant scene (e.g., wide angle-views of the court and close-up views of the players), and examine time coverage of these prototypical shots to decide the number of dominant scenes for each sports video. Mochizuki *et al.* [3] provide a baseball indexing method based on patternizing baseball scenes using a set of rectangles with image features and the motion vector.

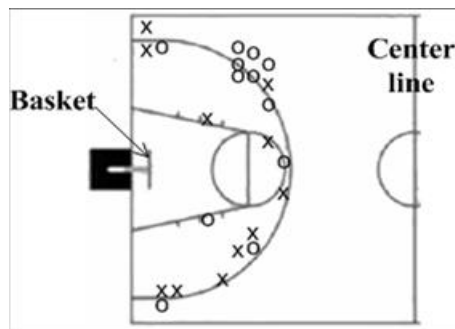
Due to broadcast requirement, *highlight extraction* attempts to abstract a long game into a compact summary to provide the audience a quick browsing of the game. Assfalg *et al.* [4] present a system for automatic annotation of highlights in soccer video. Domain knowledge is encoded into a set of finite state machines, each of which models a specific highlight. The visual cues used for highlight detection are ball motion, playfield zone, players' positions and colors of uniforms. Gong *et al.* [5] classify baseball highlights by integrating image, audio and speech cues based on maximum entropy model (MEM) and hidden Markov model (HMM). Cheng and Hsu [6] fuse visual motion information with audio features, including zero crossing rate, pitch period and Mel-frequency cepstral coefficients (MFCC), to extract baseball highlight based on hidden Markov model (HMM). Xie *et al.* [7] utilize dominant color ratio

and motion intensity to model the structure of soccer video based on the syntax and content characteristics of soccer video.

*Object tracking* is widely used in sports analysis. Since significant events are mainly caused by ball-player and player-player interactions, balls and players are tracked most frequently. Yu *et al.* [8] present a trajectory-based algorithm for ball detection and tracking in soccer video. The ball size is first proportionally estimated from salient objects (goalmouth and ellipse) to detect ball candidates. The true trajectory is extracted from potential trajectories generated from ball candidates by a verification procedure based on Kalman filter. The ball trajectory computed is applied to analyze semantic basic and complex events, team ball possession and the play-break structure. Some works of 3D trajectory reconstruction are built based on multiple cameras located on specific positions [11-14]. In addition, *computer-assisted umpiring* and *tactics inference* are burgeoning research issues of sports video analysis [11-15]. However, these can be considered as advanced applications based on ball and player tracking. Therefore, object tracking is an essential and vital issue in sports video analysis.

In this chapter, we work on the challenging issues of ball tracking and 3D trajectory reconstruction in broadcast basketball video in order to automatically gather the game statistics of *shooting locations* – the location where a player shoots the ball. Shooting location is one of the important game statistics providing abundant information about the shooting tendency of a basketball team. An example of statistical graph for shooting locations is given in Fig. 2-1, where each shooting location is marked as an **O** (score) or **X** (miss). The statistical graph for shooting locations not only provides the audience a novel insight into the game but also assists the coach in guiding the defense strategy. With the statistical graph for shooting locations, the coach is able to view the distribution of shooting locations at a glance and to quickly comprehend where the players have higher possibility of scoring by shooting. Thus, the coach can enhance the defense strategy of the team by preventing the opponents from

shooting at the locations they stand a good chance of scoring. Increasing basketball websites, such as NBA official website, provide text- and image-based web-casting, including game log, match report, shooting location and other game statistics. However, these tasks are achieved by manual efforts. It is time-consuming and inefficient to watch a whole long video, take records and gather statistics. Hence, we propose a physics-based ball tracking system for 3D trajectory reconstruction so that automatic shooting location estimation and statistics gathering can be achieved. Whether the shooting scores or not can be derived from the change of the scoreboard by close caption detection technique [16]. Thus, the statistical graph of shooting locations, as Fig. 2-1, can be generated automatically.



**Fig. 2-1.** Statistical graph of shooting locations: **O** (score) or **X** (miss).

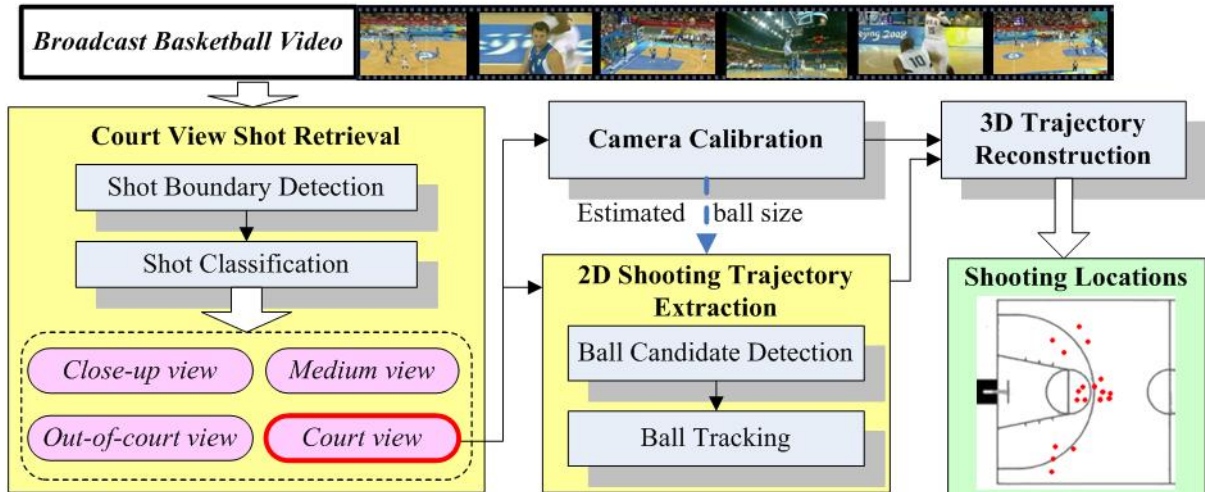
## **2.2 Overview of the Proposed Physics-Based Ball Tracking and 3D Trajectory Reconstruction System in Basketball Video**

Object tracking is usually the medium to convert the low-level features into high-level events in video processing. In spite of the long research history, it is still an arduous problem. Especially, ball tracking is a more challenging task due to the small size and fast speed. It is almost impossible to distinguish the ball within a single frame, so information over successive frames, e.g. motion information, is required to facilitate the discrimination of the ball from other objects.

To overcome the challenges of ball tracking and 3D shooting trajectory reconstruction,

an integrated system utilizing physical characteristics of ball motion is proposed, as depicted in Fig. 2-2. Basketball video contains several prototypical shots: close-up view, medium view, court view and out-of-court view. The system starts with *court shot retrieval*, because court shots can present complete shooting trajectories. Then, *2D ball trajectory extraction* is performed on the retrieved court shots. To obtain 2D ball candidates over frames, we detect ball candidates by visual features and explore potential trajectories among the ball candidates using velocity constraint. To reconstruct 3D trajectories from 2D ones, we set up the motion equations with the parameters: velocities and initial positions, to define the 3D trajectories based on physical characteristics. The 3D ball positions over frames can be represented by equations. *Camera calibration*, which provides the geometric transformation from 3D real world to 2D frames, is used to map the equation-represented 3D ball positions to 2D ball coordinates in frames. With the 2D ball coordinates over frames being known, we can approximate the parameters of the 3D motion equations. Finally, the 3D positions and velocities of the ball can be derived and the 3D trajectory is reconstructed from the 2D frame-trajectory. Having the reconstructed 3D information, the shooting locations can be estimated more accurately from 3D trajectories than from 2D trajectories, in which the z-coordinate (height) of ball is lost in camera capturing.

The major contribution of this chapter is that we reconstruct 3D information from single view 2D video sequences based on the integration of multimedia features, basketball domain knowledge and the physical characteristics of ball motion. Besides, trajectory-based high-level basketball video analysis is also provided. The 3D ball trajectories facilitate the automatic collection of game statistics about shooting locations in basketball, which greatly help the coaches and professionals to infer the *shooting tendency* of a team.

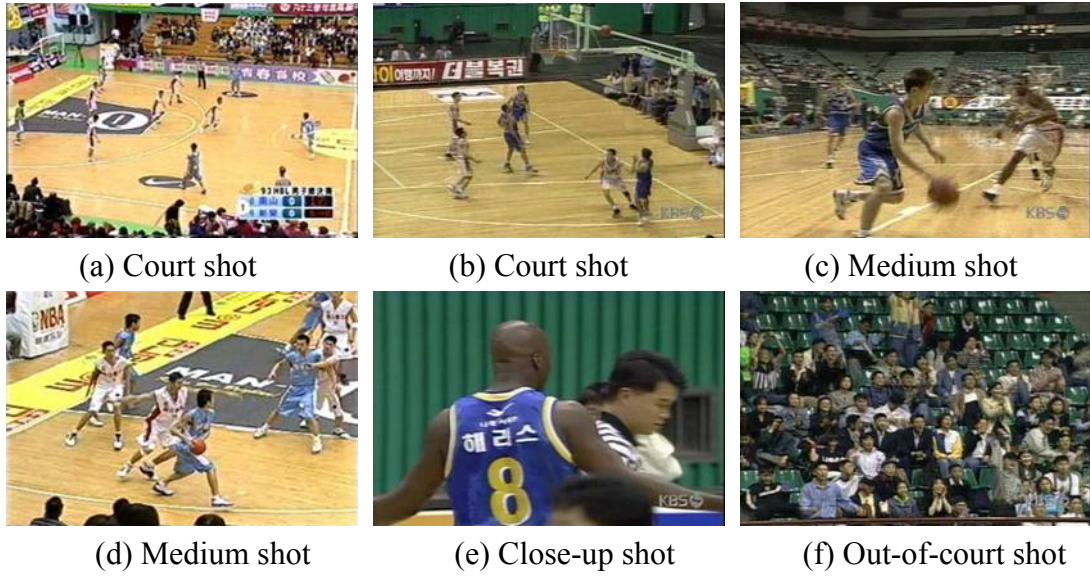


**Fig. 2-2.** Flowchart of the proposed system for ball tracking and 3D trajectory reconstruction in basketball video.

### 2.3 Court Shot Retrieval

To perform high-level analysis such as ball tracking and shooting location estimation, we should retrieve the *court shots*, which contain most of the semantic events. Shot boundary detection is usually the first step in video processing and has been extensively studied [17-19]. For computational efficiency, we apply the shot boundary detection algorithm [20,21] to segment the basketball video into shots.

To offer the proper presentation of a sports game, the camera views may switch as different events occur when a game proceeds. Thus, the information of shot types conveys important semantic cues. Motivated by this observation, basketball shots are classified into three types: 1) court view shots, 2) medium view shots, and 3) close-up view or out-of-court view shots (abbreviated as C/O shots). A *court shot* displays the global view of the court, which can present complete shooting trajectories, as shown in Fig. 2-3(a) and (b). A *medium shot*, where the player carrying the ball is focused, is a zoom-in view of a specific part of the court, as shown in Fig. 2-3(c) and (d). Containing little portion of the court, a *close-up shot* shows the above-waist view of the person(s), as shown in Fig. 2-3(e), and an *out-of-court shot* presents the audience, coach, or other places out of the court, as shown in Fig. 2-3(f).



**Fig. 2-3.** Examples of shot types in a basketball game.

Shot class can be determined from a single key frame or a set of representative frames. However, the selection of key frames or representative frames is another challenging issue. For computational simplicity, we classify every frame in a shot and assign the shot class by majority voting, which also helps to eliminate instantaneous frame misclassification.

A basketball court has one distinct dominant color—the court color. The spatial distribution of court-colored pixels and the ratio of court-colored pixels in a frame, as defined in Eq. (2-1), would vary in different view shots.

$$R = \text{\#court-colored pixels} / \text{\#pixels in a frame} \quad (2-1)$$

To compute the court-colored pixel ratio  $R$  in each frame, we apply the algorithm in [22], which learns the statistics of the court color, adapts these statistics to changing imaging and then detects the court-colored pixels. Intuitively, a high  $R$  value indicates a court view, a low  $R$  value corresponds to a C/O view, and in between, a medium view is inferred. The feature  $R$  is indeed sufficient to discriminate C/O shots from others, but medium shots with high  $R$  value might be misclassified as court shots.

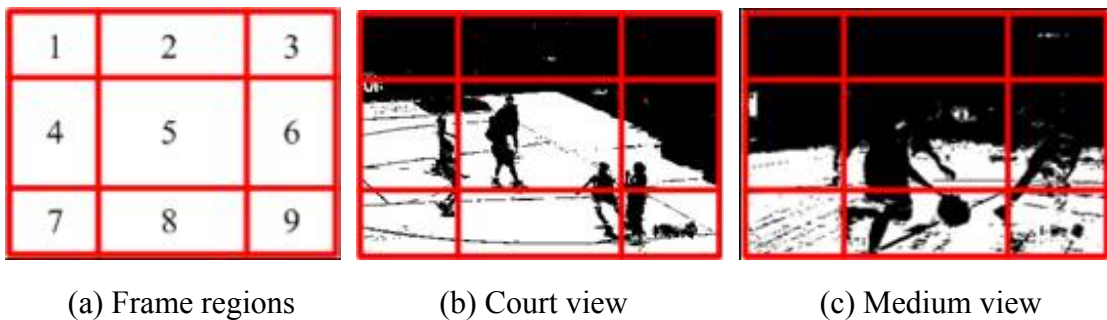
Thus, we propose a compute-easy, yet effective, algorithm to discriminate between court



shots and medium shots. As shown in Fig. 2-4, we define the nine frame regions by employing *Golden Section* spatial composition rule [23,24], which suggests dividing up a frame in 3: 5: 3 proportion in both horizontal and vertical directions. Fig. 2-4 displays the examples of the regions obtained by golden section rule on medium and court views. To distinguish medium views from court views, the feature  $R_{5\cup 8}$  defined in Eq. (2-2) is utilized on the basis of the following observation.

$$R_{5\cup 8} : \text{the } R \text{ value in the union of region 5 and region 8} \quad (2-2)$$

A medium view zooms in to focus on a specific player and usually locates the player around the frame center. Since players are composed of non-court-colored pixels, a medium view would have low  $R$  values in the center regions (region 2, 5 and 8). A court view aims at presenting the global viewing, so the players are distributed over the frames. Therefore, a court view would have higher  $R$  values in the center regions (region 2, 5 and 8) than those of a medium view. However, the upper section of a frame is usually occupied by the audience or advertising boards, so region 2 is not taken into consideration. Only the  $R$  values in region 5 and region 8 are considered for classification: court views have higher  $R_{5\cup 8}$  than that of medium views.



**Fig. 2-4.** Examples of Golden Section spatial composition.

## 2.4 Camera Calibration

Camera calibration is an essential task to provide geometric transformation mapping the positions of the ball and players in the 2D video frames to 3D real-world coordinates or vice

versa [25,26]. However, the 2D-to-2D transformation with court model known is not sufficient to reconstruct 3D trajectory due to the disregard of height information. In addition to the feature points on the court plane, some non-coplanar feature points are also taken into consideration in our system to keep the height information.

The geometric transformation from 3D real world coordinate  $(x, y, z)$  to 2D image coordinate  $(u', v')$  can be represented as Eq. (2-3):

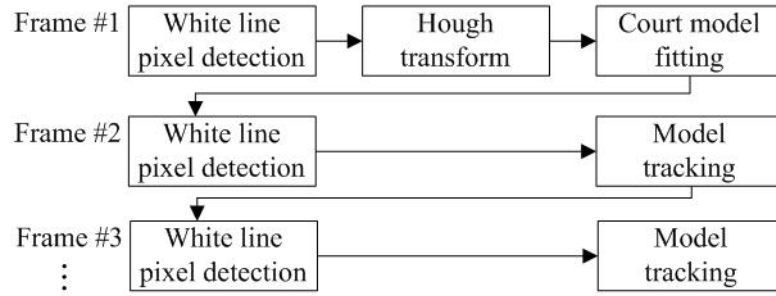
$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \triangleq \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad \text{where} \quad \begin{matrix} u' = \frac{u}{w} \\ v' = \frac{v}{w} \end{matrix} \quad (2-3)$$

The eleven camera parameters  $c_{ij}$  can be calculated from at least six non-coplanar points whose positions are both known in the court model and in frames. Since the detection of lines is more robust than locating the accurate positions of specific points, the intersections of lines are utilized to establish point-correspondence.

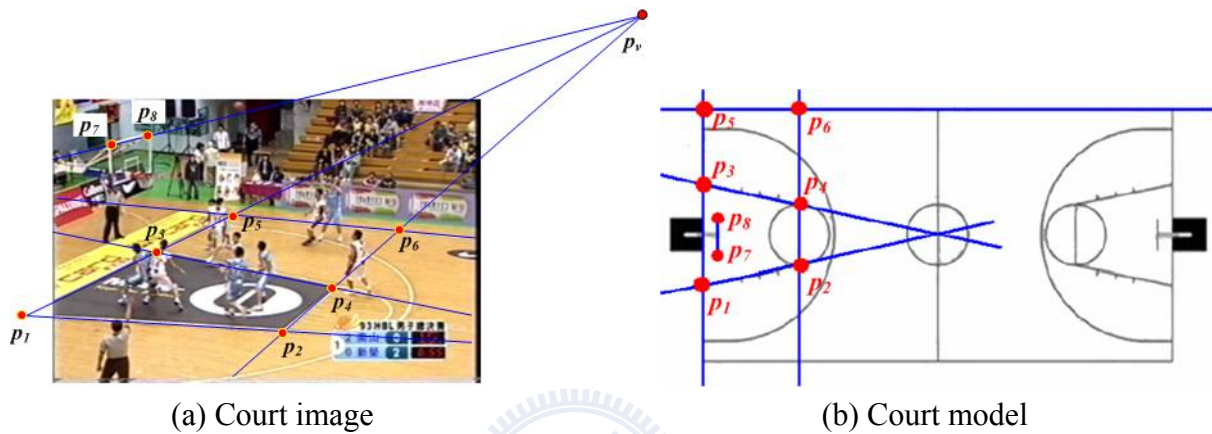
Fig. 2-5 depicts the flowchart of camera calibration. In the process, we make use of ideas in general camera calibration, such as white line pixel detection and line extraction [25]. We start with identifying the white line pixels exploiting the constraints of color and local texture. To extract feature lines, the Hough transform is applied to the detected white line pixels. Then, we compute the intersection points of court lines and end points of the backboard border. With these corresponding points whose positions are both known in 2D frame and in the court model, as shown in Fig. 2-6, the 3D-to-2D transformation can be computed and the camera parameters are then derived.

For the subsequent frames, we apply the *model tracking* mechanism [25], which predicts the camera parameters from the previous frame in spite of the camera motion, to improve the efficiency since Hough transform and court model fitting need not be performed again. For more detailed process, please refer to the paper [25].





**Fig. 2-5.** Flowchart of camera calibration.

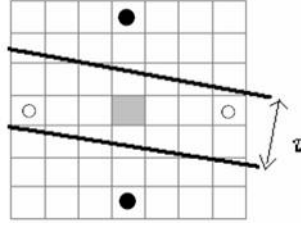


**Fig. 2-6.** Point-correspondence between the 2D frame and the basketball court model

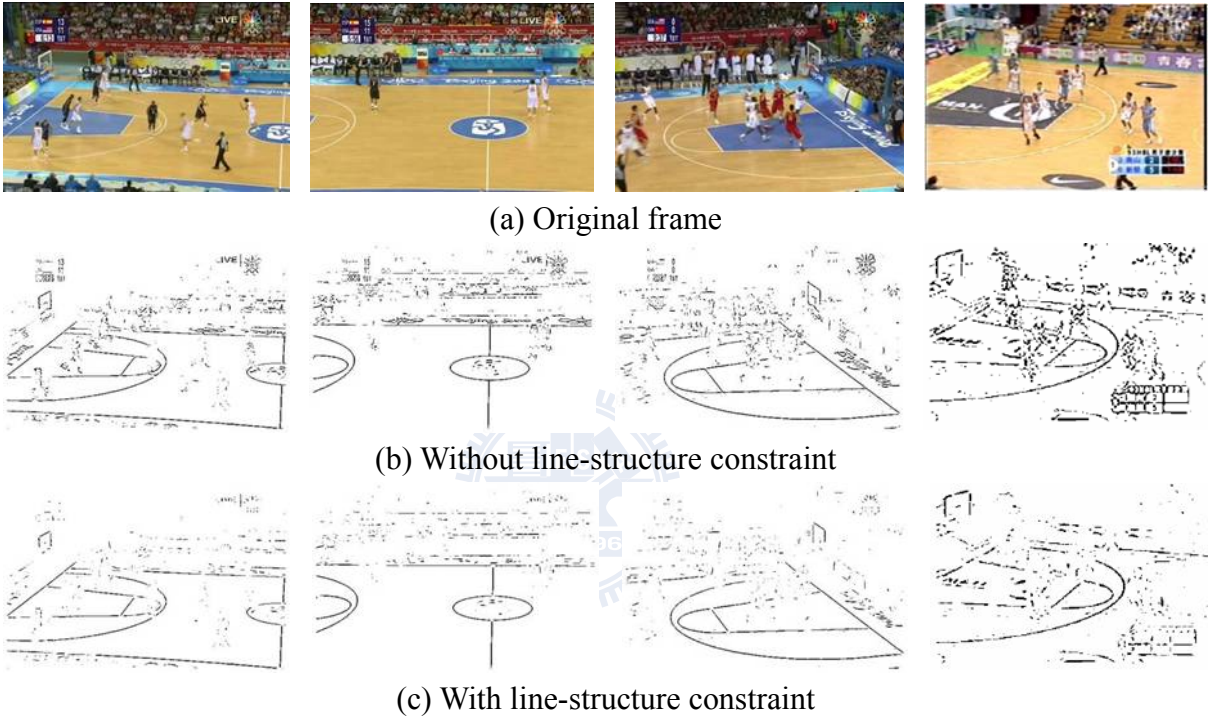
### 2.4.1 White line pixel detection

For visual clarity, the court lines and important markers are in white color, as specified in the official game rules. However, there may exist other white objects in the images such as advertisement logos and the uniforms of the players. Hence, additional criteria are needed to further constrain the set of white line pixels.

As illustrated in Fig. 2-7, each square represents one pixel and the central one drawn in gray is a candidate pixel. Assuming that white lines are typically no wider than  $\tau$  pixels ( $\tau = 6$  in our system), we check the brightness of the four pixels, marked ‘●’ and ‘○’, at a distance of  $\tau$  pixels away from the candidate pixel on the four directions. The central candidate pixel is identified as a white line pixel only if both pixels marked ‘●’ or both pixels marked ‘○’ are with lower brightness than the candidate pixel. This process prevents most of the pixels in white regions or white uniforms being detected as white line pixels, as shown in Fig. 2-8 (b).



**Fig. 2-7.** Illustration of part of an image containing a white line.



**Fig. 2-8.** Sample results of white line pixel detection.

To improve the accuracy and efficiency of the subsequent Hough transform for line detection and court model fitting, we apply the line-structure constraint [25] to exclude the white pixels in finely textured regions. The structure matrix  $S$  [27] computed over a small window of size  $2b+1$  (we use  $b=2$ ) around each candidate pixel  $(p_x, p_y)$ , as defined in Eq. (2-4), is used to recognize texture regions.

$$S = \sum_{x=p_x-b}^{p_x+b} \sum_{y=p_y-b}^{p_y+b} \nabla Y(x, y) \cdot (\nabla Y(x, y))^T \quad (2-4)$$

Depending on the two eigenvalues of matrix  $S$ , called  $\lambda_1$  and  $\lambda_2$  ( $\lambda_1 \geq \lambda_2$ ), the texture can be

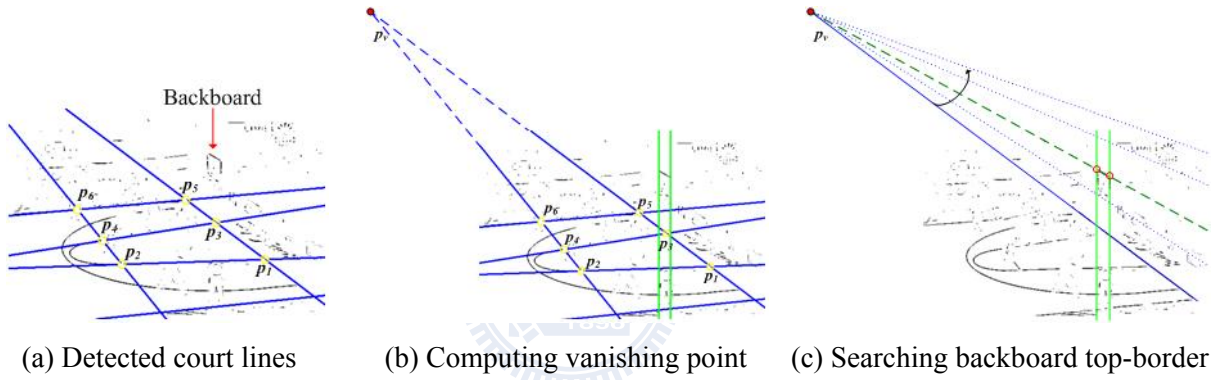
classified into *textured* ( $\lambda_1, \lambda_2$  are large), *linear* ( $\lambda_1 \gg \lambda_2$ ) and *flat* ( $\lambda_1, \lambda_2$  are small). On the straight court lines, the *linear* case will apply to retain the white pixels only if  $\lambda_1 > \alpha\lambda_2$  ( $\alpha = 4$  in our system). Fig. 2-8 demonstrates sample results of white line pixel detection. The original frames are presented in Fig. 2-8(a). In Fig. 2-8(b), although most of the white pixels in white regions or white uniforms are discarded, there are still many false detections of white line pixels occurring in the textured areas. With line-structure constraint, Fig. 2-8(c) shows that the number of false detections is reduced and white line pixel candidates are retained only if the pixel neighbor shows a linear structure.

#### 2.4.2 Line extraction

To extract the court lines and the backboard border, we perform a standard Hough transform on the detected white line pixels. The parameter space  $(\theta, d)$  is used to represent the line:  $\theta$  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  $(\theta, d)$  and sample the accumulator matrix at a resolution of one degree for  $\theta$  and one pixel for  $d$ . Since a line in  $(x, y)$  space corresponds to a point in  $(\theta, d)$  space, line candidates can be determined by extracting the local maxima in the accumulator matrix. The court line intersections on the court plane can be obtained by the algorithm of finding line-correspondences in [25], which has good performance in 2D-to-2D court model mapping. A sample result is presented in Fig. 2-9 (a).

To reconstruct 3D information of the ball movement, we need two more points which are not on the court plane to calculate the calibration parameters. The two endpoints of the *backboard top-border* ( $p_7$  and  $p_8$  as shown in Fig. 2-6) are selected because the light condition makes the white line pixels of the backboard top-border easy to detect in frames. Fig. 2-9 presents the process of the detection of backboard top-border. In 3D real world, the backboard top-border is parallel with the court lines  $(p_1, p_3, p_5)$  and  $(p_2, p_4, p_6)$ . According to

vanishing point theorem, parallel lines in 3D space viewed in a 2D frame appear to meet at a point, called *vanishing point*. Therefore, the lines  $(p_1, p_3, p_5)$ ,  $(p_2, p_4, p_6)$  and the backboard top-border in the frame will meet at the vanishing point. Utilizing this characteristic, the vanishing point  $p_v$  can be computed as the intersection of the extracted court lines  $(p_1, p_3, p_5)$  and  $(p_2, p_4, p_6)$ , as shown in Fig. 2-9(b). Besides, we also detect two vertical line segments above the court line  $(p_1, p_3, p_5)$ . Then, Hough transform is performed on the area between the two vertical lines above the court line  $(p_1, p_3, p_5)$ . The detected line segment whose extension passes the vanishing point is extracted as the backboard top-border, as shown in Fig. 2-9(c).



**Fig. 2-9.** Detection of backboard top-border.

### 2.4.3 Computation of camera calibration parameters

Multiplying out the linear system in Eq. (2-3), we obtain two equations, Eq. (2-5) and Eq. (2-6), for *each corresponding point*—the point whose coordinate is both known in the 3D court model  $(x, y, z)$  and in the frame  $(u', v')$ .

$$c_{11}x + c_{12}y + c_{13}z + c_{14} = u'(c_{31}x + c_{32}y + c_{33}z + 1) \quad (2-5)$$

$$c_{21}x + c_{22}y + c_{23}z + c_{24} = v'(c_{31}x + c_{32}y + c_{33}z + 1) \quad (2-6)$$

To compute the calibration parameters  $c_{ij}$ , we set up a linear system  $\mathbf{AC} = \mathbf{B}$  from Eq. (2-5) and Eq. (2-6):

$$\begin{bmatrix}
x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u'_1 x_1 & -u'_1 y_1 & -u'_1 z_1 \\
0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v'_1 x_1 & -v'_1 y_1 & -v'_1 z_1 \\
x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u'_2 x_2 & -u'_2 y_2 & -u'_2 z_2 \\
0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v'_2 x_2 & -v'_2 y_2 & -v'_2 z_2 \\
& & & & & & & & & & \cdot \\
& & & & & & & & & & \cdot \\
& & & & & & & & & & \cdot \\
& & & & & & & & & & \cdot \\
& & & & & & & & & & \cdot \\
x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & -u'_N x_N & -u'_N y_N & -u'_N z_N \\
0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & -v'_N x_N & -v'_N y_N & -v'_N z_N
\end{bmatrix}_{2N \times 11} \mathbf{A} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{24} \\ c_{31} \\ c_{32} \\ c_{33} \end{bmatrix}_{11 \times 1} \mathbf{C} = \begin{bmatrix} u'_1 \\ v'_1 \\ u'_2 \\ v'_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u'_N \\ v'_N \end{bmatrix}_{2N \times 1} \mathbf{B} \quad (2-7)$$

$N$  is the number of corresponding points. In our process,  $N = 8$ : six are the court line intersections and two are the endpoints of the backboard top-border. To solve  $\mathbf{C}$ , we can over-determine  $\mathbf{A}$  and find a least squares fitting for  $\mathbf{C}$  with a pseudo-inverse solution:

$$\mathbf{AC} = \mathbf{B}, \quad \mathbf{A}^T \mathbf{AC} = \mathbf{A}^T \mathbf{B}, \quad \mathbf{C} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (2-8)$$

Thus, the parameters of camera calibration can be derived to form the matrix which transforms 3D real world coordinate to 2D image coordinate.

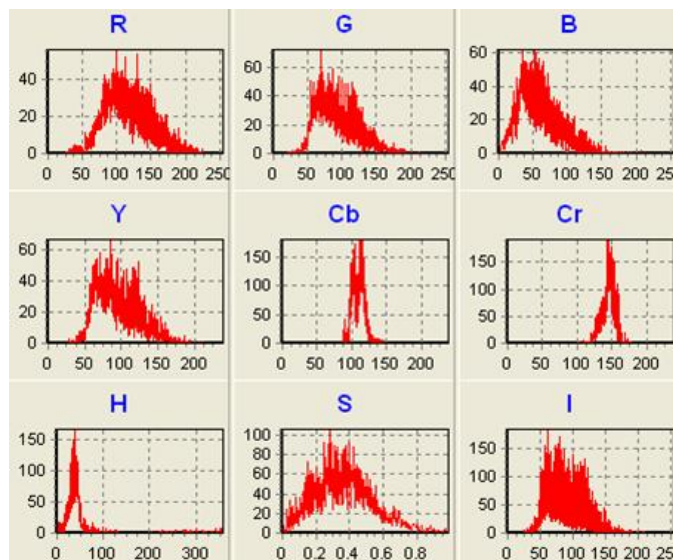
## 2.5 2D Shooting Trajectory Extraction

The ball is the most important focus of attention in basketball either for the players or for the audience. It is a challenging task to identify the ball in video frames due to its small size in court views and its fast movement. In this section, we aim at extracting the shooting trajectories in court shots. When a shooting event occurs, one of the backboards should be captured in the frames. Therefore, our system performs ball candidate detection and ball tracking on the frames with a backboard detected in court shots.

