

Ball tracking and 3D trajectory approximation with applications to tactics analysis from single-camera volleyball sequences

Hua-Tsung Chen · Wen-Jiin Tsai · Suh-Yin Lee ·
Jen-Yu Yu

Published online: 21 June 2011
© Springer Science+Business Media, LLC 2011

Abstract Providing computer-assisted tactics analysis in sports is a growing trend. This paper 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 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 coaches and players in game study but also make game watching a whole new experience. The experiments on international volleyball games show encouraging results. We believe that the proposed framework can be extended and applied to various kinds of sports games.

Keywords Object tracking · Sports video analysis · Content-based multimedia analysis · Camera calibration · 3D trajectory approximation

H.-T. Chen (✉)
Information and Communications Technology Lab, National Chiao Tung University, Hsinchu 300,
Taiwan
e-mail: huatsung@cs.nctu.edu.tw

W.-J. Tsai · S.-Y. Lee
Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan

J.-Y. Yu
ICL/Industrial Technology Research Institute, Hsinchu 300, Taiwan

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 watching a whole game in a sequential way. Various algorithms of shot classification [7, 8, 16, 23], highlight extraction [2, 9, 32] and semantic annotation [1, 3, 18] 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 [4, 5, 22, 27–29, 31, 33] to acquire the ball-player interaction and camera calibration [10, 11, 24, 29, 30] to obtain the ball/player positions in the real world coordinates.

Most existing work in sports video analysis is audience-oriented. However, the coaches and sports professionals desire to acquire tactical, statistical and professional information in game watching. 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 more and more research in sports video processing concentrates on ball tracking and trajectory-based tactics analysis, the majority of existing work focuses on tennis and soccer video [27–29, 33]. Little work was done for volleyball video. However, volleyball games attract a large amount of audience. Besides, it is very challenging to track the ball in volleyball video due to the high density of players on the court and the frequent ball-player overlaps. Hence, ball tracking and 3D trajectory approximation in volleyball video are worth in-depth investigation. In this paper, 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 fixed camera located behind the court, for tactics analysis in volleyball games.

Generally speaking, not all of the sports games are broadcasted on TV. As the rapid evolution of digital equipments, general users are able to capture multimedia data more easily. 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 [12, 17, 19]. This trend necessitates the development of computer-assisted game study system for the user-captured sports video.

The rest of the paper is organized as follows. [Section 2](#) introduces the related work on sports video analysis. [Section 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 [Sections 4, 5 and 6](#), respectively. [Section 7](#) elaborates 3D trajectory approximation. [Section 8](#) presents the trajectory-based applications to tactics analysis. [Section 9](#) reports and discusses the experimental results. Finally, [Section 10](#) concludes this paper.

2 Related work

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 [10, 11, 24]. These plane-based calibration techniques require feature points on a plane appearing in different views. Farin et al. [10, 11] propose a 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. [24] 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 to in the soccer field by utilizing the camera parameters computed via matching the wire frame model with the extracted field lines.

Yu et al. [29, 30] propose a non-plane based camera calibration method of tennis video. They approximate the projection geometry by a perspective projection model mapping from the 3D world to the 2D image. Two techniques are used: frame grouping and Hough-like search. The grouping technique clusters frames according to camera viewpoint. Then, a group-wise data analysis is used to obtain stable camera parameters. However, some of the parameters vary even if they have similar camera viewpoint. A Hough-like search is used to tune some parameters.

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. [27, 28, 31] present a trajectory-based algorithm for ball detection and tracking in soccer video. The ball size is first estimated from feature objects (the goalmouth and ellipse) to detect ball candidates. Potential trajectories are generated from ball candidates by a Kalman filter based verification procedure. Camera motion recovery helps in obtaining better candidates and forming longer ball trajectories. 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. [33] 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. Our previous work [4, 5] performs physics-based ball tracking in broadcast sports video and provide trajectory-based applications, such as pitching evaluation in baseball and shooting location estimation in basketball.

Some work focuses on 3D trajectory reconstruction based on multiple cameras located at specific positions [13, 20, 21]. Hawk-Eye system [20] produces computer-generated replays viewed through 360°. 2D ball 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 [13]

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) [21] 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.

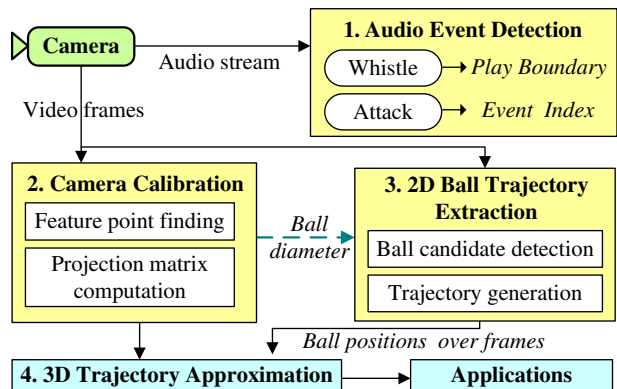
Ball tracking in sports video confronts many difficulties [28]: the small ball size in frames, the varied ball appearance (shape, color, size), the presence of many ball-like objects, the occlusion of the ball (by a player) and the mergence of the ball with lines or players. For soccer video, some previous algorithms [27, 28, 31] estimate the ball size, detect ball candidates by appearance features and extract the ball trajectory based on Kalman filter. In tennis and baseball videos, there are less ball-player occlusions but the ball moves very fast. To achieve high accuracy, multiple high-speed cameras are required to track the ball [13, 20, 21]. As to volleyball video, the high density of players on the court and the frequent ball-player occlusions make ball tracking much challenging. In this paper, we utilize the characteristic that the volleyball moves near parabolically to model the ball motion. The positions of the ball occluded by players can be recovered, and the volleyball trajectory can be extracted accurately.

3 System overview

To achieve automatic tactics analysis in volleyball games, a system called VIA (Volleyball Intelligence Agent) is designed in this paper. Based on our previous work [26], which extracts 2D trajectories for set type recognition, VIA approximates and visualizes 3D ball trajectories, 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. 1.

Whistle directly determines the start and end of each *play* in volleyball games. Thus, VIA starts with whistle detection to segment the game into plays. Moreover, VIA also

Fig. 1 Block diagram of the proposed VIA system



detects the sounds of *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. Since we use a fixed camera for video capturing and there is no camera motion, VIA performs camera calibration and projection matrix computation once in each game. 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 VIA correlates information on the ball candidates over a sequence of frames, explores potential trajectories and identifies the true ball trajectories. To approximate the 3D trajectory, we model the 3D trajectory with the parameters: velocities and initial positions 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 and approximate the 3D ball trajectory. Finally, VIA is able to present trajectory-based applications to tactics analysis.

The novelty and contribution of this paper 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. Incorporating the physical motion information and domain knowledge, we propose the premier approach capable of 3D trajectory approximation without the need of multiple cameras, based on the 2D ball tracking method of our previous work [26]. Moreover, based on the obtained 2D and 3D ball trajectories, we design several novel applications which meet professionals' demand including action detection, set type recognition, 3D virtual replay and serve placement estimation. Trajectory-based tactics analysis and statistics collection can be accomplished without manual efforts to support the coach and professionals in game study and performance evaluation. The novelty of manifold trajectory-based game analysis gives the audience brand-new experience of game watching.

4 Audio event detection

Several significant events, such as *whistle* and *attack*, are difficult to detect from visual features but can be directly traced through audio features. ZCR (Zero Crossing Rate), which counts the number of times that the signal crosses the zero axis, is a simple measurement of the frequency content of a signal. Since whistle has higher frequency than other signals, ZCR (Zero Crossing Rate) is a distinguishing and easy-to-compute feature for whistle detection [25, 26]. Thus, we perform whistle detection via picking the ZCR peaks which are greater than a threshold Z_{ZCR} .

Attack detection 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) [34], a peak can be observed when the attack occurs. Thus, we perform attack detection via picking the STE peaks which are greater than a threshold Z_{STE} .

5 Camera calibration

In the pinhole camera model, a camera is a mapping from the 3D real world to the 2D image space [15]. The real world point $(X, Y, Z)^T$ in 3D space is represented as a

homogeneous 4-vector $W=(X, Y, Z, 1)^T$ by adding a final coordinate of 1, the image point $(x, y)^T$ in 2D space is represented as a homogeneous 3-vector $(x, y, 1)^T$ by adding a final coordinate of 1, and P represents the 3×4 homogeneous camera projection matrix (for homogeneous representation of points, please refer to p27 in [15]). Then, the mapping from the 3D real world to the 2D image is written compactly as

$$m \times PW = \mathbf{0}. \tag{1}$$

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. We first segment the court region consisting of the court lines L_1 to L_7 (see Fig. 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 g_1 to g_{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 g_5, g_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 (g_{11} to g_{14}), together with the ground feature points, form a non-coplanar feature point set.

