Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

# Physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video

Hua-Tsung Chen [a,*], Ming-Chun Tien [b], Yi-Wen Chen [a], Wen-Jiin Tsai [a], Suh-Yin Lee [a,*]

[a] *College of Computer Science, National Chiao-Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan*
[b] *Graduate Institute of Networking and Multimedia, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan*

ARTICLE INFO

*Article history:*
Received 26 June 2008
Accepted 25 November 2008
Available online 10 December 2008

*Keywords:*
Shot classification
Object tracking
Sports video analysis
Camera calibration
Semantic analysis
Multimedia system
3D trajectory reconstruction

ABSTRACT

The demand for computer-assisted game study in sports is growing dramatically. This paper 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 videos 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 videos show promising results. We believe the proposed system will greatly assist intelligence collection and statistics analysis in basketball games.

## 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 easier. 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 videos. 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 videos based on the syntax and content characteristics of soccer videos.

*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 videos. 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

[*] Corresponding authors. Fax: +886 3 5721490 (H.-T. Chen).
*E-mail addresses:* huatsung@cs.nctu.edu.tw (H.-T. Chen), sylee@csie.nctu.edu.tw (S.-Y. Lee).

1047-3203/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2008.11.008

based on Kalman filter. In our previous work [9–10], the physical characteristics of the ball motion is utilized to extract the ball trajectory in sports videos. Furthermore, the extracted ball trajectory can be applied to volleyball set type recognition and baseball pitch evaluation. 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 paper, we work for the challenge of ball tracking and 3D trajectory reconstruction in broadcast basketball videos 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. 1, where each shooting location is marked as an **O** (score) or **X** (miss). The statistical graph for shooting locations not only gives 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. 1, can be generated automatically.

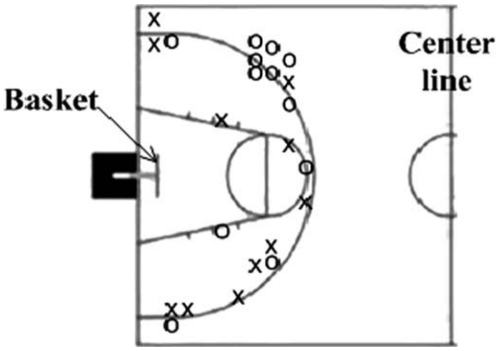

**Fig. 1.** Statistical graph of shooting locations.

The rest of this paper is organized as follows. Section 2 introduces the overview of the proposed system. Sections 3–5 present the processes of court shot retrieval, camera calibration and 2D shooting trajectory extraction, respectively. Section 6 elaborates on 3D trajectory mapping and shooting location estimation. Experimental results and discussions are presented in Section 7. Finally, Section 8 concludes this paper.

## 2. Overview of the proposed system

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. Basketball videos contain 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 to 2D, 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. Having the reconstructed 3D information, the shooting locations can be estimated more

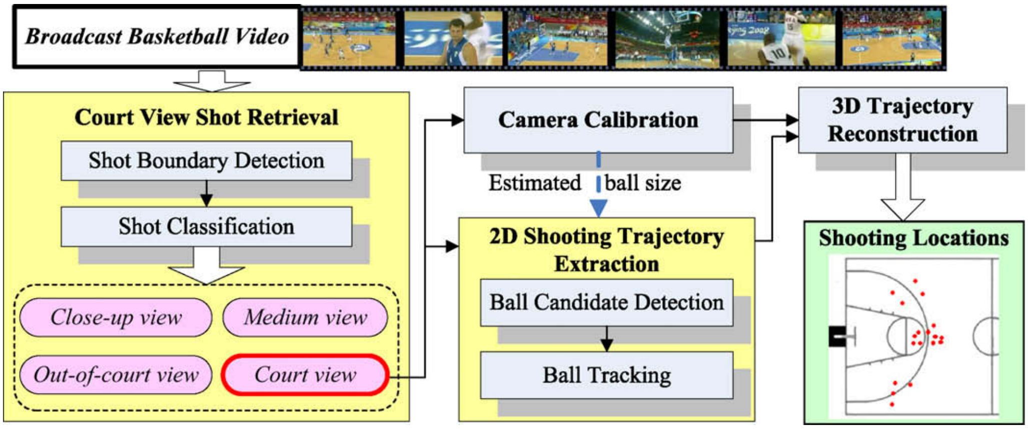

**Fig. 2.** Flowchart of the proposed system for ball tracking and 3D trajectory reconstruction in basketball videos.

accurately from 3D trajectories than from 2D trajectories, which lack the z-coordinate (height) of ball.

The major contribution of this paper 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.

## 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 our previously proposed 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 the 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 shots, (2) medium shots, and (3) close-up or out-of-court shots (abbreviated to C/O shots). A *court shot* displays the global view of the court, which can present complete shooting trajectories, as shown in Fig. 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. 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. 3(e), and an *out-of-court shot* presents the audience, coach, or other places out of the court, as shown in Fig. 3(f).

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. (1), would vary in different view shots.

$$R = \#\text{court-colored pixels}/\#\text{pixels in a frame} \tag{1}$$

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. 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. 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) is utilized on the basis of the following observation.

$$R_{5U8}: \text{the } R \text{ value in the union of region 5 and region 8} \tag{2}$$

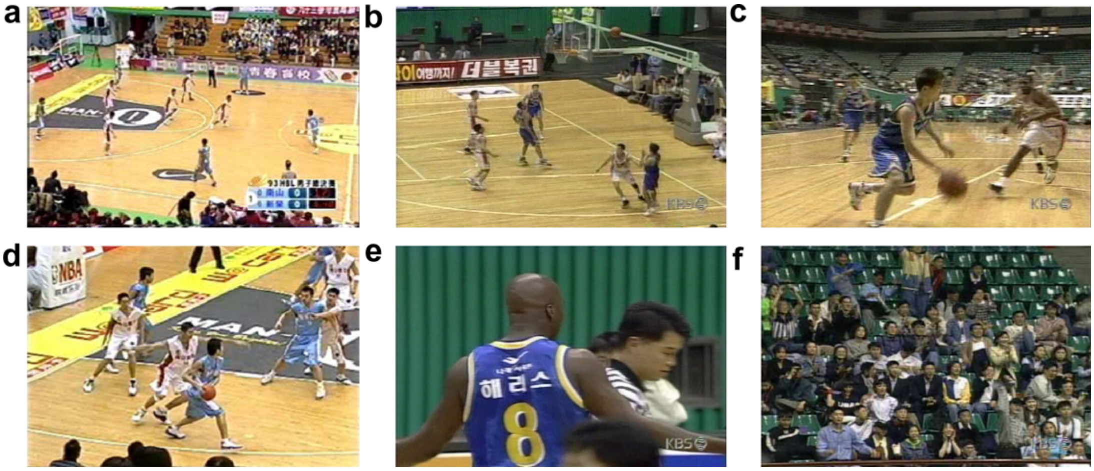

**Fig. 3.** Examples of shot types in a basketball game. (a) Court shot; (b) court shot; (c) medium shot; (d) medium shot; (e) close-up shot and (f) out-of-court shot.

