Ren, Jinchang and Xu, M. and Orwell, J. and Jones, G. (2008) Real-time modeling of 3-D soccer ball trajectories from multiple fixed cameras. IEEE Transactions on Circuits and Systems for Video Technology, 18 (3). pp. 350-362. ISSN 1051-8215 , http://dx.doi.org/10.1109/TCSVT.2008.918276

Real-time Modeling of 3D Soccer Ball Trajectories from Multiple Fixed Cameras

Jinchang Ren, James Orwell, Graeme A Jones and Ming Xu

Abstract— In this paper, model-based approaches for real-time 3D soccer ball tracking are proposed, using image sequences from multiple fixed cameras as input. The main challenges include filtering false alarms, tracking through missing observations and estimating 3D positions from single or multiple cameras. The key innovations are: i) incorporating motion cues and temporal hysteressis thresholding in ball detection; ii) modeling each ball trajectory as curve segments in successive virtual vertical planes so that the 3D position of the ball can be determined from a single camera view; iii) introducing four motion phases (rolling, flying, in possession, and out of play) and employing phase-specific models to estimate ball trajectories which enables high-level semantics applied in low-level tracking. In addition, unreliable or missing ball observations are recovered using spatio-temporal constraints and temporal filtering. The system accuracy and robustness is evaluated by comparing the estimated ball positions and phases with manual ground-truth data of real soccer sequences.

Index Terms— Motion analysis, video signal processing, geometric modeling, tracking, multiple cameras, three-dimensional vision.

I. Introduction

With the development of computer vision and multimedia technologies, many important applications have been developed in automatic soccer video analysis and content-based indexing, retrieval and visualization [1-3]. By accurately tracking players and ball, a number of innovative applications can be derived for automatic comprehension of sports events. These include annotation of video content, summarization, team strategy analysis and verification of referee decisions, as well as the 2D or 3D reconstruction and visualization of action [3-16]. In addition, some more recent work on tracking of players and the ball can be also found in [27-29].

In a soccer match, the ball is invariably the focus of attention. Although players can be successfully detected and tracked on the basis of color and shape [1, 10, 12], similar methods cannot be extended to ball detection and tracking for several reasons. First, the ball is small and exhibits irregular shape, variable size and inconsistent color when moving rapidly, as illustrated in Figure 1. Second, the ball is frequently occluded by players or is out of all camera fields of view (FOV), such as when it is kicked high in the air. Finally, the ball often leaves the ground surface, and its 3D position cannot be uniquely determined without the measurements from at least two cameras with overlapping fields of view. Therefore, 3D ball position estimation and tracking is, arguably, the most important challenge in soccer video analysis. In this paper the problem under investigation is the automatic ball tracking from multiple fixed cameras.

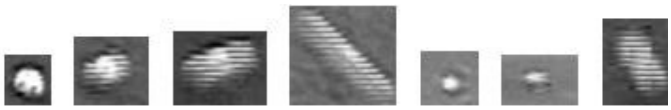


Fig. 1. Ball samples in various sizes, shapes and colors.

A. Related Work

Generally, TV broadcast cameras or fixed-cameras around the stadium are the two usual sources of soccer image streams. While TV imagery generally provides high resolution data of the ball in the image centre, the complex camera movements and partial views of the field, make it hard to obtain accurate camera parameters for on-field ball positioning. On the other hand, fixed cameras are easily calibrated, but their wide-angle field of view makes ball detection more difficult, since the ball is often represented by only a small number of pixels.

In the soccer domain, fully automatic methods for limited scene understanding have been proposed, e.g. recognition of replays from cinematic features extracted from broadcast TV data [1] and detection of the ball in broadcast TV data [1, 2, 4-9]. Gong et al adopted white color and circular shape to detect balls in image sequences [1]. In Yow et al [2], the ball is detected by template matching in each of the reference frames and then tracked between each pair of these reference frames. Seo et al applied template matching and Kalman filter to track balls after manual initialization [4]. Tong et al [5] employed indirect ball detection by eliminating non-ball regions using color and shape constraints. In Yamada et al [6], white regions

Manuscript received Dec 20, 2005. This work was supported in part by the European Commission under Project IST-2001-37422.

J. Ren is with School of Informatics, University of Bradford, BD7 1DP, U.K., on leave from the School of Computers, Northwestern Polytechnic University, Xi'an, 710072, China (email: j.ren@bradford.ac.uk; npurjc@yahoo.com).

J. Orwell and G. A. Jones are with Digital Imaging Research Centre, Kingston University, Surrey, KT1 2EE, U.K. (email: j.orwell@kingston.ac.uk; g.jones@kingston.ac.uk).

M. Xu is with Signal Processing Lab, Engineering Department, Cambridge University, CB2 1PZ, U.K. (email: mx204@cam.ac.uk).

are taken as ball candidates after removing of players and field lines. In Yu et al [7, 8], candidate balls are first identified by size range, color and shape, and then these candidates are further verified by trajectory mining with a Kalman filter. D'Orazio et al [9] detected the ball using a modified Hough transform along with a neural classifier.

Using soccer sequences from fixed cameras, usually there are two steps for the estimation and tracking of 3D ball positions. Firstly, the ball is detected and tracked in each single view independently. Then, 2D ball positions from different camera views are integrated to obtain 3D positions using known motion models [10-12]. Ohno et al arranged eight cameras to attain a full view of the pitch [10]. They modeled the 3D ball trajectory by considering air friction and gravity which depend on an unsolved initial velocity. Matsumoto et al [11] used four cameras in their optimized viewpoint determination system, in which template matching is also applied for ball detection. Bebie and Bieri [12] employed two cameras for soccer game reconstruction, and modeled 3D trajectory segments by Hermite spline curves. However, about one-fifth of the ball positions need to be set manually before estimation. In Kim et al [13] and Reid and North [14], reference players and shadows were utilized in the estimation of 3D ball positions. These are unlikely to be robust as the shadow positions depend more on light source positions than on camera projections.

B. Contributions of This Work

In this paper, a system is presented for model-based 3D ball tracking from real soccer videos. The main contributions can be summarized as follows.

Firstly, a motion-based thresholding process along with temporal filtering is used to detect the ball, which has proved to be robust to the inevitable variations in ball color and size that result from its rapid movement. Meanwhile, a probability measure is defined to capture the likelihood that any specific detected moving object represents the ball.

Secondly, the 3D ball motion is modeled as a series of planar curves each residing in a vertical virtual plane (VVP), which involves geometric based vision techniques for 3D ball positioning. To determine each vertical plane, at least two observed positions of the ball with reliable height estimate are required. These reliable estimates are obtained by either recognizing a bouncing on the ground from single view, or triangulating from multiple views. Based on these VVPs, the 3D ball positions are determined in single camera views by projections. Ball positions for frames without any valid observations are easily estimated by polynomial interpolation to allow a continuous 3D ball trajectory to be generated.

Thirdly, the ball trajectories are modeled as one of four phases of ball motion – rolling, flying, in-possession and out-of-play. These phase types were chosen because they each require different models in trajectory recovery. For the first two types, phase-specific models are employed to estimate ball positions in linear and parabolic trajectories, respectively. It is shown how two 3D points are sufficient to estimate the parabolic trajectory of a flying ball. In addition, the transitions from one phase to another also provide useful semantic insight into the progression of the game, i.e. they coincide with the passes, kicks etc. that constitute the play.

C. Structure of the Paper

The remaining part of the paper is organized as follows. In Section II, the method we used for tracking and detecting moving objects is described, using Gaussian mixtures [17] and calibrated cameras [19]. In Section III, a method is presented for identifying the ball from these objects. These methods operate in the image plane from each camera separately. In Section IV, the data from multiple cameras is integrated, to provide a segment-based model of the ball trajectory over the entire pitch, estimating 3D ball positions from either single view or multiple views. In Section V, a technique is introduced for recognizing different phases of ball motion, and for applying phase-specific models for robust ball tracking. Experimental results are presented in Section VI and the conclusions are drawn in Section VII.

II. Moving Objects Detection and Tracking

To locate and track players and the soccer ball, a multi-modal adaptive background model is utilized which provides robust foreground detection using image differencing [17]. This detection process is applied only to visible pitch pixels of the appropriate color. Grouped foreground connected-components (i.e. blobs) are tracked by a Kalman filter which estimates 2D position, velocity and object dimensions. These 2D positions and dimensions are converted to 3D coordinates on the pitch. Greater detail is given in the subsections below.

A. Determining Pitch Masks

Rather than process the whole image, a pitch mask is developed to avoid processing pixels containing spectators. This mask is defined as the intersection of the geometry-based mask M_g and the color-based mask M_c, as shown in Figure 2. The former constrains processing to only those pixels on the pitch, and can be easily derived from a coordinate transform of the position of the pitch in the ground plane to the image plane as follows. For each image pixel p, compute its corresponding ground-plane point P. If P locates within the pitch, then p is set to 255 in M_g, otherwise 0. Note, however, that parts of the pitch can be occluded by foreground spectators or parts of the stadium. Thus, a color-based mask is used to exclude these elements from the overall pitch mask (i.e. the region to be processed).