Here, we briefly describe the methods of solving for the camera projection matrix P ; the details of these methods are available in [15]. For each correspondence $W \leftrightarrow m$ we derive a relationship from Eq. (1):

$$\begin{bmatrix} 0^T & -W^T & yW^T \\ W^T & 0^T & -xW^T \\ -yW^T & xW^T & 0^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0 \tag{2}$$

where each P^{iT} is a 4-vector, the i -th row of P . Since only two of the three equations in (2) are linearly independent, each point correspondence gives two equations:

$$\begin{bmatrix} 0^T & -W^T & yW^T \\ W^T & 0^T & -xW^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0. \tag{3}$$

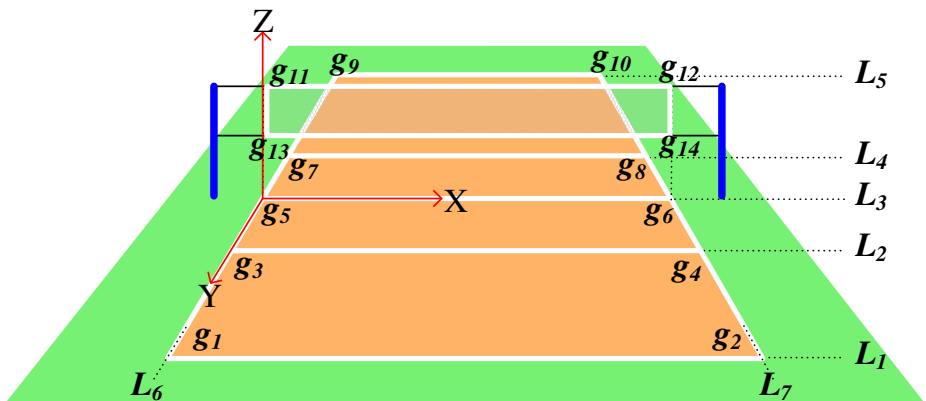


Fig. 2 Illustration of the non-coplanar feature points

The matrix P has 12 entries, and (ignoring scale) 11 degrees of freedom, so it is necessary to have at least 11 equations to solve for P . With the 14 non-coplanar point correspondences obtained, we can solve for P using the direct linear transform (DLT) algorithm (see p109, p178–184 in [15]).

6 2D ball trajectory extraction

It is a challenging task to identify the ball in frames due to its small size and fast movement. We have proposed an effective method of 2D ball trajectory extraction in [6]. In this section, we summarize the processes of 2D trajectory extraction.

6.1 Ball candidate detection

Since we focus on analyzing the video sequences captured by a fixed camera and there is no camera motion, it is sufficient to detect moving pixels via differencing successive frames for moving object segmentation. The *opening* morphological operation (erosion followed by dilation) by a 3×3 square structuring element are performed to remove noises. Please refer to [Section 9.2](#) and [Section 9.3](#) in [14] for more details about morphological operations. Then, moving objects are formed by *region growing* which iterates the following procedure until all moving pixels are assigned to regions: start with a seed (a moving pixel which is not assigned to a region) and grow the region by appending the neighboring moving pixels to the seed. However, the ball is not the only moving object, but also the audience and players. Therefore, we design the following sieves to prune non-ball objects so as to improve the efficiency of the consequent ball tracking step. The remaining objects which satisfy the constraints of size, shape and compactness are considered as the *ball candidates*.

- 1) **Size sieve:** The in-frame ball diameter D_{frm} can be proportionally estimated from the length between the court line intersections by the pinhole camera imaging principal:

$$(D_{frm}/D_{real}) = (d/D), \quad D_{frm} = D_{real}(d/D) \quad (4)$$

where D_{real} is the diameter of a real volleyball (≈ 21 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 court line intersections and calculate the in-frame distance d of the selected two points. Since the distance of the two points in the real court D is specified in the volleyball 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 filters 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 Δ is the extension for tolerance toward processing faults. In our experiments, the video resolution is 352×240 , and we set $\Delta = 10$ empirically.

- 2) **Shape sieve:** The ball in frames may have a shape different from a circle, but its height-to-width ratio R_{h-w} and width-to-height ratio R_{w-h} should be close to 1. Let $R_a = \max(R_{h-w}, R_{w-h})$. An object will be removed if its R_a is greater than a threshold Z_{Ra} .
- 3) **Compactness sieve:** The compactness sieve is built to filter out the objects with the compactness degree CD , as defined in Eq. (5), less than a threshold Z_{CD} .

$$CD = \text{Object_Size} / \text{Bounding_Box_Area} \quad (5)$$

The ball is at a distance away from other moving objects 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), and it is classified as *contacted*, otherwise.

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-time plot* (YTP) and *x-time plot* (XTP), respectively. An example of YTP and XTP is shown in Fig. 3a, where black dots and green crosses represent the detected isolated candidates and contacted candidates, respectively.

To acquire the characteristics of ball motion, we make YTPs and XTPs for 30 volleyball clips by manually locating the ball positions over frames. Three examples are shown in Fig. 4, where the vertical lines indicate the ball-player interactions. We observe that the ball

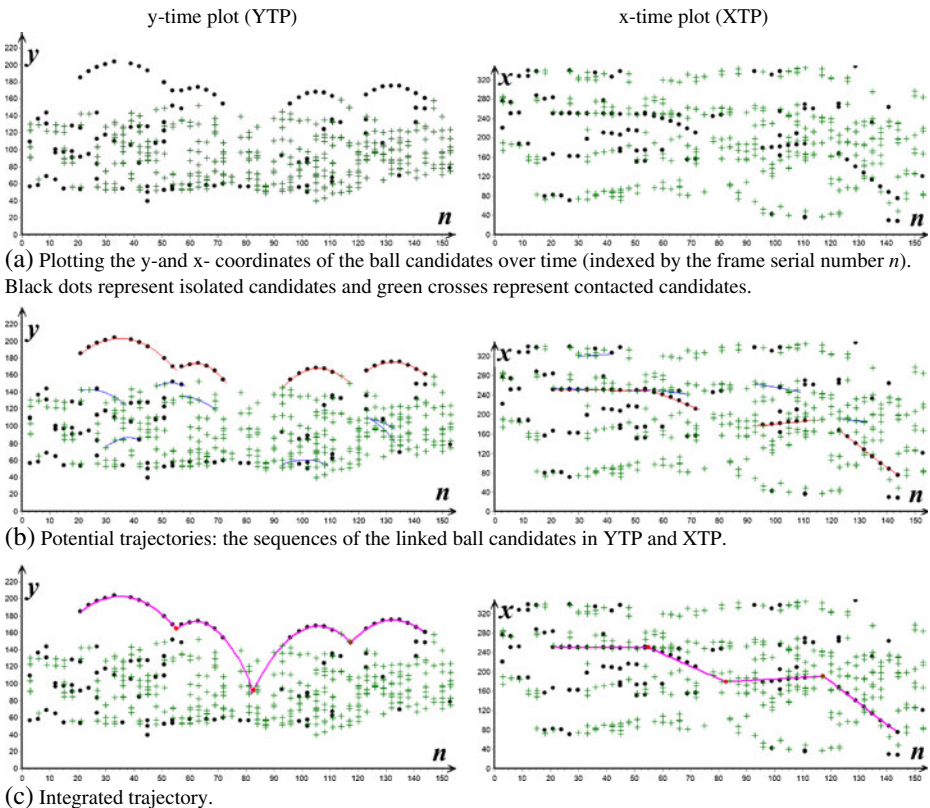


Fig. 3 Illustration of the y-time plot and x-time plot for different process stages.

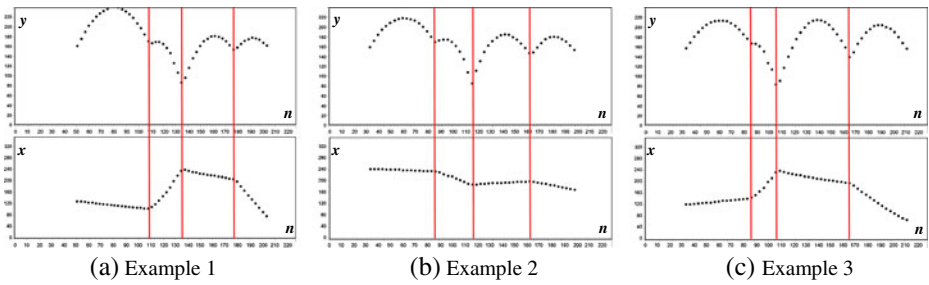


Fig. 4 YTPs(the upper row) and XTPs(the lower row) of the ball positions manually located

moves near parabolically in y-direction and moves near straightly in x-direction between a pair of successive ball-player interactions. Therefore, we design an algorithm to explore a sequence of ball candidates which form a near parabolic curve in YTP and a near straight line in XTP simultaneously as a *potential trajectory*.