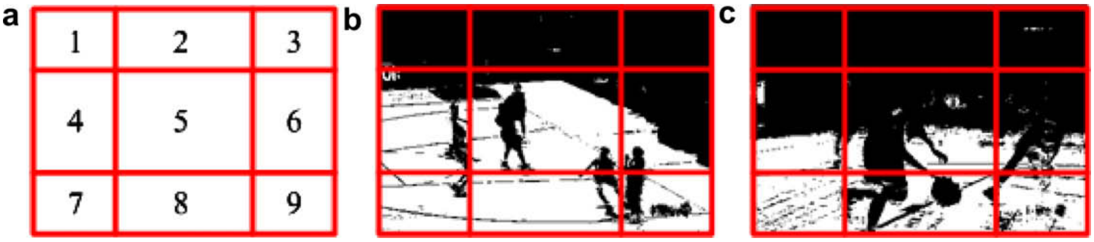

**Fig. 4.** Examples of Golden Section spatial composition. (a) Frame regions; (b) court view and (c) medium view.

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 payers 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.

## 4. Camera calibration

Camera calibration is an essential task to provide geometric transformation mapping the positions of the ball and players in the video frames to 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. (3):

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

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 the image. 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. 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. 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].

### 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. 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. 8(b).

To improve the accuracy and efficiency of the subsequent Hough transform for line detection and court model fitting, we

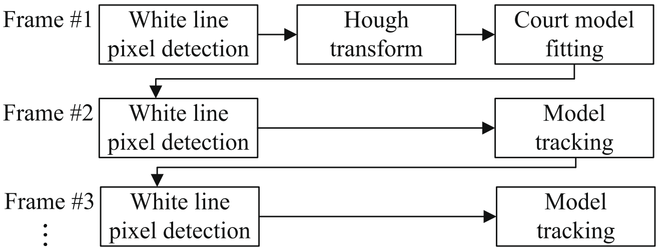

**Fig. 5.** Flowchart of camera calibration.

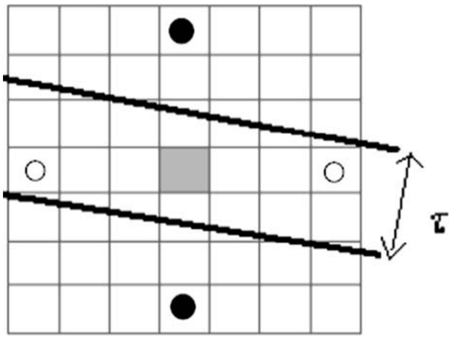

**Fig. 7.** Illustration of part of an image containing a white line.

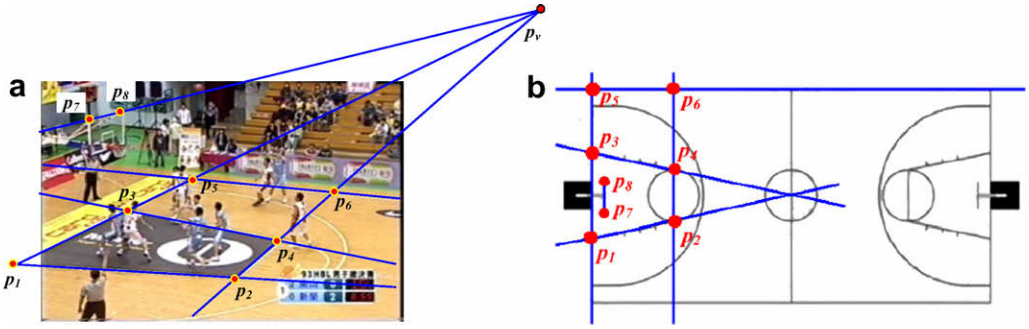

**Fig. 6.** Point-correspondence between the 2D frame and the basketball court model. (a) Court image and (b) court model.

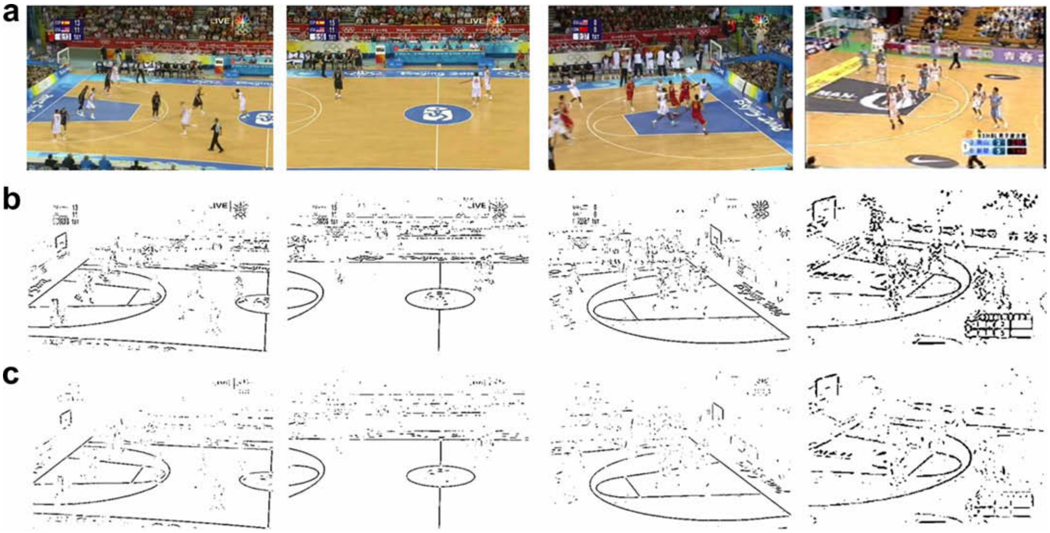

**Fig. 8.** Sample results of white line pixel detection. (a) Original frame; (b) without line-structure constraint and (c) with line-structure constraint.