The hue component of the HSV color space is used to identify the region of the background image representing the pitch, since it is robust to shadows and other variations in the appearance of the grass. As it is assumed that the pitch region has an approximately uniform color and occupies the dominant area of the background image, pixels belonging to the pitch will contribute to the largest peak in any hue histogram. Lower and upper hue thresholds H_1 and H_2 delimit an interval around the position H_0 of this maximum. Defined as the positions at

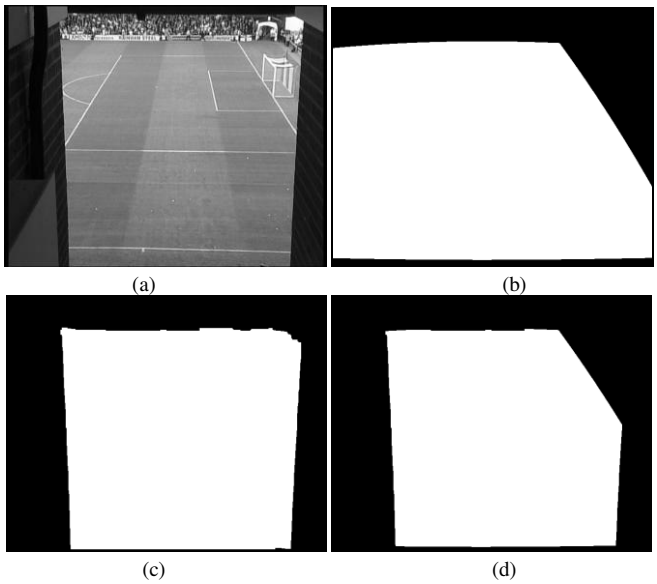
which the histogram has decreased by 90% of the peak frequency, image pixels contributing to this interval are included in the color-based mask $\mathrm{M_c}$.

A morphological closing operation is performed on $\mathrm{M_c}$ to bridge the gaps caused by the white field lines in the initial color-based mask. Thus the final mask, M, can be generated as follows:

$$\mathrm{M_c} = \{(u,v) \mid H(u,v) \in [H_1, H_2]\} \circ B \quad (1)$$

$$\mathrm{M} = \mathrm{M_g} \bigcap \mathrm{M_c} \quad (2)$$

where the morphological closing operation is denoted by $\circ$ and B is its square structuring element of size 6×6.



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

Fig. 2. Extraction of pitch masks based on both color and geometry: (a) Original background image, (b) Geometry-based mask of pitch, (c) Color-based mask of pitch, and (d) Final mask obtained.

B. Detecting Moving Objects

Over the mask M detected above, foreground pixels are located using the robust multi-modal adaptive background model [17]. Firstly, an initial background image is determined by a per-pixel Gaussian Mixture Model, and then the background image is progressively updated using a running average algorithm for efficiency.

Each per-pixel Gaussian Mixture Model is represented as $(\boldsymbol{\mu}_k^{(j)}, \sigma_k^{(j)}, \omega_k^{(j)})$, where $\boldsymbol{\mu}_k^{(j)}, \sigma_k^{(j)}$ and $\omega_k^{(j)}$ are the mean, root of the trace of covariance matrix, and weight of the j^{th} distribution at frame k. The distribution which matches each new pixel observation $\mathbf{I}_k$ is updated as follows:

$$\begin{cases} \boldsymbol{\mu}_k = (1-\rho)\boldsymbol{\mu}_{k-1} + \rho \mathbf{I}_k \\ \sigma_k^2 = (1-\rho)\sigma_{k-1}^2 + \rho(\mathbf{I}_k - \boldsymbol{\mu}_k)^T(\mathbf{I}_k - \boldsymbol{\mu}_k) \end{cases} \quad (3)$$

where ρ is the updating rate satisfying $0 < \rho < 1$. For each unmatched distribution, the parameters remain the same but its weight decreases. The initial background image is selected as the distribution with the greatest weight at each pixel.

Given the input image $\mathbf{I}_k$, the foreground binary mask $\mathrm{F_k}$ can be generated by comparing $\| \mathbf{I}_k - \boldsymbol{\mu}_{k-1} \|$ against a threshold, i.e. $2.5\sigma_k$. To accelerate the process of updating the background image, a running average algorithm is further employed after the initial background and foreground have been estimated:

$$\boldsymbol{\mu}_k = [\alpha_L \mathbf{I}_k + (1-\alpha_L)\boldsymbol{\mu}_{k-1}]\mathrm{F_k} + [\alpha_H \mathbf{I}_k + (1-\alpha_H)\boldsymbol{\mu}_{k-1}]\overline{\mathrm{F}}_k \quad (4)$$

where $\overline{\mathrm{F}}_k$ is the complement of $\mathrm{F_k}$. The use of two update weights (where $0 < \alpha_L << \alpha_H << 1$) ensures that the background image is updated slowly in the presence of foreground regions. Updating is required even when a pixel is flagged as moving to allow the system to overcome mistakes in the initial background estimate.

Inside these foreground masks, a set of foreground regions are generated using connected component analysis. Each region is represented by its centroid (r_0, c_0), area a, and bounding box where (r_1, c_1) and (r_2, c_2) are the top-left and bottom-right corners of the bounding box.

C. Tracking Moving Objects

A Kalman tracker is used in the image plane to filter noisy measurements and split merged objects because of frequent occlusions of players and the ball. The state $\mathbf{x}_I$ and measurement $\mathbf{z}_I$ are given by:

$$\mathbf{x}_I = [r_0 \quad c_0 \quad \dot{r}_0 \quad \dot{c}_0 \quad \Delta r_1 \quad \Delta c_1 \quad \Delta r_2 \quad \Delta c_2]^T \quad (5)$$

$$\mathbf{z}_I = [r_0 \quad c_0 \quad r_1 \quad c_1 \quad r_2 \quad c_2]^T \quad (6)$$

where (r_0, c_0) is the centroid, $(\dot{r}_0, \dot{c}_0)$ is the velocity, (r_1, c_1) and (r_2, c_2) are the top-left and bottom-right corners of the bounding box respectively (such that $r_1 < r_2$ and $c_1 < c_2$) and $(\Delta r_1, \Delta c_1)$ and $(\Delta r_2, \Delta c_2)$ are the relative positions of the two opposite corners to the centroid. The state transition and measurement equations in the Kalman filter are:

$$\begin{aligned} \mathbf{x}_I(k+1) &= \mathbf{A}_I \mathbf{x}_I(k) + \mathbf{w}_I(k) \\ \mathbf{z}_I(k) &= \mathbf{H}_I \mathbf{x}_I(k) + \mathbf{v}_I(k) \end{aligned} \quad (7)$$

where $\mathbf{w}_I$ and $\mathbf{v}_I$ are the image plane process noise and measurement noise, and $\mathbf{A}_I$ and $\mathbf{H}_I$ are the state transition matrix and measurement matrix, respectively.

$$\mathbf{A}_I = \begin{bmatrix} \mathbf{I}_2 & \Delta T \cdot \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix}$$

$$\mathbf{H}_I = \begin{bmatrix} \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 \\ \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \quad (8)$$

In equation (8), $\mathbf{I}_2$ and $\mathbf{O}_2$ represent 2×2 identity and zero metrics; ΔT is the time interval between frames. Further detail on the method for data association and handling of occlusions can be found in [18].

D. Computing Ground Plane Positions

Using the Tsai's algorithm for camera calibration [19], the measurements are transformed from image co-ordinates into world co-ordinates. Basically, the pin-hole model of 3D-2D perspective projection is employed in [19] to estimate totally 11 intrinsic and extrinsic camera parameters. In addition, effective dimensions of pixel in images are obtained in both horizontal and vertical directions as two fixed intrinsic constants. These two constants are then taken to calculate the world co-ordinates measurements of the objects on the basis of detected image-plane bounding boxes. Let (x, y, z) denote the 3D object position in world co-ordinates, then x and y are estimated by using the center point of the bottom line of each bounding box, and z initialized as zero. Until Section IV, all objects are assumed to lie on the ground plane. (This assumption is usually true for players, but the ball could be anywhere on the line between that ground plane point and the camera position). For each tracked object, a position and attribute measurement vector is defined as $\mathbf{p}_i = [x \quad y \quad z \quad v_x \quad v_y]^T$ and $\mathbf{a}_i = [w \quad h \quad a \quad n]^T$. In addition, a ground plane velocity (v_x, v_y) is estimated from the projection of the image-plane velocity (which is obtained from the image plane tracking process) onto the ground plane. Note that this ground-plane velocity is not intended to estimate the real velocity, in cases where the ball is off the ground. The attributes w, h and a are an object's width, height and area, also measured in meters (and meters squared), and calculated by assuming the object touches the ground plane. Besides, each object is validated before further processing provided that its size satisfies $w \geq 0.1\text{m}$, $h \geq 0.1\text{m}$ and $a \geq 0.03\text{m}^2$. Finally, n is the longevity of the tracked object, measured in frames.