### 2.5.1 Ball candidate detection

The detection of *ball candidates*, the basketball-colored moving objects, requires extracting the pixels which are 1) moving and 2) in basketball color. For moving pixel

detection, frame difference is a compute-easy and effective method. We extract the pixels with significant luminance difference between consecutive frames as moving pixels. Color is another important feature to extract ball pixels. However, the color of the basketball in frames might vary due to the different angles of view and lighting conditions. To obtain the color distribution of the basketball in video frames, 30 ball images are segmented manually from different basketball videos to produce the respective color histograms in RGB, YCbCr and HSI color spaces, as shown in Fig. 2-10. After statistical analysis, the Hue value in HSI color space has better discriminability and is selected as the color feature and the ball color range is set to  $[H_a, H_b]$ . We compute the average Hue value for each  $4 \times 4$  block in frames and discard the moving pixels in the blocks of which the average Hue values are not within the ball color range  $[H_a, H_b]$ . To remove noises and gaps, morphological operations are performed on the remaining moving pixels, called *ball pixels*. An example of ball pixel detection is shown in Fig. 2-11. Fig. 2-11(a) is the original frame and Fig. 2-11(b) shows the moving pixels detected by frame difference. The extracted ball pixels after morphological operations are presented in Fig. 2-11(c).



**Fig. 2-10.** Color histograms of 30 manually segmented basketball images

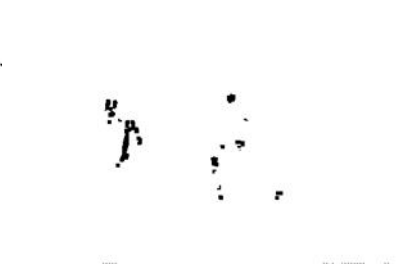




(a) Source frame



(b) Moving pixels



(c) Extracted ball pixels

**Fig. 2-11.** Illustration of ball pixel detection.

With the extracted ball pixels, objects are formed in each frame by region growing. To prune non-ball objects, we design two sieves based on visual properties:

**1) Shape sieve:** The ball in frames might have a shape different from a circle, but the deformation is not so dramatic that its aspect ratio should be within the range  $[1/R_a, R_a]$  in most frames. We set  $R_a = 3$  since the object with aspect ratio  $> 3$  or  $< 1/3$  is far from a ball and should be eliminated.

**2) Size sieve:** The in-frame ball diameter  $D_{frm}$  can be proportionally estimated from the length between the court line intersections by pinhole camera imaging principal, as Eq. (2-9):

$$(D_{frm} / D_{real}) = (L_{frm} / L_{real}) \quad , \quad D_{frm} = D_{real} (L_{frm} / L_{real}) \quad (2-9)$$

where  $D_{real}$  is the diameter of a real basketball ( $\approx 24\text{cm}$ ),  $L_{frm}$  and  $L_{real}$  are the in-frame length and the real-world length of a corresponding line segment, respectively. To compute the ratio  $(L_{frm} / L_{real})$ , we select the two points closest to the frame center from the six court line intersections and calculate the in-frame distance  $L_{frm}$  of the selected two points. Since the distance of the two points in real court  $L_{real}$  is specified in the basketball rules, the ratio  $(L_{frm} / L_{real})$  can be computed out. Thus, the planar ball size in the frame can be estimated as  $\pi \cdot (D_{frm}/2)^2$ . The size sieve filter out the objects of which the sizes are not within the range  $[\pi \cdot (D_{frm}/2)^2 - \Delta, \pi \cdot (D_{frm}/2)^2 + \Delta]$ , where  $\Delta$  is the extension for tolerance toward processing faults.

It would be a difficult task to detect and track the ball if there is camera motion. There

are two major problems we may confront. The first is that more moving pixels are detected due to the camera motion and therefore more ball candidates might exist. However, our analysis is focused on the shooting trajectories in court shots. To capture and present the large portion of the court, the camera is usually located at a distance from the court. The camera motion is not so violent in court shots except for a rapid camera transition from one half-court to the other, as can be noted in Fig. 2-12, where the left image shows the detected ball candidates marked in the yellow circles, and the right image presents the camera motion using motion history image [28], generated from 45 consecutive frames. When a shooting event occurs, one of the backboards should be captured in the frames. During the camera transition since no backboard shows on the screen, our system need not perform ball candidate detection. That is, the performance of ball candidate detection is not affected by the camera moving from one half-court to the other. Second, it is possible (although it is rare in practice) that the ball might have little motion or stay still on the screen when the camera attempts to follow the ball. However, we observe in experiments that the ball is hardly at exactly the same position in consecutive frames even if the camera follows the ball. Although there are still some misses in moving pixel detection in this case due to the mild motion of the ball in frames, the pixels of the true ball can be correctly detected in most frames. The missed ball candidate can be recovered from the ball positions in the previous and the subsequent frames by interpolation.





(a) Fewer ball candidates produced if the camera motion is small.



(b) More ball candidates would be produced if there is large camera motion.

**Fig. 2-12.** Left: detected ball candidates, marked as yellow circles. Right: motion history image to present the camera motion.

### 2.5.2. Ball tracking

Many non-ball objects might look like a ball in video frames and it is difficult to recognize which is the true one. Therefore, we integrate the physical characteristic of the ball motion into a dynamic programming-based route detection mechanism to track the ball candidates, generate potential trajectories and identify the true ball trajectory.

For ball tracking, we need to compute the ball velocity constraint first. Since the displacement of the ball in a long shoot would be larger than that in a short shoot, we take a long shoot into consideration, as diagramed in Fig. 2-13. The time duration from the ball leaving the hand to the ball reaching the peak in the trajectory  $t_1$  and the time duration of the ball moving from the peak to the basket  $t_2$  can be represented by Eq. (2-10) and Eq. (2-11), respectively:

$$H+h = g t_1^2 / 2 , \quad t_1 = [2(H+h)/g]^{1/2} \quad (2-10)$$

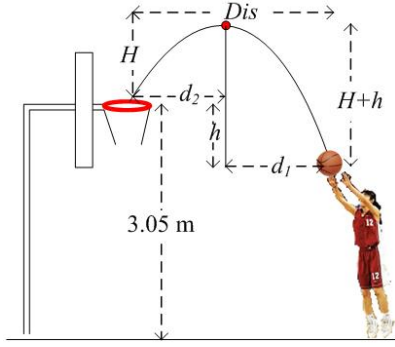
$$H = g t_2^2 / 2 , \quad t_2 = (2H/g)^{1/2} \quad (2-11)$$

where  $g$  is the gravity acceleration ( $9.8 \text{ m/s}^2$ ) and  $t$  is the time duration,  $H$  and  $h$  is the vertical distances from the basket to the trajectory peak and to the position of ball leaving the hand, respectively. Thus, the highest vertical velocity  $V_v$  of the ball in the trajectory should be  $V_v = g t_1$  and the horizontal velocity  $V_h$  can be calculated as  $V_h = Dis / (t_1+t_2)$ , where  $Dis$  is the distance from the shooter to the basket center. With the vertical and horizontal velocities, the ball velocity  $V_b$  can be derived as Eq. (2-12):

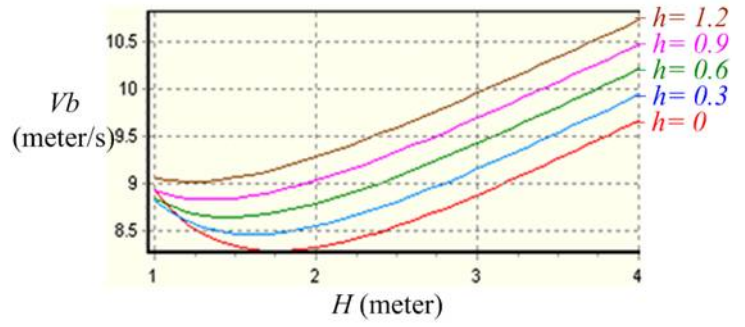
$$V_b = (V_h^2 + V_v^2)^{1/2} \quad (2-12)$$

$V_b$  value increases as  $Dis$  increases. Since our goal is to compute the upper limit of the ball velocity, we consider the distance from the 3-point line to the basket (6.25m), which is almost the longest horizontal distance from the shooter to the basket. To cover all cases, we set  $Dis = 7\text{m}$ . Considering an  $l$  meter tall player, the height of the ball leaving the hand should be higher than  $(l+0.2)$  m. Thus, the value  $h$  should be less than  $(3.05-0.2-l)$  m. To cover most players, we set  $l = 1.65$ , that is,  $h \leq 1.2$ . Besides, there are few shooting trajectories with the vertical distance  $H$  greater than 4 meters. Given different  $h$  values (0, 0.3, 0.6, 0.9 and 1.2), the values of  $V_b$  computed using Eq. (2-10)-(2-12) for  $H$  varying between 1 and 4 meters are plotted in Fig. 2-14, showing the reasonable values of  $V_b$ . It can be observed that, when  $H = 4$  m and  $h = 1.2$  m, we have the maximum value of  $V_b$  ( $\approx 10.8 \text{ m/s}$ ). Thus, we set the velocity constraint (upper limit) as  $V_b \approx 10.8 \text{ m/s} \approx 36 \text{ cm/frame}$ . Finally, similar to Eq. (2-9), the in-frame velocity constraint  $V_c$  can be proportionally estimated by applying pinhole camera imaging principle as Eq. (2-13):

$$(V_c / V_b) = (L_{frm} / L_{real}) , \quad V_c = V_b (L_{frm} / L_{real}) \quad (2-13)$$



**Fig. 2-13.** Diagram of a long shoot.



**Fig. 2-14.** Relation between  $V_b$  and  $H$ .

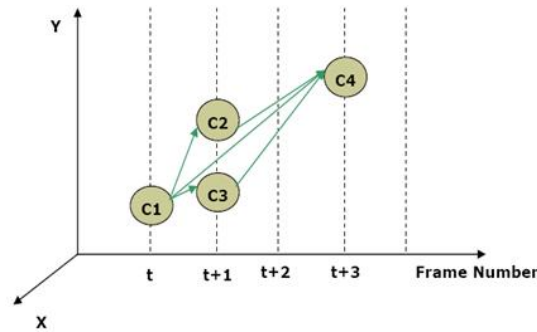
The goal of ball velocity constraint is to determine the search range for ball tracking. To avoid ball missing in ball tracking, what we want to derive is the upper limit of in-frame ball velocity. Hence, although there may be deviation of in-frame ball velocity due to the different relationship between the angle of camera shooting and the angle of player's shooting, the derived upper limit of ball velocity still significantly improves the computational efficiency and accuracy for ball tracking by setting an appropriate search range.

Fig 2-15 illustrates the ball tracking process. The X and Y axes represent the in-frame coordinates of ball candidates, and the horizontal axis indicates the frame number. The nodes C1, C2, C3 and C4 represent the ball candidates. Initially, for the first frame of a court shot, each ball candidate is considered as the root of a trajectory. For the subsequent frames, we check if any ball candidate can be added to one of the existing trajectories based on the velocity property. The in-frame ball velocity can be computed by Eq. (2-14):

$$Velocity_{i \rightarrow j} = \frac{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{t_{i \rightarrow j}} \quad (2-14)$$

where  $i$  and  $j$  are frame indexes,  $(x_i, y_i)$  and  $(x_j, y_j)$  are the coordinates of the ball candidates in frame  $i$  and frame  $j$ , respectively, and  $t_{i \rightarrow j}$  is the time duration. Trajectories grow by adding the ball candidates in the subsequent frames which satisfy the velocity constraint. Although it is possible that no ball candidate is detected in some frames, the trajectory growing process

does not terminate until no ball candidate is added to the trajectory for  $T_f$  consecutive frames (we use  $T_f = 5$ ). The missed ball position(s) can be estimated from the ball positions in the previous and the subsequent frames by interpolation.



**Fig. 2-15.** Illustration of ball tracking process. (X and Y represent ball coordinates)

To extract the shooting trajectory, we exploit the characteristic that the ball trajectories are near parabolic (or ballistic) due to the gravity, even though the trajectories are not actually parabolic curves because of the effect of the air friction, ball spin, etc. As illustrated in Fig. 2-16, we compute the best-fitting quadratic function  $f(x)$  for each route using the least-squares-fitting technique of regression analysis and determine the *distortion* as the average of the distances from ball candidate positions to the parabolic curve. A shooting trajectory is then verified according to its length and the distortion. Although the passing trajectories are often more linear in nature, still some passing trajectories in the form of parabolic (or ballistic) curves are verified as shooting trajectories. We can further identify a shooting trajectory by examining if it approaches the backboard. Thus, the passing trajectories can be discarded even though they may be parabolic (or ballistic).

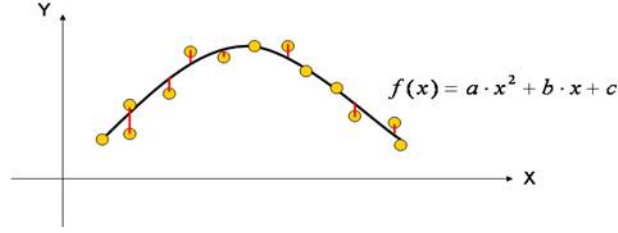


Fig. 2-16. Illustration of the best-fitting function.

## 2.6 3D Trajectory Reconstruction and Shooting Location Estimation

With the 2D trajectory extracted and the camera parameters calibrated, now we are able to employ the physical characteristics of ball motion in real world for 3D trajectory reconstruction. The relationship between each pair of corresponding points in the 2D space  $(u', v')$  and 3D space  $(X_c, Y_c, Z_c)$  is given in Eq. (2-3). Furthermore, the ball motion should fit the physical properties, so we can model the 3D trajectory as:

$$\begin{aligned} X_c &= x_0 + V_x t \\ Y_c &= y_0 + V_y t \\ Z_c &= z_0 + V_z t + gt^2/2 \end{aligned} \quad (2-15)$$

where  $(X_c, Y_c, Z_c)$  is the 3D real world coordinate,  $(x_0, y_0, z_0)$  is the initial 3D coordinate of the ball in the trajectory,  $(V_x, V_y, V_z)$  is the 3D ball velocity,  $g$  is the gravity acceleration and  $t$  is the time interval. Substituting  $X_c, Y_c$  and  $Z_c$  in Eq. (2-3) by Eq. (2-15), we obtain:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 + V_x t \\ y_0 + V_y t \\ z_0 + V_z t + \frac{1}{2} g t^2 \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \triangleq \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad \text{where} \quad \begin{aligned} u' &= \frac{u}{w} \\ v' &= \frac{v}{w} \end{aligned} \quad (2-16)$$

Multiplying out the equation with  $u = u'w$  and  $v = v'w$ , we get two equations for each ball candidate:

$$\begin{aligned} c_{11} x_0 + c_{11} V_x t + c_{12} y_0 + c_{12} V_y t + c_{13} z_0 + c_{13} V_z t + c_{13} g t^2 / 2 + c_{14} \\ = u' (c_{31} x_0 + c_{31} V_x t + c_{32} y_0 + c_{32} V_y t + c_{33} z_0 + c_{33} V_z t + c_{33} g t^2 / 2 + 1) \end{aligned} \quad (2-17)$$

$$\begin{aligned} c_{21} x_0 + c_{21} V_x t + c_{22} y_0 + c_{22} V_y t + c_{23} z_0 + c_{23} V_z t + c_{23} g t^2 / 2 + c_{24} \\ = v' (c_{31} x_0 + c_{31} V_x t + c_{32} y_0 + c_{32} V_y t + c_{33} z_0 + c_{33} V_z t + c_{33} g t^2 / 2 + 1) \end{aligned} \quad (2-18)$$

Since the eleven camera calibration parameters  $c_{ij}$  and the time of each ball candidate on the trajectory are known, we set up a linear system  $\mathbf{D}_{2N \times 6} \mathbf{E}_{6 \times 1} = \mathbf{F}_{2N \times 1}$  from Eq. (2-17) and Eq. (2-18) to compute the six unknowns ( $x_0, V_x, y_0, V_y, z_0, V_z$ ) of the parabolic (or ballistic) trajectory:

$$\begin{bmatrix} c_{11} - u_1' c_{31} & c_{11} t_1 - u_1' c_{31} t_1 & c_{12} - u_1' c_{32} & c_{12} t_1 - u_1' c_{32} t_1 & c_{13} - u_1' c_{33} & c_{13} t_1 - u_1' c_{33} t_1 \\ c_{21} - v_1' c_{31} & c_{21} t_1 - v_1' c_{31} t_1 & c_{22} - v_1' c_{32} & c_{22} t_1 - v_1' c_{32} t_1 & c_{23} - v_1' c_{33} & c_{23} t_1 - v_1' c_{33} t_1 \\ c_{11} - u_2' c_{31} & c_{11} t_2 - u_2' c_{31} t_2 & c_{12} - u_2' c_{32} & c_{12} t_2 - u_2' c_{32} t_2 & c_{13} - u_2' c_{33} & c_{13} t_2 - u_2' c_{33} t_2 \\ c_{21} - v_2' c_{31} & c_{21} t_2 - v_2' c_{31} t_2 & c_{22} - v_2' c_{32} & c_{22} t_2 - v_2' c_{32} t_2 & c_{23} - v_2' c_{33} & c_{23} t_2 - v_2' c_{33} t_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{11} - u_N' c_{31} & c_{11} t_N - u_N' c_{31} t_N & c_{12} - u_N' c_{32} & c_{12} t_N - u_N' c_{32} t_N & c_{13} - u_N' c_{33} & c_{13} t_N - u_N' c_{33} t_N \\ c_{21} - v_N' c_{31} & c_{21} t_N - v_N' c_{31} t_N & c_{22} - v_N' c_{32} & c_{22} t_N - v_N' c_{32} t_N & c_{23} - v_N' c_{33} & c_{23} t_N - v_N' c_{33} t_N \end{bmatrix} \begin{bmatrix} x_0 \\ V_x \\ y_0 \\ V_y \\ z_0 \\ V_z \end{bmatrix} = \mathbf{F}$$

$\mathbf{D}$   $2N \times 6$   $6 \times 1$

$$= \begin{bmatrix} u_1' (c_{33} g t_1^2 / 2 + 1) - (c_{13} g t_1^2 / 2 + c_{14}) \\ v_1' (c_{33} g t_1^2 / 2 + 1) - (c_{23} g t_1^2 / 2 + c_{24}) \\ u_2' (c_{33} g t_2^2 / 2 + 1) - (c_{13} g t_2^2 / 2 + c_{14}) \\ v_2' (c_{33} g t_2^2 / 2 + 1) - (c_{23} g t_2^2 / 2 + c_{24}) \\ \vdots \\ u_N' (c_{33} g t_N^2 / 2 + 1) - (c_{13} g t_N^2 / 2 + c_{14}) \\ v_N' (c_{33} g t_N^2 / 2 + 1) - (c_{23} g t_N^2 / 2 + c_{24}) \end{bmatrix}_{2N \times 1}$$

$\mathbf{F}$

(2-19)

where  $N$  is the number of ball candidates on the trajectory and  $(u_i', v_i')$  are the 2D coordinates of the candidates. Similar to Eq. (2-8), we can over-determine  $\mathbf{D}$  with three or more ball candidates on the 2D trajectory and find a least squares fitting for  $\mathbf{E}$  by pseudo-inverse. Finally, the 3D trajectory can be reconstructed from the six physical parameters ( $x_0, V_x, y_0, V_y, z_0, V_z$ ).

Originally, the definition of shooting location should be the location of the player shooting the ball. However, the starting position of the trajectory is almost the position of the ball leaving the player's hand. Thus, we can estimate the shooting location on the court model



as  $(x_0, y_0, 0)$  via projecting the starting position of the trajectory onto the court plane. Moreover, the occurring time of a shooting action can also be recorded for event indexing and retrieval.

## 2.7 Experimental Results and Discussions in Basketball Video

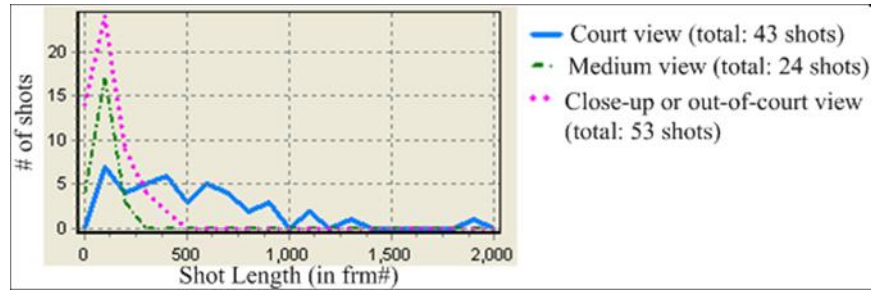
The framework elaborated in the previous sections supports shot classification, ball tracking, 3D trajectory reconstruction and shooting location estimation. For performance evaluation the proposed system has been tested on broadcast basketball video sequences: 1) the Olympics gold medal game: USA vs. Spain, 2) the Olympics game: USA vs. China, 3) one Taiwan high-school basketball league (HBL) game and 4) one Korea basketball game. The replay shots can be eliminated in advance by previous researches of replay detection [17, 29]. In the following, the parameter setting and experimental results are presented.



### 2.7.1 Parameter setting

Although the basketball courts are similar in different games, they would be captured in different lighting conditions and the quality of video would be different. Hence, the thresholds should be determined adaptively. For court shot retrieval, two thresholds  $T_{c/o}$  and  $T_{court}$  are used. A frame with the dominant color ratio  $R \leq T_{c/o}$  is assigned as a C/O view. When  $R > T_{c/o}$ , the frame is classified as a court view ( $R_{5\cup 8} > T_{court}$ ) or a medium view ( $R_{5\cup 8} \leq T_{court}$ ). The thresholds are automatically learned as explained in the following. Some court shots can be first located using shot length since the shots with long lengths are mostly court shots. This can be verified by the statistical data of the shot lengths for different shot classes, as shown in Fig. 2-17, which is constructed from 120 shots with shot classes already known. Starting with roughly initialized threshold ( $T_{c/o} = \text{average } R \text{ in all frames}$ ), each shot with long length ( $>600$  frames) and high court-colored pixel ratio ( $R > T_{c/o}$ ) is classified as a court shot.

We construct the  $R_{5\cup 8}$  histogram of those shots passing the shot length and  $R$  constraints.  $T_{court}$  is determined in such a way that the percentage of the frames with  $R_{5\cup 8} > T_{court}$  contained in the qualified shots should be  $\geq 70\%$ . Then,  $T_{clo}$  is re-adjusted to the average  $R$  of the frames excluding the frames of court shots.



**Fig. 2-17.** Statistical data of the shot lengths for different basketball shot classes.

For ball candidate detection, the ball Hue color range  $[H_a, H_b]$  is determined statistically. With the Hue histogram constructed from 30 ball images manually segmented out of different basketball sequences, as shown in Fig. 2-10, the range  $[H_a, H_b]$  is selected to cover 80% of the pixels of the 30 ball images. An alternative way to determine the ball color range is that the system provides frames of court shots for the user to locate the ball and then computes  $[H_a, H_b]$ .

### 2.7.2 Performance of basketball shot boundary detection and court shot retrieval

In sports videos, gradual transitions usually accompany replay shots. The shot boundaries are almost cut-type after replay shot elimination. Thus, we achieve good performance of overall 96.38% recall rate and 91.51% precision rate in shot boundary detection, as reported in Table 2-1. The misses are mainly caused by the strong correlation of the court color between shots, while special effects, high camera motion and the drastic action of the players in close-up view lead to false alarms.

Since our final applications are ball tracking and shooting location estimation, we favor



court shots over other shots. The results of court shots retrieval are presented in Table 2-2 (only the correctly segmented shots are used). We achieve high recall rate (98.59%) so that few shooting events are missed. The results of shot boundary detection and court shot retrieval are quite satisfactory, which allows the proposed system to perform the subsequent high-level analysis of basketball video.

**Table 2-1.** Performance of basketball shot boundary detection.

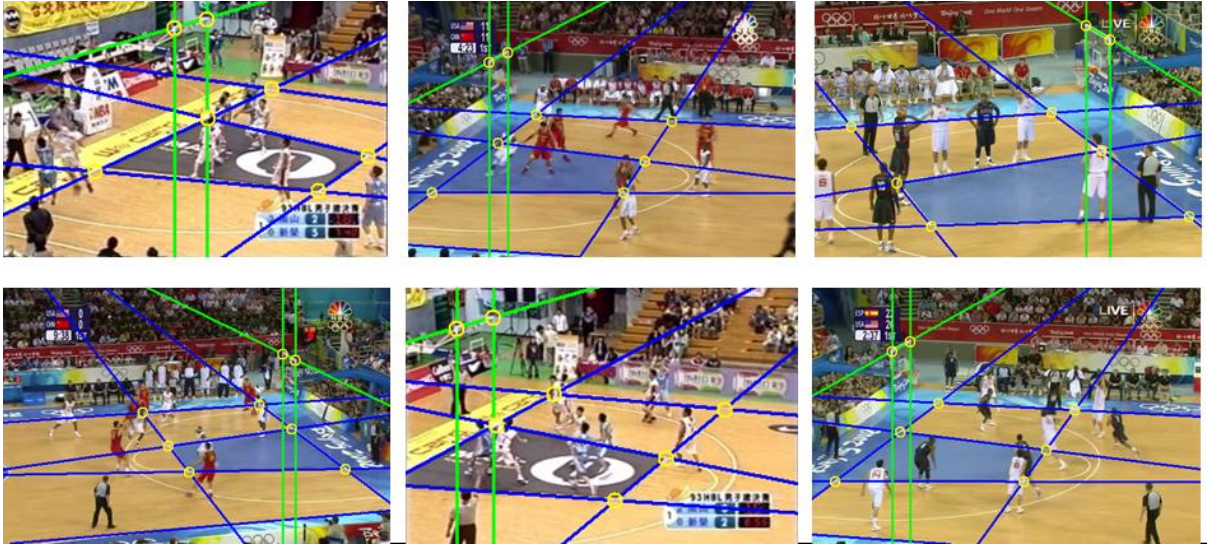
	Olympics1	Olympics2	HBL	Korea	Overall
Correct	159	103	98	66	426
Miss	4	3	6	3	16
False positive	12	10	10	7	39
Recall (%)	97.55	97.17	94.23	95.65	96.38
Precision (%)	92.98	91.15	90.74	90.41	91.61

**Table 2-2.** Performance of basketball court shot retrieval.

	Olympics1	Olympics2	HBL	Korea	Overall
Correct	52	35	32	21	140
Miss	1	0	0	1	2
False positive	3	2	2	1	8
Recall (%)	98.11	100	100	95.45	98.59
Precision (%)	94.55	94.59	94.12	95.45	94.59

### 2.7.3 Results of court line and backboard top-border detection

The proposed systems detect the court lines and the backboard top-border reliably. Fig. 2-18 demonstrates some example results, where the corresponding points are marked with yellow circles. Since the camera motion in a shot is continuous, the coordinates of corresponding points should not change dramatically in successive frames. Hence, though there might be errors in court line detection caused by the occlusion of players in some frames, the incorrect coordinates of the corresponding points can be recovered by interpolation.



**Fig. 2-18.** Example results of detecting court lines and corresponding points (marked with yellow circles).

#### **2.7.4 Performance of basketball tracking and shooting location estimation**

The performance study of ball tracking and shooting location estimation is focused on the *shooting trajectory*. The ground truth boundaries of shooting segments and ground truth ball positions are determined manually. A ball is said to be detected correctly if the system can conclude the correct position of the ball on the trajectory. The experimental results of ball tracking are presented in Table 2-3, where “ball frame” represents the number of frames containing the ball belonging to a shooting trajectory. On average, the recall and precision are up to 92.83% and 95.41%, respectively. On inspection, we find that the false alarms of ball tracking are mainly from the case when there is a ball-like object located on the extension of the ball trajectory. Tracking misses happen when the ball flies over the top boundary of the frame, as the example shown in Fig. 2-19. In this case, an actual shooting trajectory is separated into two potential trajectories and the system retains only the one approaching the backboard as shooting trajectory. The other trajectory will be eliminated, which leads to the misses of the ball candidates on it. Besides, this case (trajectory split) is also one main cause of the mistakes in shooting location estimation.

Table 2-4 reports the performance of shooting location estimation. The shooting locations estimated are judged as correct or not by an experienced basketball player and the proposed system achieves an accuracy of 86.21%. Some demonstrations of shooting location estimation are presented in Fig. 2-20. In each image, the blue circles are the ball positions over frames and the green circle represents the estimated shooting location, which is obtained by projecting the starting position of the trajectory onto the court plane. To present the camera motion, we also mark the positions of corresponding points over frames with red squares.



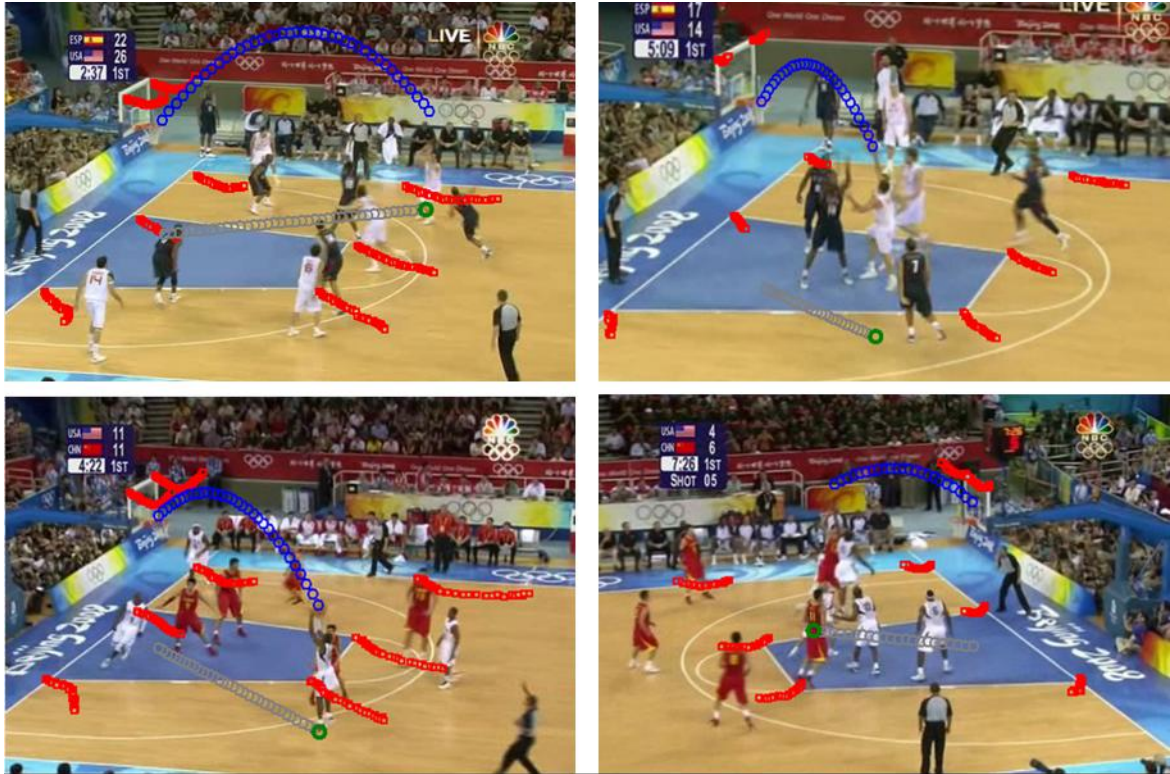
**Fig. 2-19.** Example of a shooting trajectory being separated.

**Table 2-3.** Performance of basketball tracking.

	Olympics1	Olympics2	HBL	Korea	Total
Ball frame	1509	794	643	459	3405
Correct	1421	740	598	402	3161
False alarm	57	29	32	34	152
Recall (%)	94.17	93.2	93	87.58	92.83
Precision (%)	96.14	96.23	94.92	92.2	95.41

**Table 2-4.** Performance of basketball shooting location estimation.

	Olympics1	Olympics2	HBL	Korea	Total
#shoot	48	26	26	16	116
#correct	42	23	22	13	100
Accuracy (%)	87.5	88.46	84.62	81.25	86.21



**Fig. 2-20.** Demonstration of shooting location estimation. In each image, the blue circles are the ball positions over frames, the green circle represents the estimated shooting location and the red squares show the movement of corresponding points due to the camera motion.