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 2*b* + 1 (we use b = 2) around each candidate pixel ($p_x$, $p_y$), as defined in Eq. (4), is used to recognize texture regions.

$$s = \sum_{x=p_x-b}^{p_x+b} \sum_{y=p_y-b}^{p_y+b} \nabla Y(x,y)(\nabla Y(x,y))^{\mathrm{T}} \tag{4}$$

Depending on the two eigenvalues of matrix S, called $\lambda_1$ and $\lambda_2$ ($\lambda_1 \geqslant \lambda_2$), the texture can be classified into *textured* ($\lambda_1$, $\lambda_2$ are large), *linear* ($\lambda_1$ » $\lambda_2$) 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 $\lambda_1 > \leftarrow\alpha\lambda_2$ ($\alpha = 4$ in our system). Fig. 8 demonstrates sample results of white line pixel detection. The original frames are presented in Fig. 8(a). In Fig. 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. 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.

### 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. 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. 6) are selected because the light condition makes the white line pixels of the backboard top-border easy to detect in frames. Fig. 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 fame 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. 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-boarder, as shown in Fig. 9(c).

### 4.3. Computation of camera calibration parameters

Multiplying out the linear system in Eq. (3), we obtain two equations, Eqs. (5) and (6), for *each corresponding point*–the point

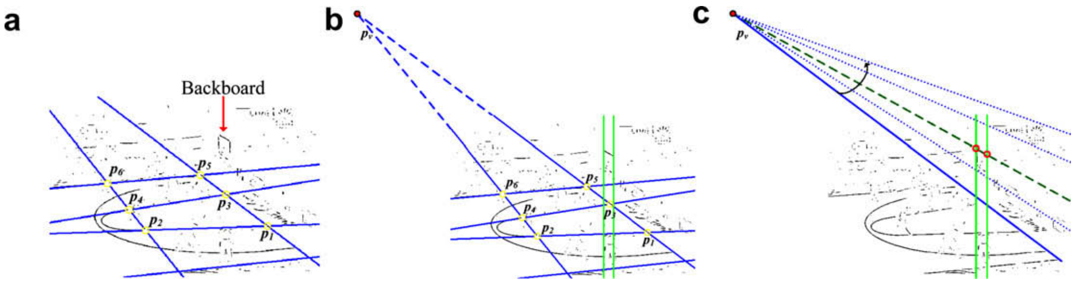

**Fig. 9.** Detection of backboard top-border. (a) Detected court lines; (b) computing vanishing point and (c) searching backboard top-border.

whose coordinate is both known in the 3D court model $(x, y, z)$ and in the frame $(u', v')$.

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

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

To compute the calibration parameters $c_{ij}$, we set up a linear system **AC** = **B** as Eq. (7) from Eqs. (5) and (6).

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u'_1x_1 & -u'_1y_1 & -u'_1z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v'_1x_1 & -v'_1y_1 & -v'_1z_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u'_2x_2 & -u'_2y_2 & -u'_2z_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v'_2x_2 & -v'_2y_2 & -v'_2z_2 \\ & & & & & & \vdots & & & & \\ x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & -u'_Nx_N & -u'_Ny_N & -u'_Nz_N \\ 0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & -v'_Nx_N & -v'_Ny_N & -v'_Nz_N \end{bmatrix}_{2N\times 11} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ \vdots \\ c_{31} \\ c_{32} \\ c_{33} \end{bmatrix}_{11\times 1} = \begin{bmatrix} u'_1 \\ v'_1 \\ u'_2 \\ v'_2 \\ \vdots \\ u'_N \\ v'_N \end{bmatrix}_{2N\times 1} \tag{7}$$

$N$ 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 **C**, we can over-determine **A** and find a least squares fitting for **C** with a pseudo-inverse solution:

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

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

## 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.

### 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 color histograms including RGB, YCbCr and HSI color spaces, as shown in Fig. 10. Due to the discriminability, the Hue value in HSI space is selected as the color feature and the ball color range $[H_a, H_b]$ is set. We compute the average H value for each 4 × 4 block in frames and discard the moving pixels in the blocks of which the average H 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. 11 and 11(a) is the original frame and Fig. 11(b) shows the moving pixels detected by frame difference. The extracted ball pixels after morphological operations are presented in Fig. 11(c).

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. (9):

$$(D_{frm}/D_{real}) = (d/D), \quad D_{frm} = D_{real}(d/D) \tag{9}$$

where $D_{real}$ is the diameter of a real basketball (≈24 cm), $d$ and $D$ are the in-frame length and the real-world length of a corresponding line segment, respectively. To compute the ratio $(d/D)$, we select the two points closest to the frame center from the six court line intersections and calculate the in-frame distance $d$ of the selected two points. Since the distance of the two points in real court $D$ is specified in the basketball rules, the ratio $(d/D)$ 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 ex-