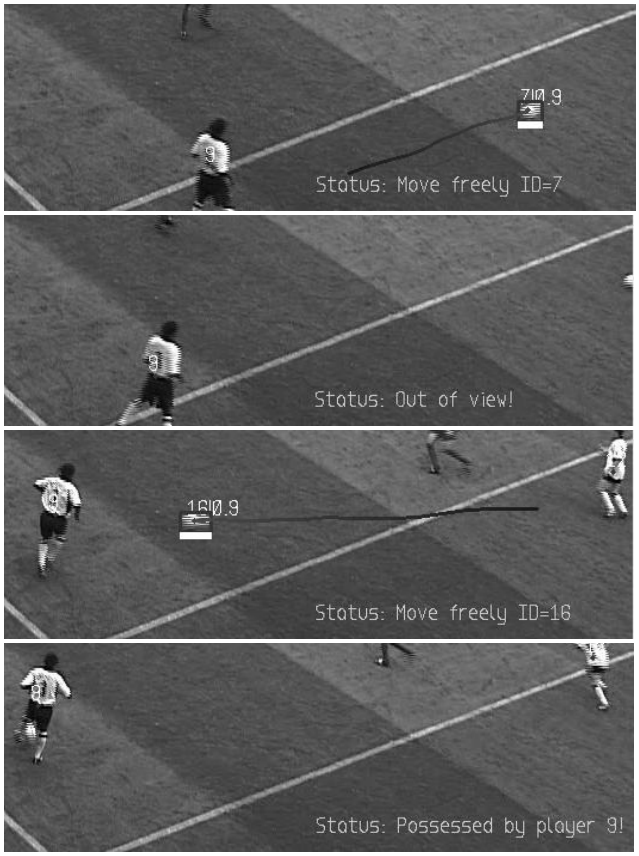
III. Detecting Ball-like Features

To identify ball-like features in a single-view process, each of the tracked objects is attributed with a likelihood l that represents the ball. The two elementary properties to distinguish the ball from players and other false alarms are its size and color. Three simple features are used to describe the size of the object, i.e. its width, height, and area, in which measurements in real-world units are adopted for robustness against variable sizes of the ball in image plane. A fourth feature derived from its color appearance, measures the proportion of the object's area that is white.

To discriminate the ball from other objects, a straightforward process is to apply fixed thresholds to these features. However, this suffers from several difficulties. Firstly, false alarms such as fragmented field lines or fragments of players (especially socks) cannot always be discriminated. Secondly, if no information is available about the height of the ball, the estimate of the dimensions may be inaccurate. For example, by assuming the ball is touching the ground plane, an airborne ball will appear to be a larger object. Thirdly, the image of a fast-moving ball is affected by motion blurring, rendering it larger and less white than a stationary (or slower moving) ball.

A key observation from soccer videos is that the ball in play is nearly always moving, which suggests that the velocity may be a useful additional discriminant. Thus, as field markings are stationary the majority of these markings can be discriminated from the ball by thresholding both the size and absolute velocity of the detected object.

Another category of false alarms is caused by a part of a player that has become temporarily disassociated from the remainder of the player. A typical cause of this phenomenon is imperfect foreground segmentation. However, such transitory artifacts do not in general persist for longer than a couple of



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

Fig. 3. Tracked ball with ID and assigned likelihood (a) Id=7, l=0.9 (b) l=0.0, the ball is moving out of current camera view (c) Id=16, l=0.9 and (d) ball is merged with player 9 in frame #977, #990, #1044, and #1056, respectively.

frames, whereupon the correct representation is resumed. Therefore, this category of false alarm can be correctly discriminated by discarding all short-lived objects, i.e. whose longevity is less than five frames.

Features describing the velocity and longevity of the observations are used to solve the three difficulties described above. These features (derived from tracking) are employed alongside size and color features to help discriminate the ball from other objects. The velocity feature is also useful when the size of the detected ball is overestimated, either through a motion-blur effect (proportional to the duration of the shutter-speed), or a range error effect (incorrectly assuming the object lies on the ground plane). Here, the key innovation is to allow the size threshold to vary as a function of the estimated ground-plane velocity. There is a simple rationale for the motion-blur effect: the expected area is also directly proportional to the image-plane speed. The range error effect is more complicated as the 3D trajectory of the ball may be directly towards the camera generating zero velocity in the image plane. However, in general it can be assumed that the ball rapidly moving in the image plane is more likely to be positioned above the ground plane, and therefore, the size threshold should be increased to accommodate the consequent over-estimation of the ball size.

As for a standard soccer ball, it has a constant diameter d_0 (between 0.216m and 0.226m) and an area (of a great circle) a_0 about 0.04m^2. Considering over-estimated ball size during fast movement, two thresholds for the width and height of the ball, w_0 and h_0, are defined by

$$\begin{aligned} w_0 &= d_0 + |v_x| \Delta T \\ h_0 &= d_0 + |v_y| \Delta T \end{aligned} \tag{9}$$

For robustness, valid size ranges of the ball are required satisfying $|w - w_0| < d_0/5$, $|h - h_0| < d_0/5$, and $|a - a_0| < a_0/8 + |v_x v_y| (\Delta T)^2$. In addition, the proportion of white color within the object is required no less than 30% of the whole area. All objects having size and color outside the prescribed thresholds are assigned a likelihood of zero and excluded from further processing. Each remaining object is classed as a ball candidate, and assigned an estimate of the likelihood that represents the ball. The proposed form for this estimate is the following equation, incorporating both its absolute velocity $\|\mathbf{v}_i\|$ and longevity n:

$$l_i = \frac{\|\mathbf{v}_i\|}{v_{max}} (1 - e^{-nt_0}) \tag{10}$$

where v_{max} is the maximum absolute velocity of all the objects detected in the given camera, at a given frame (including the ball, if visible, and also non-ball objects), and t_0 is a constant parameter. Thus, faster moving objects are considered more likely to be the ball based on the fact that, in the professional game, the ball normally moves faster than other objects.

Figure 3 shows partial views of camera #1 with detected ball at frame 977, 990, 1044 and 1056, respectively. The ball or each player is assigned with a unique ID unless it is near the

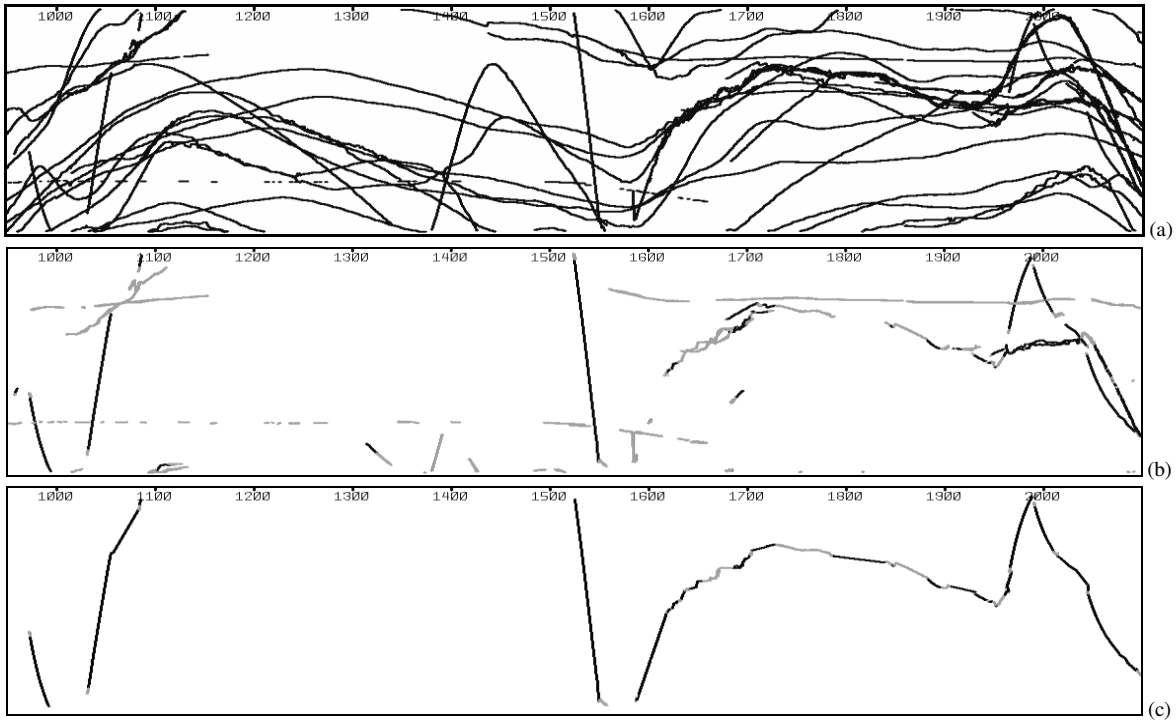


Fig. 4. Thirty seconds of single camera tracking data from camera #1 (a) and filtered results of the ball in (b) and (c), in which time t moves from left to right, and the x-coordinate of the objects c_0 is plotted up the y-axis. In (b) and (c), most-likely ball is labeled in black, (b) is the result filtering on appearance and velocity and (c) is the result after temporal filtering.

boundary of the camera view or merged with another object. When the ball is moving, its trajectory (recent history of centroid positions) is plotted, too. Text output is also utilized to show the tracking status.