In fact, the physical factors we do not involve in 3D trajectory modeling, such as air friction and ball spin rate, may cause the deviation in 3D trajectory reconstruction. In the example of Fig. 2-21(a), the estimated shooting location (the green circle) seems correct. However, we inspect the 3D virtual replay from different viewpoints and observe the directional deviation in the reconstructed 3D trajectory. In the original video clip, the basketball is shot toward the right of the basket, but the ball of the reconstructed 3D trajectory is shot toward the left in the 3D virtual replay. On the other hand, the errors in ball tracking and camera calibration also affect the accuracy of 3D trajectory reconstruction. Fig. 2-22 presents another example. As shown in Fig. 2-22(a), the backboard top border is occluded by the superimposed caption and can not be detected. The incorrect calibration parameters lead to the deviation in shooting location estimation, as shown in Fig. 2-22(b).



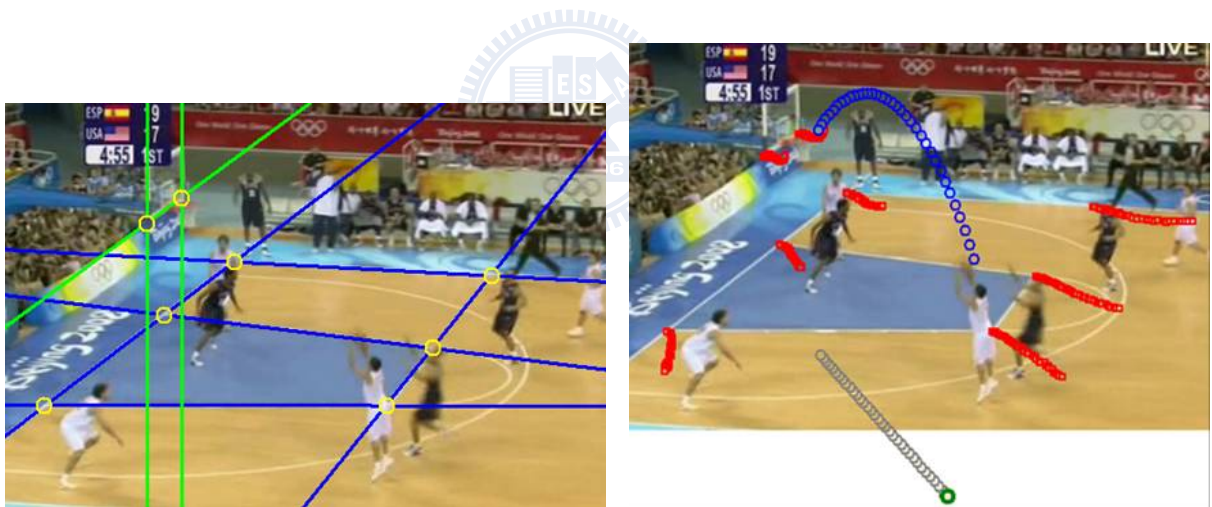
However, the court lines and the backboard top-border are detected appropriately in most frames and overall, we achieve quite encouraging results.



(a) Shooting location estimation

(b) 3D virtual replay

**Fig. 2-21.** Example of directional deviation in 3D trajectory reconstruction.



(a) Corresponding points

(b) Shooting location estimation

**Fig. 2-22.** Error case of shooting location estimation caused by the misdetection of backboard top-border.

### 2.7.5 Comparison and discussion

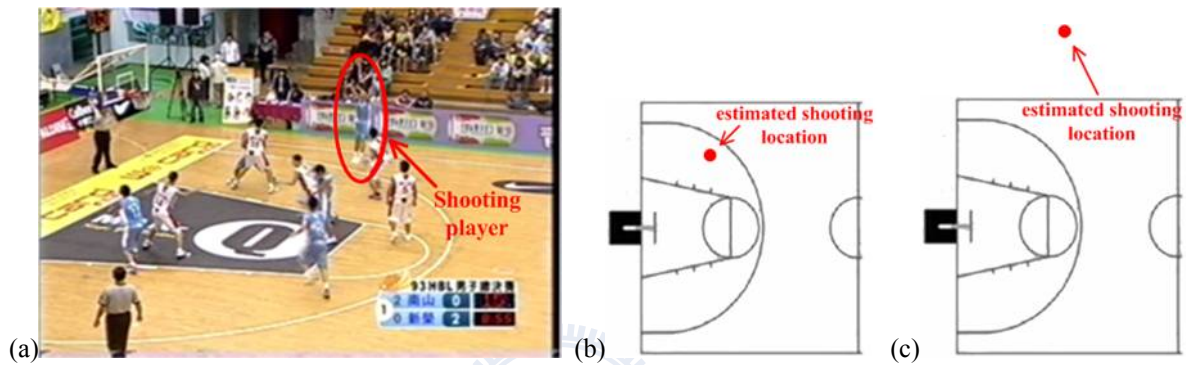
For performance comparison, we implement another ball tracking algorithm based on Kalman filter, which is widely used in moving object tracking [8,11,52]. To compare the effectiveness and efficiency of the Kalman filter-based algorithm (KF) with those of the

proposed physics-based algorithm (PB), we use the precision, recall and the number of potential trajectories (#PT) as criteria. As reported in Table 2-5, KF algorithm has a similar recall with PB algorithm but lower precision, which reveals that PB algorithm performs better in eliminating the false alarms. Besides, PB algorithm produces less potential trajectories because most of the trajectories which do not fit the physical motion characteristics would be discarded. Therefore, fewer potential trajectories need be further processed in PB algorithm, which leads to high efficiency. Overall, the proposed PB algorithm outperforms KF algorithm in both effectiveness and efficiency.

As to shooting location estimation, strictly speaking, there may be some deviation between the actual shooting location and the estimated one, due to the effects of the physical factors we do not involve, such as air friction, ball spin rate and spin axis, etc. However, owing to the consideration of 3D information in camera calibration, the automatic generated statistics of shooting locations provide strong support for the coach and players to comprehend the scoring distribution and even the general offense strategy. Compared to the plane-to-plane (2D-to-2D) mapping in [25], our system has the advantage of the 2D-to-3D inference retaining the vertical information, so the shooting location can be estimated much more precisely. An example for comparing the estimated shooting locations with/without vertical (height) information is presented in Fig. 2-23. Without the vertical information, the estimated shooting locations in Fig. 2-23(c) is far from the actual ones as in Fig. 2-23(a). That is, our system greatly reduces the deviation of shooting location estimation due to the reconstructed 3D information. Overall, the experiments show encouraging results and we believe that the proposed system would highly assist the statistics gathering and strategy inference in basketball games.

**Table 2-5.** Comparison between the proposed physics-based method and the Kalman filter-based method in basketball video. (#PT : number of potential trajectories)

<i>Ball tracking</i>	Proposed PB method			Comparative KF method		
	Recall(%)	Precision(%)	#PT	Recall(%)	Precision(%)	#PT
Olympics1	94.17	96.14	286	92.31	92.12	346
Olympics2	93.20	96.23	153	91.68	93.33	183
HBL	93.00	94.92	164	90.51	91.65	212
Korea	87.58	92.20	94	87.36	90.51	133



**Fig. 2-23.** Comparison of shooting location estimation with/without vertical (height) information: (a) Original shooting location in the frame (b) Estimated shooting location with vertical information. (c) Estimated shooting location without vertical information.

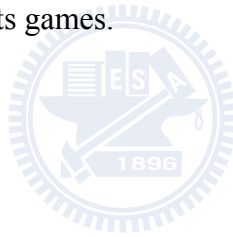
## 2.8 Summary

The more you know the opponents, the better chance of winning you stand. Thus, game study in advance of the play is an essential task for the coach and players. It is a growing trend to assist game study for intelligence collection in sports games with computer technology. To cater for this, we design a physics-based ball tracking system for 3D trajectory reconstruction and shooting location estimation.

Some key ideas and contributions in our system are as follows. The first is to utilize the domain knowledge of court specification for camera calibration. This enables the computation of 3D-to-2D transformation for single-view video sequences. The second is the development of physics-based trajectory extraction mechanism. Exploiting the physical

characteristics of ball motion assists eliminating the non-parabolic (or non-ballistic) trajectories and improves the efficiency and effectiveness of trajectory extraction. Moreover, it allows the 3D information lost in projection to 2D images in camera capturing to be reconstructed. The technical ideas presented in this chapter can also be applied to other sports, such as volleyball, baseball, etc. To the best of our knowledge, the trajectory-based application of shooting location estimation in basketball is first proposed. The experiments show encouraging results on broadcast basketball video.

The fairly good results of ball tracking and 3D trajectory reconstruction in basketball video encourage us to explore appropriate physical motion models for 3D ball trajectory reconstruction in other kinds of sports videos. It is our belief that the preliminary work presented in this chapter will lead to satisfactory solutions for automatic intelligence collection in various kinds of sports games.





## **Chapter 3. Ball Tracking and 3D Trajectory Approximation with Applications to Tactics Analysis in Volleyball Video**

Providing computer-assisted tactics analysis in sports is a growing trend. This chapter presents an automatic system for ball tracking and 3D trajectory approximation from single-camera volleyball sequences as well as demonstrates several applications to tactics analysis. Ball tracking in volleyball video has great complexity due to the high density of players on the court and the complicated overlapping of player-player or ball-player. The 2D-to-3D inference is intrinsically challenging due to the loss of 3D information in projection to 2D frames. To overcome these challenges, we propose a two-phase ball tracking algorithm in which we first detect ball candidates for each frame, and then use them to compute the ball trajectories. With the aid of camera calibration, we involve physical characteristics of ball motion to approximate the 3D ball trajectory from the 2D trajectory. The visualization of 3D trajectory and the applications to trajectory-based tactics analysis not only assist the coach and players in game study but also make game watching a whole new experience. The experiments on international volleyball games show encouraging results.

The rest of the chapter is organized as follows. Section 3.1 gives the introduction. Section 3.2 explains the related work on sports video analysis. Section 3.3 describes the overview of the proposed VIA system. The processes of audio event detection, camera calibration and 2D ball trajectory extraction are explained in section 3.4, 3.5 and 3.6, respectively. Section 3.7 elaborates 3D trajectory approximation. Section 3.8 presents the trajectory-based applications to tactics analysis. Section 3.9 reports and discusses the experimental results. Finally, section 3.10 summaries this chapter.

### 3.1 Introduction

The proliferation of multimedia data makes manual annotation of huge video databases no more practical. This trend facilitates developing automatic systems and tools for content-based multimedia information retrieval. Recently, sports video is attracting considerable attention due to the potential commercial benefits and entertaining functionalities. As the pace of life in the information society accelerates, most viewers desire to retrieve the significant events or designated scenes and players, rather than watch a whole game in a sequential way. Various algorithms of shot classification [1,2,30,31], highlight extraction [6,17,32] and semantic annotation [4,33] in sports video have been developed based on the combination of low-level visual/auditory features and game-specific rules. Furthermore, semantic content analysis of sports video requires ball/player tracking [8,10,34-37] to acquire the ball-player interaction and camera calibration [25,26,32,38,39] to obtain the ball/player positions in the real world coordinates.

Most existing work in sports video analysis is audience-oriented. However, the coach and sports professionals desire to watch a sports game not only with efficiency but also with profundity, variety and professionalism. Traditional interactive video viewing systems which provide quick browsing, indexing and summarization of sports video no longer fulfill their requirements. The professionals prefer better understanding of the tactic patterns and statistical data so that they are able to improve performance and better adapt the operational policy during the game. To achieve this purpose, the current trend is to employ some personnel for game annotation, match recording, tactics analysis and statistics collection. However, it is obviously time-consuming and labor-intensive. Hence, automatic tactics analysis and statistics collection in sports games are undoubtedly compelling.

Although increasing research effort of sports video processing concentrates on ball tracking and trajectory-based tactics analysis, the majority of existing work focuses on tennis

and soccer video [8,34-36]. Little work was done for volleyball video because it is much complex to track the ball in volleyball video due to the high density of players on the court and the frequent ball-player overlaps. However, the ball motion brings significant tactic information in volleyball games and the ball trajectory greatly contributes to tactics analysis. In this chapter, we develop an automatic system called VIA (Volleyball Intelligence Agent), which performs 2D ball tracking and 3D trajectory approximation from single view video sequences, captured by a camera set behind the court, for tactics analysis in volleyball games. Note that the “trajectory” mentioned in this chapter does not include the trajectory of the *attack* action since the attacked ball move too fast and is almost unrecognizable in video frames, even for human eyes.

It might be arguable that the proposed VIA system concentrates on the user-captured volleyball video. However, not all of the games are broadcasted on TV. As the rapid evolution of digital equipments, general users are allowed to capture multimedia data much easier. It is common nowadays for sports professionals to set up a camera to capture the video sequences of the games they are interested in for game strategy study. Visual content analysis is no longer confined to broadcast video. Content analysis in user-generated multimedia data becomes another burgeoning and critical issue [40-42]. This trend necessitates the development of computer-assisted game study system for the user-captured sports video.

## **3.2 Related Work**

### ***3.2.1 Related work on camera calibration***

Semantic analysis of sport video requires camera calibration to convert 2D positions in the video frame to 3D real world coordinates or vice versa. Various camera calibration methods are based on planar reference objects [25,26,38]. These plane-based calibration techniques require feature points on a plane appearing in different views. Farin *et al.* [25,26]

propose an camera calibration algorithm for court sports. They start with identifying the court-line pixels by exploiting the constraints of color and local texture, and then detect the court lines by the Hough transform. The intersection points of the court lines are extracted as the feature points to compute the camera projection matrix via solving a set of linear equations. For the subsequent frames, a *model tracking* mechanism is used to predict the camera parameters from the previous frame. Watanabe *et al.* [38] propose a soccer field tracking method, which extracts the field lines, defines a wire frame model based on the official layout of the soccer field lines, and finally tracks where the field area corresponds in the soccer field by utilizing the camera parameters computed via matching the wire frame model with the extracted field lines.

Yu. *et al.* [36,39] propose a non-plane based camera calibration method of tennis video. They approximate the projection geometry by a perspective projection model mapping between the 3D world and the 2D image. To compute the camera matrix, they find the tops of the net-poles together with the ground feature points to form a non-coplanar feature point set. For camera matrix refinement, they find clip-invariant parameter via finding the cluster centers of all their instances in a clip. Then, they classify all frames into groups according to the focal lengths and the 3D world points which the camera is looking at in the straight-ahead direction. A better estimate of the group-invariant parameters is obtained via a group-wise data analysis. Except for the camera center, the rest of the group-invariant parameters are further refined via a Hough-like search, i.e. frame-wise tuning.

### **3.2.2 Related work on ball/player tracking**

Since significant events are mainly caused by ball-player and player-player interactions, balls and players are the most frequently tracked objects in sports video. Yu *et al.* [34] present a trajectory-based algorithm for ball detection and tracking in soccer video. The ball size is first estimated from feature objects (goalmouth and ellipse) to detect ball candidates.

Potential trajectories are generated from ball candidates by a Kalman filter based verification procedure. The true ball trajectories are finally selected from the potential trajectories according to a confidence index, which indicates the likelihood that a potential trajectory is a ball trajectory. Zhu *et al.* [35] analyze the temporal-spatial interaction among the ball and players to construct a tactic representation called *aggregate trajectory* based on multiple trajectories. The interactive relationship with play region information and hypothesis testing for trajectory temporal-spatial distribution are exploited to analyze the tactic patterns.

Some work focuses on 3D trajectory reconstruction based on multiple cameras located at specific positions [11,13,43]. Hawk-Eye system [43] produces computer-generated replays viewed through 360 degrees. 2D tracking is first performed on each of the specifically located cameras. These 2D trajectories are then sent to a 3D reconstitution module to construct the 3D trajectories, and impact points between separate trajectories (can occur at a bounce or a strike) are determined. Finally, the complete track is visualized. ESPN K-Zone system [11] extracts the trajectory for each pitch and uses computer-generated graphics to outline the strike zone boundaries. Two cameras linked to two PCs are used to observe the ball and each PC extracts a 2D trajectory. The two pitch-tracking computers combine two 2D positions which correspond to the same time code into a 3D position. Then, the successive 3D positions are fed into a Kalman filter to determine the final trajectory. UIS (Umpire Information System) [13] uses multiple cameras to track each pitch and measure the batter's strike zone so as to support the strike/ball judgment.

Although these systems perform well in ball tracking and 3D trajectory reconstruction, they have strong limitation of view angles and require high cost of multiple high speed cameras. Moreover, the high demand for the camera installation locations and the visible areas constrains their systems to be used in a studio-like sports field. These systems are not practicable for general users.

### 3.3 Overview of the Proposed Ball Tracking and 3D Trajectory Approximation in Volleyball Video

To achieve automatic tactics analysis in volleyball games, a system called VIA (Volleyball Intelligence Agent) is designed in this chapter. VIA fuses visual cues with audio features for both syntactic and semantic analysis in volleyball games. Furthermore, VIA visualize the 3D trajectory approximation, so that not only trajectory-based game study can be presented but game watching also becomes an entirely novel experience. The system framework is illustrated in Fig. 3-1.

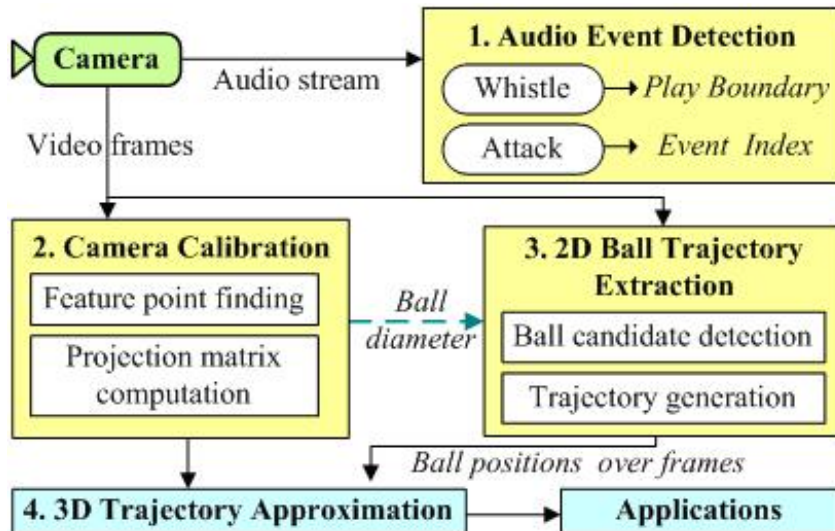


Fig. 3-1. Framework overview of the proposed VIA system.

Whistle is one of the most indicative audio events in volleyball games since whistle determines the start and end of each *play*. Thus, VIA starts with whistle detection to determine play boundaries. Moreover, VIA also detects the *attacks* for event indexing.

For video frames, VIA first performs camera calibration via finding the non-coplanar feature points to compute the projection matrix mapping 3D real world coordinates to 2D image positions. For 2D ball trajectory extraction, ball candidates are detected in each frame by the constraints of size, shape and compactness. However, it is almost impossible to

distinguish the ball within a single frame, so information from successive frames, e.g. motion information, is required for the discrimination between the ball and other objects. VIA correlates information on the ball candidates over a sequence of frames, explores potential trajectories and identifies the true ball trajectories. To approximate 3D trajectories, we set up the motion equations with the parameters: velocities and initial positions, to model the 3D trajectories based on the physical characteristics of ball motion. The 3D ball positions over frames can be represented by equations. The projection matrix computed in camera calibration is then used to map the equation-represented 3D ball positions to the 2D ball coordinates in frames. With the 2D coordinates of the extracted ball candidates being known, we can compute the parameters of the 3D motion equations. Finally, the 3D positions and velocities of the ball can be derived to approximate the 3D ball trajectory.

The novelty and contribution of this chapter are summarized as follows. The problem of 2D-to-3D inference is intrinsically challenging due to the loss of 3D information in projection to 2D frames in picture capturing. We reconstruct 3D information from single view 2D video sequences based on the integration of multimedia features, domain knowledge of court specification and physical characteristics of ball motion. Moreover, several applications using the acquired 2D and 3D trajectories are presented to demonstrate the utility of the proposed 2D ball tracking and 3D trajectory approximation scheme. The trajectory-based applications to tactics analysis greatly assist the coaches and professionals in game strategy study.

### **3.4 Audio Event Detection**

Several significant events which are difficult to detect from visual features, such as whistle, can be directly traced through audio features [44,45]. In this section, we explain the detection of whistles and attacks in volleyball video.

### 3.4.1. Whistle detection

Whistle directly determines the play boundaries in volleyball games. For whistle detection, ZCR (Zero Crossing Rate) is a distinguishing and easy-to-compute feature [46,47]. ZCR counts the number of times which an audio signal crosses its zero axis, as defined in Eq. (3-1) [44]:

$$Z_n = \frac{1}{2N_z} \sum_k |\text{sgn}[s(k)] - \text{sgn}[s(k-1)]| w(n-k) \quad (3-1)$$

where

$$\begin{aligned} s(k) &: \text{discrete time audio signal;} \\ n &: \text{time index of the zero crossing rate;} \\ w(k) &: \text{rectangle window of length } N_z; \\ \text{sgn}[s(k)] &= \begin{cases} 1, & \text{if } s(k) \geq 0 \\ -1, & \text{if } s(k) < 0 \end{cases} \end{aligned} \quad (3-2)$$

The frequency of whistle is higher than that of other signals, so a peak can be found in ZCR when the referee is blowing the whistle. By observation, the duration of a whistle is about 200 to 800 ms, so we set the window length  $N_z$  to the average—500 ms. The feature *peak index* [48] is adopted here for peak picking. The mean of ZCR over a time duration of  $\alpha$  times the window length ( $\alpha \times N_z$ ,  $\alpha = 10$ ) is computed, and then the peak index is defined as the difference between the ZCR of each window and the mean ZCR. Final decision of whistle detection is obtained by thresholding. With the whistles detected, each play can be segmented out. Besides, the *time-outs* (game breaks) can be easily eliminated by visual features since in volleyball games players are not allowed to stay in the court during a time-out.

### 3.4.2 Attack detection

Attack plays an important role in event indexing. The sound of attack is a transient signal in a very short duration. By analyzing the STE (Short-Time Energy) [44], a peak can



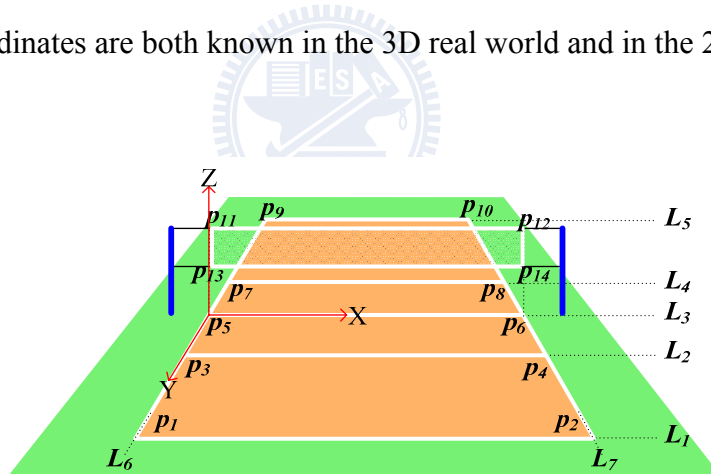
be observed when an attack is coming up. The STE is defined as Eq. (3-3):

$$E_n = \frac{1}{N_E} \sum_k [s(k)w(n-k)]^2 \quad (3-3)$$

where  $n$  is the time index of the short-time energy and  $w(k)$  is a rectangle window of length  $N_E$  (we use  $N_E = 15$  ms, since the sound of attack is a transient signal in a very short duration of about 10~20 ms). As in whistle detection, the feature peak index [48] is adopted for peak picking and final decision of attack detection is obtained by thresholding.

### 3.5 Camera Calibration

Camera calibration aims at computing the geometric transformation, Eq. (2-3) in section 2.4 on pp. 14, mapping from the 3D real-world positions to 2D video frames or vice versa. To compute the camera projection matrix, we need to extract a set of *corresponding points*—the points whose coordinates are both known in the 3D real world and in the 2D image.



**Fig. 3-2.** Illustration of the non-coplanar feature points.

We first segment the court region consisting of the court lines  $L_1$  to  $L_7$  (see Fig. 3-2) using the dominant color feature computed via color histogram. The court lines are detected by the Hough transform. Then, we can obtain the coordinates of the ground feature points  $p_1$  to  $p_{10}$  via computing the intersection of court lines. In addition to the ground feature points, the computation of camera matrix requires non-coplanar feature points. Thus, we trace vertically from the ground points  $p_5, p_6$  in the image and search the two vertical borders of the

net using the Hough transform. The endpoints of the vertical border of the net ( $p_{11}$  to  $p_{14}$ ), together with the ground feature points, form a non-coplanar feature point set. For the computation of camera calibration parameters  $c_{ij}$ , please refer to section 2.4.3 on pp. 18.

### 3.6 2D Volleyball Trajectory Extraction

It is a challenging task to identify the ball in frames due to its small size and fast movement. In this section, we explain the processes of 2D trajectory extraction.

#### 3.6.1 Ball candidate detection

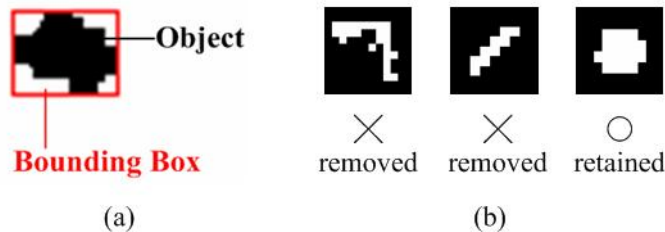
We design the following sieves to prune the non-ball objects. The objects satisfying the constraints of size, shape and compactness are retained as *ball candidates*.

1) **Size sieve:** Please refer to the description of size sieve in section 2.5.1 on pp. 21.

2) **Shape sieve:** Please refer to the description of shape sieve in section 2.5.1 on pp. 21.

3) **Compactness sieve:** The ball object should be compact, so we build a compactness sieve to filter out the objects with the degree of compactness  $D_{compact}$ , as defined in Eq. (3-4), less than a threshold  $T_c$  (we set  $T_c = 50\%$  since an object of  $D_{compact}$  less than half can not be claimed to be “compact”). The illustration of the compactness filter is presented in Fig. 3-3.

$$D_{compact} = Object\_Size / Bounding\_Box\_Area \quad (3-4)$$



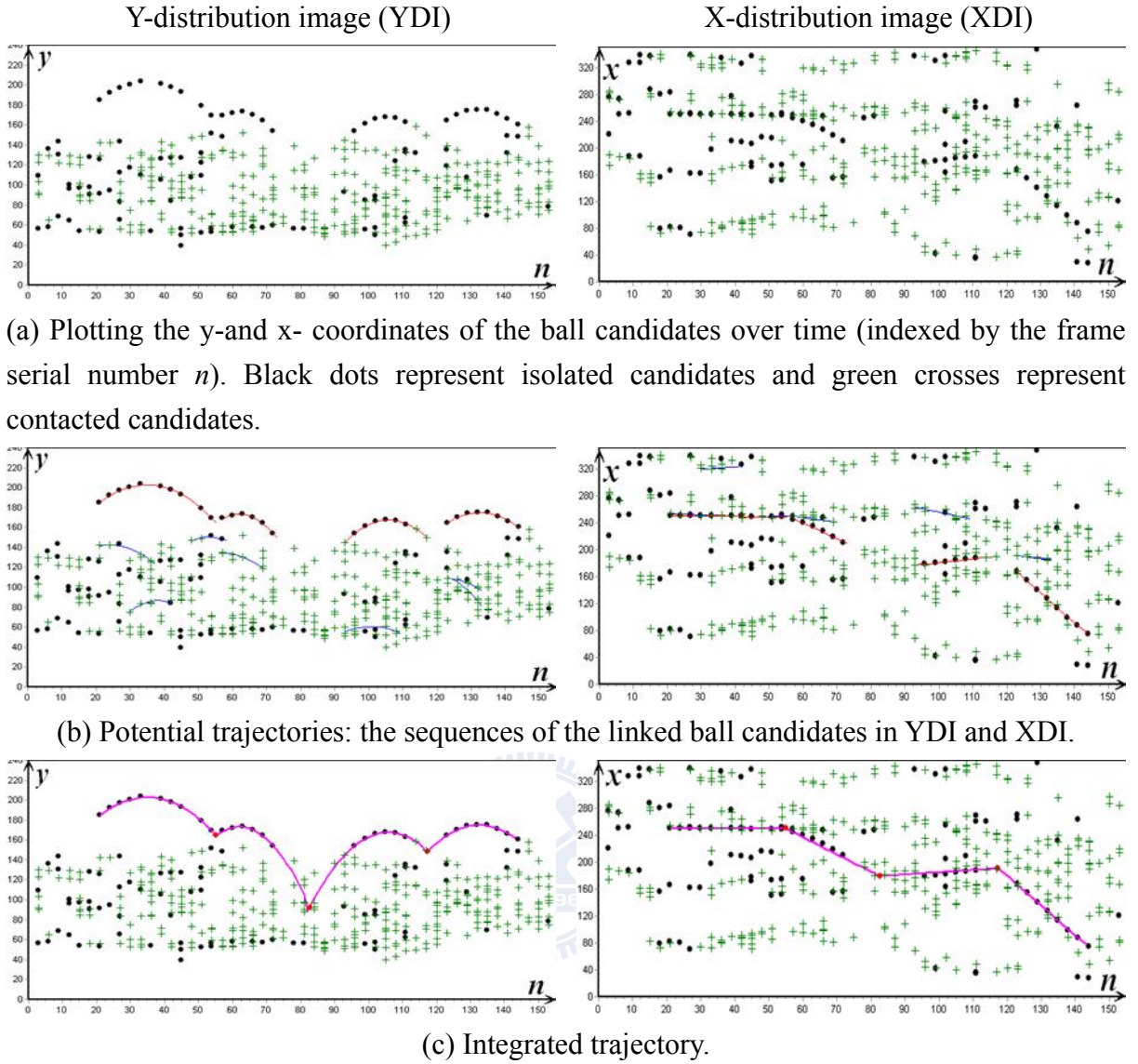
**Fig. 3-3.** Illustration of the compactness filter: (a) Compactness degree  $D_{compact}$  is defined as the ratio of the object size to the area of the bounding box. (b) Objects with low  $D_{compact}$  would be removed while objects with high  $D_{compact}$  would be retained.

The ball is at a distance away from other moving object in most frames. Thus, the ball candidates close to other moving objects might be over-segmented regions of players. To improve the accuracy of ball tracking, ball candidates are classified into *isolated* or *contacted* candidates according to their nearest objects. A candidate is classified as *isolated* if there is no neighboring object within a distance of  $D_{frm}$ , the in-frame ball diameter in Eq. (2-9) on pp. 21, and it is classified as *contacted*, otherwise.

### 3.6.2 *Potential trajectory exploration*

It is very difficult to identify the ball from the ball candidates within a single frame. Therefore, motion information over successive frames is required to discriminate the ball from other moving objects. To visualize the motion of ball candidates, we plot the y-and x-coordinates of the ball candidates over time (indexed by the frame serial number  $n$ ), called *Y-distribution image* (YDI) and *X-distribution image* (XDI), respectively. An example of YDI and XDI is shown in Fig. 3-4(a), where black dots represent isolated candidates and green crosses represent contacted candidates.

In volleyball games, players are not allowed to hold the ball during a play, so the ball trajectory comprises a sequence of near parabolic curves. We have a further observation that the ball moves near parabolically in y-direction due to the gravity and moves near straightly in x-direction in spite of the air friction. Thus, we explore a sequence of points which form a near parabolic curve in YDI and a near straight line in XDI simultaneously as a *potential trajectory*.



