Multi-camera Video Surveillance for Real-time Analysis and Reconstruction of Soccer Games

Jinchang Ren[1,2], Ming Xu[3], James Orwell[4], Graeme A Jones[4]

[1] *School of Informatics, University of Bradford, U.K.*

[2] *School of Computers, Northwestern Polytechnic University, China*

[3] *Xi'an Jiaotong Liverpool University, Suzhou, China.*

[4] *Digital Imaging Research Centre, Kingston University, U.K.*

j.ren@bradford.ac.uk ming.xu@xjtlu.edu.cn {j.orwell,g.jones}@kingston.ac.uk

Abstract Soccer analysis and reconstruction is one of the most interesting challenges for wide-area video surveillance applications. Techniques and system implementation for tracking the ball and players with multiple stationary cameras are discussed. With video data captured from a football stadium, the real-world, real-time positions of the ball and players can be generated. The whole system contains a two-stage workflow, i.e. single view and multi-view processing. The first stage includes categorizing of players and filtering of the ball after change detection against an adaptive background and image-plane tracking. Occlusion reasoning and tracking-back is applied for robust ball filtering. In the multi-view stage, multiple observations from overlapped single views are fused to refine players' positions and to estimate 3-D ball positions using geometric constraints. Experimental results on real data from long sequences are demonstrated.

Keywords: video surveillance, multiple cameras, tracking, 3-D vision, video signal processing

1. Introduction

Sports analysis, in particular soccer game reconstruction, is one of the most interesting and challenging applications in wide-area video surveillance with multiple cameras. By accurately tracking players and the ball, a number of important applications have been derived for automatic comprehension of soccer events, such as annotation of video content, summarization, team strategy analysis, verification of referee decisions and 2-D or 3-D reconstruction and visualization [1-4].

Despite frequently occlusion in clutters, players may be successfully detected and tracked on the basis of color and shape from monocular cameras, especially from TV broadcasting streams [1-6, 9-11]. However, similar methods cannot be extended to ball detection and tracking for several reasons [17]. Firstly, the ball is too small, exhibits irregular shape, variable size and unstable color when moving fast. Secondly, the ball is frequently occluded by players or flying out of filed-of-views of all the cameras. Finally, the ball is mostly traveling above the ground, which necessitates 3D tracking for accurate positioning. Consequently, using multiple stationary cameras is a good alternative approach in such a context, as it increases the overall field-of-view and enables 3-D positioning of the ball for more challenging applications. Besides, it provides video data of better resolution and can successfully reduce the effects of dynamic occlusion and improve the accuracy and robustness of estimation. Related work from either monocular or multiple sequences is reviewed below.

1.1 Tracking from monocular sequence

In monocular sequence of TV streams, the ball is mostly of good resolution in the image centre. However, due to complex camera movements and partial views of the field, it is hard to obtain accurate camera parameters for on-field ball positioning. In Gong et al [1], white color and circular shape are employed to detect the ball in image sequences. In Yow et al [2], the ball is detected by template matching in each of the reference frames and then tracked between these frames. In Seo et al [6], template matching and Kalman filter are used to track the ball after manual initialization. In Yu et al [8], candidate balls are first identified by size range, color and shape, and further verified using motion information obtained from a Kalman filter. Since color and shape varies considerably in soccer games, these methods seem unlikely to provide robust solutions.

As for tracking of players, several research projects have been reported in this field. In [6] and [9], a monocular TV sequence is used for generating panoramic views and players' trajectories. Using the concept of a closed-world, Intille and Bobick [10] track players in the broadcast TV sequences of American football games. Needham and Boyle [11], also adopt a monocular static camera to track players of an indoor 5-a-side soccer game with the CONDENSATION algorithm.

1.2 Tracking from multiple sequences

Generally, there are two steps to estimate and track 3D balls from multiple cameras. Firstly, the ball is detected and tracked in single views independently. Then, 2D ball positions from multiple camera views are integrated to attain 3D positions using known motion models [4-6]. Bebie, and Bieri [4], model 3D trajectory segments by Hermite spline curves. However, about one-fifth of the ball positions need to be set manually before estimation. In Matsumoto et al [5], 2D ball is detected by template matching and tracked by epipolar line constraints between multiple cameras. In Ohno et al [9], 3D ball trajectory is modeled by considering air friction and gravity but depends on an unsolved initial 3D velocity. In Kim et al [18] and Reid and North [19], reference players and shadows are utilized in the estimation of 3D ball positions. These are unlikely to be robust as the shadow positions depend more on light positions than camera projections.

On tracking players, multi-view data is used in several different ways. In Cai and Aggarwal [12] and Khan et al [13], each target is tracked in the best-view camera and it is passed to the neighboring camera when leaving the field-of-view of the current camera. Assuming all the targets being in the same plane (e.g. the ground plane), Stein [14] and Black et al. [15] compute the homography transformation between the coordinates of two overlapping images captured with uncalibrated and calibrated cameras, respectively.