Since the ball is frequently missed in detection, due to occlusion or in-possession, the proposed approach includes a temporal filter of the ball likelihood. The filter uses hysteresis to process the likelihood estimates into discrete labels, in an approach similar to the Canny filter [20]. Three thresholds h_1, h_2, h_3 are used, where $h_1 > h_2 > h_3$. Candidates with a likelihood above h_1 are unequivocally designated a 'ball' label; and candidates with a likelihood below h_3 are unequivocally classified as 'not ball' (i.e. false alarms). The filtering process iteratively examines likelihood values along the tracked trajectory: observations with likelihood l in the interval $h_1 > l > h_2$ are labeled as a 'ball' if there is an object has been labeled as a ball in its neighboring frame. Similarly, objects with likelihood l in the interval $h_2 > l > h_3$ are labeled as 'not ball' if the object has that label as not ball in a neighboring frame. This process is iterated for N_B frames, where N_B is the buffer size for the filter. This temporal filter significantly improves the robustness of detection and continuity of trajectory.

Figure 4 plots the positional column- coordinate c_0 of the trajectories of multiple objects tracked over the frame #950 to #2100 in camera sequence #1. The original trajectories before ball detection is shown in Figure 4(a), and the result of appearance and velocity filter is shown in Figure 4(b), which still include some false alarms. In this sequence, application of the temporal filter successfully locates the ball among these various candidates.

The above process is executed on the data from each camera, and the most likely ball candidate from each is input to the second processing stage, described below in Sections IV, V and VI, in which these observations are combined to estimate the height, phase and trajectory of the motion.

IV. Model Based 3D Position Estimation in Single and Multiple Views

In this section, the detection results of the ball from all single views are integrated for estimation of 3D position. If the ball is located on the ground, the conversion from 2D image co-ordinates to real world 3D co-ordinates is completely determined using the camera calibration parameters. Otherwise, the 2D image position can only provide constraints for the 3D line on which, somewhere, the ball is located. After a segment-based model of the ball motion is presented, two methods are provided for determining 3D ball positions. The first method is for cases in which the ball is detected from only one camera: the instant when the ball bounces on the ground is detected and the corresponding 3D position is estimated as zero. The second is for cases in which the ball is visible from at least two cameras, thus integration from multiple observations are used.

A. The Ball Motion Model

During a soccer game, the ball is moving regularly from one place to another. Its direction will change suddenly if and only if it touches the ground, a player, or a goal post, etc. It is assumed that the ball trajectory between two bounces or kicks forms a curve in a vertical virtual plane. In a special case when the ball is rolling on the ground, the curve will become a straight line. Therefore, the ball movement can be modeled as comprising a series of virtual vertical planes. In each vertical plane π, the ball will generate a single trajectory curve. The complete ball trajectory can be modeled as a sequence of adjacent planar curve segments. If it is assumed that there is no air resistance, then each plane will correspond to a single flight made by the ball. While beyond the scope of this paper, if the ball is struck to impart significant spin about an axis, then it will 'swerve' in the air and the assumption that the ball travels in a vertical plane is invalid, although the 'swerve' may be approximated by several segments, each defined by a vertical plane.

To estimate a virtual vertical plane (VVP), at least two estimates of 3D ball positions on the vertical plane must be available. These estimated 3D ball positions are described as fully determined estimates, in contract to most observations, which are only determined up to a line passing through the camera focal point. If r and s are two fully determined estimates hypothesized to lie in virtual plane π, the plane π can be simply determined as follows. Firstly, locate points r' and s' on the ground plane β with $rr' \perp \beta$ and $ss' \perp \beta$, then there is a line $r's'$ on β. Then, π is determined as the plane through $r's'$ and perpendicular to β.

The process for locating fully determined estimates from single and multiple views, and then a method for generating further 3D estimates within the corresponding VVP, are provided below.

B. Fully Determined Estimates from a Single View

From a single camera view, the strategy adopted for determining a 3D ball position, is to detect an occasion in which the ball bounces off some other object: players, ground or goal-post. If the height at which the bounce occurs can be estimated, then this height, together with its 2D image location, completely determines the 3D ball position at this time.

The ground-plane ball positions at frame n can firstly be obtained as $\mathbf{x}(n) = [x_n, y_n]^T$. Then, the velocity $\mathbf{v}(n) = [v_x, v_y]^T$ is defined as:

$$\mathbf{v}(n) = [\mathbf{x}(n+1) - \mathbf{x}(n)] / \Delta T \qquad (11)$$

where ΔT is the time interval between frames defined in Section II(C). To identify the bouncing ball, it is proposed to detect significant changes in the direction of this velocity at frame n:

$$\cos^{-1}\frac{\mathbf{v}(n-1)\cdot\mathbf{v}(n)}{\|\mathbf{v}(n-1)\|\cdot\|\mathbf{v}(n)\|}>\theta_0 \quad (12)$$

where $\theta_0 = 0.5$ is an appropriate threshold. Then, the height of the ball position is estimated as zero if there are no players or other objects near the ball. Otherwise, it is determined by the relative position of the ball and the object it touches. For example, it can be assumed the ball is two meters off the ground plane when it strikes a player's head.

C. Fully Determined Estimates from Multiple Views

When a ball is observed in multiple cameras, there are multiple projection lines from each camera position through the corresponding observation (which, in this application, can be terminated at the ground plane). The intersection point of these lines is normally taken as the estimate of the 3D ball position. However, false observations may exist which will lead to incorrect solutions. In the proposed method, firstly, 3D ball positions from each pair of cameras are estimated, and then the final estimate is calculated as the average of these estimated positions. However, some false estimates can be generated from the mis-association of the ball (in one camera) and e.g. some background clutter (from another camera). Thus, prior to this last step, some false estimates can be excluded by accepting only those estimates for which the calculated height is positive (actually, greater than a small negative value, -0.25m, to account for the small calibration error between two cameras, and less than the height of the cameras above the ground), and also estimated positions within the virtual pitch.

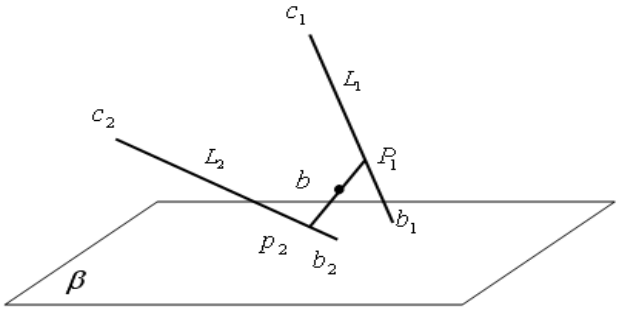


Fig. 5. 3D ball b estimation from cameras c_1 and c_2 with projected ball positions b_1 and b_2.

If the ball b is observed from two cameras c_1 and c_2 with projected positions b_1 and b_2 on the ground plane β as shown in Figure 5, b is estimated from b_1, b_2, c_1 and c_2. Let $L_1 = \overline{b_1 c_1}$ and $L_2 = \overline{b_2 c_2}$. However, L_1 and L_2 usually have no intersection point due to errors caused by image-plane measurement and inaccurate camera calibration. Therefore, it is proposed to use the solution for b which minimizes the sum of squared distances to both lines.

Let p_1 and p_2 be two points on lines L_1 and L_2, respectively, so that line $\overline{p_1 p_2}$ be a common perpendicular of L_1 and L_2. Then b should be on the line $\overline{p_1 p_2}$. Suppose $f_1[\cdot]=0$ and $f_2[\cdot]=0$ are the equations for the two lines L_1 and L_2, then the points p_1 and p_2 can be determined by:

$$f_1[X_{p_1}]=0 \quad (13)$$

$$f_2[X_{p_2}]=0 \quad (14)$$

$$[X_{p_1}-X_{p_2}]\cdot[X_{c_1}-X_{b_1}]=0 \quad (15)$$

$$[X_{p_1}-X_{p_2}]\cdot[X_{c_2}-X_{b_2}]=0 \quad (16)$$

where $X_{p_1}, X_{p_2}, X_{c_1}, X_{c_2}$ and X_b are the 3D positions of p_1, p_2, c_1, c_2 and b respectively. When the different measurement covariances for p_1 and p_2 are considered, the distances from b to p_1 and p_2 are changed into Mahalanobis distances. The measurement covariance for p_1 and p_2 is inversely proportional to its distance to the underlying camera. As a result, the estimated position of b is automatically biased to the viewing rays of the closer cameras which has the more accurate ball measurement. Assume δ_1 and δ_2 are the position covariance of the ball in cameras c_1 and c_2 respectively. Let $\lambda_1 = \delta_1 \| X_{c_1} - X_{b_1} \|^2$ and $\lambda_2 = \delta_2 \| X_{c_2} - X_{b_2} \|^2$, thus the final 3D ball position along $\overline{p_1 p_2}$ can then be estimated as

$$X_b = \frac{X_{p_1}\lambda_2 + X_{p_2}\lambda_1}{\lambda_1+\lambda_2} \quad (17)$$

For simplicity, b can be set as the middle-point of p_1 and p_2, i.e. $X_b = (X_{p_1} + X_{p_2})/2$.