Figure 5 shows the algorithm of potential trajectory exploration. Initially, each ball candidate c is linked to the nearest candidate c' in the previous frame if the distance between c and c' is smaller than D_{frm} (the in-frame ball diameter). A *growing trajectory* is formed when the number of linked ball candidates is up to three (three points form a parabolic curve), and the prediction functions are defined as Eq. (6) and Eq. (7).

$$y = a_1 \cdot n^2 + a_2 \cdot n + a_3, a_1 < 0, n : \text{frame serial number} \tag{6}$$

$$x = b_1 \cdot n + b_2 \tag{7}$$

Then, the algorithm verifies the ball position prediction of each growing trajectory. The prediction is considered *matched* if the distance between a ball candidate and the predicted position is smaller than D_{frm} . A growing trajectory T extends by adding the ball candidate which matches the predicted position, and the prediction functions are updated by re-computing the best-fitting functions for the coordinates of the ball candidates in T using the *least squares fitting*. For a growing trajectory, if no candidate matches the predicted position, the ball is considered *missed* and the predicted position is taken as the ball position. A growing trajectory T is finalized as a potential trajectory if the ball is missed for Z_f consecutive frames. A ball candidate added to no growing trajectory is linked to the nearest candidate in the previous frame for forming a new growing trajectory. The potential trajectories produced from this procedure are shown as the sequences of the linked ball candidates in YTP and XTP in Fig. 3b.

6.3 Trajectory identification and integration

For each potential trajectory T , we define its *confidential point* to measure how likely T is a ball trajectory. A potential trajectory T gains one confidential point for each satisfaction of the following criteria: 1) the trajectory length is greater than a threshold Z_{TL} , 2) the percentage of isolated candidates in the trajectory is greater than a threshold Z_{IC} , and 3) the prediction error (defined as the average of the distances from ball candidate positions to the predicted positions) is less than a threshold Z_{PE} . Obviously, we want ball trajectories to

Input: BC , the set of detected ball candidates

Output: S , the set of potential trajectories

Initialization for growing trajectories;

```

for each frame  $f$  {
  for each ball candidate  $c$  in  $f$  {
    for each growing trajectory  $T$  {
      if ( $c$  matches the ball position predicted by  $T$ ) {
        Add  $c$  to  $T$ ;
        Update the prediction functions of  $T$ ;
      }
    }
    if ( $c$  is not added to any growing trajectory and  $f$  is not the first frame) {
      Find in the previous frame the ball candidate  $c'$  closest to  $c$ ;
      if (distance( $c, c'$ ) <  $D_{f,m}$ ) {
        Link  $c$  to  $c'$ ;
        if (# of linked ball candidates reaches 3) {
          if (the 3 linked ball candidates form a line in XTP) {
            Initialize a growing trajectory  $T$ ;
            Initialize the prediction functions of  $T$ ;
          }
          else
            Remove the first ball candidate from the link;
        }
      }
    }
  }
}
for each growing trajectory  $T$  {
  if (no ball candidate matches the predicted position) {
    The ball is considered missed;
    The predicted position is taken as the ball position;
    Set  $Z_f$  = half the number of the ball candidates in  $T$ ;
    if (the ball is missed for  $Z_f$  consecutive frames )
      Move  $T$  into  $S$ ;
  }
}

```

Fig. 5 Algorithm of potential trajectory exploration

have high confidential points. However, we cannot *just* iteratively select the potential trajectory T_h with highest confidential points and discard the trajectories which overlap with T_h , because a ball trajectory may be removed wrongly if the adjacent trajectory is over extended due to a nearby spurious candidate. Hence, we first select the potential trajectories with 3 confidential points as the true trajectories (we say that these trajectories are *identified*). For two overlapped trajectories, we compute the trajectory intersection, and trim the portion after the intersection in the former trajectory and trim the portion before the intersection in the latter. Then, we iteratively select the longest 2-point potential trajectories which do not overlap with identified trajectories until all 2-point potential trajectories are processed. 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. 3c. Thus, the ball positions can be estimated even though the ball is temporarily occluded.

7 3D 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, gravitational constant, 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:

$$\begin{aligned} X_t &= X_0 + V_X t \\ Y_t &= Y_0 + V_Y t \\ Z_t &= Z_0 + V_Z t + g t^2 / 2 \end{aligned} \tag{8}$$

where (X_t, Y_t, Z_t) 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 gravitational constant.

We use $W_t = (X_t, Y_t, Z_t, 1)^T = (X_0 + V_X t, Y_0 + V_Y t, Z_0 + V_Z t + g t^2 / 2, 1)^T$ and $m_t = (x_t, y_t, 1)^T$. From Eq. (3) each point correspondence gives two equations as

$$\begin{bmatrix} 0^T & -W_t^T & y_t W_t^T \\ W_t^T & 0^T & -x_t W_t^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0. \tag{9}$$

Given N detected ball candidates, we obtain $2N$ equations. Since the entries in P (denoted as p_{ij}), the 2D image coordinate (x_t, y_t) and the occurring time t of each ball candidate are known, we can set up a linear system, as Eq. (10), to compute the six unknowns X_0, Y_0, Z_0, V_X, V_Y and V_Z . We solve for $(X_0, V_X, Y_0, V_Y, Z_0, V_Z)^T$ using the direct linear transform algorithm (see p109 in [15]). Thus, each 3D sub-trajectory can be approximated.

$$\begin{bmatrix} p_{11} - x_1 p_{31} & p_{11} t_1 - x_1 p_{31} t_1 & p_{12} - x_1 p_{32} & p_{12} t_1 - x_1 p_{32} t_1 & p_{13} - x_1 p_{33} & p_{13} t_1 - x_1 p_{33} t_1 \\ p_{21} - y_1 p_{31} & p_{21} t_1 - y_1 p_{31} t_1 & p_{22} - y_1 p_{32} & p_{22} t_1 - y_1 p_{32} t_1 & p_{23} - y_1 p_{33} & p_{23} t_1 - y_1 p_{33} t_1 \\ p_{11} - x_2 p_{31} & p_{11} t_2 - x_2 p_{31} t_2 & p_{12} - x_2 p_{32} & p_{12} t_2 - x_2 p_{32} t_2 & p_{13} - x_2 p_{33} & p_{13} t_2 - x_2 p_{33} t_2 \\ p_{21} - y_2 p_{31} & p_{21} t_2 - y_2 p_{31} t_2 & p_{21} - y_2 p_{32} & p_{21} t_2 - y_2 p_{32} t_2 & p_{23} - y_2 p_{33} & p_{23} t_2 - y_2 p_{33} t_2 \\ & & & \vdots & & \\ p_{11} - x_N p_{31} & p_{11} t_N - x_N p_{31} t_N & p_{12} - x_N p_{32} & p_{12} t_N - x_N p_{32} t_N & p_{13} - x_N p_{33} & p_{13} t_N - x_N p_{33} t_N \\ p_{21} - y_N p_{31} & p_{21} t_N - y_N p_{31} t_N & p_{22} - y_N p_{32} & p_{22} t_N - y_N p_{32} t_N & p_{23} - y_N p_{33} & p_{23} t_N - y_N p_{33} t_N \end{bmatrix} \begin{bmatrix} X_0 \\ V_X \\ Y_0 \\ V_Y \\ Z_0 \\ V_Z \end{bmatrix} = \begin{bmatrix} x_1(p_{33} g t_1^2 / 2 + 1) - (p_{13} g t_1^2 / 2 + p_{14}) \\ y_1(p_{33} g t_1^2 / 2 + 1) - (p_{23} g t_1^2 / 2 + p_{24}) \\ x_2(p_{33} g t_2^2 / 2 + 1) - (p_{13} g t_2^2 / 2 + p_{14}) \\ y_2(p_{33} g t_2^2 / 2 + 1) - (p_{23} g t_2^2 / 2 + p_{24}) \\ \vdots \\ x_N(p_{33} g t_N^2 / 2 + 1) - (p_{13} g t_N^2 / 2 + p_{14}) \\ y_N(p_{33} g t_N^2 / 2 + 1) - (p_{23} g t_N^2 / 2 + p_{24}) \end{bmatrix} \tag{10}$$

However, here comes a problem. Since each 3D sub-trajectory is approximated independently, the 3D coordinate of the transition point between a pair of 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 simultaneously.