1.3 Main contributions

In this paper, we present the algorithms and system implementation for estimating positions of soccer players and the ball from multiple stationary cameras. The novel components include a method for using partial observations to split grouped targets, tracking-back for robust filtering the ball, and fusion of multi-sensor data to improve target visibility and tracking accuracy as well as a geometric approach for 3-D ball positioning. Using observations from either multiple or a single camera(s), 3-D positions of the ball can be estimated in our trajectory model. With video data inputted from static cameras of overlapping fields-of-view at a football stadium, the system can generate estimates of the real-world, real-time (allowing several seconds of latency) positions of the ball and players during a match and this will benefit many potential applications.

2. Architecture and foreground detection

2.1 System architecture

Our system contains a two-tier architecture in which single-view and multi-view processing are implemented, respectively. For single-view processing, features are extracted in the sequences from each individual camera to describe 2-D ground plane positions of players and their category as well as likelihood and ground plane positions of ball candidates. These features are then integrated in the second tier for both 3-D ball positioning and accurate locating of players to update a single model of the state of the game. Finally, this game-state is passed through a phase of marking-up for generating the XML output used by third party applications to deliver results to their respective audiences [16].

In our system, totally eight cameras are positioned at suitable locations around the stadium, and their positions are governed by the layout of the chosen stadium and the requirement to achieve an optimal view of the football pitch: good resolution of each area, in particular the goal-mouths. Through a network of optical fibre, the output of each camera is connected to a feature server (single-view processing module). All the eight feature servers are put on a rack, and an IP/Ethernet network is used to connect their output to a multi-view tracker to provide output of game states (see Figure 1).

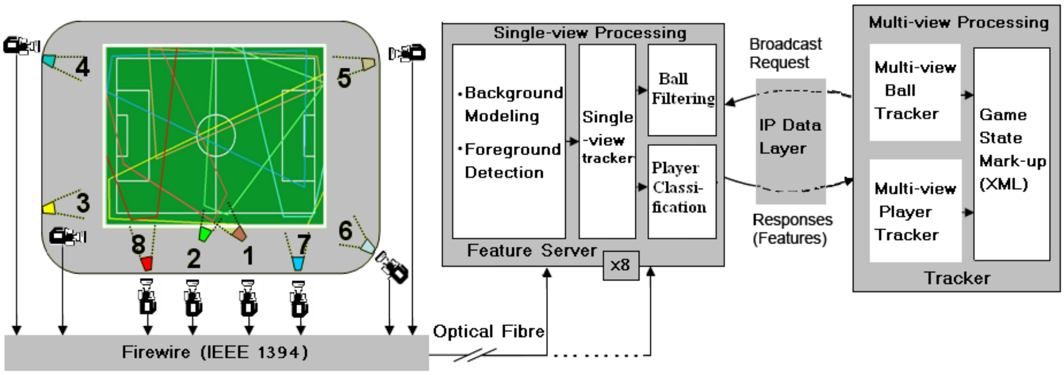


Figure 1. Architecture of our system with plotted filed-of-views from the eight cameras.

To synchronize the whole stage, both single-view and multi-view processing is associated with a time-stamp for the feature data. The multi-view Tracker is responsible for synchronization of single-view Feature Servers (first stage) in a request-response mechanism. Each iteration (or frame) of the process comprises a single (broadcast) request issued by the multi-view Tracker at a given time, and then the Feature Servers respond by taking the latest frame in the video stream, processing it and transmitting the resultant Features back to the Tracker. Synchronization of the feature sets is implied as the multi-view Tracker will record the time-stamp at which the request was made.

2.2 Foreground detection

Image subtraction of an adaptive background from incoming frames is utilized for moving object detection in the field of view (FOV) of each individual camera. During a short bootstrap stage, a per-pixel Gaussian mixture model [20] is used to estimate an initial background, in which each background candidate corresponds to a Gaussian distribution and will be favoured if supported by incoming data. The probability of observing a pixel value **I** at time k is modelled by a mixture of Gaussians:

$$P(\mathbf{I}_k)=\sum_i \omega_k^{(i)} G\left(\mathbf{I}_k,\boldsymbol{\mu}_k^{(i)},\sigma_k^{(i)}\right) \qquad (1)$$

where $\boldsymbol{\mu}_k^{(i)}$ is the mean of the i-th distribution, $\sigma_k^{(i)}$ is the square root of the covariance matrix trace, and $\omega_k^{(i)}$ is the weight reflecting the prior probability that the i-th distribution accounts for the data.