D. Estimating Internal Ball Positions from a Single View

When the location of a virtual plane π is determined, the 3D positions of the ball observed in a single view can be recovered as internal ball positions within the corresponding

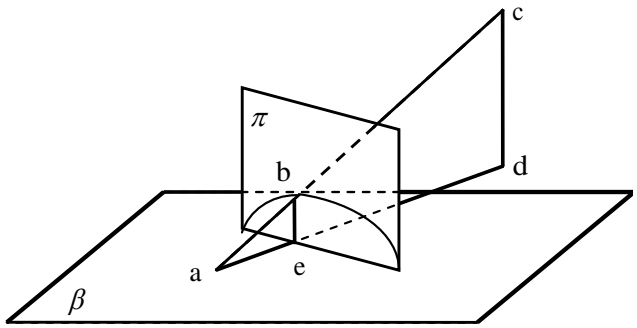


Fig. 6. Geometric relationship among the camera position c, the ball position b, as well as vertical and ground planes π and β.

curve-segment on π. This is achieved by triangulation from the ground-plane projection of the image-plane observations, as illustrated in Figure 6.

Let us represent the world coordinates of a point p by $X_p = (x_p, y_p, z_p)$. In Figure 6, c is the camera position; a is the projection of an image-plane ball observation on the ground plane; b is the required 3D position of the ball and located on the vertical plane; d and e are the normal projections of c and b onto the ground plane, respectively. Therefore, b is given from the intersection point of plane π and line $\overline{ac}$. The coordinates of points a, b, c, d and e in Figure 6 satisfy $z_a = 0$, $(x_d, y_d, z_d) = (x_c, y_c, 0)$, $(x_e, y_e, z_e) = (x_b, y_b, 0)$.

With a known plane π and points a and d, x_b and y_b can be recovered from e as the intersection point of π and line $\overline{ad}$. As the two triangles Δacd and Δabe are similar, hence

$$\frac{\|X_b - X_e\|}{\|X_c - X_d\|} = \frac{\|X_a - X_e\|}{\|X_a - X_d\|}, \tag{18}$$

and since $\|X_b - X_e\| = z_b$ and $|cd| = z_c$, then z_b can be expressed as:

$$z_b = \frac{\|X_a - X_e\|}{\|X_a - X_d\|} \cdot z_c . \tag{19}$$

Thus the 3D ball position X_b has been recovered.

E. Estimation of Missed or Uncertain Ball Positions

For those frames without ball observations in any single view or with ball observations of lower likelihood, i.e. less than a given threshold, the 3D ball positions are estimated by using polynomial interpolation in a curve on the corresponding vertical planes (see Section V). In this work, each curve is calculated from two fully determined estimates. If more fully determined estimates are available, then they could all be incorporated into the estimation of the trajectory based on a more general least squares estimator [25].

V. Recognition of Ball Motion Phases and Phase-Specific Trajectory Estimation

A. Four Phases of Ball Motion

In this work, it is proposed to model the ball motion at each instant into four phases, namely rolling (R), flying (F), in-possession (P) and out-of-play (O). A different tracking model is applicable to each phase, and furthermore the designation also provides a useful insight into the semantic progression of the game. In most cases the progression of play is reasonably straightforward to annotate, as a chain of transitions e.g. {P|F|O|P|R|P…} among these phases. A sequence of play can be annotated according to these four definitions and the transition graph given in Figure 7. However, there are sometimes ambiguities in the interpretation, e.g. between flying and rolling phases or in deciding how many touches of the ball constitute a possession.

Though some other semantic events have been analyzed for soccer video understanding [21-24], they are focused on players' motion in broadcasting context, yet phase transitions in the ball trajectory have not been discussed. For this model, it is simpler to denote even a single touch of the ball by a player as a frame of in-possession phase. This is because in-possession phases act as special periods that initialize other phases (such as rolling or flying), i.e. literally kicking the ball off in a particular direction. In fact, the ball trajectories for rolling and flying phases are determined by the last kick of a player during the preceding in-possession phase (notwithstanding gusts of wind, and other noise processes). Furthermore, the pattern of play is punctuated by periods when the ball is out-of-play, e.g. caused by fouls, ball crossing touchline, off-side or in-possession by the goal-keeper. Thus, the pattern of play is described as a list of phase-chains, always being started by an in-possession phase, and ending in an out-of-play phase. When the ball is out of play, it will be reinitialized (through several football events like throw-in, corner-kick etc.) for another cycle of phase transitions.

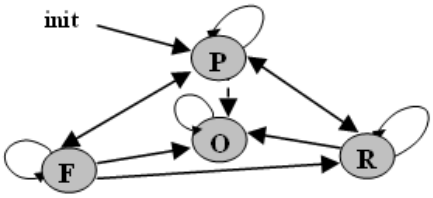


Fig. 7. Phase transition graph in soccer ball motion.

B. Estimating Motion Phases

Given observations of the ball from separate cameras and height cues obtained as described in Section IV, what follows is the estimation of the current ball phase. Prior to this stage, at each frame there is at most one estimate of ball position from each of the camera views, and each estimate is assigned a measure of the likelihood that it represents the ball. A 'soft' classification [26] of the four phases is introduced, which is then input into a decision process to determine the final estimate of the phase. These estimates are written as $G(\text{flying})$, $G(\text{rolling})$, $G(\text{in\_possession})$ and $G(\text{out\_of\_play})$.

Let (x_w, y_w, z_w) denote the final position of the ball and l_{max} the maximum likelihood among all the observations from multiple cameras. In this work, three distances are used to estimate the phase of the motion, i.e. the height z_w, distance to nearest player d_{min_p}, and distance from the edge of the playfield d_{min_b}. Smooth functions are chosen to provide a measure, bounded between 0 and 1, of the membership of each motion phase.

Firstly, $G(\text{flying})$ is determined by comparing the height of the ball to a threshold height z_f as

$$G(\text{flying}) = \begin{cases} 1 & \text{if} \quad z_w \geq z_f \\ \sin(\dfrac{\pi \cdot z_w}{2z_f}) & \text{else} \end{cases} \tag{20}$$

If $l_{max} \geq h_2$ (where h_2 is firstly defined for temporal filtering of ball likelihood in Section III), it refers to a separate ball observation hence $G(\text{in\_possession}) = 0$, and $G(\text{rolling})$ is decided by

$$G(\text{rolling}) = \begin{cases} 0 & \text{if} \quad z_w \geq z_f \\ \cos(\dfrac{\pi \cdot z_w}{2z_f}) & \text{else} \end{cases} \tag{21}$$

When $l_{max} < h_2$, this refers to a ball occluded or in-possession, and a pair of functions are then used to discriminate between in-possession and rolling phases, comparing the distance to the nearest player d_{min_p} with a scaling distance d_p:

$$G(\text{in\_possession}) = \begin{cases} 0 & \text{if} \quad d_{min\_p} \geq d_p \\ \cos(\dfrac{\pi \cdot d_{min\_p}}{2d_p}) & \text{else} \end{cases} \tag{22}$$

$$G(\text{rolling}) = \begin{cases} 1 & \text{if} \quad d_{min\_p} \geq d_p \\ \sin(\dfrac{\pi \cdot d_{min\_p}}{2d_p}) & \text{else} \end{cases} \tag{23}$$

Finally, the membership of the out of play phase is determined using a corresponding model, using d_{min_b}, the distance to the edge of the playfield:

$$G(\text{out\_of\_play}) = \begin{cases} 0 & \text{if} \quad d_{min\_b} > d_b \\ \cos(\dfrac{\pi \cdot d_{min\_b}}{2d_b}) & \text{else} \end{cases} \tag{24}$$

where $d_p = 0.4\text{m}$ and $d_b = 0.5\text{m}$ are both constant parameters.

The final designation of the motion phase at any instant is simply decided as the maximum of the four measures $G(\text{flying})$, $G(\text{rolling})$, $G(\text{in\_possession})$ and $G(\text{out\_of\_play})$. In the case of equal measures, the out of play phase has the priority, since the ball can still be flying or rolling when it moves out of the pitch.

If there is no ball observation, the phase of the previous frame is temporally maintained. For each of the in-play phases, a specific model is then employed for robust trajectory estimation below. While the above labeling formulations are somewhat arbitrary, their adoption has proved successful in accurately labeling the phases of play.