Figure 6 illustrates the procedure of 3D trajectory approximation by a sample ball trajectory. As shown in Fig. 6a, the ball trajectory contains three sub-trajectories S_0 , S_1 and S_2 . Let P_1 be the transition point between S_0 and S_1 , and P_2 between S_1 and S_2 . Let $(V_{X_i}, V_{Y_i}, V_{Z_i})$ be the 3D ball velocity of the sub-trajectory S_i , where i is the index. As shown in Fig. 6b, we consider the two adjacent sub-trajectories S_0 and S_1 to derive $(V_{X_0}, V_{Y_0}, V_{Z_0}, X_1, Y_1, Z_1, V_{X_1}, V_{Y_1}, V_{Z_1})$. Taking P_1 as the initial point, the 3D sub-trajectories of S_0 and S_1 are expressed as Eq. (11) and Eq. (12), respectively.

$$\begin{aligned} X_t &= X_1 - V_{X_0}t \\ Y_t &= Y_1 - V_{Y_0}t \\ Z_t &= Z_1 - V_{Z_0}t - gt^2/2 \end{aligned} \tag{11}$$

$$\begin{aligned} X_t &= X_1 + V_{X_1}t \\ Y_t &= Y_1 + V_{Y_1}t \\ Z_t &= Z_1 + V_{Z_1}t + gt^2/2 \end{aligned} \tag{12}$$

Using $W_t = (X_t, Y_t, Z_t, 1)^T = (X_1 - V_{X_0}t, Y_1 - V_{Y_0}t, Z_1 - V_{Z_0}t - gt^2/2)$ for S_0 and $W_t = (X_t, Y_t, Z_t, 1)^T = (X_1 + V_{X_1}t, Y_1 + V_{Y_1}t, Z_1 + V_{Z_1}t + gt^2/2)$ for S_1 , we obtain two equations for each ball candidate, as Eq. (9). The $2N$ equations produced from N ball

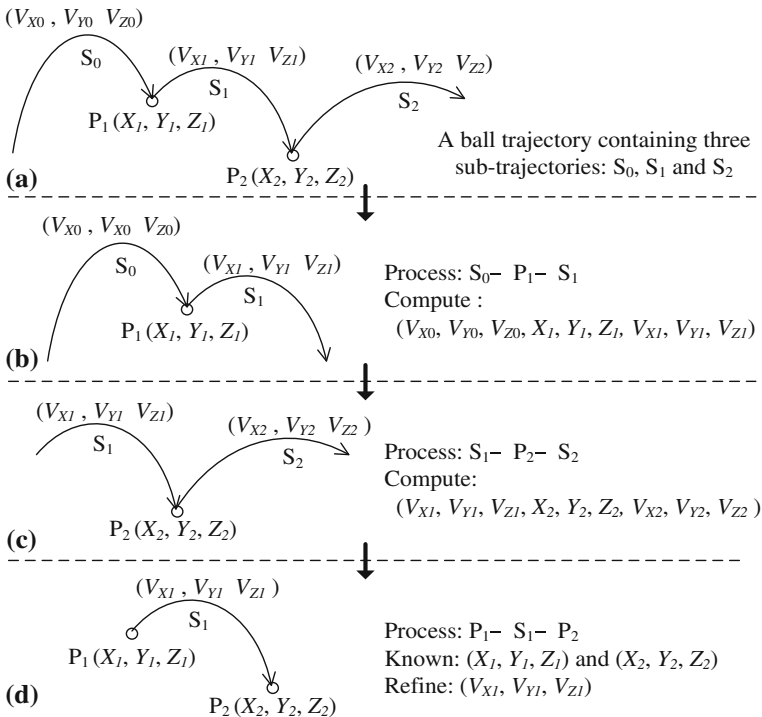


Fig. 6 Procedure of 3D trajectory approximation

candidates form a linear system, as Eq. (13). Assume that the S_0 and S_1 consist of k and $N-k$ ball candidates, respectively.

$$\begin{bmatrix}
 p_{11} - x_1 p_{31} & p_{11} t_1 + x_1 p_{31} t_1 & 0 & \cdots & p_{13} - x_1 p_{33} & p_{13} t_1 + x_1 p_{33} t_1 & 0 \\
 p_{21} - y_1 p_{31} & p_{21} t_1 + y_1 p_{31} t_1 & 0 & \cdots & p_{23} - y_1 p_{33} & p_{23} t_1 + y_1 p_{33} t_1 & 0 \\
 & & & \vdots & & & \\
 p_{11} - x_k p_{31} & p_{11} t_k + x_k p_{31} t_k & 0 & \cdots & p_{13} - x_k p_{33} & p_{13} t_k + x_k p_{33} t_k & 0 \\
 p_{21} - y_k p_{31} & p_{21} t_k + y_k p_{31} t_k & 0 & \cdots & p_{23} - y_k p_{33} & p_{23} t_k + y_k p_{33} t_k & 0 \\
 p_{11} - x_{k+1} p_{31} & 0 & p_{11} t_{k+1} - x_{k+1} p_{31} t_{k+1} & \cdots & p_{13} - x_{k+1} p_{33} & 0 & p_{13} t_{k+1} + x_{k+1} p_{33} t_{k+1} \\
 p_{21} - y_{k+1} p_{31} & 0 & p_{21} t_{k+1} - y_{k+1} p_{31} t_{k+1} & \cdots & p_{23} - y_{k+1} p_{33} & 0 & p_{23} t_{k+1} + y_{k+1} p_{33} t_{k+1} \\
 & & & \vdots & & & \\
 p_{11} - x_N p_{31} & 0 & p_{11} t_N - x_N p_{31} t_N & \cdots & p_{13} - x_N p_{33} & 0 & p_{13} t_N - x_N p_{33} t_N \\
 p_{21} - y_N p_{31} & 0 & p_{21} t_N - y_N p_{31} t_N & \cdots & p_{23} - y_N p_{33} & 0 & p_{23} t_N - y_N p_{33} t_N
 \end{bmatrix}
 \begin{bmatrix}
 X_1 \\
 V_{X0} \\
 V_{X1} \\
 Y_1 \\
 V_{Y0} \\
 V_{Y1} \\
 Z_1 \\
 V_{Z0} \\
 V_{Z1}
 \end{bmatrix}
 =
 \begin{bmatrix}
 -x_1(p_{33} g_{t_1}^2/2 + 1) + (p_{13} g_{t_1}^2/2 + p_{14}) \\
 -y_1(p_{33} g_{t_1}^2/2 + 1) + (p_{23} g_{t_1}^2/2 + p_{24}) \\
 \vdots \\
 -x_k(p_{33} g_{t_k}^2/2 + 1) + (p_{13} g_{t_k}^2/2 + p_{14}) \\
 -y_k(p_{33} g_{t_k}^2/2 + 1) + (p_{23} g_{t_k}^2/2 + p_{24}) \\
 x_{k+1}(p_{33} g_{t_{k+1}}^2/2 + 1) + (p_{13} g_{t_{k+1}}^2/2 + p_{14}) \\
 y_{k+1}(p_{33} g_{t_{k+1}}^2/2 + 1) + (p_{23} g_{t_{k+1}}^2/2 + p_{24}) \\
 \vdots \\
 x_N(p_{33} g_{t_N}^2/2 + 1) - (p_{13} g_{t_N}^2/2 + p_{14}) \\
 y_N(p_{33} g_{t_N}^2/2 + 1) - (p_{23} g_{t_N}^2/2 + p_{24})
 \end{bmatrix}
 \tag{13}$$

We solve for $(X_I, V_{X0}, V_{X1}, Y_I, V_{Y0}, V_{Y1}, Z_I, V_{Z0}, V_{Z1})^T$ using the direct linear transform algorithm. Thus, the coordinate of $P_1 (X_I, Y_I, Z_I)$ 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 computed by processing S_1 and S_2 simultaneously, as shown in Fig. 6c.

For the sub-trajectory S_1 between two transition points P_1 and P_2 , 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 $S_1-P_2-S_2$. Thus, we refine (V_{X1}, V_{Y1}, V_{Z1}) via solving for Eq. (12) with $(X_t, Y_t, Z_t)=(X_2, Y_2, Z_2)$. Finally, the 3D trajectory can be approximated using the 3D ball velocity (V_{Xi}, V_{Yi}, V_{Zi}) on each sub-trajectory S_i and the coordinate (X_j, Y_j, Z_j) of each transition point P_j .

8 Trajectory-based applications

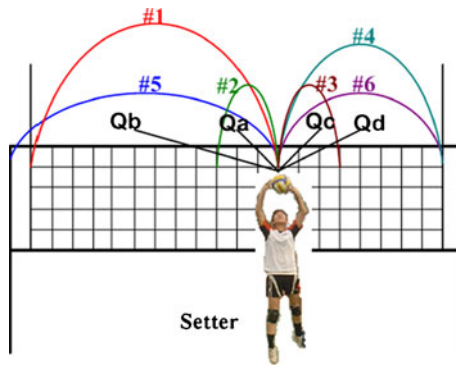
This section presents several applications based on the acquired 2D and 3D trajectories to demonstrate the utility of the proposed 2D ball tracking and 3D trajectory approximation scheme. The trajectory-based applications to tactics analysis greatly assist the coaches and players in game strategy study.

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 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.

Since in volleyball games the *set type* brings most tactical information and typically dominates whether a team can score or not, we focus on the *set* action and further recognize the *set type*. Figure 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.

Fig. 7 Illustration of set type diagram



We define the discriminants as Table 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. (6) and Eq. (7), and γ_1 – γ_3 are thresholds. We use 200 set curves (20 curves per set type) as training data and manually label the set types. The thresholds γ_1 – γ_3 are determined by seeking for the values which best classify the set curves in the training data.

8.2 3D virtual replay 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).