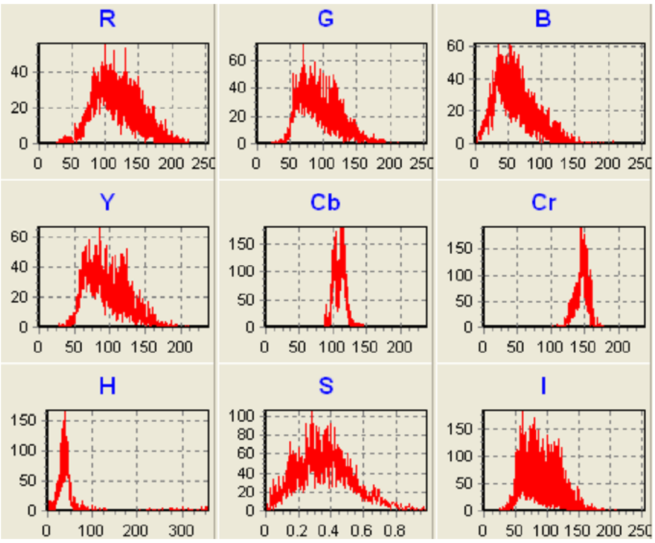

**Fig. 10.** Color histograms of 30 manually segmented basketball images.

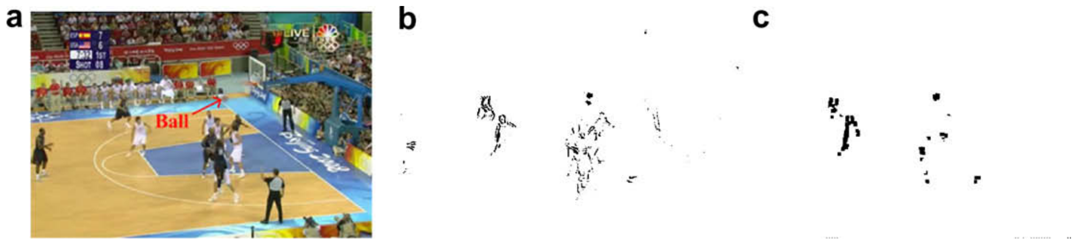

**Fig. 11.** Illustration of ball pixel detection. (a) Source frame; (b) moving pixels and (c) extracted ball pixels.

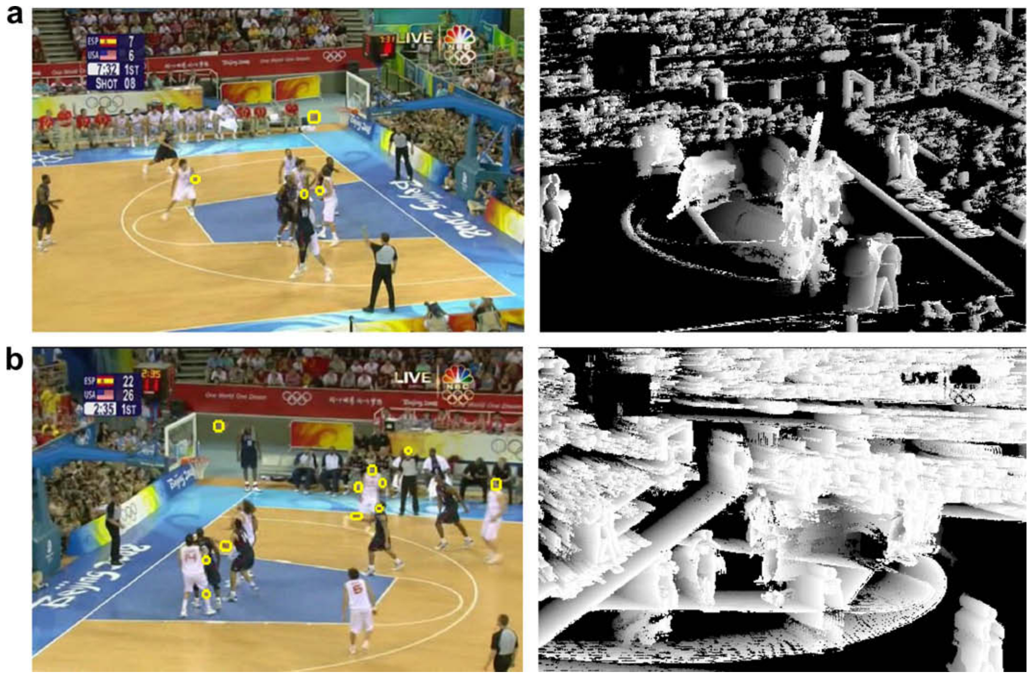

**Fig. 12.** Left: detected ball candidates, marked as yellow circles. Right: motion history image to present the camera motion. (a) Fewer ball candidates produced if the camera motion is small. (b) Fewer ball candidates produced if the camera motion is small.

ist. 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 the rapid camera transition from one half-court to the other, and there are not too many ball candidates, as shown in Fig. 12, where the left image shows the detected ball candidates, marked as the yellow circles, and the right image presents the camera motion using motion history image (MHI, please refer to [28]), generated from 45 consecutive frames. When a shooting event occurs, one of the backboards should be captured in the frames. During the 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.

### 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.

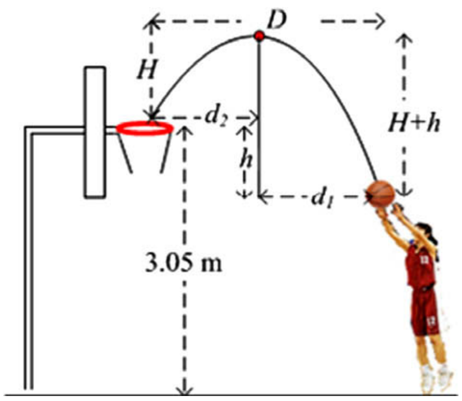

**Fig. 13.** Diagram of a long shoot.

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. 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 Eqs. (10) and (11), respectively:

$$H + h = gt_1^2/2, \quad t_1 = [2(H + h)/g]^{1/2} \tag{10}$$

$$H = gt_2^2/2, \quad t_2 = (2H/g)^{1/2} \tag{11}$$

where $g$ is the gravity acceleration (9.8 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 $Vv$ of the ball in the trajectory should be $Vv = g\ t_1$ and the horizontal velocity $Vh$ can be calculated as $Vh = D/(t_1 + t_2)$, where $D$ is the distance from the shooter to the basket center. With the vertical and horizontal velocities, the ball velocity $Vb$ can be derived as Eq. (12):

$$Vb = (Vh^2 + Vv^2)^{1/2} \tag{12}$$

$Vb$ value increases as $D$ 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 (6.25 m), which is almost the longest horizontal distance from the shooter to the basket. To cover all cases, we set $D = 7$ 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, $h \leqslant 1.2$. 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 $Vb$ computed using Eqs. (10)–(12) for $H$ varying between 1 and 4 are plotted in Fig. 14, showing the reasonable values of $Vb$. It can be observed that, when $H = 4$ m and $h = 1.2$ m, we have the maximum value of $Vb$ ($\approx$10.8 m/s). Thus, we set the velocity constraint (upper limit) as $Vb \approx 10.8$ m/s $\approx 36$ cm/frm. Finally, similar to Eq. (9), the in-frame velocity constraint $Vc$ can be proportionally estimated by applying pinhole camera imaging principle as Eq. (13):

$$(Vc/Vb) = (d/D), \quad Vc = Vb(d/D) \tag{13}$$

The goal of ball velocity constraint is to determine the search range for ball tracking. To avoid 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.