C. Phase-specific Trajectory Estimation

Finally, in this section, the three different in-play models of ball motion are described, starting with the flying trajectory. Disregarding air friction, the velocity parallel to the ground plane is constant and thus the ball follows a single parabolic trajectory. Let p_1 and p_2 be two known 3D ball positions on this trajectory such as the two fully determined estimates. The points $X_1 = (x_1, y_1, z_1)$ and $X_2 = (x_2, y_2, z_2)$ are the 3D co-ordinates of p_1 and p_2, and t_1 and t_2 are their corresponding moments in time.

Let $X = (x(t), y(t), z(t))$ denote the 3D position of the ball at time t. Disregarding all friction, $x(t)$ and $y(t)$ will satisfy the following equations, whether the ball is rolling or flying:

$$\begin{cases} x(t) = x_1 + \dfrac{x_2 - x_1}{t_2 - t_1}(t - t_1) \\ y(t) = y_1 + \dfrac{y_2 - y_1}{t_2 - t_1}(t - t_1) \end{cases} \tag{25}$$

Moreover, when the ball is rolling or in-possession, the approximation is made that $z(t) = 0$.

For the flying ball phase, the parabolic trajectory is decided by the two known 3D points p_1 and p_2 using the standard equation of motion under gravitational acceleration g.

$$z(t) = -\frac{g}{2}(t - t_1)(t - t_2) + \frac{z_2 - z_1}{t_2 - t_1}t + \frac{z_1 t_2 - z_2 t_1}{t_2 - t_1} \tag{26}$$

Moreover, if more than two ball positions have been decided within a curve-segment, then a least-squares calculation of the trajectory segment can be used to provide a more robust estimate [25].

VI. Experimental Results and Discussions

A. System Architecture

The proposed system was tested on data captured from matches played at Fulham Football Club, U.K. in the 2001 Premiership Season captured by eight fixed cameras. The camera positions are constrained by the layout of the stadium and the requirement to obtain the best resolution view of the football pitch - in particular the goalmouth. All the cameras are manually calibrated before tracking and 3D estimation. In the video processing stage, all the moving objects are detected and tracked in each camera to generate the ground plane positions of players (including the referee and linesmen) and the ball (by assuming it is on the ground). These 2D positions along with category information are collected as features and integrated by a centralized tracker. Using the multi-view information, 3D ball positions are then estimated in world coordinates on the basis of the proposed model. The 3D ball trajectory is

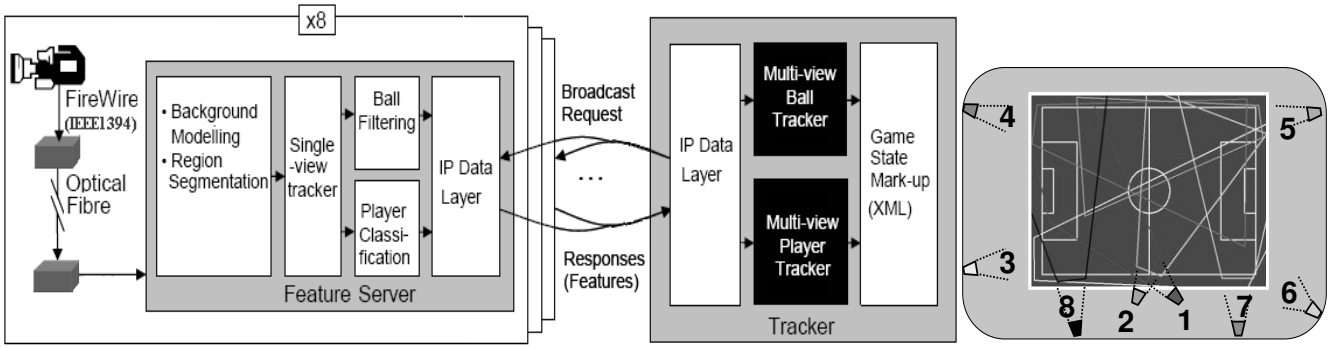


Fig. 8. Architecture of the proposed tracking system and arrangements of eight cameras with their field-of-views.

visualized along with tracked players on a virtual playfield. The final output generated will be a single model of the game state at a given time marked up in XML for third party applications e.g. the delivery of football services to specific cell-phone audiences. The system architecture and arrangements of the cameras with their field-of-views are shown in Figure 8.

All the cameras are connected to a rack of eight processors (named Feature Servers) through a network of fiber optics, with each optical fiber terminating in a centralized location that houses all of the processing hardware. An IP/Ethernet network is employed to connect these processors. For communication and synchronization, a "request-response" mechanism is utilized to manage eight simultaneous streams of data across the network. It is the multi-view tracker which is responsible for orchestrating the process by which (single-view) Feature Servers generate their results of features. During each iteration of the process, the tracker sends a single broadcast request with time stamp when it takes place. Then, all the Feature Servers will respond to this request by generating features using the latest frame, and these features will be naturally synchronized by the time stamp obtained from the tracker. Detailed discussion on player tracking and classification as well as communication and synchronization are given in [16].

Eight cameras were statically mounted around the stadium as described in Figure 8. All cameras recorded in mini DV format, in which four Canon XM1 cameras and four Sony cameras. The fields of view were adjusted to ensure all areas of the pitch were covered by at least one camera, which implied most cameras were almost fully zoomed out. The white balance was set to automatic on all cameras.

B. Data Preparation and Results

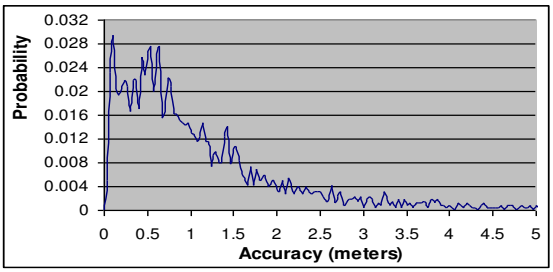
The proposed model has been tested in several sequences with up to 8 cameras, and each sequence has over 5500 frames. In the experiments, the frame images are at 720×576 using 24bits full color (RGB) format. In a single-view process, ball candidates are filtered after foreground detection and image-plane tracking. Then, all the ball candidates detected from 8 sequences are integrated for multi-view tracking of the ball and 3D positioning. The output of the system will lag up to several seconds behind the input observations, as the tracking process accumulates evidence for the temporal filter process (in single-view processing) and awaits the detection of a second fully determined point (in multi-view processing). When two fully determined estimates are available, a virtual vertical plane is generated. As discussed in Section IV(C), internal ball positions within this virtual plane can be even recovered from single-view observations. With the estimated 3D positions of the ball, four motion phases are determined for phase-specific trajectory generation. A list of important thresholds and parameters used in the proposed system is provided in Table 1.

Table 1. List of important constant parameters and thresholds.

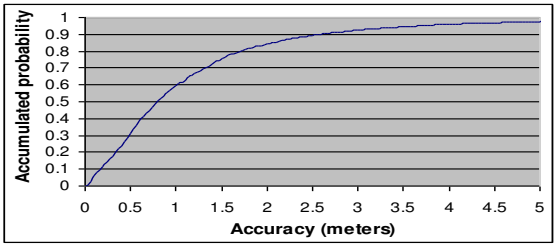
Location	Symbol	Value	Description
Eq. (3)	ρ	0.02	Updating rate in estimating the original background
Eq. (4)	α_L	0.002	Updating rate in dealing with foreground
	α_H	0.02	Updating rate in dealing with background
Eq. (9)	d_0	0.22	Diameter of a football in meters
After Eq. (9)	a_0	0.04	Projected area of a football in square meters
Section III for temporal filtering of ball likelihood	h_1	0.75	Maximum likelihood in the temporal filter
	h_2	0.55	Medium likelihood in the temporal filter
	h_3	0.35	Minimum likelihood in the temporal filter
	N_B	50	Buffer size in frames for temporal ball filtering
Eq. (12)	θ_0	0.5	Threshold of angle in radian for a bouncing ball
Eq. (20)	z_f	1.5	Height threshold of a flying ball in meters

Ground truth ball positions were manually extracted at every 25th frame from the common ground plane, providing 239 GT positions in 5500 frames. To determine 3D ground truth data of the ball, we need to locate image-plane ball positions in multiple sequences by hand. Then, its 3D position is estimated by using multi-view geometry constraints. For the frames between two GT frames, the estimated GT positions are linearly interpolated. At the same time, motion phases within the GT data were also extracted whenever there was a phase transition among the four phases. Based on the manually derived ground truth ball positions and motion phases, two types of evaluation are presented. The first is the distance (in meters) between estimated and ground-truth (GT) ball

positions, in which only 2D distance in x-y plane is used. The second is a comparison of the transitions of motion phases between estimated and GT data. Details of these two evaluations are discussed in the next two Sections below.



(a)



(b)

Fig. 9. Probability distribution of tracking accuracy compared the estimated ball positions with GT in 5500 frames: (a) PDF of the accuracy, (b) is accumulated probability of (a).