9 Experimental results and discussion

The proposed algorithms of audio event detection, 2D ball tracking and 3D trajectory approximation are implemented in Borland C++ Builder 6.0. For performance evaluation,

Table 1 Discriminants of ten common set types

Set	Discriminant	Set	Discriminant
#1(long)	$b_1 < 0, b_1 > \gamma_1, b_1 /a_2 \leq \gamma_2$	Qa (quick)	$b_1 < 0, b_1 < \gamma_1, \gamma_2 < b_1 /a_2 \leq \gamma_3$
#2(short)	$b_1 < 0, b_1 < \gamma_1, b_1 /a_2 \leq \gamma_2$	Qb (quick)	$b_1 < 0, b_1 /a_2 > \gamma_3$
#3(short)	$b_1 > 0, b_1 < \gamma_1, b_1 /a_2 \leq \gamma_2$	Qc (quick)	$b_1 > 0, b_1 < \gamma_1, \gamma_2 < b_1 /a_2 \leq \gamma_3$
#4(long)	$b_1 > 0, b_1 > \gamma_1, b_1 /a_2 \leq \gamma_2$	Qd (quick)	$b_1 > 0, b_1 /a_2 > \gamma_3$
#5(long)	$b_1 < 0, b_1 > \gamma_1, \gamma_2 < b_1 /a_2 \leq \gamma_3$		
#6 (long)	$b_1 > 0, b_1 > \gamma_1, \gamma_2 < b_1 /a_2 \leq \gamma_3$		

the proposed system is tested on the volleyball video sequences (MPEG-1, 352×240 , 29.97 fps, Audio: 44.1 kHz, 16 bits, stereo) which we captured in *2005 Asia Men's Volleyball Challenge Cup*: 1) Taiwan vs. Korea, 2) China vs. Japan, and 3) Japan vs. Korea.

9.1 Parameter setting

In ball candidate detection, an object will be removed by the shape sieve or the compactness sieve if its R_a (the higher value between AR and $1/AR$, where AR is the aspect ratio) is greater than a threshold Z_{Ra} or its compactness degree CD is less than Z_{CD} , respectively. To determine the values of Z_{Ra} and Z_{CD} , we use a training data set containing 300 ball candidates which are manually recognized as the true ball. We compute the R_a and CD of the 300 ball candidates, and construct the relative cumulative frequency distributions, as shown in Fig. 8. For high recall rate, the thresholds Z_{Ra} and Z_{CD} are determined in such a way that no more than 5% of the ball candidates in the training data set will be removed by the shape and compactness sieves.

In potential trajectory exploration, a growing trajectory T is finalized as a potential trajectory if the ball is missed for Z_f consecutive frames. We set $Z_f = \text{half the number of the ball candidates in } T$, since a potential trajectory with more than half the ball positions missed is not reliable. In trajectory identification, we compute the trajectory lengths, the percentages of isolated candidates in the trajectory and the prediction errors (in pixel) of a training data set containing 100 true ball trajectories and 100 false trajectories to construct the relative frequency distributions, as shown in Fig. 9, where the solid lines are for true ball trajectories and the dotted lines are for false trajectories. The thresholds Z_{TL} , Z_{IC} and Z_{PE} are determined at the intersections of the solid lines and the dotted lines.

9.2 Results of 2D trajectory extraction

We achieve good performance of overall 96.5% precision rate and 98.57% recall rate in whistle detection. 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 ball is in the region of the frame even though the ball is occluded), 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 ball position within a distance of D_{frm} (the in-frame ball diameter) from the ground truth 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

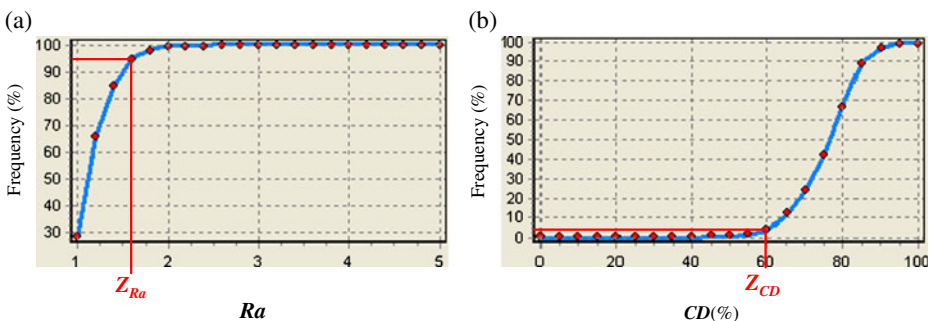


Fig. 8 Relative cumulative frequency distributions of (a) R_a and (b) CD

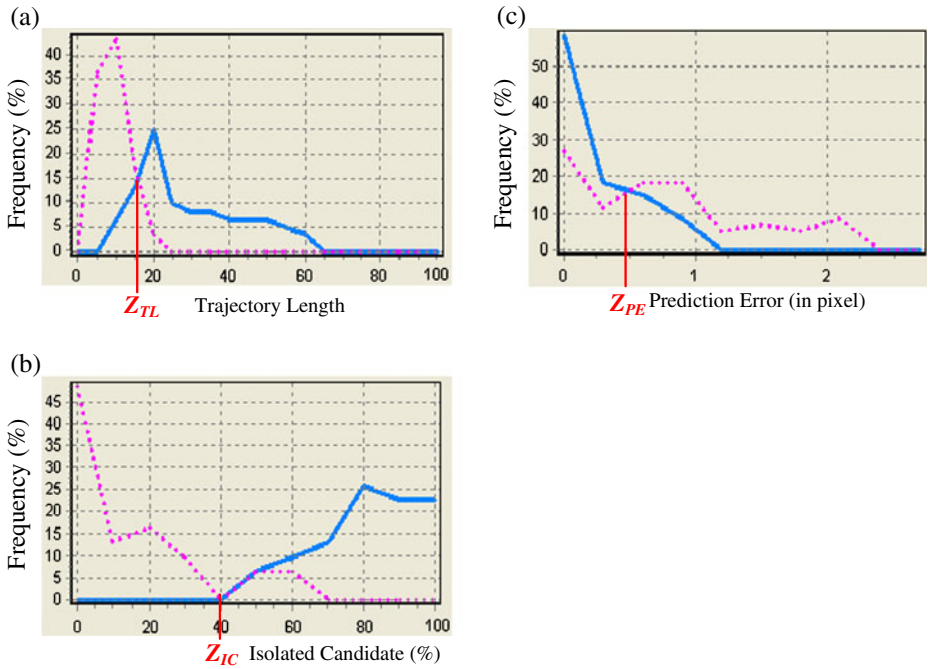


Fig. 9 Relative frequency distributions of (a) trajectory length, (b) percentage of isolated candidates in the trajectory, (c) prediction errors (in pixel)

or it detects a ball in a no-ball frame. Let #frm and #ball-frm denote the numbers of frames and ball frames in the sequence, respectively. Let #correct be the number of frames which the system identifies correctly and #false be the number of false alarms.

The results of ball detection and tracking are presented in Table 2. 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. 10. Figure 10a shows the original frame. In Fig. 10b, 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. 10c. We achieve the accuracy (#correct/#frm) of 71.85% 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

Table 2 Performance of ball detection and tracking (accuracy=#correct/#frm)

Sequence	Ground truth		Detection result			Tracking result		
	#frm	#ball-frm	#correct	#false	accuracy	#correct	#false	accuracy
TWN-KOR	15824	11508	11643	426	73.58%	13632	403	86.15%
CHN-JPN	14835	10520	10712	422	72.21%	12877	411	86.80%
JPN-KOR	19241	13147	13499	518	70.16%	16964	506	88.17%
Total	49900	35175	35854	1366	71.85%	43473	1320	87.12%

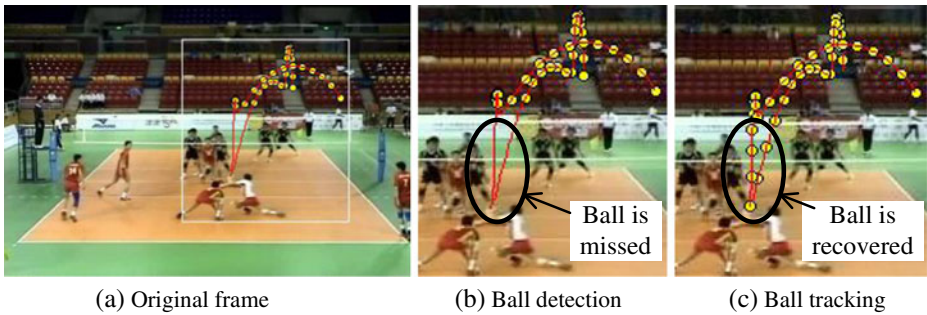


Fig. 10 Illustration of ball detection and ball tracking

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 ball tracking method is able to correct most errors and promotes the final accuracy up to 87.12% on average. Besides, the rate of false alarm ($\#false/\#frm$) is very low—an average of 2.65%, 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 to tactics analysis and 3D trajectory approximation.

Table 3 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.

Figure 11 demonstrates 2D ball trajectory extraction and action detection. The detected action: *serve*, *reception*, *set* and *attack* are shown in Fig. 11a–d, respectively. *Set* is one of the actions and the *set* type is further recognized. In each of Fig. 11a–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.