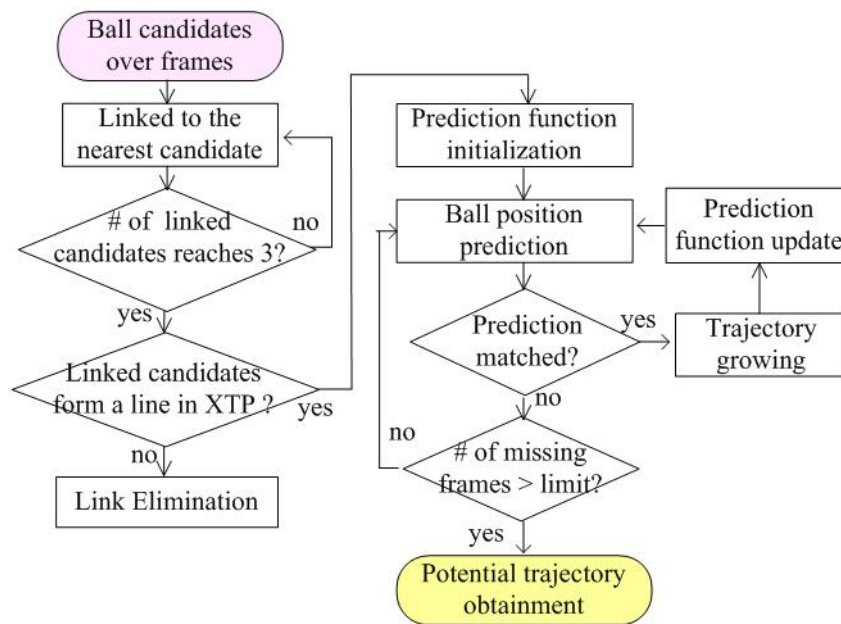
**Fig. 3-4.** Illustration of the Y- and X-distribution images for different process stages of a volleyball game.

The procedure of potential trajectory exploration is depicted in Fig. 3-5. Each ball candidate is first linked to the nearest neighbor in the next frame. Since in frames the ball moves (near) parabolically in y-direction and (near) straightly in x-direction, we define the prediction functions for YDI and XDI as Eq. (3-5) and Eq. (3-6), respectively.

$$y = a_1 \cdot n^2 + a_2 \cdot n + a_3, \quad a_1 < 0, \quad n : \text{frame serial number} \quad (3-5)$$

$$x = b_1 \cdot n + b_2 \quad (3-6)$$

The ball position in the next frame is predicted by the prediction functions. The prediction is considered *matched* if a ball candidate close to the predicted position is found. The trajectory then grows by adding the candidate found and the prediction functions are updated by re-computing the best-fitting functions for the coordinates of the candidates detected so far using the *least squares fitting* technique of regression analysis. If there exists no candidate close to the predicted position, the frame is regarded as a *missing frame* and the predicted position is taken as the ball position. The trajectory growing terminates when the number of consecutive missing frames reaches a predefined limit (4 in our experiments). The potential trajectories produced from this procedure are shown as the sequences of the linked ball candidates in YDI and XDI, as depicted in Fig. 3-4(b).



**Fig. 3-5.** Procedure of potential trajectory exploration.

### 3.6.3. Trajectory identification and integration

Given the set of potential trajectories, the next step is to identify the true ball trajectories. For each potential trajectory  $T$ , we define its confidence degree, denoted by  $F(T)$ , which measures how likely  $T$  is a ball trajectory:

$$F(T) = \lambda_1 L + \lambda_2 R - \lambda_3 E \quad (3-7)$$

where  $L$  is the trajectory length,  $R$  is the ratio of isolated candidates to contacted candidates in  $T$ ,  $E$  is the prediction error (defined as the average of the distances from ball candidate positions to the predicted positions) and the parameters  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  are used to adjust the relative importance. We design an algorithm which iteratively selects the potential trajectory with the highest confidence degree as the ball trajectory  $T_b$ , and then discards the trajectories which overlap with  $T_b$ , until all trajectories are processed, as shown in the following pseudo code. Finally, the gaps between two successive identified trajectories can be patched by extending these two trajectories based on their respective prediction functions, as shown in Fig. 3-4(c). Thus, the ball positions can be estimated even though the ball is temporarily occluded.

**Algorithm of trajectory identification**

**Input:**  $S$ , the set of potential trajectories;

**Output:**  $I$ , the set of identified ball trajectories;

Initialize  $I$  to be empty;

**while** ( $S$  is not empty) **do**{

    Let  $T_b$  be the trajectory with highest confidence degree in  $S$ ;

    Move  $T_b$  into  $I$ ;

    Eliminate the trajectories which overlap with  $T_b$  in  $S$ ;

}

### 3.7 3D Volleyball Trajectory Approximation

In volleyball games, the ball trajectory comprises a sequence of near parabolic curves, even though many factors affect the ball motion, such as velocity, gravity acceleration, spin axis, spin rate, air friction, etc. We call each near parabolic curve in the ball trajectory a *sub-trajectory* and roughly model a 3D sub-trajectory as:

$$\begin{aligned}
x &= x_0 + V_x t \\
y &= y_0 + V_y t \\
z &= z_0 + V_z t + gt^2/2
\end{aligned} \tag{3-8}$$

where  $(x, y, z)$  is the 3D ball coordinate at time  $t$ ,  $(x_0, y_0, z_0)$  is the 3D ball coordinate of the starting position in the sub-trajectory,  $(V_x, V_y, V_z)$  is the 3D ball velocity and  $g$  is the gravity acceleration.

The relationship between each pair of corresponding points in the 2D and 3D spaces is given in Eq. (2-3) on pp. 14. Substituting Eq. (3-8) into Eq. (2-3), we obtain:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 + V_x t \\ y_0 + V_y t \\ z_0 + V_z t + \frac{1}{2} g t^2 \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \triangleq \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad \text{where} \tag{3-9}$$

$$\begin{aligned}
u' &= \frac{u}{w} \\
v' &= \frac{v}{w}
\end{aligned}$$

Multiplying out the equation with  $u = u'w$  and  $v = v'w$ , we get two equations for each ball candidate  $(u', v')$ :

$$\begin{aligned}
&c_{11} x_0 + c_{11} V_x t + c_{12} y_0 + c_{12} V_y t + c_{13} z_0 + c_{13} V_z t + c_{13} g t^2/2 + c_{14} \\
&= u' (c_{31} x_0 + c_{31} V_x t + c_{32} y_0 + c_{32} V_y t + c_{33} z_0 + c_{33} V_z t + c_{33} g t^2/2 + 1)
\end{aligned} \tag{3-10}$$

$$\begin{aligned}
&c_{21} x_0 + c_{21} V_x t + c_{22} y_0 + c_{22} V_y t + c_{23} z_0 + c_{23} V_z t + c_{23} g t^2/2 + c_{24} \\
&= v' (c_{31} x_0 + c_{31} V_x t + c_{32} y_0 + c_{32} V_y t + c_{33} z_0 + c_{33} V_z t + c_{33} g t^2/2 + 1)
\end{aligned} \tag{3-11}$$

Since the eleven camera calibration parameters  $c_{ij}$  and the time of each ball candidate on the sub-trajectory are known, we set up a linear system  $\mathbf{D}_{2N \times 6} \mathbf{E}_{6 \times 1} = \mathbf{F}_{2N \times 1}$  from Eq. (3-10) and Eq. (3-11) to compute the six unknowns  $(x_0, V_x, y_0, V_y, z_0, V_z)$  of the parabolic trajectory:

$$\begin{bmatrix} c_{11} - u'_1 c_{31} & c_{11} t_1 - u'_1 c_{31} t_1 & c_{12} - u'_1 c_{32} & c_{12} t_1 - u'_1 c_{32} t_1 & c_{13} - u'_1 c_{33} & c_{13} t_1 - u'_1 c_{33} t_1 \\ c_{21} - v'_1 c_{31} & c_{21} t_1 - v'_1 c_{31} t_1 & c_{22} - v'_1 c_{32} & c_{22} t_1 - v'_1 c_{32} t_1 & c_{23} - v'_1 c_{33} & c_{23} t_1 - v'_1 c_{33} t_1 \\ c_{11} - u'_2 c_{31} & c_{11} t_2 - u'_2 c_{31} t_2 & c_{12} - u'_2 c_{32} & c_{12} t_2 - u'_2 c_{32} t_2 & c_{13} - u'_2 c_{33} & c_{13} t_2 - u'_2 c_{33} t_2 \\ c_{21} - v'_2 c_{31} & c_{21} t_2 - v'_2 c_{31} t_2 & c_{22} - v'_2 c_{32} & c_{22} t_2 - v'_2 c_{32} t_2 & c_{23} - v'_2 c_{33} & c_{23} t_2 - v'_2 c_{33} t_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{11} - u'_N c_{31} & c_{11} t_N - u'_N c_{31} t_N & c_{12} - u'_N c_{32} & c_{12} t_N - u'_N c_{32} t_N & c_{13} - u'_N c_{33} & c_{13} t_N - u'_N c_{33} t_N \\ c_{21} - v'_N c_{31} & c_{21} t_N - v'_N c_{31} t_N & c_{22} - v'_N c_{32} & c_{22} t_N - v'_N c_{32} t_N & c_{23} - v'_N c_{33} & c_{23} t_N - v'_N c_{33} t_N \end{bmatrix} \begin{bmatrix} x_0 \\ V_x \\ y_0 \\ V_y \\ z_0 \\ V_z \end{bmatrix} = \begin{bmatrix} u'_1 \\ v'_1 \\ u'_2 \\ v'_2 \\ \vdots \\ u'_N \\ v'_N \end{bmatrix} \tag{3-12}$$

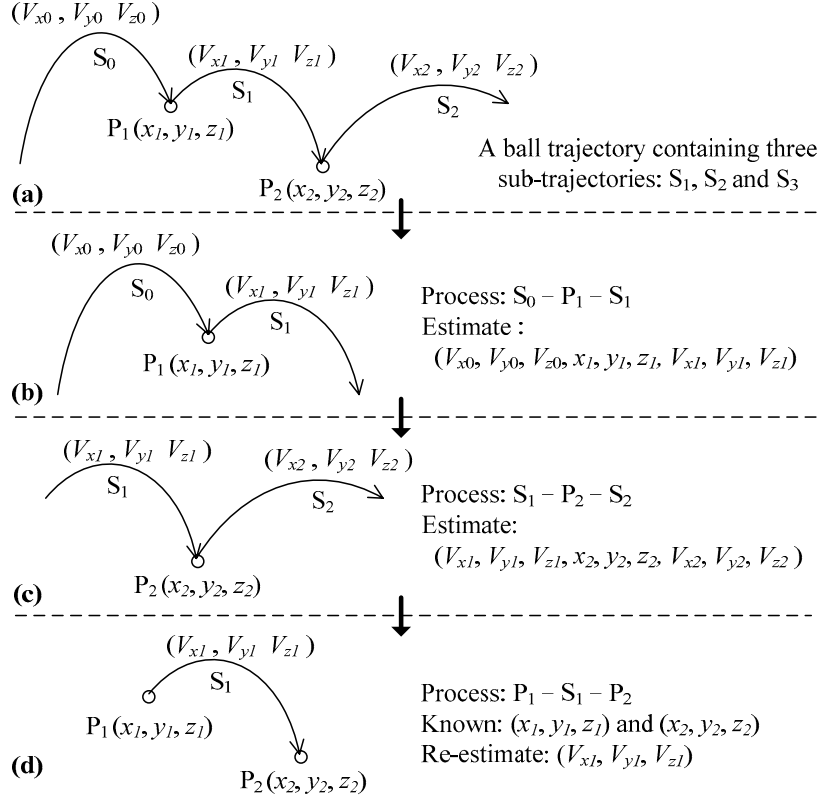
$\mathbf{D}$   $\mathbf{E}$   $6 \times 1$

$$\begin{aligned}
& \begin{bmatrix} \bar{u}_1' (c_{33} g t_1^2/2 + 1) - (c_{13} g t_1^2/2 + c_{14}) \\ v_1' (c_{33} g t_1^2/2 + 1) - (c_{23} g t_1^2/2 + c_{24}) \\ u_2' (c_{33} g t_2^2/2 + 1) - (c_{13} g t_2^2/2 + c_{14}) \\ v_2' (c_{33} g t_2^2/2 + 1) - (c_{23} g t_2^2/2 + c_{24}) \\ \vdots \\ u_N' (c_{33} g t_N^2/2 + 1) - (c_{13} g t_N^2/2 + c_{14}) \\ v_N' (c_{33} g t_N^2/2 + 1) - (c_{23} g t_N^2/2 + c_{24}) \end{bmatrix} \\
= & \begin{bmatrix} \mathbf{F} \\ \mathbf{E} \end{bmatrix}_{2N \times 1} \tag{3-12}
\end{aligned}$$

where  $N$  is the number of ball candidates on the sub-trajectory and  $(u_i', v_i')$  is the 2D coordinate of the candidates. Then, similar to Eq. (2-8) on pp. 19, we can over-determine  $\mathbf{D}$  and find a least squares fitting for  $\mathbf{E}$  by pseudo-inverse. Finally, each 3D sub-trajectory can be reconstructed from the six physical parameters  $(x_0, V_x, y_0, V_y, z_0, V_z)$ .

However, here comes a problem. Since each 3D sub-trajectory is reconstructed independently, the 3D coordinate of the transition point between two adjacent sub-trajectories computed from the preceding sub-trajectory is not always consistent with the one computed from the succeeding sub-trajectory. To overcome this problem, we enhance the algorithm by taking two adjacent sub-trajectories into consideration at one time. Fig. 3-6 illustrates the procedure of 3D trajectory reconstruction by a sample ball trajectory.





**Fig. 3-6.** Procedure of 3D trajectory approximation.

As shown in Fig. 3-6(a), the ball trajectory contains three sub-trajectories  $S_0$ ,  $S_1$  and  $S_2$ . Let  $P_1$  be the transition point of  $S_0$  and  $S_1$ , on which the 3D ball velocities are  $(V_{x0}, V_{y0}, V_{z0})$  and  $(V_{x1}, V_{y1}, V_{z1})$ , respectively. As shown in Fig. 3-6(b), to derive  $(V_{x0}, V_{y0}, V_{z0}, x_1, y_1, z_1, V_{x1}, V_{y1}, V_{z1})$ , we consider the two adjacent sub-trajectories  $S_0$  and  $S_1$ . Taking  $P_1$  as the initial point, the 3D trajectories of  $S_0$  and  $S_1$  are modeled as Eq. (3-13) and Eq. (3-14), respectively:

$$\begin{aligned}
 X &= x_1 - V_{x0} t \\
 Y &= y_1 - V_{y0} t \\
 Z &= z_1 - V_{z0} t - g t^2 / 2
 \end{aligned} \tag{3-13}$$

$$\begin{aligned}
 X &= x_1 + V_{x1} t \\
 Y &= y_1 + V_{y1} t \\
 Z &= z_1 + V_{z1} t + g t^2 / 2
 \end{aligned} \tag{3-14}$$

Similar to Eq. (3-10) and Eq. (3-11), we can derive two equations, Eq. (3-15) and Eq. (3-16), for each ball candidate on  $S_0$ , and two equations, Eq. (3-17) and Eq. (3-18), for each ball candidate on  $S_1$ .  $(u', v')$  is the 2D coordinate of the ball candidate.

$$\begin{aligned}
& c_{11} x_1 - c_{11} V_{x0} t + c_{12} y_1 - c_{12} V_{y0} t + c_{13} z_1 - c_{13} V_{z0} t - c_{13} g t^2 / 2 + c_{14} \\
& = u' (c_{31} x_1 - c_{31} V_{x0} t + c_{32} y_1 - c_{32} V_{y0} t + c_{33} z_1 - c_{33} V_{z0} t - c_{33} g t^2 / 2 + 1) \quad (3-15)
\end{aligned}$$

$$\begin{aligned}
& c_{21} x_1 - c_{21} V_{x0} t + c_{22} y_1 - c_{22} V_{y0} t + c_{23} z_1 - c_{23} V_{z0} t - c_{23} g t^2 / 2 + c_{24} \\
& = v' (c_{31} x_1 - c_{31} V_{x0} t + c_{32} y_1 - c_{32} V_{y0} t + c_{33} z_1 - c_{33} V_{z0} t - c_{33} g t^2 / 2 + 1) \quad (3-16)
\end{aligned}$$

$$\begin{aligned}
& c_{11} x_1 + c_{11} V_{x1} t + c_{12} y_1 + c_{12} V_{y1} t + c_{13} z_1 + c_{13} V_{z1} t + c_{13} g t^2 / 2 + c_{14} \\
& = u' (c_{31} x_1 + c_{31} V_{x1} t + c_{32} y_1 + c_{32} V_{y1} t + c_{33} z_1 + c_{33} V_{z1} t + c_{33} g t^2 / 2 + 1) \quad (3-17)
\end{aligned}$$

$$\begin{aligned}
& c_{21} x_1 + c_{21} V_{x1} t + c_{22} y_1 + c_{22} V_{y1} t + c_{23} z_1 + c_{23} V_{z1} t + c_{23} g t^2 / 2 + c_{24} \\
& = v' (c_{31} x_1 + c_{31} V_{x1} t + c_{32} y_1 + c_{32} V_{y1} t + c_{33} z_1 + c_{33} V_{z1} t + c_{33} g t^2 / 2 + 1) \quad (3-18)
\end{aligned}$$

The equations produced by ball candidates can be represented as a linear system:  $\mathbf{G}_{2N \times 9} \mathbf{H}_{9 \times 1} = \mathbf{L}_{2N \times 1}$ , where  $\mathbf{H} = (V_{x0}, V_{y0}, V_{z0}, x_1, y_1, z_1, V_{x1}, V_{y1}, V_{z1})^T$  and  $N$  is the number of ball candidates on the  $S_0$  and  $S_1$  [please refer to Eq. (3-12)]. With  $\mathbf{G}$  over-determined, we can find a least squares fitting for  $\mathbf{H}$  by pseudo-inverse. Thus, the coordinate of  $P_1 (x_1, y_1, z_1)$  is obtained. In the same way, the nine parameters  $(V_{x1}, V_{y1}, V_{z1}, x_2, y_2, z_2, V_{x2}, V_{y2}, V_{z2})$  can be derived by processing  $S_1$  and  $S_2$  simultaneously, as shown in Fig. 3-6(c).

For the sub-trajectory ( $S_1$  in this example) between two transition points, its 3D velocity  $(V_{x1}, V_{y1}, V_{z1})$  is computed twice: one when processing  $S_0$ - $P_1$ - $S_1$  and the other when processing  $S_1$ - $P_2$ - $S_2$ . For consistence, we take  $(x_0, y_0, z_0)$  and  $(x_1, y_1, z_1)$  as known parameters and estimate  $(V_{x1}, V_{y1}, V_{z1})$  again, as shown in Fig. 3-6(d). Thus, each ball candidate on  $S_1$  would produce four equations: Eq. (3-17), Eq. (3-18), Eq. (3-19) and Eq. (3-20).

$$\begin{aligned}
& c_{11} x_2 - c_{11} V_{x1} t + c_{12} y_2 - c_{12} V_{y1} t + c_{13} z_2 - c_{13} V_{z1} t - c_{13} g t^2 / 2 + c_{14} \\
& = u' (c_{31} x_2 - c_{31} V_{x1} t + c_{32} y_2 - c_{32} V_{y1} t + c_{33} z_2 - c_{33} V_{z1} t - c_{33} g t^2 / 2 + 1) \quad (3-19)
\end{aligned}$$

$$\begin{aligned}
& c_{21} x_2 - c_{21} V_{x1} t + c_{22} y_2 - c_{22} V_{y1} t + c_{23} z_2 - c_{23} V_{z1} t - c_{23} g t^2 / 2 + c_{24} \\
& = v' (c_{31} x_2 - c_{31} V_{x1} t + c_{32} y_2 - c_{32} V_{y1} t + c_{33} z_2 - c_{33} V_{z1} t - c_{33} g t^2 / 2 + 1) \quad (3-20)
\end{aligned}$$

Once again, the equations produced by ball candidates can be represented as a linear system:  $\mathbf{M}_{4N \times 3} \mathbf{Q}_{3 \times 1} = \mathbf{R}_{4N \times 1}$ , where  $\mathbf{Q} = (V_{x1}, V_{y1}, V_{z1})^T$  and  $N$  is the number of ball candidates on  $S_1$ .  $\mathbf{Q}$  can be estimated by pseudo-inverse. Finally, the complete 3D trajectory can be

reconstructed by the 3D ball velocity  $(V_{xi}, V_{yi}, V_{zi})$  on each sub-trajectory  $S_i$  and the coordinate  $(x_i, y_i, z_i)$  of each transition point  $P_j$ .

### 3.8 Trajectory-Based Applications in Volleyball Games

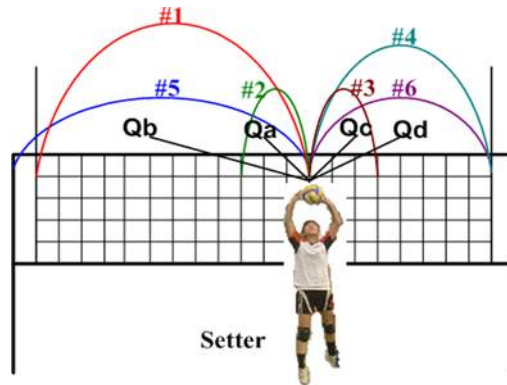
This section presents several application based on the acquired 3D and 3D trajectories to demonstrate the utility of the proposed 2D ball tracking and 3D trajectory approximation scheme. the trajectory-based applications of tactics analysis greatly assist the coaches and players in game strategy study.

#### 3.8.1 Action detection and set type recognition using 2D trajectory

In volleyball games, a play begins with a *serve* followed by the iterative actions: *reception*, *set* and *attack*. By game rules, players are not allowed to hold the ball during a play. Thus, the ball changes its motion only when interacting with a player. The turning points of the ball trajectory can be detected and recognized as *serve*, *reception*, *set* and *attack* in order.

The *set* action can be further analyzed for *set type* recognition, which is crucial for tactics inference because *attack* is the most effective way to gain points and the set type dominates an attack. Fig. 3-7 illustrates ten common set types. A set type is determined according to its direction (forward or backward), the horizontal and vertical displacements of the ball. We define the discriminants as Table 3-1. Set  $Qa$ ,  $Qb$ ,  $Qc$  and  $Qd$  are *quick sets* which players try to hit the ball as soon as possible. Set #2 and #3 are *short sets* next to the setter while set #1, #4, #5 and #6 are *long sets* toward the two sides of the net. A set type can be recognized by classifying the *set curve* (the sub-trajectory after the *set* action) into one of the ten types by the discriminants, where  $a_2$  and  $b_1$  are coefficients in Eq. (3-5) and Eq. (3-6), and  $T1\sim T3$  are thresholds. We use 200 set curves (20 curves per set type) as training data and manually label the set types. The thresholds  $T1\sim T3$  are determined by seeking for the values

which best classify the set curves in the training data.



**Fig. 3-7.** Illustration of set type diagram.

**Table 3-1.** Discriminants of ten common set types.

Set	Discriminant	Set	Discriminant
#1(long)	$b_1 < 0,  b_1  > T1,  b_1 /a_2 \leq T2$	Qa (quick)	$b_1 < 0,  b_1  < T1, T2 <  b_1 /a_2 \leq T3$
#2(short)	$b_1 < 0,  b_1  < T1,  b_1 /a_2 \leq T2$	Qb (quick)	$b_1 < 0,  b_1 /a_2 > T3$
#3(short)	$b_1 > 0,  b_1  < T1,  b_1 /a_2 \leq T2$	Qc (quick)	$b_1 > 0,  b_1  < T1, T2 <  b_1 /a_2 \leq T3$
#4(long)	$b_1 > 0,  b_1  > T1,  b_1 /a_2 \leq T2$	Qd (quick)	$b_1 > 0,  b_1 /a_2 > T3$
#5(long)	$b_1 < 0,  b_1  > T1, T2 <  b_1 /a_2 \leq T3$		
#6 (long)	$b_1 > 0,  b_1  > T1, T2 <  b_1 /a_2 \leq T3$		

### 3.8.2 3D virtual replays and serve placement estimation using 3D trajectory

3D trajectory approximation facilitates the enriched visual presentation of 3D virtual replays. The ball movement can be watched on a virtual court from any viewpoint. This visualization is exciting and practical that the viewpoints can be switched among the receiver, setter, attacker or the players opposite the net, which cannot be captured from any camera on the court.

*Serve placement* (landing position) offers a valuable insight into the game strategy because the serve-reception directs the first attack in a play. With the 3D trajectory approximated, we extend the sub-trajectory of a serve and the serve placement can be estimated when the sub-trajectory reaches the ground (the z-coordinate of the ball equals

zero).

### **3.9 Experimental Results and Discussion in Volleyball Video**

The framework elaborated in the previous sections supports audio event detection, 2D ball tracking and 3D trajectory approximation. For performance evaluation, the proposed system has been tested on the volleyball video sequences (MPEG-1, 352x240, 29.97 fps, Audio: 44.1 kHz, 16 bits, stereo) captured in the Asia Men's Volleyball Challenge Cup: 1) Taiwan vs. Korea, 2) China vs Japan, and 3) Japan vs. Korea.

#### **3.9.1 Results of audio event detection**

To segment a game into plays, VIA detects the whistles using ZCR (Zero Crossing Rate). The frequency of whistle is higher than that of other signals, so a peak of ZCR can be found when the referee is blowing the whistle. As to the attack detection, STE (Short-Time Energy) is used. A peak of STE occurs when an attack is taking place. To obtain the threshold values for peak picking in whistle detection and attack detection, a training set containing the sounds of whistle and attack in different games is used and an iterative procedure of modifying and testing the threshold values is conducted. We achieve a fairly good performance of overall 96.5% precision rate and 98.57% recall rate in whistle detection, as reported in Table 3-2, so that most plays can be segmented appropriately.

For attack detection, the sound energy highly depends on the power of the player hitting the ball. Not all of the attacks are very powerful, so some attacks with less power would be missed. The precision and recall of attack detection using STE are 84.62% and 88%, respectively, as presented in Table 3-3. The accuracy of attack detection can be improved with the assistance of visual information, such as the transition of ball motion (to be discussed in section 3.9.2).

**Table 3-2.** Performance of whistle detection.

	#correct	#false positive	#miss	Precision	Recall
TWN-KOR	93	4	1	95.88%	98.94%
CHN-JPN	89	3	1	96.74%	98.89%
JPN-KOR	94	3	2	96.91%	97.92%
<b>Total</b>	<b>276</b>	<b>10</b>	<b>4</b>	<b>96.5%</b>	<b>98.57%</b>

**Table 3-3.** Performance of attack detection.

	#correct	#false positive	#miss	Precision	Recall
TWN-KOR	37	6	5	86.05%	88.1%
CHN-JPN	33	5	6	86.84%	84.62%
JPN-KOR	40	9	4	81.63%	90.91%
<b>Total</b>	<b>110</b>	<b>20</b>	<b>15</b>	<b>84.62%</b>	<b>88%</b>

### 3.9.2 Results of 2D volleyball trajectory extraction

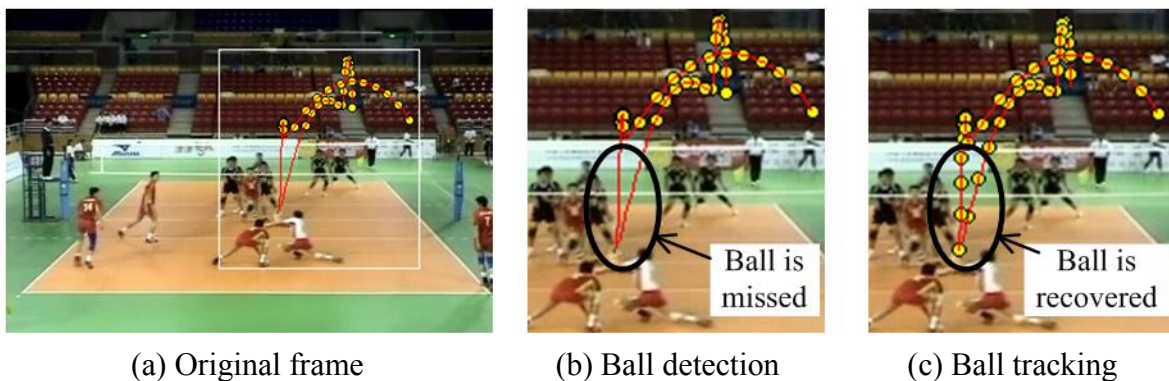
The experiment of 2D trajectory extraction is conducted on the shots which are correctly segmented by whistle detection. The following conventions and notations are used in presenting the results. For each *ball frame* (the frame contains the ball), the ground truth of ball position is obtained by manual inspection. The system is said to *correctly identify a frame  $f$*  if: 1) it concludes the correct ball position when  $f$  is a ball frame or 2) it concludes that there is no ball when  $f$  is a no-ball frame. The system is said to give a *false alarm* if it concludes the incorrect ball position in a ball frame or it detects a ball in a no-ball frame. Let  $\#frm$ ,  $\#ball-frm$  be the number of frames and ball frames in the sequence, respectively. Let  $\#correct$  denote the number of frames which the system correctly identifies the ball and  $\#false$  denote the number of false alarms.

The results of ball detection and tracking are presented in Table 3-4. A ball is said to be *detected* correctly if it matches a ball candidate. A ball is said to be *tracked* if the system can conclude the correct position of the ball on the derived trajectory. An example is given in Fig. 3-8. Fig. 3-8(a) shows the original frame. In Fig. 3-8(b), the ball is missed when the ball is

occluded by or close to the player(s). However, the system can still compute the ball trajectory and track the ball positions, as shown in Fig. 3-8(c). We achieve an accuracy of 71.84% on average in ball detection. By inspecting the error cases, we observe that the ball might be missed before serving in some plays, because the player who is serving does not toss up the ball high enough. Consequently, the ball which is too close to or occluded by the player is hard to detect. On the other hand, the tracking might fail if too many ball candidates are missed and not enough ball candidates are detected. However, the proposed physics-based ball tracking method is able to correct most errors and promotes the final accuracy up to 87.1% on average. Besides, the rate of false alarm ( $\#false/\#frm$ ) is very low—an average of 2.68%, which takes a very small portion in the trajectory. Hence, the high reliability of the extracted trajectories significantly promotes the feasibility of the subsequent trajectory-based applications for tactics analysis and 3D trajectory approximation.

**Table 3-4.** Performance of volleyball detection and tracking.

Sequence	Ground truth		Detection result			Tracking result		
	#frm	#ball-frm	#correct	#false	accuracy	#correct	#false	accuracy
TWN-KOR	15824	11508	11626	430	73.47%	13620	408	86.07%
CHN-JPN	14835	10520	10728	418	72.32%	12885	410	86.86%
JPN-KOR	19241	13147	13492	521	70.12%	16959	519	88.14%
<b>Total</b>	<b>49900</b>	<b>35175</b>	<b>35846</b>	<b>1369</b>	<b>71.84%</b>	<b>43464</b>	<b>1337</b>	<b>87.1%</b>



**Fig. 3-8.** Illustration of volleyball detection and ball tracking.

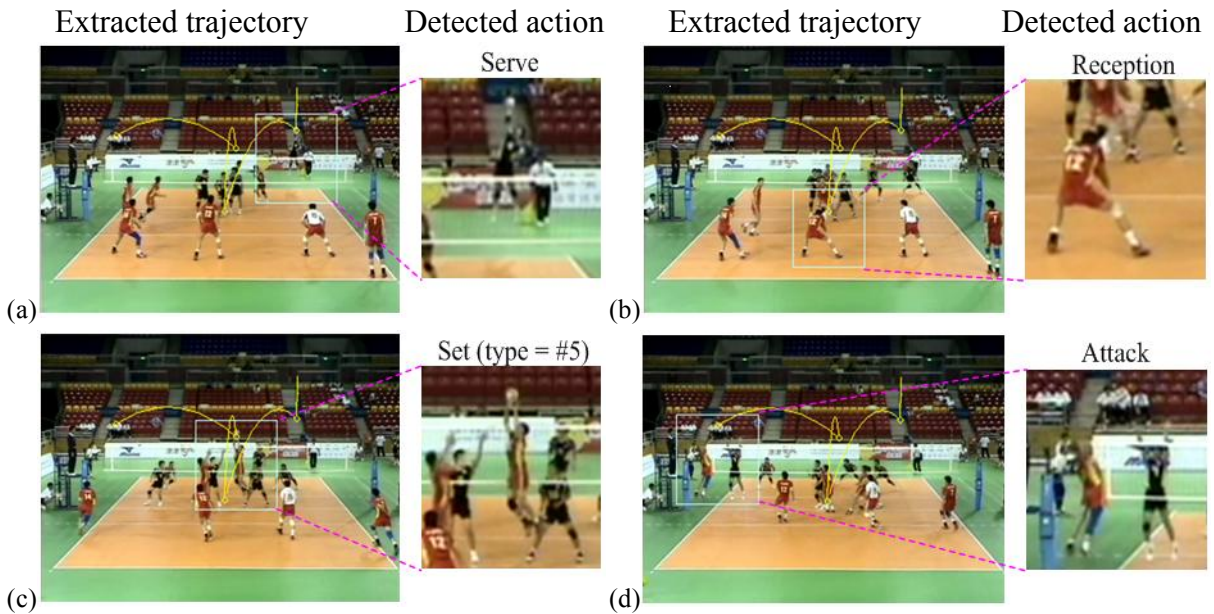


Table 3-5 presents the performance of action detection. Most actions, except *serve*, are detected well and the accuracies are about 90%. The misses of *serve* detection are mainly caused by the failures in ball tracking before a *serve* (as mentioned in the previous paragraph). The 6th column “attack(+audio)” reports the result of attack detection using both the trajectory and audio information. A peak in STE (Short-Time Energy) after the *set* action is recognized as an attack action. Combination of the trajectory and audio information improves the accuracy of attack detection in two ways: 1) the peaks in STE before the *set* action should be false alarms and can be eliminated, and 2) some misses in trajectory-based attack detection due to the tracking error can be recovered by STE.