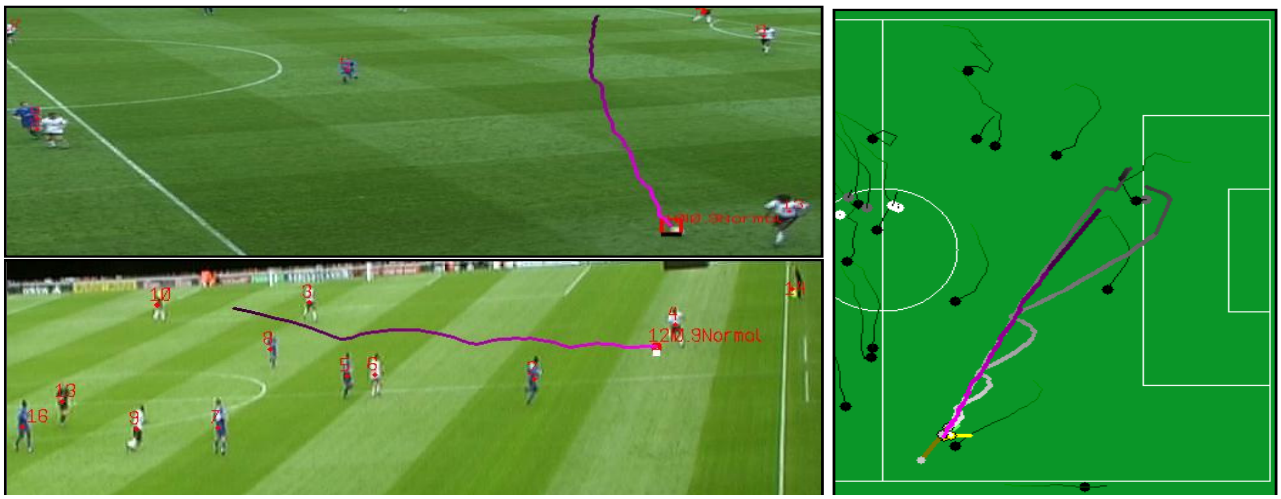
C. Evaluation of Tracking Accuracy versus Latency

In eight testing sequences of 5500 frames each, 3D ball positions are estimated in about 3720 frames. Excluding the 1131 frames in which the ball is out of play, approximately 85% of in-play ball positions are correctly detected using the proposed method. Figure 9 presents the accuracy measured as distance between ground truth and estimate in the ground plane. The estimate is accurate to within 3m of the ground truth position for more than 90% of the recovered ball positions. It is not trivial to further improve the accuracy of the proposed method, given the errors and inconsistencies among calibration parameters at this order of magnitude. The ground-plane errors among calibration projections are estimated to be between 0.1 and 2.5 meters, depending on the distance of the ground point to the cameras. In addition, the model used to estimate 3D position for the flying ball does not take into account all factors (such as air resistance). Hence, with these considerations, the proposed method demonstrates promising performance for real-time automated ball tracking.

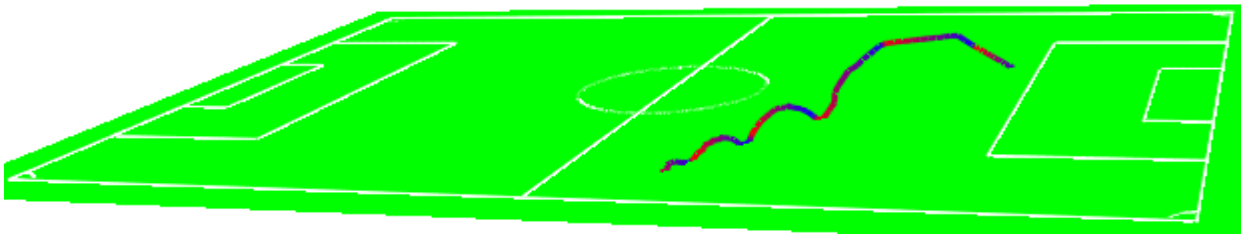
Table 2. Tracking accuracy versus latency (buffering size)

Items / Buffer/Latency	Recover rate	Distribution of recovered ball samples					
		<0.5m	<1.0m	<1.5m	<2.0m	<2.5m	<3.0m
0/0	34.5%	14.7%	25.6%	29.1%	30.6%	32.8%	33.4%
25 frames/1s	72.5%	26.0%	48.4%	57.8%	62.8%	67.4%	69.0%
50 frames/2s	85.2%	26.7%	50.6%	64.4%	71.8%	76.2%	78.6%
75 frames/3s	87.1%	26.9%	50.8%	65.1%	72.3%	77.0%	78.7%

Figure 10(a) demonstrates a plane view of the estimated 3D ball positions when a ball was kicked out by the goalkeeper. Estimated ball positions are shown as a magenta trajectory. The grey trajectories are ground plane projections of ball positions from individual camera views. The brown line in front of the 3D ball trajectory refers to path of ground truth, along which the actual ball trajectory should follow. Player positions are



(a)



(b)

Fig. 10. Estimated 3D ball trajectory compared with GT and two 2D trajectories (a), and the visualization of the 3D trajectory from frame #755 to #826 shown in (b).

marked by black or white circles with tails representing recent trajectory. For comparison, two single-view ball trajectories are given in image planes and projected ground plane, respectively. The 3D visualization of the ball trajectory from frame 755 to 826 is also shown in Figure 10(b).

In the above experiment, a buffer of 50 frames (2 seconds latency) is used for temporal filtering. Table 2 illustrates performance under various size of buffer (latency) in the proposed system. From this table it can be observed that, without temporal filtering, only 34.5% ball positions can be recovered. Allowing a latency of 1, 2 and 3 seconds, the overall recover rates are significantly improved to 72.5%, 85.1% and 87.1%, respectively. At the same time, the corresponding tracking accuracies within 2.5 meters, the maximum measurement errors of the system output are 32.8%, 67.4%, 76.2% and 77.0%. This demonstrates that although longer latency or buffer size is helpful to attain higher recovery rate, it seems that no further significant improvements in tracking accuracy can be achieved with latency of more than 2 seconds. In other words, a buffer of 50 frames or a latency of 2 seconds is an acceptable trade-off between a high recovery rate and a reasonable delay for a broadcast system. Consequently, the overall latency is 3 seconds alongside one second delay (25 frames buffering) for the multi-view tracking of the ball. However, in a broadcast environment, this delay is of the size as other common digital processing operations; furthermore, the method can operate on locally generated streamed media without accumulating an increasing delay.

Most of the estimates additionally provided by the temporal filter are occluded ball observations. If the ball is occluded by several players or in a crowd, whether its trajectory can be recovered depends on if the ball can be observed again within the given latency, even from separate camera views. A ball in-possession is expected to emerge eventually from the player(s) by whom it is being occluded. When the ball is found within temporal filter window, its trajectory is approximated by using the trajectory of the corresponding player(s), hence the lower accuracy of these estimations. However, in the limiting case of severely occluded situations lasting for several seconds or more, the system will fail. The challenge remains to design a system as robust as a human observer.

D. Evaluation of Phase Transition Accuracy

Figure 11 illustrates a complete 3D trajectory history (ground plane projection) from frame 0 to 954 and its corresponding phase transitions. In Fig 11(a), the frame numbers at some of the phase transition points are marked and player trajectories are given in black. Compared with GT, the phase transitions are successfully extracted as shown in Fig 11(b).

The analysis of the frame-by-frame phases can be presented as a confusion matrix, as in Table 3, from which several facts can be observed. Firstly, during the period, in over 50% of samples, the ball is in-possession; and in 33% of samples the ball is rolling, thus 2D models can be applied to 83% of cases. Secondly, about 25% rolling and 13% in-possession balls are misjudged from each other, which happens when a rolling ball cannot be observed in a crowd or an in-possession ball is rolling near the player who possessed the ball. This misjudgment affects the accuracy of the ground-truth as well as the estimate from the proposed method. Disregarding the confusion between these two phases, the average correct rate of phase transitions will increase from 82.6% to 98.3%. Interestingly, about 11% flying balls are incorrectly classified as rolling. One explanation for this error is that a low-flying ball has a similar appearance to a rolling ball. Calculation of the height z_w is sensitive to errors in camera calibration and motion detection; hence the threshold z_f has to be tolerant to these errors. Heights below z_f will not be recognized correctly.

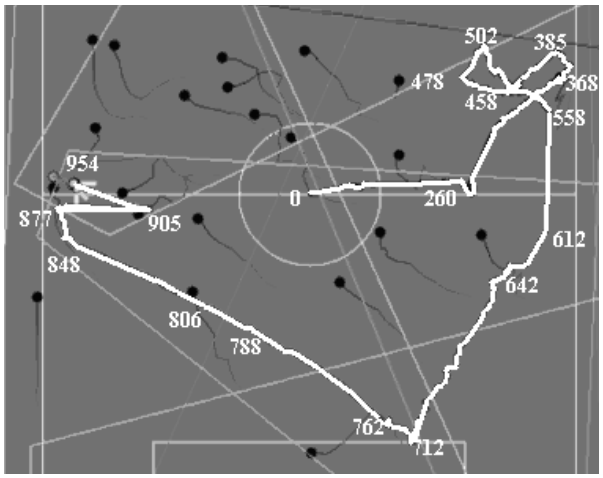


Fig. 11. Overall trajectory of the ball from frame #0 to frame #954 (left), and its corresponding phase transition graph in a complete CPT compared with ground truth with four y-axis positions represent the four phases (right).