9.3 Simulation results of 3D 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 Figs. 12, 13, and 14. Take Fig. 12 for explanation. Figure 12a displays the frame at the moment when a *serve* is

Table 3 Performance of action detection

Action	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%

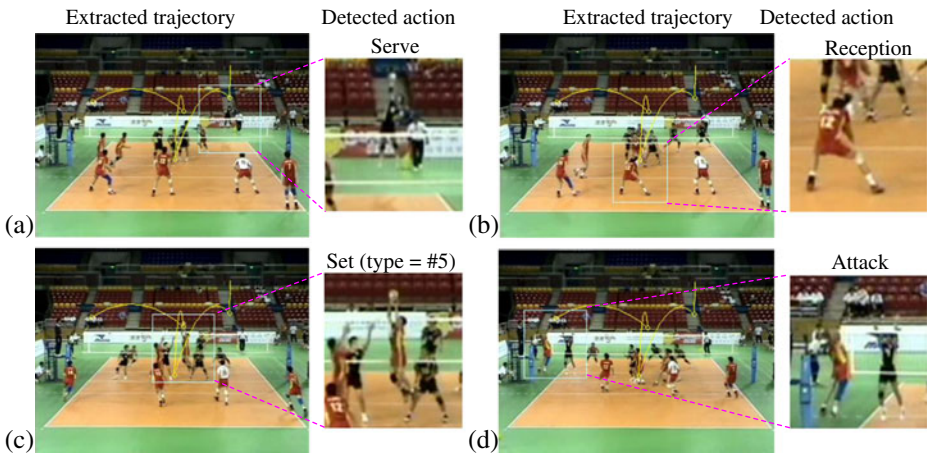


Fig. 11 Ball trajectory extraction and action detection: **a** serve, **b** reception, **c** set and **d** attack

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. 12b–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. 12e, enables the audience or professionals to comprehend the transition of ball motion much easily. Figure 12f displays the

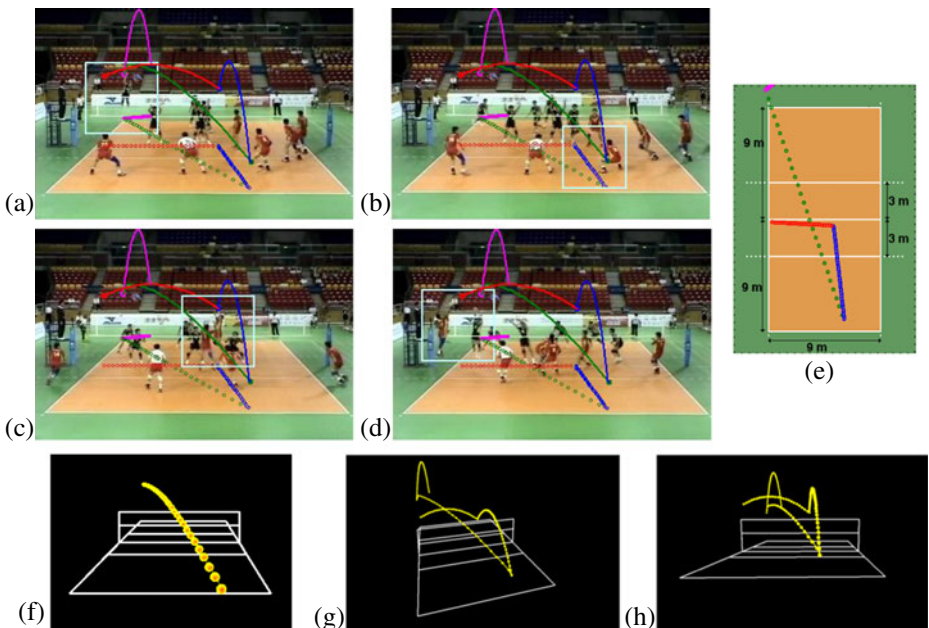


Fig. 12 3D trajectory approximation of sample 1 (set type=#5 and the set action is close to the net): **(a)–(d)** The enriched frames for serve, reception, set and attack, respectively **(e)** Ball trajectory projected on the court model **(f)** Serve placement estimation **(g)–(h)** 3D virtual replays from different viewpoints

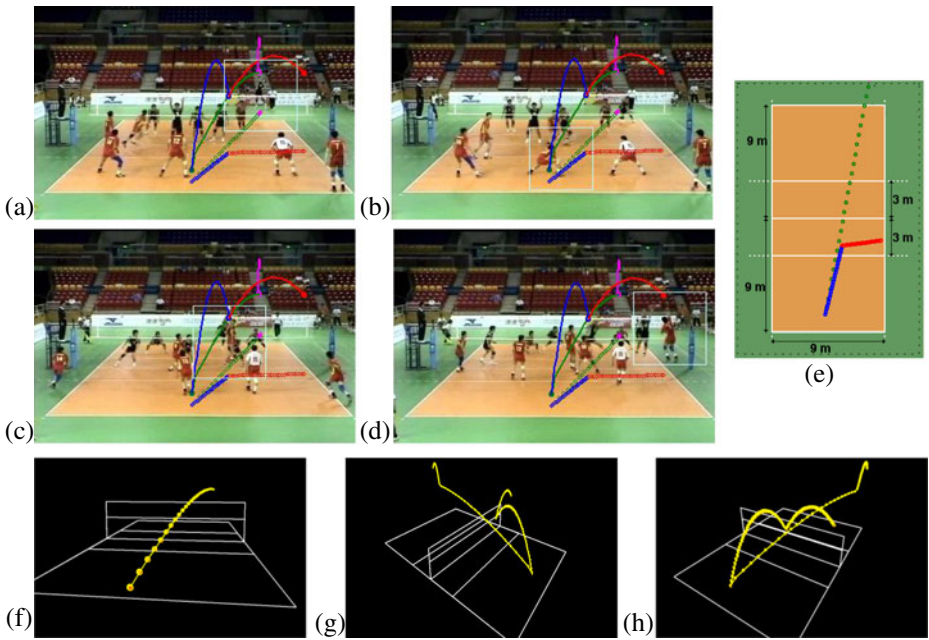


Fig. 13 3D trajectory approximation of sample 2 (set type=#4 and the set action is far from the net): (a)–(d) The enriched frames for serve, reception, set and attack, respectively (e) Ball trajectory projected on the court model (f) Serve placement estimation (g)–(h) 3D virtual replays from different viewpoints

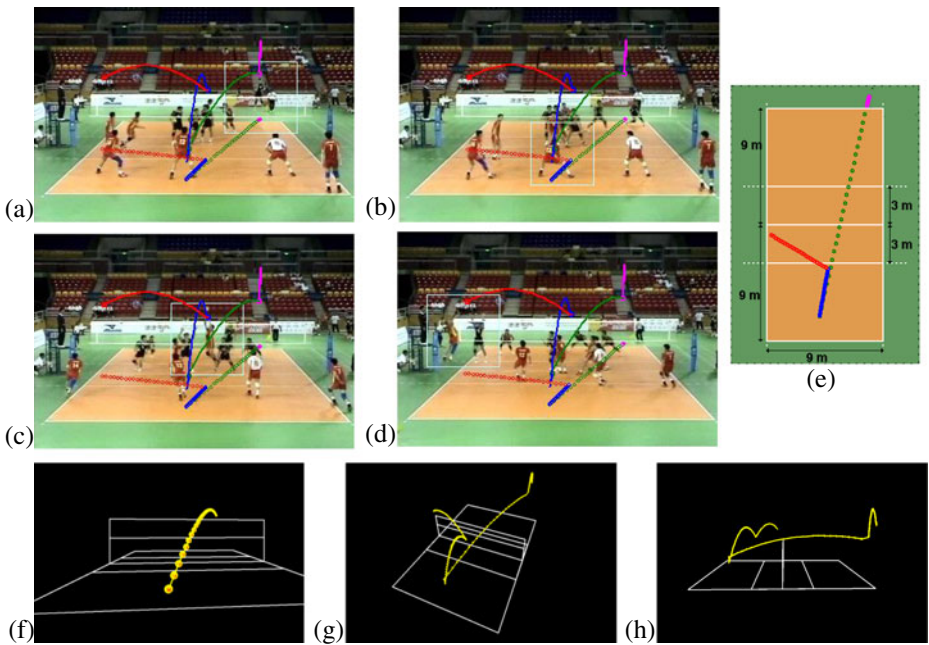


Fig. 14 3D trajectory approximation of sample 3 (set type=#5 and the set action is far from the net): (a)–(d) The enriched frames for serve, reception, set and attack, respectively (e) Ball trajectory projected on the court model (f) Serve placement estimation (g)–(h) 3D virtual replays from different viewpoints

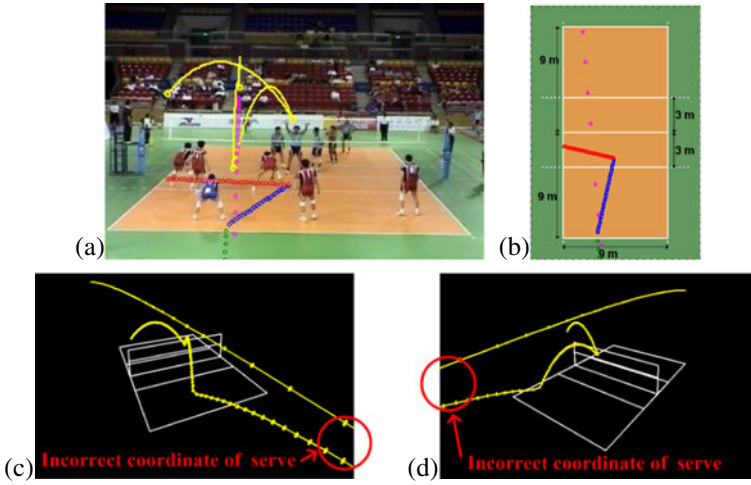


Fig. 15 Example of false 3D ball trajectory approximation: (a) 2D ball trajectory superimposed on the frame, (b) ball trajectory projected on the court model, (c) and (d) 3D virtual replays from different viewpoints

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. 12g–h. In the play of Fig. 12, the receiver passes the ball close to the net and the setter sets the ball (type=#5) along the net, which can be seen in Fig. 12e. In Fig. 13, the receiver passes the ball to the setter about two meters away from the net (see Fig. 13e), and the set type is #4. In Fig. 14, the receiver passes the ball to the setter about three meters away from the net (see Fig. 14e), and the setter sets the ball (type #5) toward the net so that the attacker can hit the ball at a position close to the net.