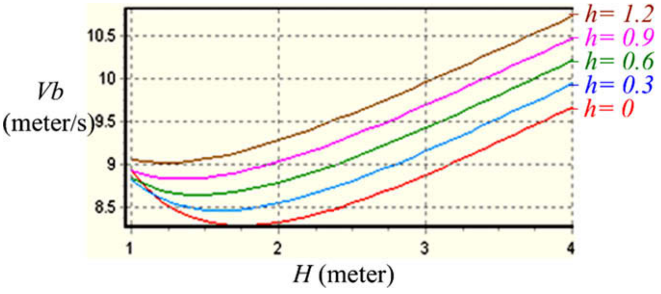

**Fig. 14.** Relation between *Vb* and *H*.

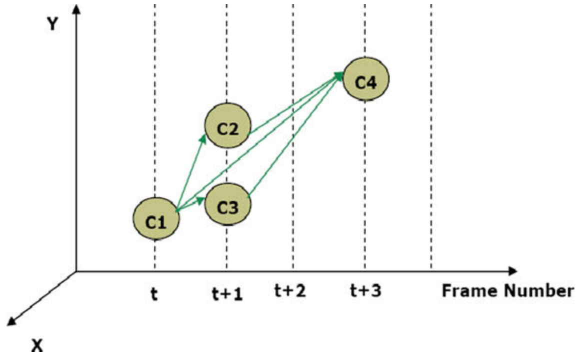

**Fig. 15.** Illustration of ball tracking process.

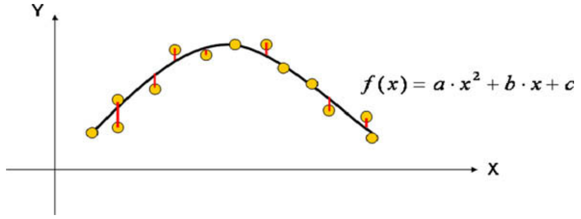

**Fig. 16.** Illustration of the best-fitting function.

Fig 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. (14):

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

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 $t_{i \to j}$ 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.

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. 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).

## 6. 3D trajectory mapping 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. (3). Furthermore, the ball motion should fit the physical properties, so we can model the 3D trajectory as:

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

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. (3) by Eq. (15), we obtain:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} x_0 + v_x t \\ y_0 + v_y t \\ z_0 + v_z t + \frac{1}{2} g t^2 \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \begin{matrix} \mathit{where} \\ u' = \frac{u}{w} \\ v' = \frac{v}{w} \end{matrix} \tag{16}$$

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

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

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

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 $\mathbf{D}_{2N\mathrm{x}6}$ $\mathbf{E}_{6\mathrm{x}1} = \mathbf{F}_{2N\mathrm{x}1}$, as Eq. (19), from Eq. (17) and Eq. (18) to compute the six unknowns ($x_0$, $V_x$, $y_0$, $V_y$, $z_0$, $V_z$) of the parabolic (or ballistic trajectory), 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. (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 ($x_0$, $V_x$, $y_0$, $V_y$, $z_0$, $V_z$).

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 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.

## 7. Experimental results and discussions

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.

### 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 $R \leqslant T_{c/o}$ 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 ($R_{5\cup 8} \leqslant T_{court}$). 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. 17, which is constructed from 120 shots with shot classes already known. Starting with roughly initialized threshold ($T_{c/o}$ = average $R$ 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

$$\begin{bmatrix} C_{11} - \boldsymbol{u}_1'c_{31} & C_{11}\boldsymbol{t}_1 - \boldsymbol{u}_1'c_{31}\boldsymbol{t}_1 & c_{12} - \boldsymbol{u}_1'c_{32} & c_{12}\boldsymbol{t}_1 - \boldsymbol{u}_1'c_{32}\boldsymbol{t}_1 & c_{13} - \boldsymbol{u}_1'c_{33} & c_{13}\boldsymbol{t}_1 - \boldsymbol{u}_1'c_{33}\boldsymbol{t}_1 \\ C_{21} - v_1'c_{31} & C_{21}\boldsymbol{t}_1 - v_1'c_{31}\boldsymbol{t}_1 & c_{22} - v_1'c_{32} & c_{22}\boldsymbol{t}_1 - v_1'c_{32}\boldsymbol{t}_1 & c_{23} - v_1'c_{33} & c_{23}\boldsymbol{t}_1 - v_1'c_{33}\boldsymbol{t}_1 \\ C_{11} - \boldsymbol{u}_2'c_{31} & C_{11}\boldsymbol{t}_2 - \boldsymbol{u}_2'c_{31}\boldsymbol{t}_2 & c_{12} - \boldsymbol{u}_2'c_{32} & c_{12}\boldsymbol{t}_2 - \boldsymbol{u}_2'c_{32}\boldsymbol{t}_2 & c_{13} - \boldsymbol{u}_2'c_{33} & c_{13}\boldsymbol{t}_2 - \boldsymbol{u}_2'c_{33}\boldsymbol{t}_2 \\ C_{21} - v_2'c_{31} & C_{21}\boldsymbol{t}_2 - v_2'c_{21}\boldsymbol{t}_2 & c_{22} - v_2'c_{32} & c_{22}\boldsymbol{t}_2 - v_2'c_{32}\boldsymbol{t}_2 & c_{23} - v_2'c_{33} & c_{23}\boldsymbol{t}_2 - v_2'c_{33}\boldsymbol{t}_2 \\ & & & \vdots & & \\ C_{11} - \boldsymbol{u}_N'c_{31} & C_{11}\boldsymbol{t}_1 - \boldsymbol{u}_N'c_{31}\boldsymbol{t}_1 & c_{12} - \boldsymbol{u}_N'c_{32} & c_{12}\boldsymbol{t}_1 - \boldsymbol{u}_N'c_{32}\boldsymbol{t}_1 & c_{13} - \boldsymbol{u}_N'c_{33} & c_{13}\boldsymbol{t}_1 - \boldsymbol{u}_N'c_{33}\boldsymbol{t}_1 \\ C_{21} - v_N'c_{31} & C_{21}\boldsymbol{t}_1 - v_N'c_{31}\boldsymbol{t}_1 & c_{22} - v_N'c_{32} & c_{22}\boldsymbol{t}_1 - v_N'c_{32}\boldsymbol{t}_1 & c_{23} - v_N'c_{33} & c_{23}\boldsymbol{t}_1 - v_1'c_{33}\boldsymbol{t}_1 \\ & & & \mathbf{D} & & \end{bmatrix}_{2N\times 6} \begin{bmatrix} X_0 \\ V_x \\ Y_0 \\ V_y \\ z_0 \\ V_z \\ \mathbf{E} \end{bmatrix}_{6\times 1}$$