**Table 3-5.** Performance of action detection.

<b>Action</b>	Serve	Reception	Set	Attack	Attack(+audio)
#action	133	133	130	125	125
#correct	110	119	120	112	115
Accuracy	82.71%	89.47%	92.31%	89.6%	92%

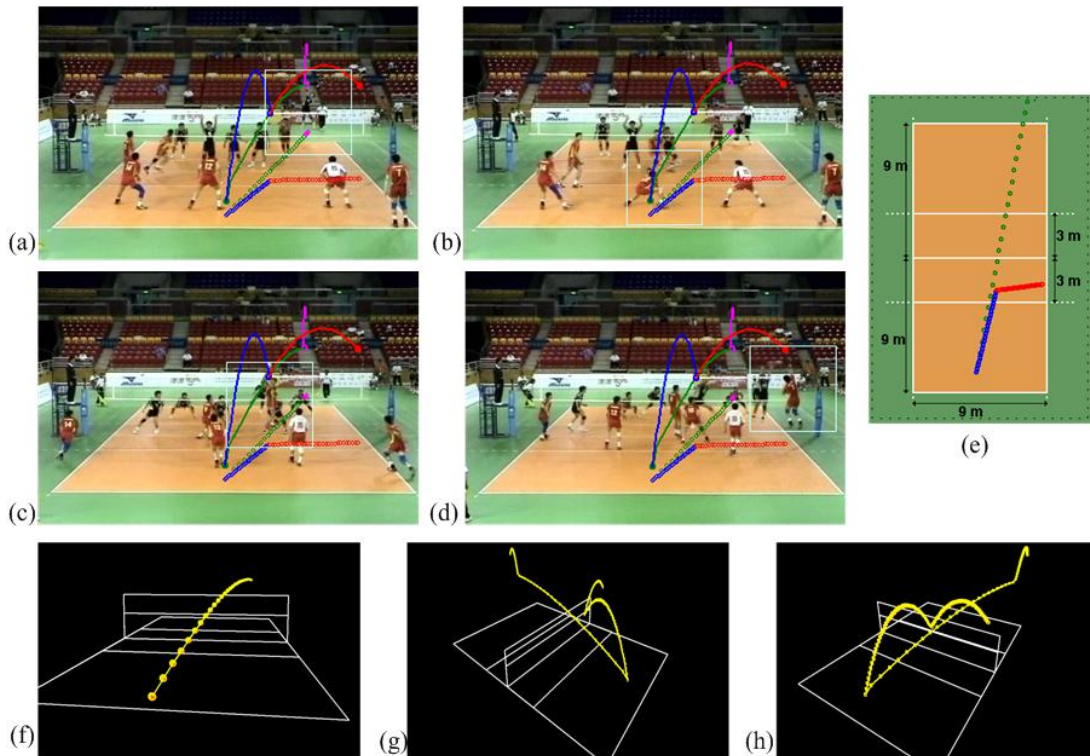
Fig. 3-9 demonstrates examples of 2D ball trajectory extraction and action detection in volleyball video. The detected action: *serve*, *reception*, *set* and *attack* are shown in Fig. 3-9(a) ~ (d), respectively. *Set* is one of the actions and the *set* type is further recognized. In each of Fig. 3-9(a) ~ (d), the left image displays the frame at the moment when the action is detected, with the trajectory superimposed on the frame. The right image shows the automatic generated close-up for the detected action.



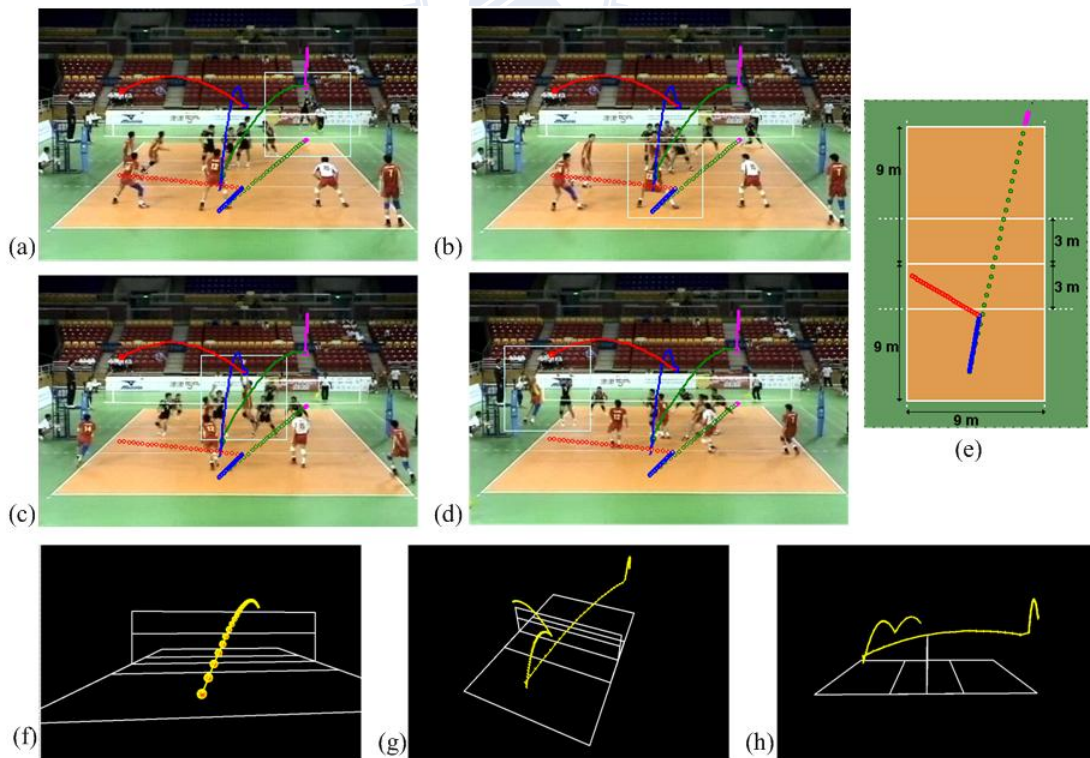
**Fig. 3-9.** Demonstration of ball trajectory extraction and action detection in volleyball video: (a) Serve, (b) Reception, (c) Set and (d) Attack.

### 3.9.3 Simulation results of 3D volleyball trajectory approximation

The estimation of 3D ball positions highly relies on the 2D ball positions extracted. Owing to the high accuracy of the proposed 2D ball tracking scheme, VIA is able to approximate the 3D trajectory well. Sample simulation results are demonstrated in Fig. 3-10 and Fig. 3-11. Take Fig. 3-10 for explanation. Fig. 3-10(a) displays the frame at the moment when a serve is occurring. The frame is enriched by superimposing the extracted ball trajectory on the frame and projecting the 3D trajectory on the court plane. Similarly, the enriched frames for reception, set and attack are shown in Fig. 3-10(b)~(d), respectively. It can be observed that the transition positions of the 3D trajectory are almost the locations of the actions occurring, which verifies the feasibility of the proposed 3D trajectory approximation method. The trajectory projected on the court model, as shown in Fig. 3-10(e), enables the audience or professionals to comprehend the transition of ball motion much easily. Fig. 3-10(f) displays the serve placement estimation. Furthermore, virtual replays can be provided and the ball trajectory in each play can be viewed from any viewpoint, as presented in Fig. 3-10(g)~(h).



**Fig. 3-10.** Demonstration of 3D trajectory approximation. (a)~(d) The enriched frames for serve, reception, set and attack, respectively. (e) Ball trajectory projected on the court model. (f) Serve placement estimation. (g)~(h) 3D virtual replays from different viewpoints.

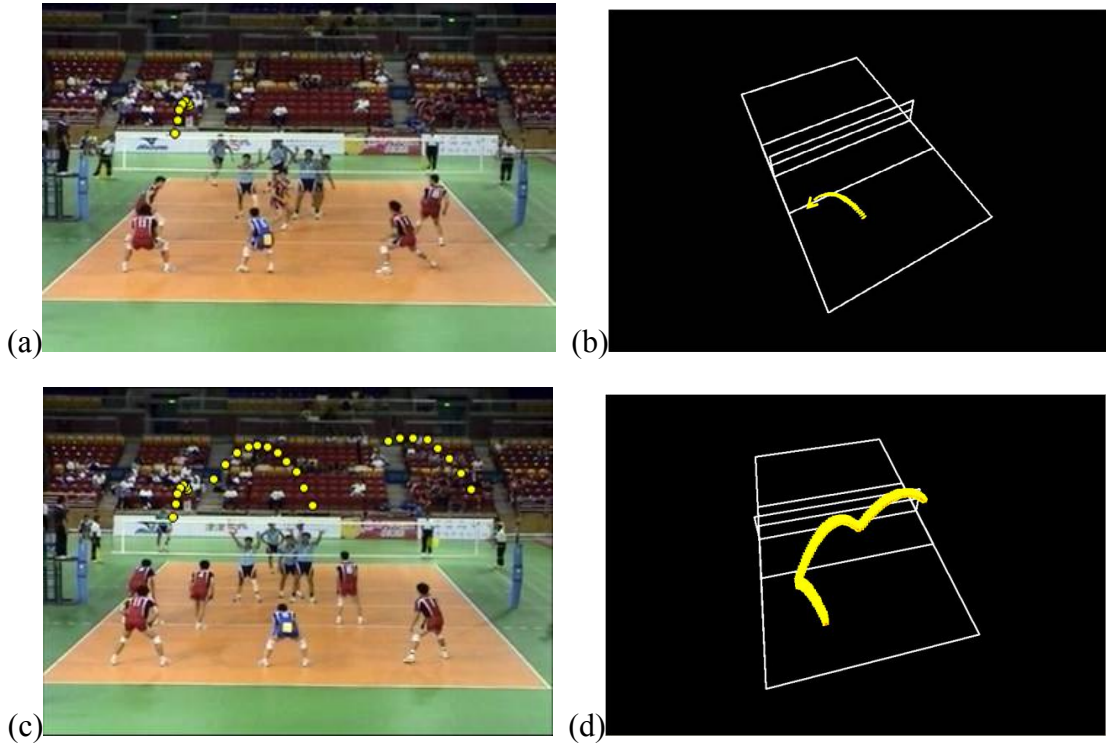


**Fig. 3-11.** Demonstration of 3D trajectory approximation. (a)~(d) The enriched frames for serve, reception, set and attack, respectively. (e) Ball trajectory projected on the court model. (f) Serve placement estimation. (g)~(h) 3D virtual replays from different viewpoints.

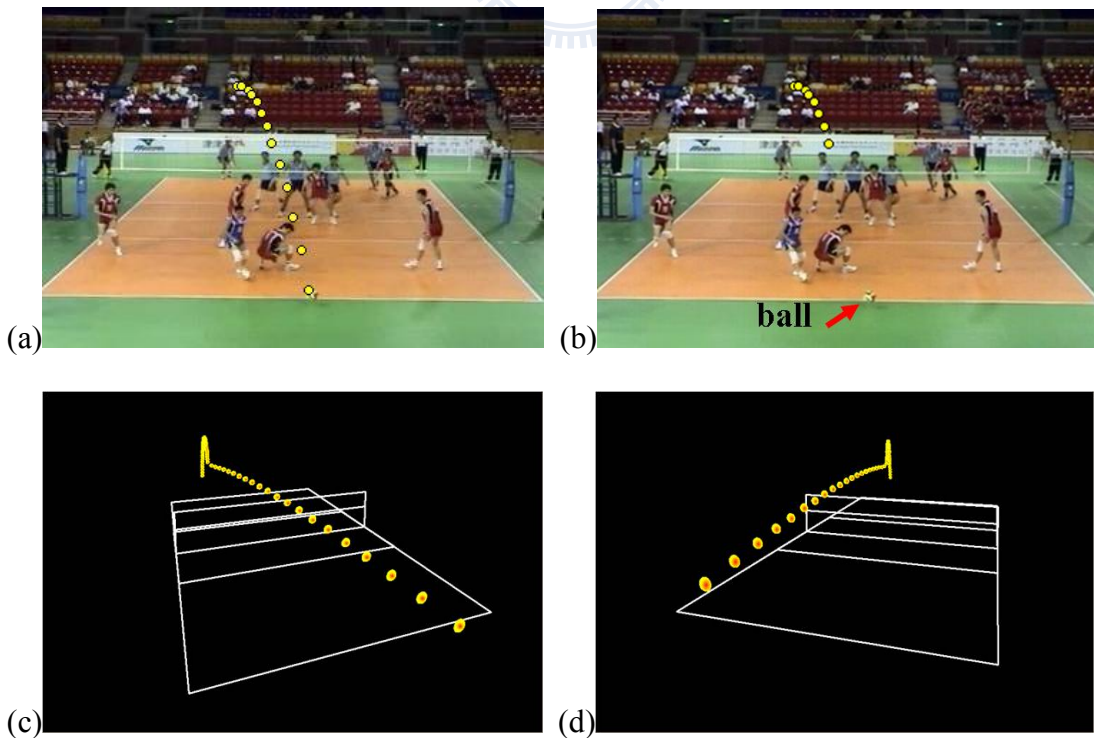
Inspecting the error cases, we find that improper segmentation of the ball might lead to the deviation of the 2D ball candidate coordinate. If there are not enough ball candidates detected to rectify the deviation, the system might misjudge a far-to-near trajectory as a near-to-far one, and vice versa. An example is shown in Fig. 3-12. Only five ball candidates detected for the serve, as shown in Fig. 3-12(a), and the far-to-near serve trajectory is miscomputed as a near-to-far trajectory, as shown in Fig. 3-12(b). All ball candidates detected in the video sequence and the complete 3D trajectory approximated are presented in Fig. 3-12(c) and (d), respectively. Fig. 3-13 gives another example, in which the ball served directly falls to the ground without being touched by any player. The ground truth 2D ball positions and the ball candidates detected are shown in Fig. 3-13(a) and (b), respectively. In the sequence, the ball drops suddenly due to the effects of air friction and ball spin, and the serve placement is near the end-line, as shown in Fig. 3-13(b). However, we do not consider those physical factors. Moreover, we miss the later ball candidates in the serve. Thus, the 3D trajectory approximated is not accurate and the serve placement is over the end-line, as shown in Fig. 3-13(c) and (d).

Strictly speaking, there may be some deviation between the actual ball trajectory and the approximated 3D trajectory, due to the effects of the physical factors we do not involve, such as air friction, ball spin rate and spin axis, etc. However, our experimental results show that the proposed physics-based method is able to approximate the 3D ball trajectory pretty well for tactics analysis.





**Fig. 3-12.** Error case in 3D trajectory approximation. (a) Detected ball candidates of serve. (b) The 3D approximated trajectory of serve. (c) Detect ball candidates in the video sequence. (d) The 3D trajectory approximated from the ball candidates in (c).



**Fig. 3-13.** Error case in 3D trajectory approximation. (a) Ground truth 2D ball positions. (b) Ball candidates detected. (c) and (d) 3D virtual replays from different viewpoints.

### 3.9.4 Comparison with Kalman filter-based algorithm

For performance comparison, we implement another ball tracking algorithm based on Kalman filter, which is widely used in moving object tracking [8,11,34]. To compare the effectiveness and efficiency of the Kalman filter-based algorithm (KF) with those of the proposed physics-based algorithm (PB), we use #correct (the number of frames which the system correctly identifies the ball), #false (the number of false alarms), accuracy and #PT (the number of potential trajectories) as criteria, as reported in Table 3-6. It can be observed that PB algorithm performs better in eliminating the false alarms. Consequently, PB algorithm has a higher accuracy of about 87% compared to about 80% for KF algorithm. Moreover, PB algorithm produces less potential trajectories because most of the trajectories which do not fit the motion characteristics would be discarded. Therefore, PB algorithm achieves higher efficiency since fewer potential trajectories need further processing. The results show that the proposed PB algorithm outperforms KF algorithm in both effectiveness and efficiency.

**Table 3-6.** Comparison between the proposed physics-based method and the Kalman filter-based method in volleyball video. (#false: number of false positive, #PT: number of potential trajectories)

Sequence	Proposed PB method				Comparative KF method			
	#correct	#false	accuracy	#PT	#correct	#false	accuracy	#PT
TWN-KOR	13620	408	86.07%	1116	12890	775	81.46%	1774
CHN-JPN	12885	410	86.86%	945	11818	701	79.66%	1550
JPN-KOR	16959	519	88.14%	1233	15427	1019	80.18%	1981
<b>Total</b>	<b>43464</b>	<b>1337</b>	<b>87.1%</b>	<b>3294</b>	<b>40135</b>	<b>2495</b>	<b>80.43%</b>	<b>5305</b>

### 3.10 Summary

The more you know the opponents, the better chance you stand of winning. Therefore, game strategy study before the play is of vital importance for the coach and players. To assist game strategy study and extract tactic information, we design a physics-based system VIA (Volleyball Intelligence Agent) for ball tracking, 3D trajectory approximation and providing applications to tactics analysis based on the 2D and 3D trajectories. The problem of 2D-to-3D inference is intrinsically challenging due to the loss of 3D information in projection to 2D frames in picture capturing. One significant contribution is the integrated scheme which utilizes the domain knowledge of court specification for camera calibration and encapsulates physical characteristics of ball motion into object tracking to achieve 3D trajectory approximation from single view video sequences. Moreover, the VIA system has illustrated some of the numerous trajectory-based applications made possible by this scheme, including: action detection, set type recognition, 3D virtual replays and serve placement estimation. These applications significantly assist the coach, players and the audience to have a novel insight into the game.

It might be arguable that the proposed VIA system concentrates on the user-captured volleyball video. However, not all of the games are broadcasted on TV. It is a growing trend that the coach and players set up a camera to capture the game they want to analyze. This trend necessitates the development of computer-assisted game study system like the proposed VIA system. Currently, we are also trying to adapt the proposed scheme to broadcast volleyball video. On the other hand, we are exploring appropriate physical motion models for 3D ball trajectory approximation in other sports. It is our belief that the preliminary work presented in this chapter will lead to satisfactory solutions for automatic tactics analysis in various kinds of sports.



## Chapter 4. Sports Information Retrieval in Baseball Video

The pitcher/batter confrontation and the defense process play the key roles in the resultant victory or defeat in a baseball game. In this chapter we present a 3-phased scheme of semantic content analysis, video annotation, information retrieval and enriched visual presentation in baseball video. In the first phase, we present a trajectory-based framework for automatic ball tracking and pitching evaluation based on the physical characteristic of ball motion. The task of ball detection and tracking in broadcast baseball video is very challenging because many objects may look like a ball, the ball size is small, and the ball may deform due to its high speed movement. To overcome these challenges, we first define a set of filters to prune most non-ball objects but retain the ball, even if it is deformed. In ball position prediction and trajectory extraction, we analyze the 2D distribution of ball candidates and exploit the characteristic that the ball trajectory presents in a near parabolic curve in video frames. Most of the non-qualified trajectories are pruned, which greatly improves the computational efficiency. The missed ball positions can also be recovered in the trajectory by applying the position prediction.


In the second phase, we design an effective yet compute-easy algorithm for strike zone determination. The strike zone is a conceptual rectangular area through which a pitch passing would be judged as a *strike*. The strike zone can offer a reference for positioning the *pitch location*, the relative location of the ball in/around the strike zone when the ball passes by the batter, which plays an important role in determining the batting result. In the third phase, we propose a framework to automatically summarize the defense process and ball motion after the ball is batted into the field. Utilizing the strictly-defined specifications of the baseball field, we recognize the spatial patterns in each frame and identify the region of the baseball field being currently focused. Finally, an annotation string which abstracts the ball routing

patterns is generated.

With ball trajectory extraction, strike zone determination and play region classification, the system is able to make informative descriptions about the game, generate enriched visual presentation and provide exploration, so that users can have a further insight into the game with both semantic and tactical understanding. The experiments on various broadcast baseball sequences captured from different TV channels show promising results.

The rest of this chapter is organized as follows. Section 4.1 gives the introduction. Section 4.2 presents trajectory-based ball tracking framework. Section 4.3 explains the process of automatic strike zone determination. Section 4.4 shows baseball exploration using spatial pattern recognition. Experimental results are given in Section 4.5, and finally Section 4.6 summarizes this chapter.

#### **4.1 Introduction**



Sports video has been bringing considerable commercial benefits and entertaining functionalities. Hence, more and more research efforts are invested into sports video analysis. Sports games with specific rules and broadcasting characteristics draw various research issues in video analysis. Applications and technologies from many aspects are developed, including shot classification [1, 2, 50], highlight extraction [4, 5, 6, 51] and object tracking [8, 12, 13, 15, 52, 53]. Popular sports such as soccer [4, 7, 8, 17, 34, 38, 46], tennis [12, 14, 15, 36, 39, 48], basketball [31, 37], volleyball [9] and baseball [3, 5, 6, 50, 51, 52, 56, 69, 71, 72, 73] are widely studied.

More keenly than ever, the audience desire professional insights into the games. The coach and the players demand automatic tactics analysis and performance evaluation with the aid of video analysis technologies. The sports fans and professionals are no longer satisfied with traditional interactive video viewing systems for quick browsing, indexing and summarization of sports video. In this chapter we present a 3-phased scheme of semantic

content analysis, video annotation, information retrieval and enriched visual presentation in baseball video.

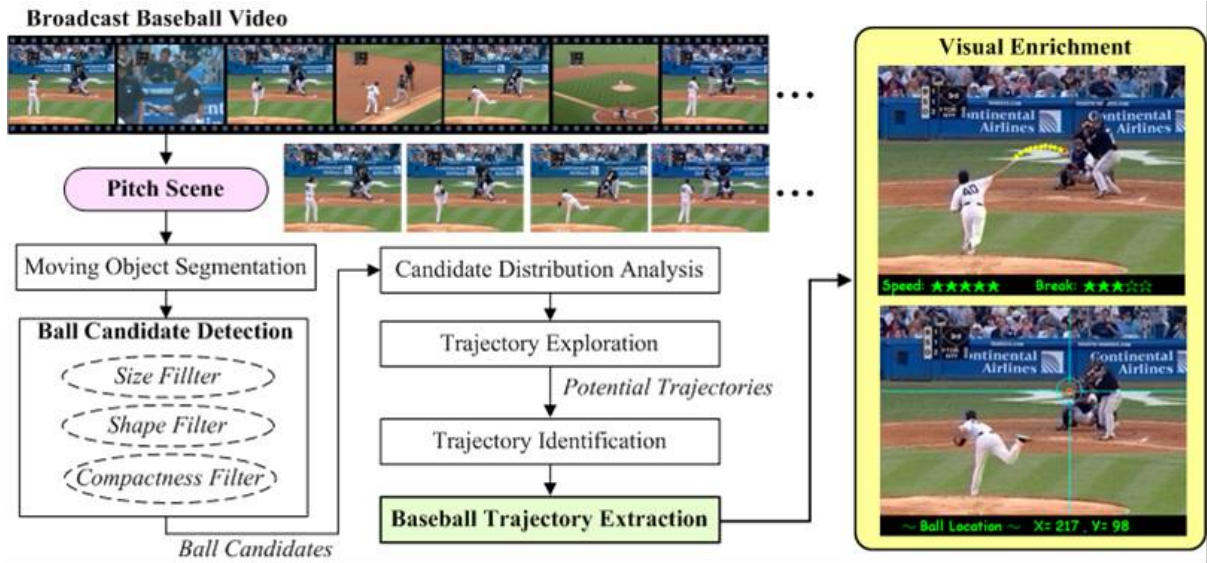
The pitches usually dominate the game situation on the baseball field. Ball speed and trajectory curvature are two main factors in determining how difficult of the pitched ball being hit. Thus, we propose a physics-based ball tracking method to extract the ball trajectory in pitch scenes. Ball speed and trajectory curvature can be computed for evaluating the pitch. Due to the capturing viewpoint and the frame rate constraint, the ball speed and trajectory curvature might not be very precise. The proposed pitch analysis is not for grading, but for entertainment effects, enriched visual presentation and sports information retrieval.

In addition to the ball speed and trajectory curvature, the pitch location (the relative location of the ball in/around the strike zone when the ball passes by the batter) also has influence on the moving direction of the ball batted out. For example, a batter who swings at a lower pitch has a good chance of hitting a ground ball, while a batter who swings at a higher pitch has a great chance of hitting the ball in the air. Since the strike zone provides reference for determining the pitch location, we propose a contour-based method to shape the strike zone according to the batter's stance. Strike/ball judgment can also be visualized on the video frames by the shaped strike zone. Besides the confrontation of the pitch vs. the batter, the ball motion and the defense process after the ball is batted into the field is another focus of attention. With the field specifications, we design algorithms to recognize the spatial patterns (field lines and field objects) in frames. Then, the active regions of event occurrence in the field are classified by the spatial patterns. We can infer the ball routing patterns and defense process from the transitions of the active regions captured in the video. From ball tracking, strike zone shaping to play region classification for ball routing inference, we have fairly extensive analysis on baseball video. Content understanding and annotation are achieved to provide rich information about the games.

## 4.2 Trajectory-Based Baseball Tracking Framework

Ball tracking in baseball video is a challenging task since the high speed of the ball may cause ball deformation in video frames and the small size of the ball leads to tracking losses. Based on the observation, the baseball trajectory presents in a near parabolic curve in pitch scenes. We analyze the vertical and horizontal motion of the ball. Ideally, in the vertical direction, the ball moves parabolically due to the gravity, while in the horizontal direction, the ball moves in a straight line in spite of the air friction. In fact, the ball motion is not exactly a parabolic curve vertically and a straight line horizontally in video frames, but the characteristic of the near- parabolic/straight motion is sufficient for ball position prediction and trajectory extraction. The missed balls can also be recovered over the trajectory by applying the position prediction.

In this section, we develop a 2D trajectory-based ball tracking framework for broadcast baseball video, as depicted in Fig. 4-1. First, the moving objects of each frame are segmented in the pitch shots. Each frame then generates *ball candidates* including the ball and some ball-like objects which satisfy the constraints of size, shape and compactness. Because of ball deformation caused by its speed, it is quite difficult to identify whether a single object is a ball. Hence, we utilize the physical characteristic of ball motion that the ball moves parabolically due to the gravity and identify whether a *potential trajectory* is the true ball trajectory. The X- and Y-distributions of ball candidates in a sequence of frames are analyzed to explore the trajectory which fulfills the physical characteristic. Finally, the baseball trajectory is extracted and the ball position in each frame can be located. In addition, visual enrichment and pitching evaluation can be presented based on the extracted ball trajectory.



**Fig. 4-1.** Block diagram of the proposed baseball tracking framework.

Now we describe in turn the components of the proposed framework: moving object segmentation, ball candidate detection, candidate distribution analysis, trajectory exploration, trajectory identification and finally, baseball trajectory extraction. Shot classification and indexing in sports videos has been researched well in the literature [1,2,50,54,55]. We adopt the method in [55] and extract pitch shots using dominant color matching, region segmentation and dominant color layout analysis.

#### 4.2.1 Moving object segmentation

Based on observation, there is usually no camera motion in pitch scenes, so frame difference method is applied to moving object segmentation. A *Frame Difference Image* (FDI) is a binary image formed by comparing every two successive frames (the intensity information is used). A pixel value of FDI is set to 255 if a significant difference occurs at the pixel location, and otherwise, the pixel value of FDI is set to 0, as defined in Eq. (4-1), where  $n$  is the frame sequence number and  $T_d$  is a threshold.

$$FDI_n(x, y) = \begin{cases} 255, & \text{if } |Intensity_n(x, y) - Intensity_{n-1}(x, y)| > T_d \\ 0, & \text{otherwise} \end{cases} \quad (4-1)$$

Fig. 4-2 presents an example of segmenting the moving objects where the ball is included. Fig. 4-2(a) gives the original frame and Fig. 4-2(b) shows the FDI. It can be observed that the ball is included in a white region larger than the original ball size. This is because FDI takes the absolute value of intensity difference between frames. Since the baseball in the video is white and bright, the intensity of the ball in a frame should be higher. That is, the baseball is included in the positive regions of intensity difference between frames. Thus, the *Positive Frame Difference Image* (PFDI), defined as Eq. (4-2), is used to effectively segment positive regions of intensity difference which contain the ball, as shown in Fig. 4-2(c). Morphological operations are then performed to remove noises and make the regions filled. Regions formed by region growing and ball candidates will be detected among these regions.

$$PFDI_n(x, y) = \begin{cases} 255, & \text{if } Intensity_n(x, y) - Intensity_{n-1}(x, y) > T_d \\ 0, & \text{otherwise} \end{cases} \quad (4-2)$$



(a) Original frame

(b) FDI

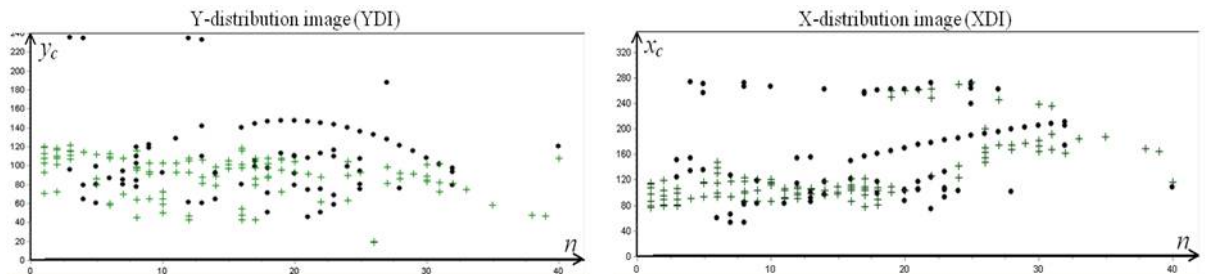
(c) PFDI

**Fig. 4-2.** Illustration of segmenting the moving objects where the ball is included.

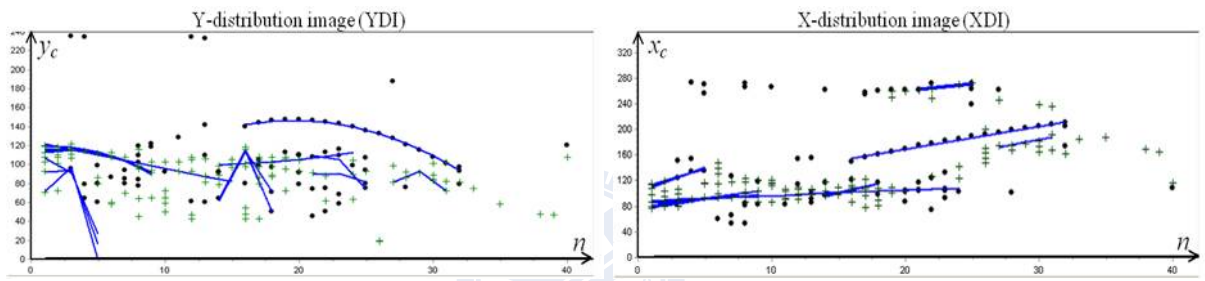
#### 4.2.2 *Ball candidate detection*

Many non-ball objects might look like a ball in video frames and it is difficult to recognize which is the true one. On the other hand, the ball might be presented in a shape different from a circle because of deformation. We use the size, shape and compactness sieves to extract the ball candidates from the moving objects segmented (please refer to

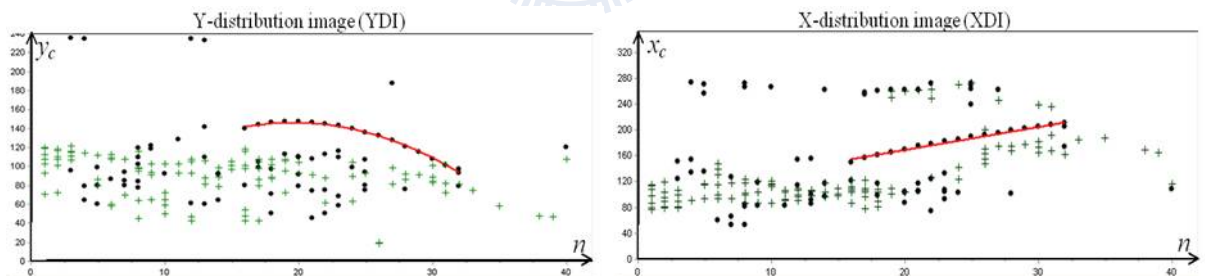
Section 3.6.1 on pp. 48). After sieving, the remaining objects which satisfy the constraints are considered as the *ball candidates*.