Inspecting 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, as shown in Fig. 15. Thus, the 3D ball positions would not be estimated correctly. 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.

Since the 3D ball coordinates in the real world cannot be obtained, we prepare synthesis data where the ground truth is known and use them to perform quantitative analysis. Table 4 shows the number of ball frames (#ball-frame), the 2D ball tracking accuracy, the average and the maximum 3D distances between the ground truth and the computed 3D ball positions. The average 3D distances are about 0.14 m and 0.18 m. The maximum 3D

Table 4 Quantitative analysis of 3D trajectory approximation

	#ball-firm	2D ball tracking accuracy	Avg. 3D distance	Max. 3D distance
Synthesis 1	1409	91.41%	0.14 m	0.31 m
Synthesis 2	1242	90.18%	0.18 m	0.39 m

Table 5 3D trajectory approximation performance on different tracking accuracies

	2D ball tracking accuracy (Manually reduced)	Avg. 3D distance	Max. 3D distance
Synthesis 1	91.41%	0.14 m	0.31 m
Synthesis R1	73.10%	0.28 m	0.63 m
Synthesis R2	60.89%	0.48 m	0.97 m
Synthesis R3	45.71%	0.91 m	1.76 m

distances are no more than 0.4 m. To further evaluate the robustness of the proposed 3D trajectory approximation approach against 2D ball tracking error, we manually reduced the tracking accuracy via discarding some tracked 2D ball positions and then compute the average and the maximum 3D distances, as show in Table 5 and Fig. 16. It can be observed that the average and the maximum distances increase dramatically as the 2D ball tracking accuracy reduces. That is, the proposed 3D trajectory approximation approach highly relies on the tracked 2D ball positions.

9.4 Discussion and comparison

The experiments are conducted on an IBM ThinkPad X60 notebook computer (CPU: Intel Core Duo T2400 1.83 GHz, RAM: 1 GB). Table 6 presents the computing time of each process stage: τ_1 for object segmentation, τ_2 for ball candidate detection, τ_3 for ball tracking and τ_4 for 3D trajectory approximation. The percentage of the computing time of each process stage to the total computing time $\tau_{all}(\tau_{all} = \tau_1 + \tau_2 + \tau_3 + \tau_4)$ is given in the parentheses. The last two rows present the number of frames in the sequence (#frm) and the average computing time for each frame ($\tau_{all}/\#frm$).

Table 6 shows that the computation of 3D trajectory approximation is very efficient, which takes a quite low percentage of the total computing time (about 1.3%). The computational cost is mainly from ball tracking (up to 70%). In our statistics, 6.89 ball candidates are detected from 59.43 objects produced in each frame on average, as presented in Table 7. The *effectiveness* is defined as Eq. (14):

$$\text{Effectiveness} = (N_b - N_a)/N_b \quad (14)$$

where N_b and N_a are the object numbers *before* and *after* applying a sieve to remove non-ball objects, respectively. The most effective sieve is the size sieve, which is able to remove 68.16% of non-ball objects. The shape and compactness sieves can remove 34.07% and 45.89%, respectively. The computational efficiency of ball tracking can be improved if we tighten the sieve constraints to remove more non-ball objects. However, more misses will occur accordingly. Overall, we achieve the average computing time of about 25 ms/frame, that is, the proposed VIA system is able to extract 2D trajectories and approximate 3D trajectories *in real time*.

Fig. 16 3D trajectory approximation performance on different tracking accuracies

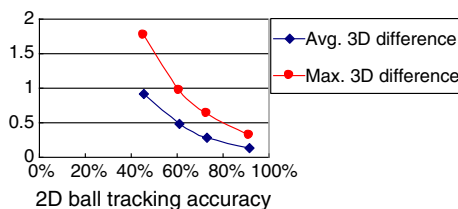


Table 6 Computing time of each process stage (τ_1 : object segmentation, τ_2 : ball candidate detection, τ_3 : ball tracking, τ_4 : 3D trajectory approximation, $\tau_{all} = \tau_1 + \tau_2 + \tau_3 + \tau_4$)

Computing time	TWN-KOR	CHN-JPN	JPN-KOR
τ_1 (τ_1/τ_{all})	81353 ms (21.77%)	78593 ms (19.61%)	96649 ms (19.93%)
τ_2 (τ_2/τ_{all})	23729 ms (6.35%)	24485 ms (6.11%)	37799 ms (7.79%)
τ_3 (τ_3/τ_{all})	264010 ms (70.63%)	292547 ms (72.99%)	343944 ms (70.93%)
τ_4 (τ_4/τ_{all})	4656 ms (1.25%)	5184 ms (1.29%)	6542 ms (1.35%)
τ_{all}	373748 ms	400809 ms	484934 ms
#frm	15824	14835	19241
$\tau_{all}/\#frm$	23.62 ms	27.02 ms	25.2 ms

For performance comparison, we implement another ball tracking algorithm based on Kalman filter, which is widely used in object tracking [13, 27, 28]. To compare the effectiveness and efficiency of the Kalman filter-based algorithm (KF) with those of our algorithm, we use #correct (the number of the frames in which the system correctly identifies the ball), #false (the number of false alarms), accuracy (#correct/#frm) and CT (the computing time) as criteria, as reported in Table 8. The comparison shows that our algorithm performs better in eliminating false alarms. Consequently, our algorithm achieves higher accuracy of about 87% compared to about 80% for KF algorithm. Moreover, our algorithm requires less computing time. This is because the use of the ball motion characteristics can prevent searching many false trajectories. In conclusion, our algorithm outperforms KF algorithm in both effectiveness and efficiency.

As to 3D trajectory approximation, Table 9 shows the comparison between the famous existing systems [13, 20, 21] and our proposed VIA system. To the best of our knowledge, there are few researches on 3D trajectory approximation from single-camera video sequences. The 2D-to-3D inference is also one of the main contribution and novelty of this paper. Most of the existing 3D trajectory reconstruction systems work on multiple cameras located on specific positions. The Hawk-Eye system [20] completed its debut at the Wimbledon Championships 2007 and has been applied to official games for years. It is claimed that the Hawk-Eye system can provide instant replay within 2–3 seconds and shows an average error of only 0.36 cm. ESPN K-Zone system [13] officially debuted in 2001 and is claimed to be accurate to within four-tenths of an inch (1.016 cm). The UIS (Umpire Information System) [21] was first seen on air during the 1997 Baseball World Series. The UIS claims that each pitch can be tracked and recorded within a half-inch (1.27 cm) of its actual location. These existing systems have outstanding performance in 3D trajectory approximation. However, they require high cost of multiple high speed cameras

Table 7 Effectiveness of the size, shape and compactness sieves

Process stage	#object (per frame)	Effectiveness
Initial	59.43	
Apply the size sieve only	18.92	68.16%
Apply the shape sieve only	39.19	34.07%
Apply the compactness sieve only	32.16	45.89%
Apply the above three sieves	6.89	88.41%

Table 8 Comparison between the Kalman filter-based algorithm and our algorithm (#false: number of false positive, CT: computing time)

Ball tracking	KF algorithm				Our algorithm			
	#correct	#false	accuracy	CT (ms)	#correct	#false	accuracy	CT (ms)
TWN-KOR	12890	775	81.46%	340308	13632	403	86.15%	264010
CHN-JPN	11818	701	79.66%	392012	12877	411	86.80%	292547
JPN-KOR	15427	1019	80.18%	449190	16964	506	88.17%	343944
Total	40135	2495	80.43%	1181510	43473	1320	87.12%	900501

and have strong limitation of view angles. In a way our proposed VIA system better meet the practical requirement and general users' needs.

10 Conclusions and future work

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 coaches 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. 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 coaches, players and the audience to have a novel insight into the game.