$$= \begin{bmatrix} u_1'(c_{33}g\boldsymbol{t}_1^2/2 + 1) - (c_{13}g\boldsymbol{t}_1^2/2 + c_{14}) \\ v_1'(c_{33}g\boldsymbol{t}_1^2/2 + 1) - (c_{23}g\boldsymbol{t}_1^2/2 + c_{24}) \\ u_2'(c_{33}g\boldsymbol{t}_1^2/2 + 1) - (c_{13}g\boldsymbol{t}_2^2/2 + c_{14}) \\ v_2'(c_{33}g\boldsymbol{t}_2^2/2 + 1) - (c_{23}g\boldsymbol{t}_2^2/2 + c_{24}) \\ \vdots \\ u_N'(c_{33}g\boldsymbol{t}_1^2/2 + 1) - (c_{13}g\boldsymbol{t}_N^2/2 + c_{14}) \\ v_N'(c_{33}g\boldsymbol{t}_2^2/2 + 1) - (c_{23}g\boldsymbol{t}_N^2/2 + c_{24}) \\ \mathbf{F} \end{bmatrix}_{2N\times 1} \tag{19}$$

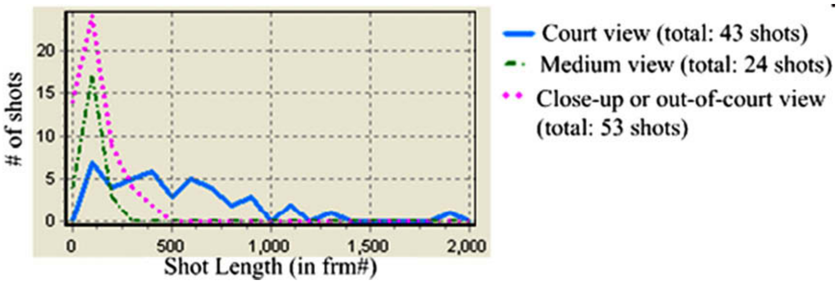

**Fig. 17.** Statistical data of the shot lengths for different shot classes.

be ⩾70%. Then, $T_{c/o}$ is re-adjusted to the average $R$ of the frames excluding the frames of court shots.

For ball candidate detection, the ball color range [Ha, Hb] is determined statistically. With the Hue histogram constructed from 30 ball images manually segmented out of different basketball sequences, as shown in Fig. 10, the range [Ha, Hb] 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 [Ha, Hb].

### 7.2. Performance of 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 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 (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 videos.

**Table 1**
Performance of 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**
Performance of 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 |

### 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. 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.

### 7.4. Performance of ball tracking and shooting location estimation

The performance study of ball tracking and shooting location estimation are 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

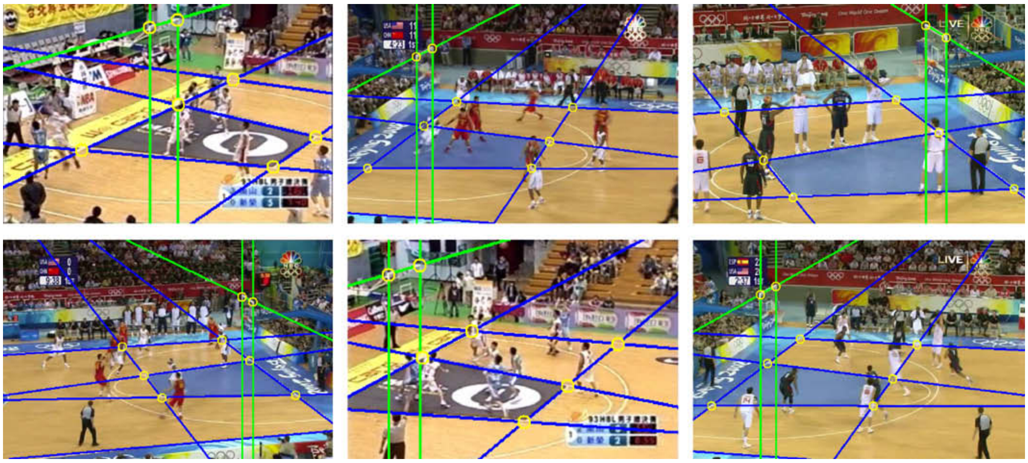

**Fig. 18.** Detection of court lines and corresponding points (marked with yellow circles).

**Table 3**
Performance of ball 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 |

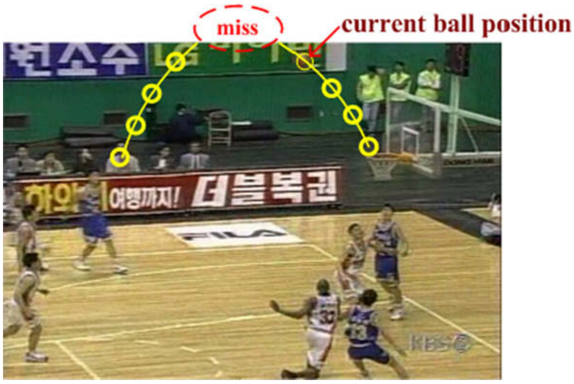

**Fig. 19.** Example of a shooting trajectory being separated. For legibility, not all of the ball candidates are drawn.

**Table 4**
Performance of 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 |

Table 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. 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 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. 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 presenting the camera motion, we also mark the positions of corresponding points over frames with red squares.

In addition to the case of trajectory split (as mentioned above), misdetection of the court lines or the backboard top-border is another cause of the mistakes in shooting location estimation. Fig. 21 presents an example. As shown in Fig. 21(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. 21(b). However, the court lines and the backboard top-border are detected appropriately in most frames and overall, we achieve quite encouraging results.