(a) Ball candidate distribution analysis. Black dots represent isolated candidates and green crosses represent contacted ones.



(b) Trajectory exploration. Potential trajectories are shown as the linking of ball candidates.



(c) Trajectory identification. The ball trajectory identified is shown as the parabolic curve in YDI and the straight line in XDI.

**Fig. 4-3.** Illustration of the Y- and X-distribution images for different processing stages of trajectory extraction, where  $n$  is the frame serial number,  $y_c$  in YDI and  $x_c$  in XDI are the y- and x-coordinates of each candidate in the original frame, respectively.

#### 4.2.3 Candidate distribution analysis and potential trajectory exploration

Here, we applied the method presented in 3.6.2 (on pp.49) to extract potential trajectories. Fig. 4-3(a) shows an example of the Y- and X-distribution images of a baseball



sequence, and the potential trajectories produced from this procedure are shown as the linking of ball candidates in YDI and XDI, as depicted in Fig. 4-3(b).

#### 4.2.4 *Trajectory identification*

After trajectory exploration, we obtain a set of potential trajectories. To identify the true ball trajectory from potential trajectories, we first prune the false ones to lower the computational complexity. For each potential trajectory, we have maintained the best-fitting function of the trajectory, the component ball candidates linked, and their associated coordinates and categories (isolated or contacted). The following properties are utilized to eliminate the potential trajectories which cannot be the true ball trajectory.

***Trajectory length:*** The distance from the pitcher to the catcher in a baseball field is about 18.39 meters, and it can be derived that a ball flying from the pitcher to the catcher at the speed of 180 km/h would last for at least 11 frames. (The detailed equation of ball speed estimation is described in section 4.4.) To the best of our knowledge, the highest ball speed in baseball games is no more than 170 km/h. Hence, the potential trajectories shorter than  $L$  ( $L = 11$  here) frames could not possibly be a true trajectory and should be discarded.

***Prediction error:*** The average distance (in pixel) of each ball candidate position from the predicted position is considered as *prediction error*. The potential trajectories with prediction error greater than a threshold  $T_e$  are eliminated.

***Ratio of isolated candidates over all candidates on the trajectory:*** Since the pitched ball is at a distance away from other moving objects in most frames in a pitch scene, the ball trajectory should contain more isolated candidates than contacted ones. On a potential trajectory, if the ratio of the isolated candidates over all candidates is less than 50%, the trajectory could not be the true one and should be discarded.

After elimination, much fewer potential trajectories remain. For each remaining trajectory, we compute the *length of consecutive isolated ball candidates*. The trajectory with the longest *length of consecutive isolated candidates* is finalized and extracted as the ball trajectory. The following pseudo code explains the procedure of trajectory identification and Fig. 4-3(c) shows the final ball trajectory after the procedure of trajectory identification.

**Trajectory Identification**

**Definitions**

*S*: the set of potential trajectories  
*T*: a potential trajectory  
*T.length*: trajectory length  
*T.error*: prediction error  
*T.ratio*: ratio of isolated candidates over all candidates on the trajectory  
*T.LCIC*: length of consecutive isolated candidates  
*I*: the identified ball trajectory

**Algorithm of Trajectory Identification**

**Input:** *S*      **Output:** *I*

**For** each trajectory *T* in *S*

{

**If** *T.length* < *L*    remove *T* from *S*; (*L* = 11 here)

**Else If** *T.error* > *T<sub>e</sub>*    remove *T* from *S*;

**Else If** *T.ratio* < 50%    remove *T* from *S*;

}

*I* = the trajectory *T* with the highest *T.LCIC* in *S*;

#### 4.2.5 Baseball trajectory extraction

The scheme of baseball trajectory extraction is summarized and an example is demonstrated in Fig. 4-4. First, the moving objects with high intensity are segmented out. Utilizing the constraints of size, shape and compactness, ball candidates are detected from the segmented moving objects. The distributions of ball candidates in both Y- and X-directions are analyzed. From the potential trajectories which form parabolic curves in YDI and straight

lines in XDI, the ball trajectory is identified based on the properties of trajectory length, prediction error, the ratio of isolated candidates over all candidates on the trajectory and the length of consecutive isolated candidates. Finally, the ball position in each frame can be obtained and the extracted trajectory can be superimposed on the frame to provide the audience an insight into the pitching content.

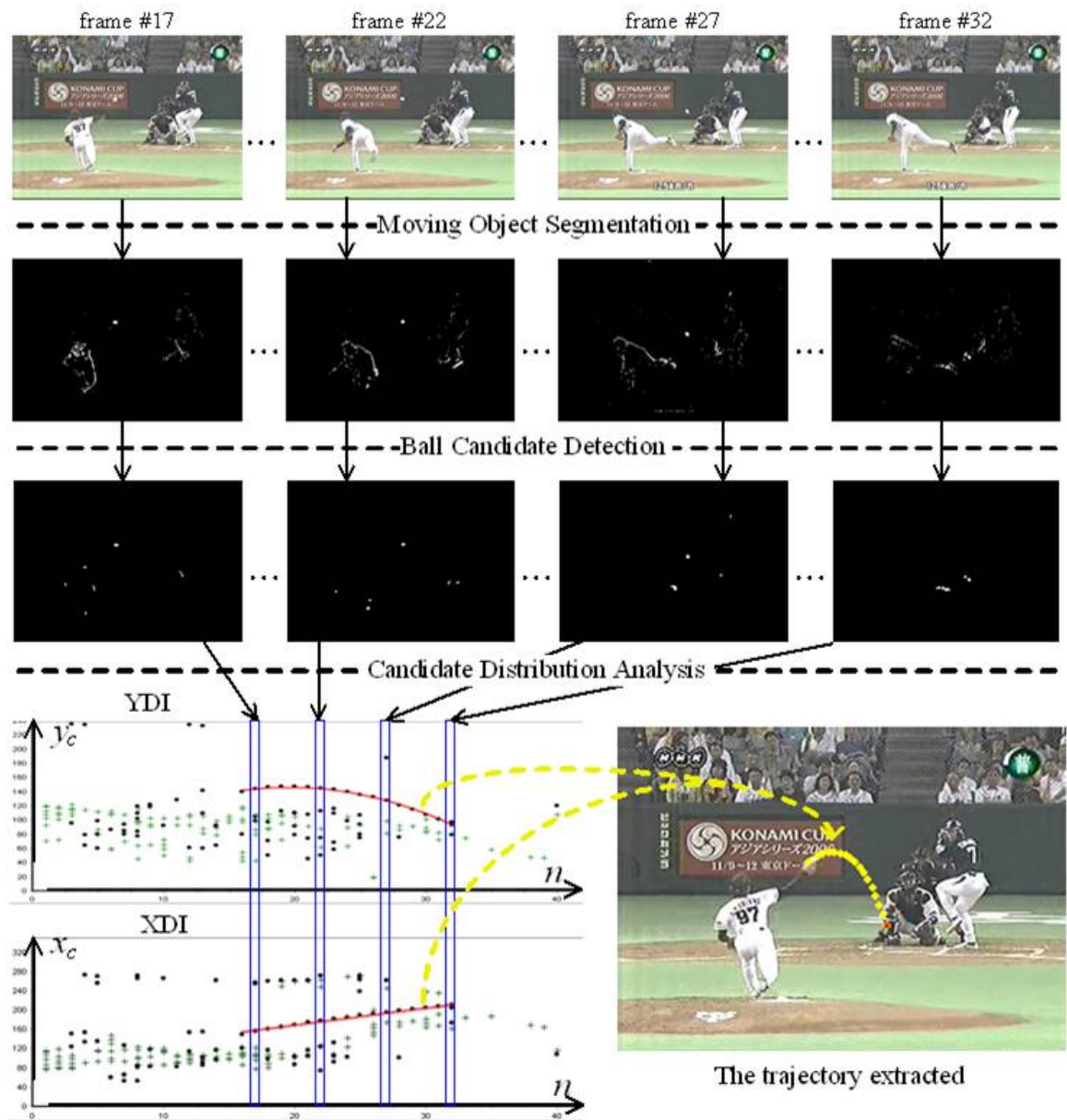


Fig. 4-4. Summarized demonstration of baseball trajectory extraction.

#### 4.2.6 Trajectory-based pitching evaluation and visual enrichment

More keenly than ever, the audience desires to perceive more comprehensive information about games. In this section, we apply the extracted baseball trajectory to pitching evaluation, such as *ball speed estimation* and *trajectory curvature measurement*, and use five-star evaluation to rank each pitch according to its speed and breaking degree. Although there might be deviation in the speed estimation due to the frame rate constraint of the capturing device, it does not influence our evaluation.

**Ball Speed Estimation.** The distance from the pitcher's mound to the home plate is strictly defined in the game rules. Hence, as defined in Eq. (4-3), the ball speed (*BallSpeed* in km/h) can be estimated as the distance from the pitcher's mound to the home plate (18.39 m = 0.01839 km) divided by the time interval of the ball trajectory (*#frm* in frame). The ball speed estimation and the five-star evaluation are given in Table 4-1, which lists the time interval of the trajectory, the estimated ball speed and the respective evaluation.

$$BallSpeed \text{ (km/h)} = \frac{0.01839 \text{ (km)}}{(\#frm / 30) / 3600 \text{ (h)}} \quad (4-3)$$

**Trajectory Curvature Measurement.** A *breaking ball* is a pitch which does not travel straightly like a *fastball*, and it would have a sudden drop when approaching the batter. The more the drop height is, the harder the batter can hit the ball. Furthermore, the drop height rises as the curvature of the trajectory increases. Hence, we measure the curvature of the parabolic curve in YDI (Y-distribution image), the coefficient  $a_1$  in Eq. (3-5). A breaking ball with larger curvature  $|a_1|$  will gain higher ranking, that is, more stars. The trajectory curvature measurement and the five-star evaluation are given in Table 4-2.

**Table 4-1.** Ball speed estimation with five-star evaluation using the ball trajectory.

#frm	BallSpeed(km/h)	Evaluation	#frm	BallSpeed(km/h)	Evaluation
12	164	★★★★★	17	116	★★☆☆☆
13	151	★★★★☆	18	109	★★☆☆☆
14	141	★★★☆☆	19	104	★☆☆☆☆
15	131	★★★☆☆	20	98	★☆☆☆☆
16	123	★★☆☆☆	21	94	★☆☆☆☆

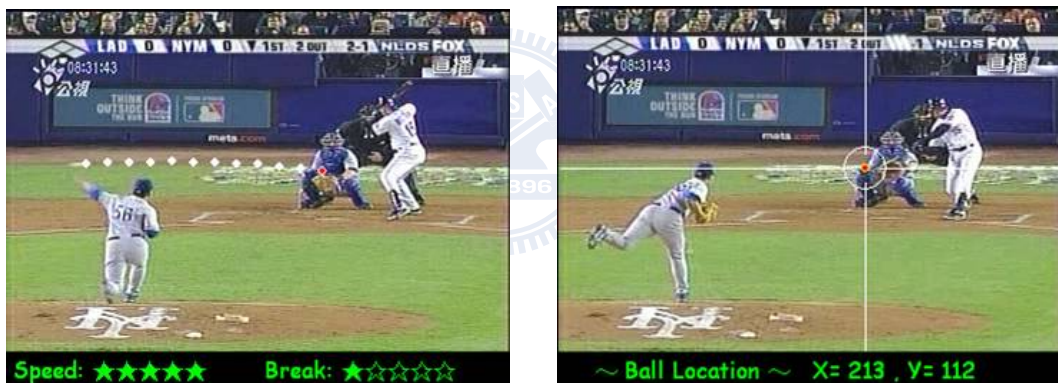
**Table 4-2.** Trajectory curvature measurement with five-star evaluation.

Curvature: $ a_I $	Evaluation
$ a_I  > 0.5$	★★★★★
$0.4 <  a_I  \leq 0.5$	★★★★☆
$0.3 <  a_I  \leq 0.4$	★★★☆☆
$0.2 <  a_I  \leq 0.3$	★★☆☆☆
$ a_I  \leq 0.2$	★☆☆☆☆

The pitching evaluation in this chapter aims at providing visual enrichment for entertainment effects based on the ball trajectory. Actually, in baseball rules there are no regulations about how fast a pitched ball can be considered as five-star or what the curvature of a five-star breaking ball is. Thus, the parameter settings, supported by two experienced experts in baseball games, in Table 4-1 and Table 4-2 for speed estimation and breaking measurement are comparative values, not absolute values.

Two examples of the trajectory-based pitching evaluation and visual enrichment are demonstrated in Fig. 4-5, where Fig. 4-5(a) is a MLB (Major League Baseball) pitch shot with a left-handed pitcher and Fig. 4-5(b) is a JPB (Japan Professional Baseball) pitch shot with a right-handed pitcher. In the left picture of each example, the enriched frame presents the sight when the pitcher is about to throw the ball. The superimposed trajectory clearly depicts the sequence of ball motion for the pitch. In addition, the pitching evaluation displayed at the bottom of the frame provides more details about the pitch. In the right picture of each example, the final ball location of the trajectory is spotlighted with a crosshair (or reticle). If the batter swings at the pitched ball, the enriched frame catches up and reflects the

situation how the ball is hit or missed, as demonstrated in the right picture of Fig. 4-5(a). On the other hand, in baseball rules the *strike zone* is defined as that area over the home plate the upper limit of which is a horizontal line at the midpoint between the shoulders and the belt, and the lower limit is a line at the knees. Hence, if the batter does not swing, the crosshair can provide the reference for the strike/ball judgment, as shown in the right picture of Fig. 4-5(b). Moreover, the ball trajectory and the final ball location can also provide assistant information for the professional personnel to infer the tactics which each pitcher usually adopts in specific situations, such as “the pitcher prefers throwing a breaking ball to the inside corner of the strike zone when there are runner(s) on the base(s) and a fast ball to the outside corner when there is no runner.”



(a) Example of a MLB (Major League Baseball) pitch shot with a left-handed pitcher.



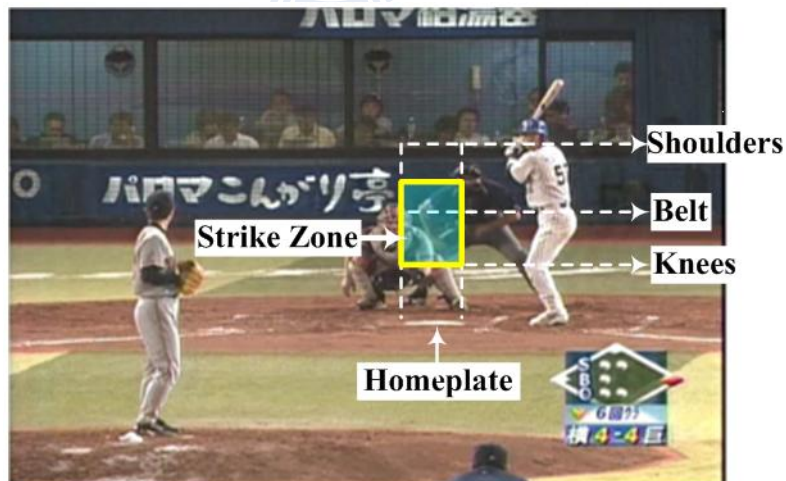
(b) Example of a JPB (Japan Professional Baseball) pitch shot with a right-handed pitcher.

**Fig. 4-5.** Demonstration of trajectory-based pitching evaluation and visual enrichment. Left: the superimposed ball trajectory and pitching evaluation. Right: the final ball location spotlighted with a crosshair.



### 4.3 Automatic Strike Zone Determination

In this section, we design an effective yet compute-easy stance-based algorithm for automatic strike zone determination. *Strike zone* is a conceptual rectangular area over *home plate* defining the boundaries through which a *pitch* must pass in order to count as a *strike* when the batter does not swing [57]. A pitch which does not pass through the strike zone is called a *ball* if the batter does not swing. (The definition and explanation of baseball terminology can be referred in [58].) Strike zone plays a crucial role in baseball since the strike/ball judgment of every pitch must rely on the strike zone. In official baseball rules, the strike zone is defined as that area over the home plate the upper limit of which is a horizontal line at the midpoint between the shoulders and the belt, and the lower limit is a line at the knees [59]. The illustration of strike zone is presented in Fig. 4-6, where the left and right boundaries are decided by the home plate while the upper and lower boundaries are decided according to the dominant points of the batter: the shoulders, belt and knees.

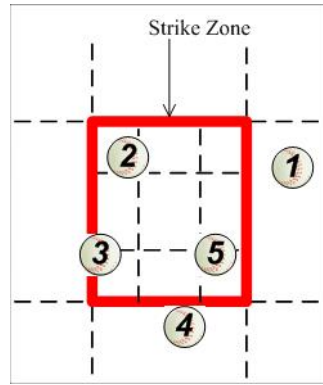


**Fig. 4-6.** Illustration of strike zone definition.

In addition to providing the criteria in strike/ball judgments, the strike zone can also offer a reference for positioning the *pitch location*, the relative location of the ball in/around the strike zone when the ball passes by the batter. The pitch location is an important factor in



determining the batting result. A batter who swings at a lower pitch has a good chance of hitting a *ground ball*, while a batter who swings at a higher pitch has a great chance of hitting the ball in the air. Therefore, recording each pitch location in a sequence of pitches is an important task for baseball analysts, since the pitch location record could provide referential information for the next match and would promote the performance of the pitcher/batter. However, manual recording of the pitch locations is short of efficiency and accuracy. With the baseball trajectory extracted and the strike zone determined by computer vision technologies without manual operation, the task of automatic pitch location recording will be achieved. Then, the sequence of pitches with which a pitcher uses to face a batter can be summarized as a *pitch location image*, as shown in Fig. 4-7(a), where the thick rectangle represents the strike zone, the circles mark the pitch locations and the numbers represent the order of the pitches. The pitch location image not only helps the professional personnel to inspect the match for flaws, but also provides information to the audience for advanced understanding of the pitching content. After accumulating a mass of pitches, we will have the statistical data, as the example shown in Fig. 4-7(b), where the number in each region is simply the count of pitches thrown in the region. These statistical data help to infer whether the pitcher prefers inside or outside (lower or higher) pitches. The ratio of strikes to balls, which reveals the *control* ability of a pitcher, can also be calculated from the statistical data. Furthermore, we would be able to predict the next pitch location by mining the regular patterns of the order of pitch locations, such as “two *upper inside* pitches are usually followed by a *lower outside* pitch”. Hence, in order to achieve the results as shown in Fig. 4-8, trajectory extraction and strike zone determination are the essential tasks which we need to elaborate on.



(a) Pitch location image

174	525	103			
358	532	475			
919	536	892	882	1339	
349	668	829	119	804	1105

(b) Count of pitches in each region

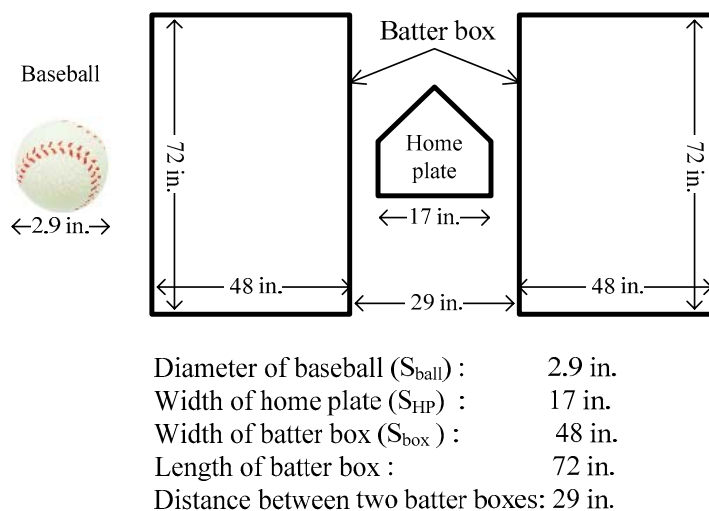
**Fig. 4-7.** Applications for strike zone determination.



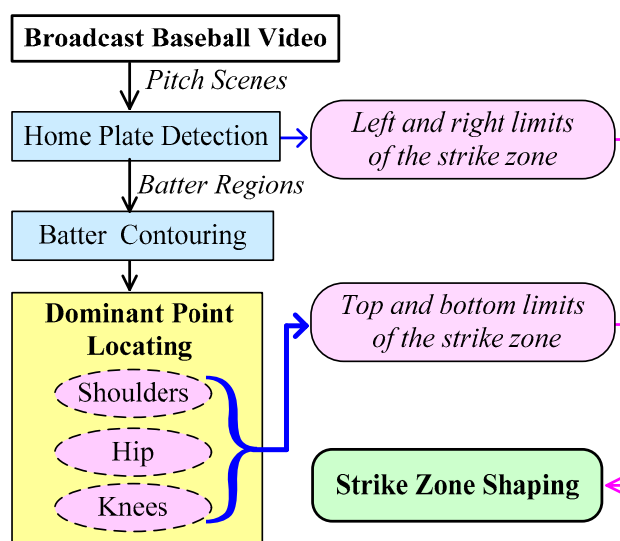
**Fig. 4-8.** Sample results of combining ball trajectory extraction with strike zone

#### 4.3.1 Overview of the proposed strike zone determination algorithm

In baseball, the field is characterized by the strictly-defined layout. The specifications for the equipments and the field are clearly defined in the rule. The domain-specific knowledge can provide much information which can make our system perform more efficiently and reliably. Fig. 4-9 depicts the specifications of the baseball, home plate and batter's boxes, which are utilized in our system. (For more details about the specifications, please refer to [61].) Exploiting the game-specific properties we propose a strike zone determination system containing four major steps: 1) Home plate detection, 2) Batter contouring, 3) Dominant point locating, and 4) Strike zone shaping, as diagramed in Fig. 4-10.



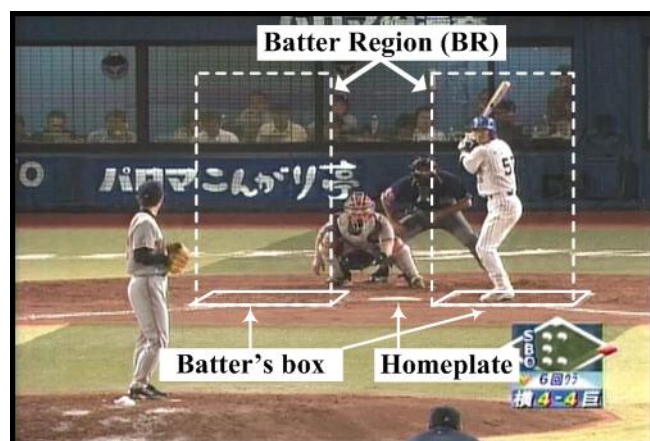
**Fig. 4-9.** Specifications of the baseball, home plate and batter's boxes [61].



**Fig. 4-10.** Block diagram of the proposed strike zone determination system.

To determine the vertical boundaries of the strike zone, the home plate in the pitch scene is first detected utilizing the characteristics: location, intensity and shape. The obtained baseball diameter is also contributive to the estimation of the home plate width in the pitch scene. Since the relative locations and sizes of the home plate and the batter's boxes are clearly defined in the rules, the width of *batter boxes* in a pitch scene can be proportionally

estimated from the width of the home plate. Then, the *batter regions* (BRs), the regions which the batter may stand in, can be circumscribed above the batter boxes so that the batter contouring processing can be limited to the batter region only, therefore with better efficiency. The layout of the home plate, batter's boxes and batter regions in the pitch scene is presented in Fig. 4-11, where the batter regions are virtual rectangles above the batter's boxes to outline the regions containing the batter. As described in the previous section, the strike zone is defined as that area over the home plate the upper limit of which is a horizontal line at the midpoint between the shoulders and the belt, and the lower limit is a line at the knees. To determine the horizontal boundaries of the strike zone, we should locate the dominant points: shoulders, belt and knees. Due to the characteristics of the batting posture, points of curvature extremes are good candidates for dominant points. Hence, we contour the batter in the batter region based on the batter's motion when he is preparing to swing, and locate the dominant points by analyzing the curvature of the contour. With the detected home plate and the located dominant points, the strike zone can take shape finally.



**Fig. 4-11.** Layout of home plate, batter's boxes and batter regions

The proposed strike zone shaping method has the advantages: 1) the strike zone can be shaped adaptively to each batter's stance; 2) no additional camera setting is required; 3) the proposed method is robust to the batter's uniform color; and 4) the proposed method is

applicable to both right- and left-handed batters. In the following, we use a right-handed batter for example to explain the processing stages.

#### 4.3.2 Home plate detection

In the pitch scene, the home plate shows in the form of a short horizontal line segment in a light gray color close to white, as shown in Fig. 4-11. Besides, the home plate is mostly located around the frame center, because the best presentation of the pitch vs. batter can be provided in this viewpoint. Based on these visual properties, we design a compute-easy yet effective algorithm for home plate detection. The procedure is presented in Fig. 4-12.



**Fig. 4-12.** Procedure of home plate detection: (a) Original frame of a pitch scene. (b) Pixels with high intensity around the frame center. (c) Detected home plate.

We remove the pixels with low intensity, and then retain only the pixels in the center quarter region of the frame, as shown in Fig. 4-12(b). Objects are formed from the remaining pixels by region growing. Utilizing the specifications of the baseball and the home plate defined in the rules (see Fig. 4-9), the in-frame width of the home plate  $W_{in}$  can be proportionally estimated from the in-frame diameter of the baseball (computed by ball tracking). Thus, the object in the form of a short horizontal line segment with the width closest to  $W_{in}$  (the estimated in-frame width of the home plate) is extracted as the home plate, as shown in Fig. 4-12(c).

### 4.3.3 Batter region (BR) outlining

The batter is restricted (by game rules) to stand in one of the batter boxes when he is preparing to bat. To reduce the processing area for efficiency and accuracy, we outline the *batter region* above each *batter box* (see Fig. 4-11) based on the relative locations and sizes of the home plate and the batter boxes, as presented in Fig. 4-9. The in-frame width of the batter box  $W_{box}$  can be proportionally estimated from the width of the detected home plate  $W_{hp}$  by Eq. (4-4).

$$W_{box} = S_{box} \times W_{hp} / S_{HP} \quad (4-4)$$

$S_{box}$  and  $S_{HP}$  are the standard widths of the batter box and the home plate, respectively. Then, the batter region (BR) is outlined above each batter box with the height  $H_{region}$  computed by Eq. (4-5).

$$H_{region} = H_{batter} \times W_{hp} / S_{HP} \quad (4-5)$$

We set  $H_{batter} = 78$  in. (about 200 cm) so that the BR could cover almost all batters. Whether the batter is right-handed or left-handed, that is, whether the batter stands in the right or left BR can be judged by the intensity difference between frames within each BR. The BR with the batter would have larger intensity difference.

### 4.3.4 Batter contouring

After batter region outlining and the recognition of the BR with the batter, we are able to contour the batter within a specific region efficiently. To extract the moving edges of the batter, we adopt the algorithm in [60], which incorporates the spatial edge information in the motion detection stage by exploiting double-edge map derived from the difference between two successive frames. Here we give a brief review of the algorithm with the example presented in Fig. 4-13. First, the edge map  $E_n$  of current frame  $I_n$  (gray level image) is calculated as Eq. (4-6):

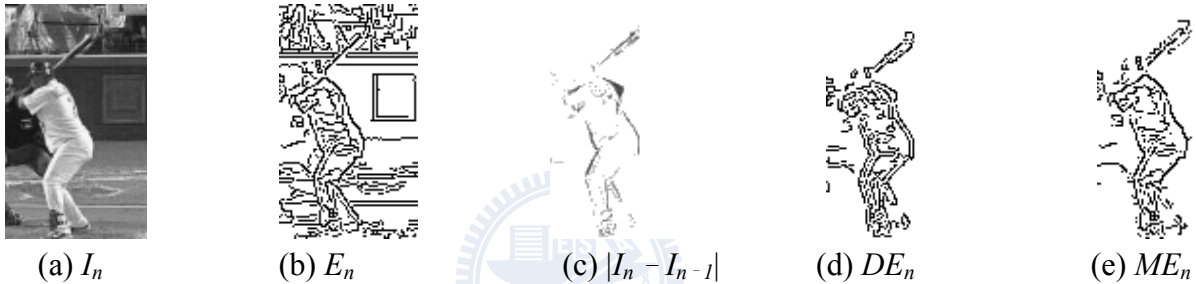
$$E_n = \Phi(I_n) \quad (4-6)$$

where  $\Phi(\cdot)$  is the Canny edge detector and  $n$  is the frame sequence number. The difference edge map  $DE_n$  is calculated by applying the Canny edge detector to the luminance difference image  $|I_n - I_{n-1}|$  of successive frames, as defined in Eq. (4-7). The Gaussian convolution included in the Canny operator suppresses the noise in the luminance difference.

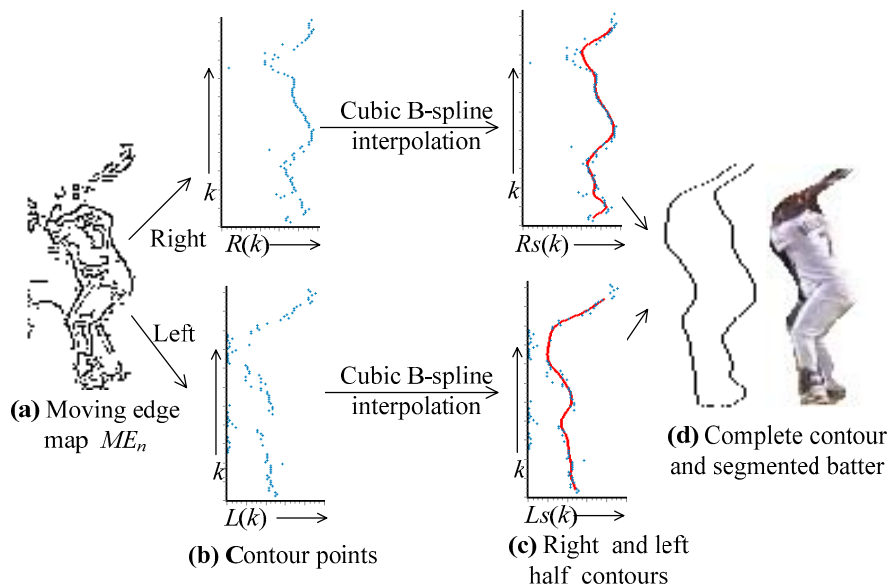
$$DE_n = \Phi(|I_n - I_{n-1}|) \quad (4-7)$$

Finally, the moving edge map  $ME_n$  is generated by selecting the edge pixels in  $E_n$  with at least one neighboring pixel in  $DE_n$ , i.e.

$$ME_n = \{e \in E_n \mid \exists p \in DE_n, e \text{ and } p \text{ are neighboring pixels}\} \quad (4-8)$$



**Fig. 4-13.** Example of moving edge extraction (within the OBR): (a) Gray level image  $I_n$  (b) Edge map  $E_n$  (c) Luminance difference image  $|I_n - I_{n-1}|$  (d) Difference edge map  $DE_n$  (e) Moving edge map  $ME_n$ .