Every sport has its respective game rules, domain knowledge, court model and tactics. Therefore, tactics analysis is specially designed for a specific sport. The trajectory-based applications to tactics analysis presented in this paper cannot be directly applied to other sports. However, we propose a generalized approach for 2D ball tracking and 3D trajectory approximation. With domain knowledge-based adaption, the proposed approach can be readily applied to the sports which have sufficient court information captured in the video and require ball trajectory extraction for tactics analysis, such as basketball, soccer, tennis and table tennis. On the other hand, it is also considered as part of our future work to apply the proposed system to the video of higher resolution, and we believe that better performance can be achieved.

Table 9 Comparison between the famous existing systems [13, 20, 21] and our proposed VIA system on 3D trajectory approximation

	ESPN K-Zone System [13]	Hawk-Eye System [20]	QuesTec UIS [21]	Our proposed VIA system
Avg. 3D error	1.016 cm	0.36 cm	1.27 cm	14 cm
# of camera(s)	2	6	4	1
High speed camera required?	yes	yes	yes	no

Generally speaking, not all of the sports games are broadcasted on TV. It is a growing trend that the coaches 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 for the user-captured sports video like the proposed VIA system. The main difference between the user-captured video and the broadcast video is that the former has no camera motion while the latter has. To avoid the effect of camera motion, in this paper we use user-captured video to verify the proposed 3D trajectory approximation approach. The experiments show convincing results.

The limitation of our proposed system is that only the video sequences captured by a fixed camera are analyzed. For broadcast video sequences, camera motion increases the complexity in moving object segmentation. The frame differencing method is no longer adequate to segment objects when the background keeps changing. On the other hand, the current system considers the physical effect of gravity acceleration to model 3D ball trajectories as parabolic curves. However, there are still other factors affecting the ball motion, such as air friction, ball spin axis, ball spin rate, etc. In the future, we will take camera motion into consideration for ball tracking in broadcast video sequences. Moreover, we will involve more physical factors to model the 3D ball trajectory more precisely.

Acknowledgement 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. Assfalg J, Bertini M, Colombo C, Bimbo AD, Nunziati W (2003) Semantic annotation of soccer videos: automatic highlights identification. *Comput Vis Image Underst* 92(23):285–305
2. Cheng CC, Hsu CT (2006) Fusion of audio and motion information on HMM-based highlight extraction for baseball games. *IEEE Trans Multimed* 8(3):585–599
3. Chen HT, Hsiao MH, Chen HS, Tsai WJ, Lee SY (2008) A baseball exploration system using spatial pattern recognition. In: *Proc IEEE Int Symp Circuits and Systems* 2008:3522–3525
4. Chen HT, Chen HS, Hsiao MH, Tsai WJ, Lee SY (2008) A trajectory-based ball tracking framework with enrichment for broadcast baseball videos. *J Inf Sci Eng* 24(1):143–157
5. Chen HT, Tien MC, Chen YW, Tsai WJ, Lee SY (2009) Physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video. *J Vis Commun Image Representation* 20(3):204–216
6. Chen HT, Chen HS, Lee SY (2007) Physics-based ball tracking in volleyball videos with its applications to set type recognition and action detection. In: *Proc. IEEE Int. Conf. on Acoustic Speech Signal Process* 2007, pp. 1-1097–1100
7. Duan LY, Xu M, Tian Q (2005) A unified framework for semantic shot classification in sports video. *IEEE Trans Multimed* 7(6):1066–1083
8. Duan LY, Xu M, Chua TS, Tian Q, Xu CS (2003) A mid-level representation framework for semantic sports video analysis. In: *Proc. 11th ACM Int. Conf. Multimedia*, pp.33–44
9. Ekin A, Tekalp AM, Mehrotra R (2003) Automatic soccer video analysis and summarization. *IEEE Trans Image Process* 12(7):796–807
10. Farin D, Krabbe S, Peter HN, Effelsberg W (2004) Robust camera calibration for sport videos using court models. *SPIE Storage Retrieval Meth Appl Multimedia* 5307:80–91
11. Farin D, Han J, Peter HN (2005) Fast camera calibration for the analysis of sport sequences. In: *Proc IEEE Int Conf Multimedia and Expo* 2005:482–485
12. Forlines C, Peker KA, Divakaran A (2006) Subjective assessment of consumer video summarization. In: *Proc. SPIE Int. Soc. Opt. Eng.* (6073), pp. 170–177
13. Gueziec A (2002) Tracking pitches for broadcast television. *Computer* 35:38–43
14. Gonzalez RC, Woods RE (2002), *Digital image processing*, Prentice Hall (2nd edition)
15. Hartley R, Zisserman A (2003) *Multiple view geometry in computer vision*, 2nd edn. Cambridge University Press, UK

16. Lu H, Tan YP (2003) Unsupervised clustering of dominant scenes in sports video. *Pattern Recogn Lett* 24(15):2651–2662
17. Loui A, Luo J, Chang S, Ellis D, Jiang W, Kennedy L, Lee K, Yanagawa A (2007) Kodak 's consumer video benchmark data set: concept definition and annotation. In: *Proc Int Workshop Multimedia Inform Retrieval 2007*:245–254
18. Mei T, Hua XS (2008) Structure and event mining in sports video with efficient mosaic. *Multimedia Tools Appl* 40(1):89–110
19. Oami R, Benitez AB, Chang SF, Dimitrova N (2004) Understanding and modeling user interests in consumer videos. In: *Proc IEEE Int Conf Multimedia and Expo 2004*:1475–1478
20. Owens N, Harris C, Stennett C (2003) Hawk-eye tennis system. In: *Proc Inf Conf Visual Information Engineering 2003*:182–185
21. QuesTec-Umpire Information System. [Online]. Available: http://www.questec.com/q2001/prod_uis.htm
22. Seo Y, Choi S, Kim H, Hong KS (1997) Where are the ball and players? Soccer game analysis with color-based tracking and image mosaic In: *Proc Image Analysis and Processing 1997*(1331):196–203
23. Tien MC, Chen HT, Chen YW, Hsiao MH, Lee SY (2007) Shot classification of basketball videos and its applications in shooting position extraction. In: *Proc. IEEE Int. Conf. Acoust. Speech Signal Process 2007*, pp. I-1085–1088.
24. Watanabe T, Haseyama M, Kitajima H (2004) A soccer field tracking method with wire frame model from TV images. In: *Proc IEEE Int Conf Image Process 2004*:1633–1636
25. Xu M, Maddage NC, Xu C, Kankanhalli M, Tian Q (2003) Creating audio keywords for event detection in soccer video. In: *Proc. IEEE Int. Conf. Multimedia and Expo 2003*, pp. II-281–284
26. Xu M, Duan L, Chia L, Xu C (2004), Audio keyword generation for sports video analysis. In: *Proc. 12th Annual ACM Int. Conf. Multimedia*, pp.758–759
27. Yu X, Xu C, Leong HW, Tian Q, Tang Q, Wan KW (2003) Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In: *Proc. 11th ACM Int. Conf. Multimedia*, pp.11–20
28. Yu X, Leong HW, Xu C, Tian Q (2006) Trajectory-based ball detection and tracking in broadcast soccer video. *IEEE Trans Multimed* 8(6):1164–1178
29. Yu X, Jiang N, Cheong LF, Leong HW, Yan X (2008) Automatic camera calibration of broadcast tennis video with applications to 3 d virtual content insertion and ball detection and tracking. *Comput Vis Image Underst* 113(5):643–652
30. Yu X, Jiang N, Cheong LF (2007) Accurate and stable camera calibration of broadcast tennis video. In: *Proc Int IEEE Conf Image Process 2007*:93–96
31. Yu X, Tu X, Ang EL (2007) Trajectory-based ball detection and tracking in broadcast soccer video with the aid of camera motion recovery. In: *Proc IEEE ICME 2007*:1543–1546
32. Zhu G, Huang Q, Xu C, Xing L, Gao W, Yao H (2007) Human behavior analysis for highlight ranking in broadcast racket sports video. *IEEE Trans Multimed* 9(6):1167–1182
33. Zhu G, Huang Q, Xu C, Rui Y, Jiang S, Gao W, Yao H (2007) Trajectory based event tactics analysis in broadcast sports video. In: *Proc. 15th ACM Int. Conf. Multimedia*, pp.58–67
34. Zhang T, Kuo J (2001) Audio content analysis for online audiovisual data segmentation and classification. *IEEE Trans Speech Audio Process* 9(4):441–457



Hua-Tsung Chen received the B.S. M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan in 2001, 2003 and 2009, respectively. He

is current Assistant Research Fellow with Information and Communications Technology Lab, National Chiao Tung University, Hsinchu, Taiwan. His research interests include computer vision, video signal processing, content-based video indexing and retrieval, multimedia information system and music signal processing.



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 from 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.



Jen-Yu Yu received the B.S. and M.S. degrees in Computer Science and Information Engineering from National Chiao Tung University in 2000 and 2002, respectively. He joined the Information and Communications Research Laboratories of the Industrial Technology Research Institute in 2003, where he is currently the deputy division director of division for Network-based Services Technology. His major research areas include mobile video streaming system, distributed multimedia system, peer-to-peer system, content-based retrieval and video compression.