### 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]. To compare the effectiveness and efficiency of the Kalman filter-based algorithm (KF) with those of

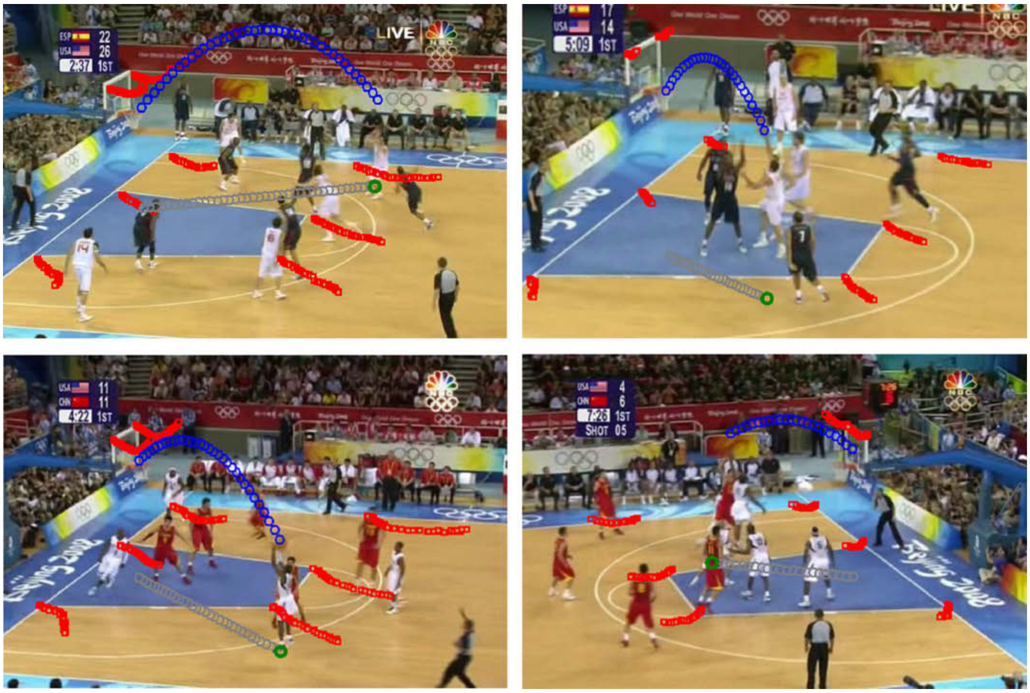

**Fig. 20.** Demonstration of shooting location estimation. In each image, the blue circles are the ball positions over frames, the green circle represents the estimated shooting location and the red squares show the movement of corresponding points due to the camera motion.

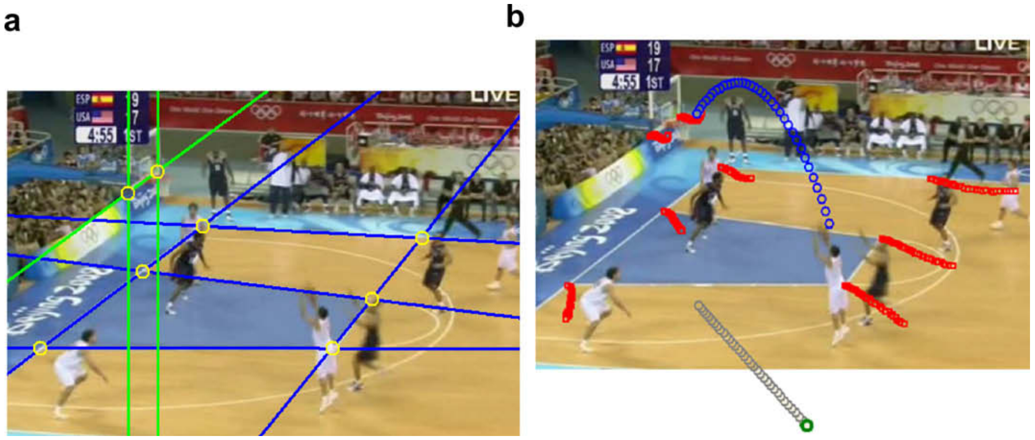

**Fig. 21.** Deviation of shooting location estimation caused by the misdetection of backboard top-border. (a) Corresponding points extraction (b) shooting location estimation.

**Table 5**
Comparison between the proposed physics-based method and the Kalman filer-based method. (#PT: number of potential trajectories).

| Ball tracking | 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 |

the proposed physics-based algorithm (PB), we use the precision, recall and the number of potential trajectories (#PT) as criteria. As reported in Table 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. 22. Without the vertical information, the estimated shooting locations in Fig. 22(c) is far from the actual ones as in Fig. 22(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.

## 8. Conclusions

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 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 below. The first is to utilize the domain knowledge of court specification for camera calibration. This enables the computation of 3D-to-2D transformation from 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 to be reconstructed. The technical ideas presented in this paper can also be applied to other sports, such as volleyball, baseball, etc. To the best of our knowledge, the trajectory-based

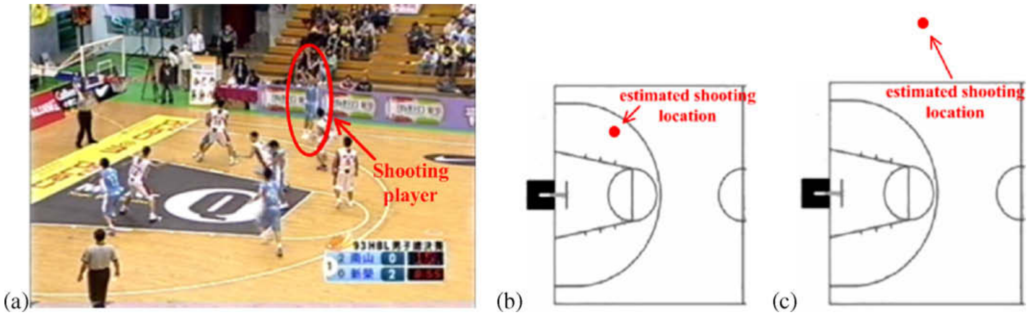

**Fig. 22.** Comparison of shooting location estimation with/without vertical (height) information: (a) original shooting location in the frame; (b) estimated shooting location with vertical information and (c) estimated shooting location without vertical information.

application of shooting location estimation in basketball is first proposed in our paper. The experiments show encouraging results on broadcast basketball videos.