**Fig. 4-14.** Procedure of batter contouring.

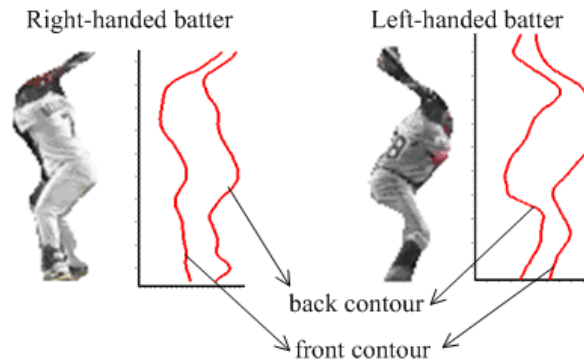


The procedure of batter contouring is illustrated in Fig. 4-14. We extract the rightmost moving edge points along the y-direction as the contour points of the right half contour. The x-coordinate  $x_k$  of each contour point forms a one dimensional discrete function  $R(k) = x_k$ , where  $k$  is the vertical index of each contour point [see Fig. 4-14(b)]. Similarly, the contour points of the left half contour are extracted from the leftmost moving edge points along the y-direction, and the x-coordinate of each contour point forms a one dimensional discrete function  $L(k)$ .

In the subsequent process, we have to locate the dominant points for strike zone shaping. Due to the human kinematic constraints, the sharp turns on the body contour are usually at the joints. Hence, points of curvature extremes are good candidates for the dominant points. The contour curvature can be obtained via computing the partial derivatives on the extracted contour. However, the zigzag contour due to the imperfect moving edge extraction may result in false alarms of curvature extremes. Thus, to achieve spatial continuity and ignore fragments, the cubic B-spline interpolation [64] is used to transform the discrete contour to a continuous one, as shown in Fig. 4-14(c). The complete contour is finalized by combining the smoothed right and left half contours as shown in Fig. 4-14(d), and the points of curvature extremes can be easily obtained by computing the second order partial derivatives on the parameterized contour.

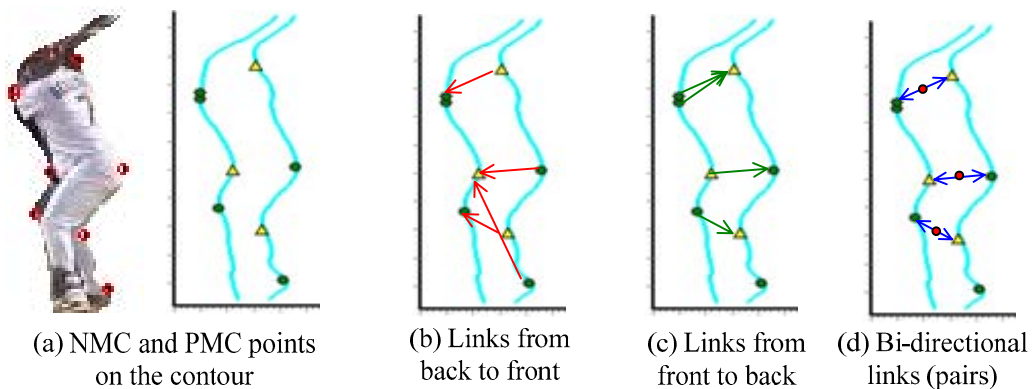
#### **4.3.5 Dominant point locating**

To determine the top and bottom boundaries of the strike zone, a curvature-based method is designed to locate the dominant points on the batter's contour: the hip, shoulders and knees. In the following, the terms *back contour* and *front contour*, as depicted in Fig. 4-15, are used to avoid the confusion of the left or right half contour for a left- or right-handed batter.



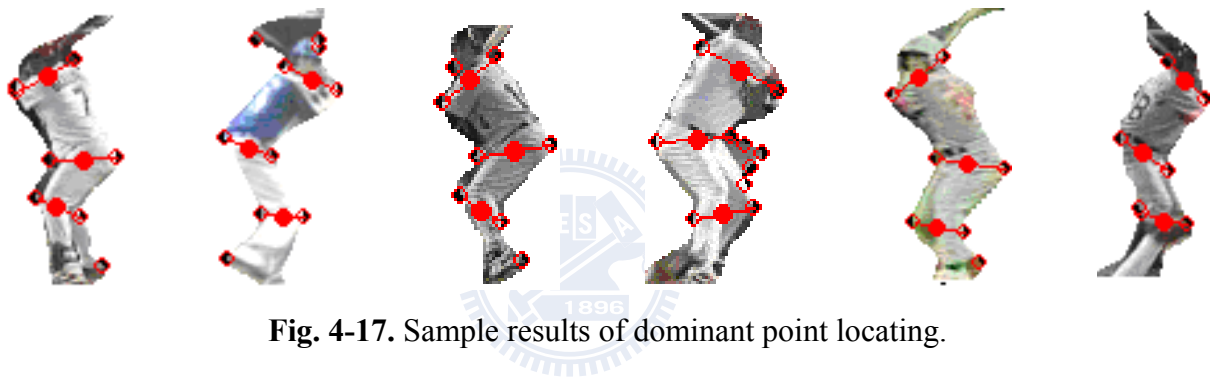
**Fig. 4-15.** Back and front contours for right- and left-handed batters.

We compute the NMC (negative minimum curvature) and PMC (positive maximum curvature) points of the contour, as shown in Fig. 4-16(a). Each NMC (or PMC) point on the back contour is linked to the nearest PMC (or NMC) point on the front contour, and vice versa, as shown in Fig. 4-16(b) and (c). Two points which are bi-directionally linked are deemed as a *pair*, as shown in Fig. 4-16(d). In the following, we use the term “NP point” to denote the midpoint of the pair of a NMC point on the front contour and a PMC point on the back contour, and the term “PN point” to denote the midpoint of the pair of a PMC point on the front contour and a NMC point on the back contour.



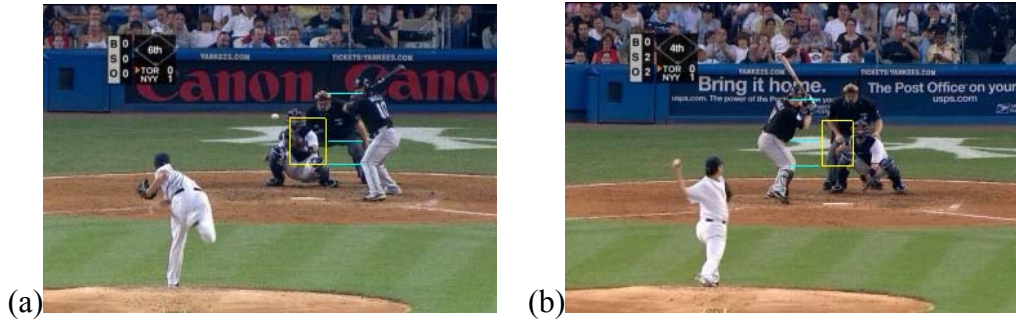
**Fig. 4-16.** Dominant point locating using the points of curvature extremes.

Due to the kinematic constraints of body joints and the balance of the gravity center during the batting action, the trunk and knees tilt forward as the hip is pushed backward. Since the hip forms a salient curve at about half of the batter's height, we choose the NP point closest to the midpoint of the contoured batter's height as the hip point. Then, the PN point under the hip point and with the longest horizontal distance to the hip point is chosen as the knee point, while the PN point above the hip point and with the longest horizontal distance to the hip point is chosen as the shoulder point. Fig. 4-17 demonstrates sample results of the extracted pairs (the red lines) and the located dominant points (the solid red circles).



**Fig. 4-17.** Sample results of dominant point locating.

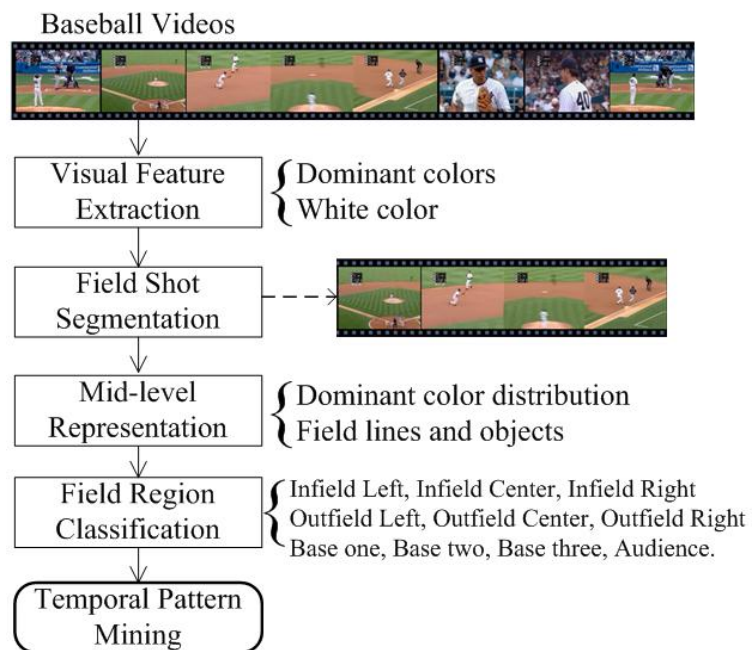
With the home plate detected and the dominant points located, now we are ready to shape the strike zone. The left and right limits of the strike zone are vertical lines at both sides of the detected home plate. The top limit is the horizontal line located at the midpoint between the batter's hip and shoulders, and the bottom limit is located at the batter's knees. Sample results of strike zone shaping and visualization for a right-handed batter and a left-handed batter are presented in Fig. 4-18.



**Fig. 4-18.** Strike zone shaping and visualization: (a) a right-handed batter and (b) a left-handed batter.

#### 4.4 Baseball Exploration using Spatial Pattern Recognition

In addition to the pitcher-batter confrontation, the ball motion and defense process after the ball is batted out is another focus of interest. In this section, we propose algorithms to recognize the spatial patterns (field lines and field objects). We identify the *play region*, the currently focused region of the baseball field, and then *annotation strings* can be generated by analyzing the transition of the identified play regions. Fig. 4-19 presents the overview of the proposed baseball exploration framework.



**Fig. 4-19.** Overview of the proposed baseball exploration framework.

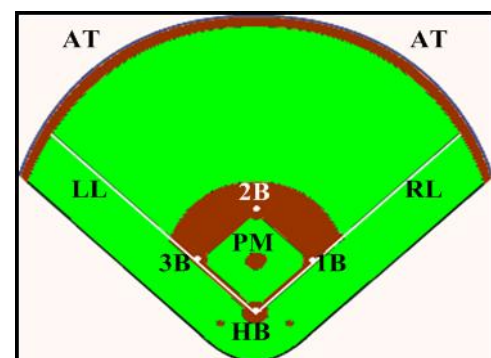
For a given baseball video, the *field shots*, in which the camera follows the batted ball in the field, are first segmented. Then, we extract the visual features in a field shot to analyze the distribution of dominant colors and white pixels. With baseball domain knowledge, the spatial patterns of *field lines* and *field objects* are recognized to classify play region types, such as infield left, outfield right, audience, etc. Finally, from each field shot, an annotation string which describes the transition of play regions is generated to abstract the content of the batting for baseball exploration. In the following, we in turn describe the major components of the proposed system: visual feature extraction, spatial pattern recognition and play region type classification.

#### 4.4.1 Visual feature extraction

As depicted in Fig. 4-20, the baseball field is characterized by a well-defined layout of specific colors. Moreover, important lines and the bases are in white color to provide visual assistance for players, umpires and audience. Therefore, color is an effective visual cue in baseball video analysis. The spatial distribution of dominant colors and white pixels are exploited to detect field objects and lines.



(a) Full view of a real baseball field



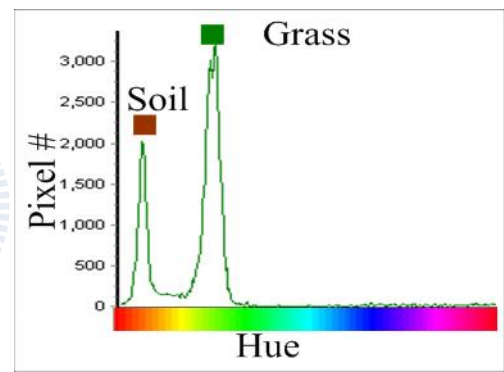
(b) Illustration of field objects and lines.

**Fig. 4-20.** Prototypical baseball field.

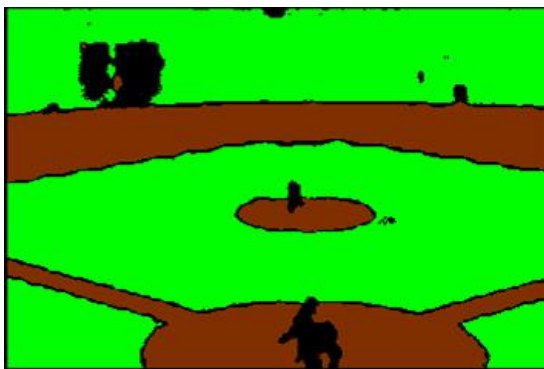
The soil color and grass color are the dominant colors in the baseball field. However, the appearance of the grass and soil colors would vary with the field condition and capturing device. We have observed that within one game, the *hue* value in the HSI (Hue-Saturation-Intensity) color space is relatively stable despite lighting variations. Hence, the hue value is adequate to define the dominant colors. In addition, the *intensity* value is applicable for white pixel extraction. In a field shot, the first frame mainly contains the baseball field, while the later frames, which might zoom in on a player or move to the audience, contain less proportion of the field. Therefore, it is reasonable to define the dominant colors at the first frame of a field shot.



(a) First field frame



(b) Hue histogram



(c) Segmented regions



(d) Extracted white pixels

**Fig. 4-21.** Spatial distribution of dominant colors and white pixels.

Fig. 4-21 demonstrates the spatial distribution of dominant colors and white pixels. The first field frame and its hue histogram are shown in Fig. 4-21(a) and (b), respectively. In the



hue histogram, dominant colors can be defined as the peak of small hue value representing the soil color and the peak of large hue value representing the grass color. The regions segmented by dominant colors are depicted in Fig. 4-21(c), where grass regions are shown in green, soil regions in brown and others in black. The white pixels extracted are presented in Fig. 4-21(d).

#### 4.4.2 Spatial pattern recognition

We focus on the analysis of field shots and attempt to recognize the spatial patterns of field lines and field objects: *left line (LL)*, *right line (RL)*, *pitcher's mound (PM)*, *home base (HB)*, *first base (1B)*, *second base (2B)*, *third base (3B)* and *auditorium (AT)*, as depicted in Fig. 4-20(b). Since the baseball field has a strictly-defined layout, the field lines and objects can be recognized based on the distribution of dominant colors and white pixels. In Fig. 4-22, the top row gives the original frames and the bottom row illustrates the following detection of the field lines and objects.

1) *Left line (LL) and right line (RL)*: A growing algorithm, which produces a vector representation of the line segments [70], is applied to the extracted white pixels. The field lines (*left line* and *right line*) are then obtained by joining together the line segments which are close and collinear, as the oblique lines in Fig. 4-22(a), (b) and (c).

2) *Pitcher's mound (PM)*: An elliptical soil region surrounded by a grass region would be recognized as *pitcher's mound*, as the red rectangle in Fig. 4-22(a) and (c).

3) *Home base (HB)*: *Home base* can be located at the intersection of left line and right line, as shown in Fig. 4-22(a), if both field lines are detected.

4) *First base (1B) and third base (3B)*: The white square located on the right line, if detected, in a soil region would be identified as *first base*, as depicted in Fig. 4-22(a). Similarly, the white square on the left line, if detected, in a soil region would be identified as *third base*, as depicted in Fig. 4-22(b).



5) *Second base (2B)*: In a soil region, a white square on neither field line would be recognized as *second base*, as the white square in Fig. 4-22(a) and (c).

6) *Auditorium (AT)*: The top area which contains high texture and no dominant colors is considered as the *auditorium*, as the black area above the white horizontal line in Fig. 4-22(c).

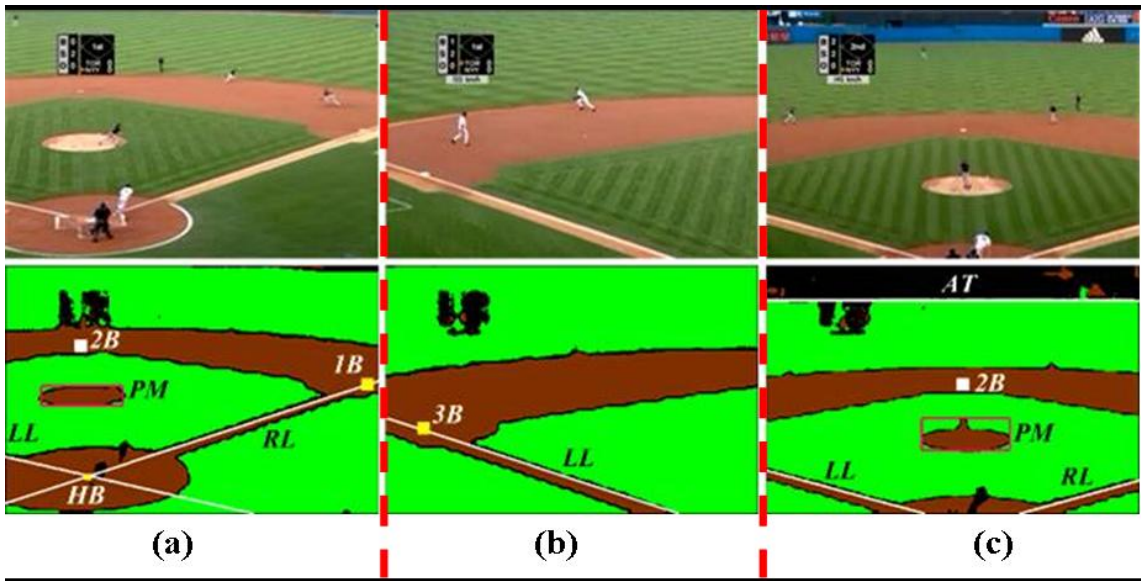
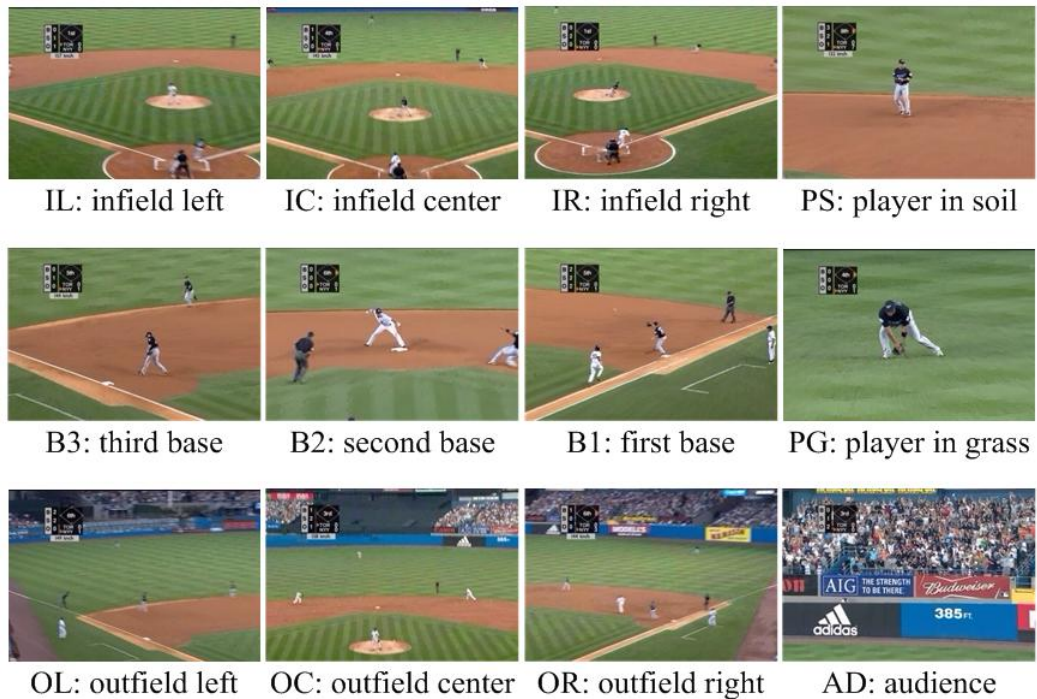


Fig. 4-22. Detection of field lines and field objects.

#### 4.4.3 Play region type classification

In order to comprehend the detailed content of ball movement and region transition, we have to recognize the *play region*, the currently focused region in the baseball field, of each field frame. With baseball domain knowledge, we utilize the detected field objects and lines to classify each field frame into one of the twelve typical play region types: IL (infield left), IC (infield center), IR (infield right), B1 (first base), B2 (second base), B3 (third base), OL (outfield left), OC (outfield center), OR (outfield right), PS (player in soil), PG (player in grass) and AD (audience), as shown in Fig. 4-23. Note that B1, B2 and B3 here represent *play region types* while *1B*, *2B* and *3B* in section 4.4.2 represent *field objects*.



**Fig. 4-23.** Twelve typical play region types.

The rules of play region type classification are list in Table 4-3, where  $W_f$  is the frame width. The function  $P(A)$  returns the percentage of the area  $A$  in a frame,  $X(Obj)$  returns the x-coordinate of the center of the field object  $Obj$ , and  $E(Obj)$  returns whether the field object (or line)  $Obj$  exists or not. Each field frame is classified into one of the twelve play region types by applying the rules on the spatial patterns. Take IL (infield left) as an example. A field frame would be identified as IL under the following conditions:

1. The percentage of  $AT$  area in a frame is no more than 10%,  $PM$  exists and the x-coordinates of  $PM$  center is greater than two-third of the frame width  $W_f$  ( $PM$  is located at the right one-third of a frame).
2. The percentage of  $AT$  area in a frame is no more than 10%,  $PM$  does not exist,  $LL$  exists and  $3B$  does not exist.
3. The percentage of  $AT$  area in a frame is no more than 10%,  $PM$  does not exist,  $LL$  exists,  $3B$  exists and the percentage of soil area is no more than 30%.

**Table 4-3. Rules of play region type classification**

<b>IL:</b>	$\{P(AT) \leq 10\%, E(PM), X(PM) > W_f \times 2/3\} \parallel$ $\{P(AT) \leq 10\%, \sim E(PM), E(LL), \sim E(3B)\} \parallel$ $\{P(AT) \leq 10\%, \sim E(PM), E(LL), E(3B), P(\text{soil}) \leq 30\%\}$
<b>IC:</b>	$\{P(AT) \leq 10\%, E(PM), W_f/3 < X(PM) \leq W_f \times 2/3\} \parallel$ $\{P(AT) \leq 10\%, \sim E(PM), \sim E(RL), \sim E(LL), E(2B), P(\text{soil}) \leq 30\%\}$
<b>IR:</b>	$\{P(AT) \leq 10\%, E(PM), X(PM) \leq W_f/3\} \parallel$ $\{P(AT) \leq 10\%, \sim E(PM), E(RL), \sim E(1B)\} \parallel$ $\{P(AT) \leq 10\%, \sim E(PM), E(RL), E(1B), P(\text{soil}) \leq 30\%\}$
<b>B1:</b>	$\{P(AT) \leq 10\%, \sim E(PM), E(RL), E(1B), P(\text{soil}) > 30\%\}$
<b>B2:</b>	$\{P(AT) \leq 10\%, \sim E(PM), \sim E(RL), \sim E(LL), E(2B), P(\text{soil}) > 30\%\}$
<b>B3:</b>	$\{P(AT) \leq 10\%, \sim E(PM), E(LL), E(3B), P(\text{soil}) > 30\%\}$
<b>OL:</b>	$\{10\% < P(AT) \leq 80\%, E(PM), X(PM) > W_f \times 2/3\} \parallel$ $\{10\% < P(AT) \leq 80\%, \sim E(PM), E(2B), X(2B) > W_f \times 2/3\} \parallel$ $\{10\% < P(AT) \leq 80\%, \sim E(PM), \sim E(2B), E(LL), \sim E(RL)\}$
<b>OC:</b>	$\{10\% < P(AT) \leq 80\%, E(PM), W_f/3 < X(PM) \leq W_f \times 2/3\} \parallel$ $\{10\% < P(AT) \leq 80\%, \sim E(PM), E(2B), W_f/3 < X(2B) \leq W_f \times 2/3\}$
<b>OR:</b>	$\{10\% < P(AT) \leq 80\%, E(PM), X(PM) \leq W_f/3\} \parallel$ $\{10\% < P(AT) \leq 80\%, \sim E(PM), E(2B), X(2B) \leq W_f/3\} \parallel$ $\{10\% < P(AT) \leq 80\%, \sim E(PM), \sim E(2B), E(RL), \sim E(LL)\}$
<b>AD:</b>	$\{P(AT) > 80\%\}$
<b>PS:</b>	$\{P(AT) \leq 10\%, \sim E(PM), \sim E(2B), \sim E(RL), \sim E(LL), P(\text{soil}) > 30\%\}$
<b>PG:</b>	$\{10\% < P(AT) \leq 80\%, \sim E(PM), \sim E(2B), \sim E(RL), \sim E(LL)\}$
<b>Unknown:</b>	others

The scheme of play region type classification within a field shot is illustrated in Fig. 4-24. The spatial patterns are first recognized by the distribution of dominant colors and white pixels in field frames. According to the rules on the spatial patterns, each field frame is then classified into one of the twelve typical play region types. To filter out instantaneous misclassifications of play region types within a field shot, a fixed length temporal window and majority voting are applied. Thus, an annotation string which describes the transition of play regions contained in a field shot can be obtained. The content of the sample field shot in Fig. 4-24 says that the ball is first batted into the left infield. Then, the shortstop picks up the

ball and throws it to the first baseman. The batting process can be appropriately abstracted by the output annotation string: IL (infield left) → PS (player in soil) → IR (infield right) → B1 (first base).

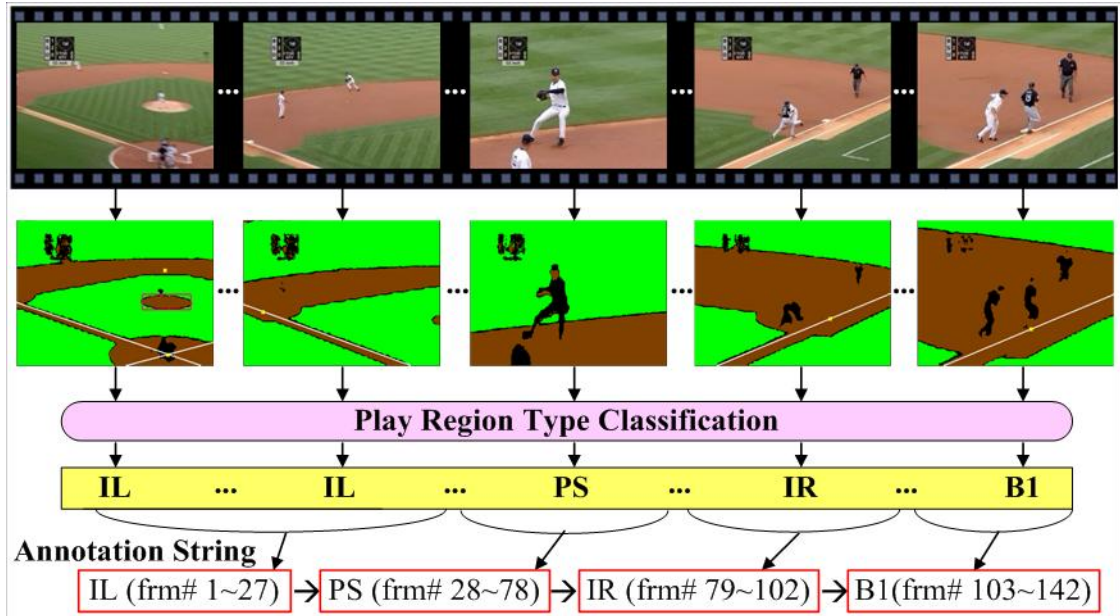


Fig. 4-24. Scheme of play region type classification within a field shot.

#### 4.5 Experimental Results

The scheme elaborated in the previous sections supports trajectory-based ball tracking, strike zone determination and batting result analysis. For performance evaluation, we conduct the experiments on broadcast baseball video (352 × 240, MPEG-1) captured from different sports channels, as listed in Table 4-4.

Table 4-4. Testing data used in the experiments.

Baseball Video	Source channels
1. MLB (Major League Baseball)	PTS channel of Taiwan
2. JPB (Japan Professional Baseball)	NHK channel of Japan
3. CPBL (Chinese Professional Baseball League)	VL sports channel of Taiwan

#### 4.5.1 Results of trajectory-based baseball tracking

For trajectory-based ball tracking, some parameters are used.  $T_d$  is the threshold of frame difference in moving object segmentation. Since the intensity of the baseball should be much higher than the background or other objects in the frames, we adaptively set  $T_d$  by Eq. (4-9), which can eliminate many noises and still retain the ball.

$$T_d = \text{Average\_intensity\_of\_the\_frame} \times 50\% \quad (4-9)$$

As to the range of size filter, up to 95% of the baseball sizes (in pixel) in the frames of the resolution  $352 \times 240$  are within the range  $[8, 50]$  by statistical results, so  $[R_{min}, R_{max}]$  is set to  $[8, 50]$ . The parameter  $R_a$  is the threshold of shape filter. Generally speaking, the aspect ratio of the baseball should equal 1. Due to the high speed movement, the ball may deform over frames. Thus, for tolerance of deformation, the constraint of shape filter is loosened. Since an object with aspect ratio greater than 3 is far from a ball,  $R_a$  is set to 3. Since an object of compactness degree  $D_c$  less than half cannot be claimed to be “compact”, the threshold of compactness filter  $T_c$  is set to 50%. Furthermore, though the ball trajectory over frames is not exactly a parabolic curve, a trajectory with great prediction error cannot be the ball trajectory. Thus, for reasonable error tolerance, the threshold of prediction error  $T_e$  is set to 2 (in pixel).

The ball position in each video frame is manually recognized as *ground truth*. A ground truth ball is called “detected” if it matches a ball candidate. A ground truth ball falling on the obtained trajectory is called “tracked”, since the ball position can be predicted on the trajectory by the motion characteristics even though it does not match a ball candidate. The experimental results of ball detection and tracking are listed in Table 4-5, where *#clip* shows the number of pitch shots, *#frm* represents the total number of frames in all the pitch shots and *#bf* represents the number of the frames containing the ball. The column “*#detected (%)*” gives the number of balls detected and the detection rate (*#detected / #bf*), “*#tracked (%)*”



gives the number of balls tracked and the tracking rate ( $\#tracked / \#bf$ ),  $\#false$  (%) gives the number of false alarms and the false alarm rate ( $\#false / \#frm$ ).

It can be found that there are some misses because the ball might not be detected when it passes over a left-handed batter dressed in a white uniform. Fortunately, the positions of missed balls can be recovered by applying the ball position prediction. An example of ball detection is shown in Fig. 4-25(a), where the ball is missed in two frames when passing over the white uniform. The result of ball tracking is presented in Fig. 4-25(b) where the missed ball positions can be recovered by applying the predicted positions of the obtained trajectory. Although some tracking errors might exist, the proposed scheme promotes the overall accuracy of ball tracking up to 96%. The ball tracking with visual enrichment of some example pitch shots are demonstrated in Fig. 4-26. It is convincing that the proposed framework performs well in baseball clips from different channels, no matter whether the pitcher/batter is left- or right- handed.

**Table 4-5.** Performance of baseball detection and tracking

Baseball	<i>#clip</i>	<i>#frm</i>	<i>#bf</i>	<i>#detected</i> (%)	<i>#tracked</i> (%)	<i>#false</i> (%)
MLB	30	1380	424	387 (91.27%)	409 (96.46%)	11 (0.80%)
JPB	32	2089	466	435 (93.35%)	453 (97.21%)	12 (0.57%)
CPBL	24	942	352	326 (92.61%)	338 (96.02%)	7 (0.74%)
<b>Total</b>	<b>86</b>	<b>4411</b>	<b>1242</b>	<b>1148 (92.43%)</b>	<b>1200 (96.62 %)</b>	<b>30 (0.68%)</b>

\* detection rate (%) =  $\#detected / \#bf$ , tracking rate (%) =  $\#tracked / \#bf$ , and false alarm rate =  $\#false / \#frm$ .



(a) Ball detection (a) Ball tracking

**Fig. 4-25.** Illustration of ball detection and ball tracking in baseball video. (a) Ball detection. Two ball positions are missed when passing over the white uniform. (b) Ball tracking. Positions of missed balls can be recovered.