Table 3. Quantitative analysis of Figure 11(a) using ground truth and estimated results**.**

Results \ GT		Flying	Rolling	Possessed	Out	Sum
Flying	Frame	56	3	1	0	**60**
	%	**88.9**	0.8	0.2	0.0	**5.4**
Rolling	Frame	7	273	77	0	**357**
	%	11.1	**73.4**	13.6	0.0	**32.4**
Possessed	Frame	0	96	480	0	**576**
	%	0.0	25.8	**84.8**	0.0	**52.3**
Out	Frame	0	0	8	100	**108**
	%	0.0	0.0	1.4	**100.0**	**9.8**
Sum	Frame	63	372	566	100	**901**
	%	**5.7**	**33.8**	**51.4**	**9.1**	**100.0**

It is worth noting that there are some phase transitions missing from the estimated trajectory. The reason here is not phase transition in a short period, but the lack of sufficient observations. For example, the phase transition from possessed to rolling between frame #413 and #450 is successfully detected. During that period, the ball is rolling within a crowd thus cannot be identified in the appearance filtering procedure. Although the ball trajectory can be estimated from the player trajectory of the player(s) using temporal filtering, the corresponding phase is incorrectly classified as in-possession. To solve this problem and thereby allow accurate estimation of the phase, high, well-separated and perhaps more numerous cameras will need to be deployed.

E. System Limitations

As discussed above, there are two main drawbacks in our system in terms of tracking accuracy and phase transition accuracy owing to severe occlusions or insufficient observations. In principle, most of these problems may be resolved by putting additional cameras, even capturing images over the pitch. However, occlusions are still unavoidable in the soccer context which constraints the overall recovery rate and accuracy. Moreover, our system ignores air friction and cannot model some complex movements of the ball, such as the 'swerve', and this may be an interesting topic for further investigation.

VII. Conclusions

A method has been described for real-time 3D trajectory estimation of the ball in a soccer game. In the proposed system, video data is captured from multiple fixed and calibrated cameras. Size, color, and speed are features that discriminate the ball from other moving objects. Temporal filtering of the ball likelihood is also proved essential in robust ball detection and tracking. We model the ball trajectory as curve segments in consecutive virtual vertical planes, which can accurately approximate the real cases even in complex situation. Using geometric reconstruction techniques, we can successfully estimate 3D ball positions from a single view.

One interesting feature of the approach is that it uses high-level phase transition information to aid low-level tracking. Through recognition of the four phases, phase-specific models are successfully applied in estimating 3D position of the ball. Unlike existing models proposed in the literature, the proposed model can fulfill automatic 3D tracking without shadow information and manual assistance. The results obtained from the proposed model are very encouraging. Simple mechanisms for classifying the phase of the ball and estimating its trajectory are demonstrated to be effective. There is an excellent scope for building more sophisticated models into this innovative approach for tracking the ball and content-based understanding of soccer videos.

Acknowledgment

The authors would like to thank Miss Yan Li at Kingston University for the evaluation using parabola estimation. The authors would also like to thank anonymous reviewers for their constructive comments that significantly improved this paper.

References

[1] Y. Gong, T. S. Lim, H. C. Chua, H. J. Zhang, and M. Sakauchi, "Automatic parsing of TV soccer programs," in Proc. IEEE Multimedia Computing and Systems, pp. 167-174, Washington D. C., 1995.

[2] D. Yow, B. L. Yeo, M. Yeung, and B. Liu, "Analysis and presentation of soccer highlights from digital video," in Proc. 2nd Asian Conference on Computer Vision, pp. 499-503, Singapore, 1995.

[3] A. Ekin, M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," IEEE Trans. on Image Processing, vol. 12, no. 7, pp. 796-807, 2003.

[4] Y. Seo, S. Choi, H. Kim, and K. S. Hong, "Where are the ball and players?: soccer game analysis with color based tracking and image mosaick," in Proc. Int. Conf. on Image Analysis and Processing, pp.196-203, Florence, Italy, 1997.

[5] X. F. Tong, H.Q. Lu, and Q.S. Liu, "An effective and fast soccer ball detection and tracking method," in Proc. IEEE Int. Conf. on Pattern Recognition, vol. IV, pp.795-798, Cambridge, England, 2004.

[6] A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games," in Proc. IEEE Int. Conf. on Pattern Recognition, vol. I, pp.303-306, Québec City, Canada, 2002.

[7] X. Yu, C. Xu, Q. Tian, and H.W. Leong, "A ball tracking framework for broadcast soccer video," in Proc. IEEE Int. Conf. on Multimedia and Expo, vol. II, pp. 273-276, Baltimore, Washington D.C., 2003.

[8] X. Yu, Q. Tian, and K.W. Wan, "A novel ball detection framework for real soccer video," in Proc. IEEE Int. Conf. on Multimedia and Expo, vol. II, pp. 265-268, Baltimore, Washington D.C., 2003.

[9] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distante, "A new algorithm for ball recognition using circle Hough transform and neural classifier," Pattern Recognition, vol. 37, no. 3, pp. 393-408, 2004.

[10] Y. Ohno, J. Miura, and Y. Shirai, "Tracking players and estimation of the 3D position of a ball in soccer games," in Proc. IEEE Int. Conf. on Pattern Recognition, vol. I, pp. 145-148, Barcelona, Spain, 2000.

[11] K. Matsumoto, S. Sudo, H. Saito, and S. Ozawa, "Optimized camera viewpoint determination system for soccer game broadcasting," in Proc. IAPR Workshop on Machine Vision Applications, pp. 115-118, Tokyo, 2000.

[12] T. Bebie and H. Bieri, "SoccerMan – reconstructing soccer game from video sequence," in Proc. IEEE Int. Conf. on Image Processing, vol. I, pp. 898-902, Chicago, 1998.

[13] T. Kim, Y. Seo, and K. S. Hong, "Physics-based 3D position analysis of a soccer ball from monocular image sequences," in Proc. IEEE Int. Conf. on Computer Vision, pp. 721-726, Bombay, India,, 1998.

[14] I. Reid and A. North, "3D trajectories from a single viewpoint using shadows," in Proc. British Machine Vision Conference, pp. 863-872, Southampton, England,1998.

[15] J. Ren, J. Orwell, G. A. Jones, and M. Xu, "A novel framework for 3D soccer ball estimation and tracking," in Proc. IEEE Int. Conf. on Image Processing, vol. 3, pp. 1935-1938, Singapore, 2004.

[16] M. Xu, J. Orwell, L. Lowey, and D.J. Thirde, "Architecture and algorithms for tracking football players with multiple cameras," IEE Proceedings-Vision, Image and Signal Processing, vol. 152, no. 2, pp. 232-241, 2005.
[17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 246-252, Ft. Collins, CO, USA, 1999.
[18] M. Xu and T. Ellis, "Partial observation vs. blind tracking through occlusion," in Proc. British Machine Vision Conference, pp. 777-786, Cardiff, 2002.
[19] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 364-374, Miami Beach (FL), USA, 1986.
[20] J. Canny, "A computational approach to edge detection," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, 1986.
[21] V. Tovinkere and R. J. Qian, "Detecting semantic events in soccer games: toward a complete solution," in Proc. IEEE Int. Conf. on Multimedia and Expo, pp. 1040-1043, Tokyo, 2001.
[22] N. Babaguchi, Y. Kawai, and T. Kitashi, "Event based indexing of broadcasted sports video by intermodal collaboration," IEEE Trans. Multimedia, vol. 4, no. 1, pp. 68-75, 2002.
[23] B. Li and M. I. Sezan, "Event detection and summarization in American football broadcast video," in Proc. Int. Conf. on Electronic Imaging: Storage and Retrieval for Media Databases, pp. 202-213, San Jose (CA), USA, 2002.
[24] R. Leonardi, P. Migliorati, and M. Prandini, "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains," IEEE Trans. on Circuits and Systems for Video Technology, vol. 14, no. 5, pp. 634-643, 2004.
[25] P. R. Bevington, Data Reduction and Error Analysis for the Physical Sciences, McGraw-Hill, New York, 1969.
[26] E. Cox, The Fuzzy System Handbook, AP Professional, Cambridge, England, 1994.
[27] Y. Liu, D. Liang, Q. Huang, and W. Gao, "Extracting 3D information from broadcast video," Image and Vision Computing, vol. 24, no. 10, pp. 1146-1162, 2006.
[28] P. J. Figueroa, N. J. Leite, and R. M. L. Barros, "tracking soccer players aiming their kinematical motion analysis," Computer Vision and Image Understanding, vol. 101, no. 2, pp. 122-135, 2006.
[29] T. Shimawaki, T. Sakiyama, J. Miura, and Y. Shirai, "Estimation of ball route under overlapping with players and lines in soccer video image sequence," in: Proc. ICPR, pp. 359-362, 2006.