Currently, we are exploring appropriate physical motion models for 3D ball trajectory reconstruction in other sports. It is our belief that the preliminary work presented in this paper will lead to satisfactory solutions for automatic intelligence collection in various kinds of sports games.

## Acknowledgment

The research is partially supported by the National Science Council of Taiwan, R.O.C, under the grant No. NSC 95-2221-E-009-076-MY3 and partially supported by Lee and MTI center for Networking Research at National Chiao Tung University, Taiwan.

## References

[1] L.Y. Duan, M. Xu, Q. Tian, C.S. Xu, J.S. Jin, A unified framework for semantic shot classification in sports video, IEEE Trans. Multimedia 7 (2005) 1066–1083.
[2] H. Lu, Y.P. Tan, Unsupervised clustering of dominant scenes in sports video, Pattern Recogn. Lett. 24 (15) (2003) 2651–2662.
[3] T. Mochizuki, M. Tadenuma, N. Yagi, Baseball video indexing using patternization of scenes and hidden Markov model, Proc. IEEE Int. Conf. Image Process. 3 (2005) 1212–1215.
[4] J. Assfalg, M. Bertini, C. Colombo, A.D. Bimbo, W. Nunziati, Semantic annotation of soccer videos: automatic highlights identification, Comput Vis Image Understand 92 (2-3) (2003) 285–305.
[5] Y. Gong, M. Han, W. Hua, W. Xu, Maximum entropy model-based baseball highlight detection and classification, Comput Vision Image Understand 96 (2) (2004) 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. Multimedia 8 (2006) 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 Recogn. Lett. 25 (7) (2004) 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. 11th ACM Int. Conf. Multimedia, 2003, pp. 11–20.
[9] H.T. Chen, H.S. Chen, S.Y. Lee, Physics-based ball tracking in volleyball videos with its applications to set type recognition and action detection, Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (2007) I-1097–I-1100.
[10] H.T. Chen, H.S. Chen, M.H. Hsiao, W.J. Tsai, S.Y. Lee, A trajectory-based ball tracking framework with visual enrichment for broadcast baseball videos, J. Inform. Sci. Eng. 24 (1) (2008) 143–157.
[11] A. Gueziec, Tracking pitches for broadcast television, Computer 35 (2002) 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.
[14] G. Pingali, A. Opalach, Y. Jean, Ball tracking and virtual replays for innovative tennis broadcasts, in Proc. 15th Int. Conf. Pattern Recogn., vol. 4, 2000, pp. 152–156.
[15] J.R. Wang, N. Parameswaran, Detecting tactics patterns for archiving tennis video clips, in: Proc. IEEE 6th Int. Symp. Multimedia Software Eng., 2004, pp. 186–192.
[16] D.Y. Chen, M.H. Hsiao, S.Y. Lee, Automatic closed caption detection and filtering in MPEG videos for video structuring, J. Inform. Sci. Eng. 22 (5) (2006) 1145–1162.
[17] A. Ekin, A.M. Tekalp, R. Mehrotra, Automatic soccer video analysis and summarization, IEEE Trans. Image Process. 12 (2003) 796–807.
[18] W.J. Heng, K.N. Ngan, Shot boundary refinement for long transition in digital video sequence, IEEE Trans. Multimedia 4 (4) (2002) 434–445.
[19] A. Hanjalic, Shot-boundary detection: unraveled and resolved?, IEEE Trans Circuits Syst. Video Technol. 12 (2) (2002) 90–105.
[20] D.Y. Chen, S.Y. Lee, H.Y. Mark Liao, Robust video sequence retrieval using a novel object-based T2D-histogram descriptor, J. Visual Commun. Image Represent. 16 (2) (2005) 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. Inform. Commun. Signal Process., 2001.
[22] A. Ekin, A.M. Tekalp, Robust dominant color region detection and color-based applications for sports video, Proc. IEEE Int. Conf. Image Process. 1 (2003) 21–24.
[23] G. Millerson, The Technique of Television Production, 12th ed., Focal, New York, 1990.
[24] A.M. Ferman, A.M. Tekalp, A fuzzy framework for unsupervised video content characterization shot classification, J. Electron. Imag. 10 (4) (2001) 917–929.
[25] D. Farin, S. Krabbe, P.H.N. de With, W. Effelsberg, Robust camera calibration for sport videos using court models, SPIE Storage Retrieval Meth. Appl. Multimedia 5307 (2004) 80–91.
[26] D. Farin, J. Han, P.H.N. de With, Fast camera calibration for the analysis of sport sequences, in: Proc. IEEE Int. Conf. Multimedia Expo, 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. Pattern Anal. Machine Intell. 23 (3) (2001) 257–267.
[29] L.Y. Duan, M. Xu, Q. Tian, C.S. Xu, Mean shift based video segment representation and applications to replay detection, Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (2004) V-709–V-712

.

**Hua-Tsung Chen** received his B.S. and M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Taiwan in 2001 and 2003, respectively. Currently, he is currently a Ph.D. candidate of Computer Science and Information Engineering in National Chiao Tung University, Taiwan. His research interests include computer vision, video signal processing, content-based video indexing and retrieval, multimedia information system and music signal processing.

**Ming-Chun Tien** received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively. She is currently pursuing the Ph.D. degree in the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan. Her research interest includes video processing, digital content analysis, and multimedia information retrieval.

**Yi-Wen Chen** is currently a Ph.D. candidate of Computer Science and Information Engineering in National Chiao Tung University, Taiwan. He received the B.S. and M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 2000 and 2002, respectively. He has been engaged in the research areas of computer vision and video/image compression.

**Wen-Jiin Tsai** received the B.S., M.S. and Ph.D. degrees in computer science and information engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1992, 1993 and 1997, respectively. She was a software manager at the DTV R&D Department of Zinwell Corporation, Hsinchu, Taiwan, during 1999–2005. She has been an Assistant Professor at the Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan, since February 2005. Her research interests include video compression, video transmission, digital TV, and content-based video retrieval.

**Suh-Yin Lee** received the B.S. degree in electrical engineering from National Chiao Tung University, Taiwan, in 1972, and the M.S. degree in computer science from University of Washington, Seattle, U.S.A., in 1975, and the Ph.D. degree in computer science form Institute of Electronics, National Chiao Tung University. Her research interests include content-based indexing and retrieval, distributed multimedia information system, mobile computing, and data mining.