(a) MLB pitch shot (b) JPB pitch shot (c) CPBL pitch shot

**Fig. 4-26.** Examples of ball tracking and visual enrichment for various baseball clips.

The experiments run on an IBM ThinkPad X60 notebook computer (CPU: Intel Core Duo T2400 1.83GHz, RAM: 1GB). For a pitch shot of 2 seconds, the required processing time is about 8~10 seconds. In baseball games, the duration between two successive pitches is usually longer than 10 seconds. That is, the proposed framework is able to compute the ball trajectory of a pitch shot and superimpose the trajectory over the video before the next pitch coming up in near real-time. The application of enriching the live broadcast baseball video for entertainment effects becomes feasible.



It is difficult to perform a head-to-head comparison with other algorithms since there exist differences in the actual setup and the implementation. As a reasonable comparison, we divide the process into two stages: potential trajectory exploration and ball trajectory identification, and make the discussion.

#### *A. Potential Trajectory Exploration*

Kalman filter and particle filter are widely used in moving object tracking. However, particle filter is usually applied to tracking large objects with salient characteristics of edges or colors, such as cars and people [55]. Though particle filter can also be used in ball tracking, it is applicable to ball of big size, such as basketball [55], for which a distinguished target model can be built. Since most of the ball tracking algorithms in the literature [8,11,52] are Kalman filter-based, we make a comparison focusing on Kalman filter. We compare the performance between the Kalman filter-based algorithm (KF) and the proposed parabola-based algorithm (PB). The performance metrics include the number of potential trajectories produced and the number of the ball candidates linked on the potential trajectories. For each pitch sequence, fewer ball candidates linked on the potential trajectories need fewer updates of the prediction function or Kalman filter. The fewer number of the potential trajectories is, the less computation in trajectory identification is.

Using the 86 testing sequences as in Table 4-5, the comparison is presented in Table 4-6. The notations #Seq, #PT and Avg. #PT represent the number of testing pitch sequences, the total number of potential trajectories produced in the pitch sequences and the average number of potential trajectories produced per pitch sequence. #Cand and Avg. #Cand denote the total number of ball candidates linked over all the potential trajectories and the average number of ball candidates linked per pitch sequence. It can be observed that KF algorithm produces more potential trajectories with more ball candidates linked, because KF algorithm may link neighboring non-ball objects in consecutive frames and form many potential trajectories

which are not parabolic and need to be eliminated. However, the proposed PB algorithm aims at extracting only the potential trajectories which form (near) straight lines in X-direction and (near) parabolic curves in Y-directions, simultaneously. Therefore, the proposed parabola-based algorithm is more efficient in potential trajectory exploration since fewer ball candidates linked cause fewer updates of prediction function, and it will save more time in trajectory identification due to the fewer potential trajectories need to be validated.

**Table 4-6.** Comparison between the Kalman filter-based algorithm and the proposed physics-based algorithm in baseball video

		KF algorithm				Proposed PB algorithm			
Video	#Seq	#PT	Avg. #PT	#Cand	Avg. #Cand	#PT	Avg. #PT	#Cand	Avg. #Cand
MLB	30	645	21.5	3819	127.3	520	17.33	2803	93.43
JPB	32	1120	35	6835	213.59	557	17.41	3352	104.75
CPBL	24	510	21.25	3297	137.38	234	9.75	1435	59.79
Total	86	2275	26.45	13951	162.22	1311	15.24	7590	88.26

### B. Trajectory Identification

Extracting the true ball trajectory from lots of potential trajectories needs some identification mechanism. Chu *et al.* [52] simulate all the possible trajectories of ball pitching varying in different beginning velocities, releasing angles and spin rates to derive physical limitation for trajectory identification, which is time-consuming. To transform 2D trajectories into 3D trajectories for validation, they compute the ratio of “the vertical movement distance of pitches in the real world” (1 meter, assumed by the authors) to “the average vertical movement distance of pitches in the video frames of their dataset”, and then estimate the depth of each ball candidate proportionally. However, the positions of pitchers releasing the ball and catchers catching the ball vary. The variation in the vertical movements of numerous pitches should be large and a pitch with the vertical movement far from the average, *e.g.* an underhand pitch, may not be identified reliably.

In our proposed scheme, we maintain the best-fitting function of the trajectory, the component ball candidates linked and their associated coordinates and categories (isolated or contacted) for each potential trajectory. Then, the properties for pruning the false trajectories and extracting the true ball trajectory, including *trajectory length*, *prediction error*, the *ratio of isolated candidates over all candidates on the trajectory*, and the *length of consecutive isolated candidates*, can be computed quickly. Therefore, the ball trajectory can be identified efficiently and reliably.

#### 4.5.2 Results of strike zone determination

The results of home plate detection are presented in Table 4-7, which lists the correct, missed and false detections. The home plate can be correctly detected in most clips. Only two misses occur as the home plate is stained with soil and is not clear in the frame, as shown in Fig. 4-27. For the clarity of strike/ball decision, the plate umpire has the responsibility to clean the home plate when the home plate is stained. Therefore, the home plate is clear in most of the sequences. On the other hand, there are seldom objects similar to the home plate in the center region of the pitch scene. Thus, we achieve a fairly good performance in home plate detection.

**Table 4-7.** Performance of home plate detection.

Video	#Seq	#Correct	#Missed	#False
1. MLB	33	32	1	0
2. JPB	33	33	0	0
3. CPBL	34	33	1	0
<b>Overall</b>	<b>100</b>	<b>98</b>	<b>2</b>	<b>0</b>



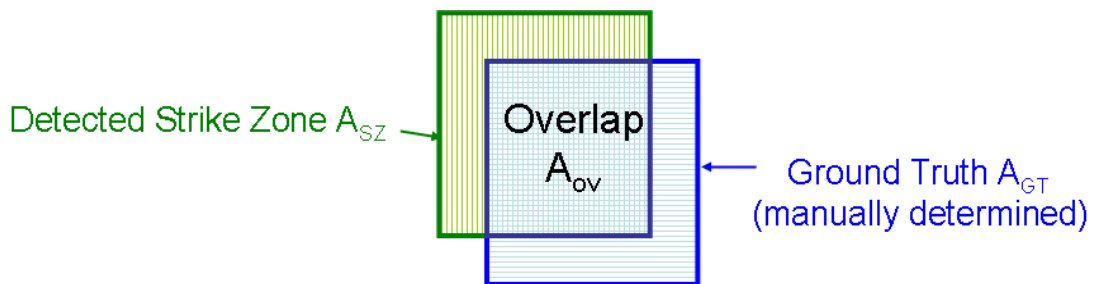
**Fig. 4-27.** Error cases of home plate detection.

The *ground truth* of strike zone is established by manual positioning of the two sides of the home plate and the batter's shoulders, hip and knees. For performance evaluation of the proposed strike zone determination system, two degrees P and R, as defined in Eq. (4-10) and Eq. (4-11) respectively, are proposed to measure the degree of the overlap between the computer-generated strike zone and the ground truth.

$$P = A_{ov} / A_{SZ} \quad (4-10)$$

$$R = A_{ov} / A_{gt} \quad (4-11)$$

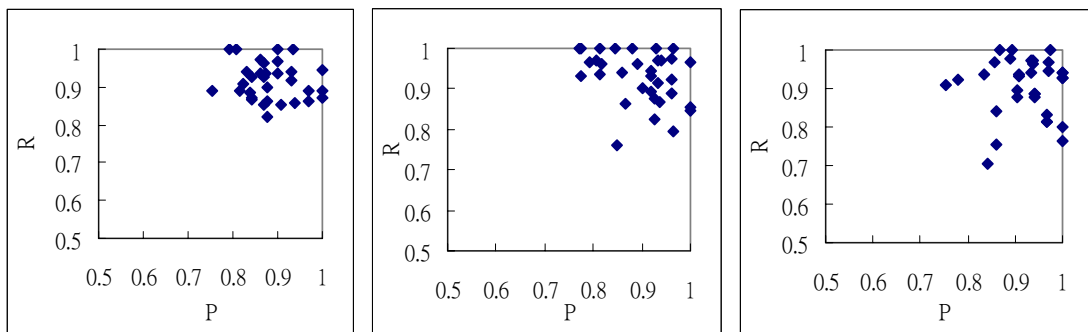
As illustrated in Fig. 4-28,  $A_{SZ}$  is the area of the computer-generated strike zone,  $A_{gt}$  is the area of the ground truth strike zone and  $A_{ov}$  is the area of the overlap between the computer-generated strike zone and the ground truth.



**Fig. 4-28.** Illustration of  $A_{SZ}$ ,  $A_{gt}$  and  $A_{ov}$ .

The P-R distributions of the sequences with correct home plate detection are presented in Fig. 4-29, where each point represents a testing sequence, the horizontal and vertical axes are the P and R degrees, respectively. It can be seen that for most of the sequences, we

achieve high P and high R degrees, both of which are around 0.9. That is, the computer-generated strike zone has great overlap with the ground truth in most of the sequences. Table 4-8 reports the average P and R degrees on the sequences with correct home plate detection. We can see good performance of the proposed strike zone system. Overall, both the average P and R degrees are over 0.9. Some examples of correctly determined strike zones (the superimposed rectangles) demonstrated in Fig. 4-30 make it convincing that the proposed framework performs well in broadcast baseball video captured from different channels, no matter whether the batter is right- or left-handed and no matter what color of uniform the batter is dressed in.



(a) MLB

(b) JPB

(c) CPBL

**Fig. 4-29.** P-R distributions of MLB, JPB and CPBL sequences.

**Table 4-8.** Performance of strike zone determination

Video	#sequence	Avg. P	Avg. R
1. MLB	32	0.884	0.914
2. JPB	33	0.895	0.928
3. CPBL	33	0.923	0.904
Overall	98	0.901	0.915



Fig. 4-30. Example results of strike zone determination.

Although we achieve quite good performance in automatic strike zone determination, the computer-generated strike zone is not so satisfying for a few sequences. We examine the experimental results and find that the inappropriate determinations of the strike zone are mainly caused by the faults in batter contouring. The dynamic advertising board, the audience moving or other noises behind the batter would lead to some errors in object segmentation. Fig. 4-31 gives two examples of the unsatisfying strike zone determination. In Fig. 4-31(a), the change of the dynamic advertising board behind the batter leads to the defect in the batter contouring and the dominant points cannot be located appropriately. However, the dynamic advertising board does not change very often. The influence of the dynamic advertising board is not so notable. In Fig. 4-31(b), the front contour of the batter is not extracted appropriately because the plate umpire behind the batter makes a sudden movement. Nevertheless, for the clarity of strike/ball decision, the plate umpire makes little movement in most cases, and we



can achieve good results in batter contouring. To the best of our knowledge, there is still no perfect solution to object segmentation in the dynamic background. However, our proposed object segmentation (batter contouring) algorithm is appropriate in most sequences and is sufficient to support strike zone determination.



(a) Improper batter contouring due to the dynamic advertising board.



(b) Improper batter contouring due to the movement of the plate umpire.

**Fig. 4-31.** Examples of the improper strike zone determination

### 4.5.3 Results of play region classification

The ground truths of the play region types contained in each field shot are identified manually. Table 4-9 presents the experimental results. The second column “#Ground truth” represents the total number of field shots containing the play region type designated in the first column. Note that a field shot might comprise more than one play region type. The “Correct detection” and “False alarm” represent the number of correct detections and false alarms. Both the precision and recall are about 90% except for the precision of PS (player in



soil) and the recall of B2 (second base region). The false alarms of PS might result from no field object detected in the infield, and the missed detection of play region type B2 might result from the missed detection of field object 2B. These could be improved by enhancing field object detection and refining the rules of play region type classification. Overall, we achieve good performance.

**Table 4-9.** Performance of play region classification.

<b>Play region type</b>	<b>#Ground truth</b>	<b>Correct detection</b>	<b>False alarm</b>	<b>Precision (%)</b>	<b>Recall (%)</b>
<b>IL</b>	34	33	2	94.3	97.1
<b>IC</b>	31	30	2	93.8	96.8
<b>IR</b>	51	49	1	98.0	96.1
<b>B1</b>	48	47	2	95.9	97.9
<b>B2</b>	12	10	1	90.9	83.3
<b>B3</b>	9	8	0	100.0	88.9
<b>OL</b>	18	18	2	90.0	100.0
<b>OC</b>	17	15	2	88.2	88.2
<b>OR</b>	25	25	1	96.2	100.0
<b>AD</b>	18	18	2	90.0	100.0
<b>PS</b>	38	34	7	82.9	89.5
<b>PG</b>	54	52	7	88.1	96.3

#### 4.6 Summary

In this chapter, we present a trajectory-based method for ball tracking in baseball video. Pitch evaluation and enriched visual presentation can be provided to the sports fans and professionals. To assist strike/ball judgment and position the pitch locations, a stance-based strike zone shaping algorithm is designed. Furthermore, we utilize the strictly-defined specifications of the baseball field to recognize the spatial patterns in each frame and identify what region of the baseball field is currently focused. Thus, we can infer the ball routing

patterns and defense process from the transitions of play regions. From ball tracking, strike zone shaping to play region classification for ball routing pattern inference, we have fairly extensive analysis for baseball video. Informative annotation and sports information enable the sports fans and professionals to go deep into the game.



## Chapter 5. Conclusions and Future Work

There is a saying, “To know both the opponent and yourself, and you can fight with no danger of defeat.” We can stand a better chance if we know the opponent more. Hence, game study before the play is a task of vital importance for the coach and players. However, manual game logging, annotation and analysis via watching the whole sports video are laborious and time-consuming. Therefore, the coach and players keenly desire the assistance of computer technology in game study. On the other hand, the audience or sports fans recently are no longer satisfied with the video viewing systems providing quick browsing, indexing and summarization of sports video. They demand informative data to have a further insight into the games. Hence, our research in this thesis focuses on sports video content analysis, understanding and annotation so as to provide computer-assisted game study and content-based sports information retrieval.

In sports games, significant events are mainly caused by the ball-player interaction and the ball trajectory brings much semantic/tactical information contributive to content understanding. Hence, we propose a physics-based ball tracking scheme to compute the ball trajectory, and furthermore design an innovative approach capable of reconstructing the 3D trajectory from single camera video. Since the ball is small and usually moves fast in frames, recognizing the ball within a single frame is almost impossible. We identify the ball trajectory via judging whether the trajectory conforms to the ball motion characteristics rather than recognize which object is the ball in each frame. The ball positions missed can also be recovered by the obtained trajectory. Moreover, the 2D-to-3D inference is intrinsically a challenging problem due to the loss of the depth information in picture capturing. Incorporating the court specifications for camera calibration and the physical characteristics of ball motion for 3D trajectory modeling, we are able to compute the motion parameter of

the modeled 3D trajectory and approximate the depth information. The challenge of 3D trajectory reconstruction from single camera video is thus overcome. Manifold trajectory-based applications are designed to comprehend the semantic or tactical content, including: shooting location estimation in basketball, pitch analysis in baseball, set type recognition, serve placement estimation and 3D virtual replay in volleyball, etc. Game watching becomes an entirely novel and exciting experience.

The strike zone plays a crucial role in each pitch of the baseball games since the strike zone not only supports the strike/ball judgment but also provides the reference for determining the pitch location. Thus, we design a stance-based strike zone shaping scheme which integrates efficient algorithms of home plate detection, object contour and dominant point locating. No matter the batter is right- or left-handed, the strike zone can be shaped adaptively to the batter's stance. In addition to the confrontation of the pitch vs. the batter, the ball motion and the defense process after the ball is batted into the field also catch the attention of the audience. Thus, we recognize the spatial patterns in the frames of the field shot, classify the play regions (the active regions of event occurrence), and infer the ball routing patterns and defense process from the transitions of play regions. From ball tracking, strike zone shaping to play region classification for ball routing pattern inference, we have fairly extensive analysis for baseball video. Informative annotation and sports information enable the sports fans and professionals to go deep into the game.

Comprehensive experiments on basketball, volleyball and baseball videos have been conducted. The experimental results show that the proposed methods perform well in retrieving game information and even reconstructing 3D information from single camera video for different kinds of sports. The features and techniques proposed in this thesis lead to satisfactory solution for content understanding, tactics analysis sports information retrieval and computer-assisted game study in many kinds of sports videos. Although the 3D trajectory reconstruction method proposed in this thesis has good results, there is still some deviation

between the real-world ball trajectory and the reconstructed trajectory. This may result from the effects of the physical factors we do not involve, such as air friction, ball spin rate and spin axis, etc., and the intrinsic constraints, such as the loss of 3D information or depth information, lighting conditions, noises, the capturing angle and the frame rate of the capturing device, etc. Hence, one direction of our future work is to involve more physical factors in modeling the 3D trajectory for better approximating the reconstructed 3D trajectory to the real world ball trajectory. Moreover, single-view video analysis may be limited by the incomplete 3D information due to object occlusion and the loss of depth information. Another direction is to extend the proposed approaches to multi-view video analysis. In the future, we will integrate the information from multiple cameras to reinforce multimedia content analysis, understanding, indexing and annotation.

On the other hand, we are currently working on deriving the intrinsic rules of region transitions for different defense patterns in baseball games, using temporal pattern mining based on the play region classification proposed in this thesis. The cooperation of players to achieve a successful defense is exciting and inspiring. Hence, we will utilize the transitions of the play region classified for ball route pattern deducing, content-based defense process recognition and similarity event retrieval with concise content presentation, so that the sports fans and professionals will be greatly assisted in game strategy studying, statistics collection, tactics analysis and even improving their own skills.

## Reference

- [1] L. Y. Duan, M. Xu, Q. Tian, C. S. Xu and J. S. Jin, "A unified framework for semantic shot classification in sports video," *IEEE Trans. on Multimedia*, vol. 7, 2005, pp. 1066-1083.
- [2] H. Lu, Y. P. Tan, "Unsupervised clustering of dominant scenes in sports video," *Pattern Recognition Letters*, vol.24 ,issue 15, 2003, pp. 2651-2662.
- [3] T. Mochizuki, M. Tadenuma and N. Yagi, "Baseball video indexing using patternization of scenes and hidden Markov model," in: *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 3, 2005, pp.1212-1215.
- [4] J. Assfalg, M. Bertini, C. Colombo, A. D. Bimbo, W. Nunziati, "Semantic annotation of soccer videos: automatic highlights identification," *Computer Vision and Image Understanding*, vol. 92, issue 2-3, 2003, pp. 285-305.
- [5] Y. Gong, M. Han, W. Hua, W. Xu, "Maximum entropy model-based baseball highlight detection and classification," *Computer Vision and Image Understanding*, vol. 96, issue 2, 2004, pp. 181-199.
- [6] C. C. Cheng, C. T. Hsu, "Fusion of audio and motion information on HMM-based highlight extraction for baseball games," *IEEE Trans. on Multimedia*, vol. 8, 2006, pp. 585-599.
- [7] L. Xie, P. Xu, S. F. Chang, A. Divakaran, H. Sun, "Structure analysis of soccer video with domain knowledge and hidden Markov models," *Pattern Recognition Letters*, vol. 25, issue 7, 2004, pp. 767-775.
- [8] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, K. W. Wan, "Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video," in: *Proc. of the 11th ACM Int. Conf. on Multimedia*, 2003, pp. 11-20.
- [9] H. T. Chen, H. S. Chen and S. Y. Lee, "Physics-based ball tracking in volleyball videos with its applications to set type recognition and action detection," in: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2007, pp. I-1097-I-1100.
- [10] H. T. Chen, H. S. Chen, M. H. Hsiao, W. J. Tsai and S. Y. Lee, "A trajectory-based ball tracking framework with visual enrichment for broadcast baseball videos," *Journal of Information Science and Engineering*, 24 (1), 2008, pp. 143-157.
- [11] A. Gueziec, "Tracking pitches for broadcast television," *Computer*, vol.35, 2002, pp.38-43.

- [12] Hawk-Eye, <http://news.bbc.co.uk/sport1/hi/tennis/2977068.stm>.
- [13] QUESTEC, [http://www.questec.com/q2001/prod\\_uis.htm](http://www.questec.com/q2001/prod_uis.htm).
- [14] G. Pingali, A. Opalach, Y. Jean, "Ball tracking and virtual replays for innovative tennis broadcasts," in: *Proc. of the 15th Int. Conf. on Pattern Recognition*, vol. 4, 2000, pp. 152-156.
- [15] J. R. Wang, N. Parameswaran, "Detecting tactics patterns for archiving tennis video clips," in: *Proc. of IEEE 6th Int. Symp. on Multimedia Software Engineering*, 2004, pp. 186-192.
- [16] D. Y. Chen, M. H. Hsiao, and S. Y. Lee, "Automatic closed caption detection and filtering in MPEG videos for video structuring," *Journal of Information Science and Engineering*, vol. 22, issue 5, 2006, pp. 1145-1162.
- [17] A. Ekin, A. M. Tekalp, R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Trans. on Image Processing*, vol. 12, 2003, pp. 796-807.
- [18] W. J. Heng and K. N. Ngan, "Shot boundary refinement for long transition in digital video sequence," *IEEE Trans. on Multimedia*, vol. 4, issue 4, 2002, pp. 434 – 445.
- [19] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?" *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, issue 2, 2002, pp. 90-105.
- [20] D. Y. Chen, S. Y. Lee and H. Y. Mark Liao, "Robust video sequence retrieval using a novel object-based T2D-histogram descriptor," *Journal of Visual Communication and Image Representation*, vol. 16, issue 2, 2005, pp. 212-232.
- [21] S. Y. Lee, J. L. Lian, D. Y. Chen, "Video summary and browsing based on story-unit for video-on-demand service," in: *Proc. Int. Conf. on Information, Communications and Signal Processing*, 2001.
- [22] A. Ekin and A.M. Tekalp, "Robust dominant color region detection and color-based applications for sports video," in: *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2003, pp. 21-24.
- [23] G. Millerson, *The technique of television production*, 12th ed., New York: Focal, March 1990.
- [24] A. M. Ferman and A. M. Tekalp, "A fuzzy framework for unsupervised video content characterization and shot classification," *Journal of Electronic Imaging*, vol. 10, no. 4, 2001, pp. 917–929.



- [25] D. Farin, S. Krabbe, P. H. N. de With and W. Effelsberg, "Robust camera calibration for sport videos using court models," *SPIE Storage and Retrieval Methods and Applications for Multimedia*, 2004, vol.5307, pp. 80-91.
- [26] D. Farin, J. Han and P. H. N. de With, "Fast camera calibration for the analysis of sport sequences," in: *Proc. IEEE Int. Conf. on Multimedia and Expo 2005*, pp.-, 2005.
- [27] B. Jähne, *Digital Image Processing*, Springer Verlag, 2002.
- [28] J. W. Davis, A. F. Bobick, "The recognition of human movement using temporal templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23 (3), 2001, pp. 257-267.
- [29] L. Y. Duan, M. Xu, Q. Tian and C. S. Xu, "Mean shift based video segment representation and applications to replay detection," in: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2004, pp. V- 709-712.
- [30] L. Y. Duan, M. Xu, T. S. Chua, Q. Tian, and C. S. Xu, "A mid-level representation framework for semantic sports video analysis," in: *Proc. 11th ACM Int. Conf. on Multimedia*, 2003, pp. 33- 44.
- [31] M. C. Tien, H. T. Chen, Y. W. Chen, M. H. Hsiao and S. Y. Lee, "Shot classification of basketball videos and its applications in shooting position extraction," in: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. I-1085-I-1088, 2007.
- [32] G. Zhu, Q. Huang, C. Xu, L. Xing, W. Gao, and H. Yao, "Human behavior analysis for highlight ranking in broadcast racket sports video," *IEEE Trans. on Multimedia*, vol. 9, no.6, pp.1167-1182, Oct. 2007.
- [33] H. T. Chen, M. H. Hsiao, H. S. Chen, W. J. Tsai and S. Y. Lee, "A baseball exploration system using spatial pattern recognition," in *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 3522-3525, 2008
- [34] X. Yu, H. W. Leong, C. Xu, and Q. Tian, "Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video," *IEEE Trans. on Multimedia*, vol. 8, no.6, pp.1164-1178, Dec. 2006.
- [35] G. Zhu, Q. Huang, C. Xu, Y. Rui, S. Jiang, W. Gao and H. Yao, "Trajectory based event tactics analysis in broadcast sports video," in: *Proc. 15th ACM Int. Conf. on Multimedia*, pp.58-67, 2007.
- [36] X. Yu, N. Jiang, L. F. Cheong, H. W. Leong and X. Yan, "Automatic camera calibration of broadcast tennis video with applications to 3d virtual content insertion and ball detection and tracking," *Computer Vision and Image Understanding*, 2008.

- [37] H. T. Chen, M. C. Tien, Y. W. Chen, W. J. Tsai and S. Y. Lee, "Physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video," *Journal of Visual Communication and Image Representation*, vol. 20, pp. 204-216, 2009 .
- [38] T. Watanabe, M. Haseyama, and H. Kitajima, "A soccer field tracking method with wire frame model from TV images," in: *Proc. IEEE Int. Conf. on Image Processing*, vol. 3, pp. 1633-1636, 2004
- [39] X. Yu, N. Jiang and L. F. Cheong, "Accurate and stable camera calibration of broadcast tennis video," in: *Proc Int. IEEE Conf. Image Processing*, vol. 3, pp. 93-96, 2007.
- [40] A. Loui, J. Luo, S. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa, "Kodak 's consumer video benchmark data set: concept definition and annotation," in: *Proc. Int. Workshop on Multimedia information Retrieval*, pp.245-254, 2007.
- [41] R. Oami, A.B. Benitez, S.-F. Chang and N. Dimitrova, "Understanding and modeling user interests in consumer videos," in: *Proc. IEEE Int. Conf. on Multimedia and Expo*, vol.2, pp.1475-1478, 2004.
- [42] C. Forlines, K.A. Peker and A. Divakaran "Subjective assessment of consumer video summarization," in: *Proc. SPIE Int. Soc. Opt. Eng.* 6073, 60730J, 2006.
- [43] N. Owens, C. Harris, and C. Stennett, "Hawk-eye tennis system," in: *Proc. VIE 2003*, pp.182- 185, 2003.
- [44] T. Zhang and C. C. Jay Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 4, pp.441-457, May 2001.
- [45] Y. Wang, Z. Liu, and J.C. Huang, "Multimedia content analysis using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, Issue 6, pp.12-36, Nov. 2000.
- [46] M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, and Q. Tian, "Creating audio keywords for event detection in soccer video," in: *Proc. IEEE Int. Conf. on Multimedia and Expo*, vol.2, pp.II- 281-284, 2003.
- [47] M. Xu, L. Duan, L. Chia, and C. Xu, "Audio keyword generation for sports video analysis," in: *Proc. 12th Annual ACM Int. Conf. on Multimedia*, pp. 758-759, 2004.
- [48] B. Zhang, W. Dou, and L. Chen, "Ball Hit Detection in Table Tennis Games Based on Audio Analysis," in: *Proc. 18th IEEE Int. Conf. on Pattern Recognition*, vol. 3, pp. 220-223, 2006.

- [49] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, *Cambridge University Press 2003*(2nd edition), UK.
- [50] W. Hua, M. Han, and Y. Gong, “Baseball scene classification using multimedia features,” in: *Proc. of IEEE Int. Conf. on Multimedia and Expo 2002*, vol. 1, pp. 821-824, 2002
- [51] Y. Rui, A. Gupta, and A. Acero, “Automatically extracting highlights for TV baseball programs,” in: *Proc. of the 8th ACM Int. Conf. on Multimedia*, pp. 105-115, 2000.
- [52] W. T. Chu, C. W. Wang, and J. L. Wu, “Extraction of baseball trajectory and physics-based validation for single-view baseball video sequences,” in: *Proc. of IEEE Int. Conf. on Multimedia and Expo 2006*, pp. 1813-1816, 2006.
- [53] S. C. Pei and F. Chen, “Semantic scenes detection and classification in sports videos,” in: *Proc. of IPPR Conf. on Computer Vision, Graphics and Image Processing (CVGIP)*, 2003, pp. 210-217.
- [54] D. Zhong and S. F. Chang, “Structure analysis of sports video using domain models,” in: *Proc. of IEEE Int. Conf. on Multimedia and Expo 2001*, pp. 713-716, 2001.
- [55] K. Nummiaro, E. Koller-Meier, and L. V. Gool, “An adaptive color-based particle filter,” *Image and Vision Computing*, vol. 21, pp. 99-110, 2003.
- [56] M. Kumano, Y. Ariki, K. Tsukada, S. Hamaguchi and H. Kiyose, “Automatic extraction of PC scenes based on feature mining for a real time delivery system of baseball highlight scenes,” in: *Proc. IEEE Int. Conf. on Multimedia and Expo*, vol. 1, pp. 277-280, 2004.
- [57] [http://en.wikipedia.org/wiki/Strike\\_zone](http://en.wikipedia.org/wiki/Strike_zone)
- [58] [http://en.wikipedia.org/wiki/Category:Baseball\\_terminology](http://en.wikipedia.org/wiki/Category:Baseball_terminology)
- [59] [http://mlb.mlb.com/mlb/official\\_info/umpires/strike\\_zone.jsp](http://mlb.mlb.com/mlb/official_info/umpires/strike_zone.jsp)
- [60] C. Kim, J. N. Hwang, “Fast and automatic video object segmentation and tracking for content-based applications,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, pp.122-129, 2002.
- [61] [http://mlb.mlb.com/mlb/official\\_info/official\\_rules/objectives\\_1.jsp](http://mlb.mlb.com/mlb/official_info/official_rules/objectives_1.jsp)
- [62] Z. Xiong, R. Radhakrishnan, A. Divakaran, T. S. Huang, “Highlights extraction from sports video based on an audio-visual marker detection framework,” in: *Proc. of IEEE Int. Conf. on Multimedia and Expo 2005*, pp. 29-32, 2005.
- [63] G. Welch and G. Bishop, “An introduction to the Kalman filter,” Technical Report no. TR

95-041, University of North Carolina at Capel Hill, 2004.

- [64] L. Piegl, W. Tiller, "The NURBS Book," Springer, ISBN 3-540-61545-8, 1997.
- [65] D. Hoffman, W. Richards, "Parts of recognition," *Cognition*, vol.18, pp. 65-96, 1984.
- [66] K. Siddiqi and B. B. Kimia, "Parts of visual form: computational aspects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 239-251, 1995.
- [67] X. Tong, H. Lu, Q. Liu and H. Jin, "Replay detection in broadcasting sports video," in: *Proc. of 3rd IEEE Int. Conf. on Image and Graphics*, pp. 337-304, 2004.
- [68] H. Shum and T. Komura, "Tracking the Translational and Rotational Movement of the Ball Using High-Speed Camera Movies," in: *Proc. of the IEEE Int. Conf. on Image Processing 2005*, vol. 3, pp.1084-1087, 2005.
- [69] P. Chang, M. Han and Y. Gong, "Extract highlights from baseball game video with hidden Markov models," in: *Proc. of the IEEE Int. Conf. on Image Processing 2002*, vol. 1, pp.609-612, 2002.
- [70] R. C. Nelson, "Finding Line Segments by Stick Growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.16, pp. 519-523,1994.
- [71] W. T. Chu and J. L. Wu "Explicit semantic events detection and development of realistic applications for broadcasting baseball videos," *Multimedia Tools and Applications*, vol. 38, No. 1, pp. 27-50, 2008.
- [72] C. H. Liang, W. .T Chu, J. H. Kuo, J. L. Wu, W. H. Cheng, "Baseball event detection using game-specific feature sets and rules," in: *Proc. IEEE Int. Symp. on Circuits and System2005*, vol. 4, pp. 3829-3832, 2005.
- [72] W. T Chu, J. L. Wu, "Integration of rule-based and model-based decision methods for baseball event detection," In: *Proc. IEEE Int. Conf. on Multimedia and Expo 2005*, pp.-, 2005.
- [73] S. F. Chang, "The holy grail of content-based media analysis," *IEEE Multimedia*, vol. 9, no.2, pp.6-10, 2002.
- [74] A. Kokaram, N. Rea, R. Dahyot, M. Tekalp, P. Bouthemy, P. Gros, and I. Sezan, "Browsing sports video: trends in sports-related indexing and retrieval work," *IEEE Signal Processing Magazine*, vol. 23, pp. 47-58, Mar. 2006.