Instead of using the Expectation Maximization (EM) algorithm to maximize the likelihood of the observed data, an on-line K-means approximation is implemented to accelarate the estimation and cope with the non-stationary pixel processes. For a new pixel observation, a matched distribution is updated with increasing weight as follows:

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

where ρ is the updating rate satisfying $\rho \in (0, 1)$. For unmatched distributions, the parameters remain the same but the weights decrease. The initial background image is selected as the distribution with the greatest weight at each pixel.

Then, this initial background image is continuously updated using a faster running average algorithm for efficiency:

$$\boldsymbol{\mu}_k = [\alpha_L \mathbf{I}_k + (1-\alpha_L)\boldsymbol{\mu}_{k-1}]F_k + [\alpha_H \mathbf{I}_k + (1-\alpha_H)\boldsymbol{\mu}_{k-1}]\overline{F_k} \tag{3}$$

where $0 < \alpha_L << \alpha_H << 1$, and F_k is the foreground binary mask. This method avoids locking any detection error by slowly updating the background image even in foreground regions.

Given the input image $\mathbf{I}_k$, we can decide the foreground binary mask F_k by comparing $\| \mathbf{I}_k - \boldsymbol{\mu}_{k-1} \|$ against a threshold. From the foreground masks, we can obtain a series of foreground regions representing candidate objects after a connected component analysis and thresholding by size. Each foreground region is represented by its centroid, bounding box and area.

2.3 Tracking in image plane

An image-plane Kalman tracker is used to filter noisy measurements and split merged objects, in which the state $\mathbf{x}_I$ and measurement $\mathbf{z}_I$ are given by:

$$\begin{cases} \mathbf{x}_I = [r_0 \;\; c_0 \;\; \dot{r}_0 \;\; \dot{c}_0 \;\; \Delta r_1 \;\; \Delta c_1 \;\; \Delta r_2 \;\; \Delta c_2]^{\mathrm{T}} \\ \mathbf{z}_I = [r_0 \;\; c_0 \;\; r_1 \;\; c_1 \;\; r_2 \;\; c_2]^{\mathrm{T}} \end{cases} \tag{4}$$

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 ($r_1 < r_2$ and $c_1 < c_2$); $(\Delta r_1, \Delta c_1)$ and $(\Delta r_2, \Delta c_2)$ are the relative positions of (r_1, c_1) and (r_2, c_2) to (r_0, c_0).

The state transition and measurement equations in the Kalman filter are:

$$\begin{cases} \mathbf{x}_{I,k+1} = \mathbf{A}_I \mathbf{x}_{I,k} + \mathbf{w}_{I,k} \\ \mathbf{z}_{I,k} = \mathbf{H}_I \mathbf{x}_{I,k} + \mathbf{v}_{I,k} \end{cases} \tag{5}$$

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. Given ΔT as the time interval between two successive frames (for image formation), $\mathbf{I}_2$ and $\mathbf{O}_2$ represent 2×2 identity and zero metrics; we have $\mathbf{A}_I$ and $\mathbf{H}_I$ defined as

$$\mathbf{A}_I = \begin{bmatrix} \mathbf{I}_2 & \Delta T \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \quad \mathbf{H}_I = \begin{bmatrix} \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{O}_2 \\ \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{I}_2 & \mathbf{O}_2 \\ \mathbf{I}_2 & \mathbf{O}_2 & \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \tag{6}$$

Using the Tsai's algorithm for camera calibration [21], the measurements are transformed into world co-ordinates. Until Section 4, all objects are assumed to lie on the ground plane, which 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 group of measurement vectors are defined, including the object's width, height and area measured in meters (and square meters), and calculated by assuming it is touching the ground plane. 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.

3. Filtering players and the ball

3.1 Filtering and categorizing of players

Since players are larger than the ball, normally they appear with bigger bounding boxes. However, within each bounding box there are perhaps several players due to occlusion. In this Section, separating each group of players, in particular groups of two, are discussed by assuming that each target has a slowly varying height and width [22].

Once some bounding edge of a target is decided to be observable, its opposite, unobservable bounding edge can be roughly estimated. The decisions about which targets are grouped and which bounding edges are observable, are based on the relative positions of the foreground region and the predictions of individual targets. If the predicted centroids of multiple targets are within the bounding box of the same foreground region, then these targets are considered to be in group. Each bounding edge of a target, which is outer-most in its group, is treated as 'observed', and associated with the corresponding edges of the foreground region bounding box. The centroids of each player, and the remaining bounding edges of the individual players are not observed at this time step. For these variables, the corresponding elements in the covariance matrices are increased so that they contribute little to the estimation process. Therefore, the estimate depends more on the observable variables. Because the estimate is updated using partial measurements whenever available, it is more robust and accurate than using prediction only. An example of single-view tracking with two grouped players is shown in Figure 2.

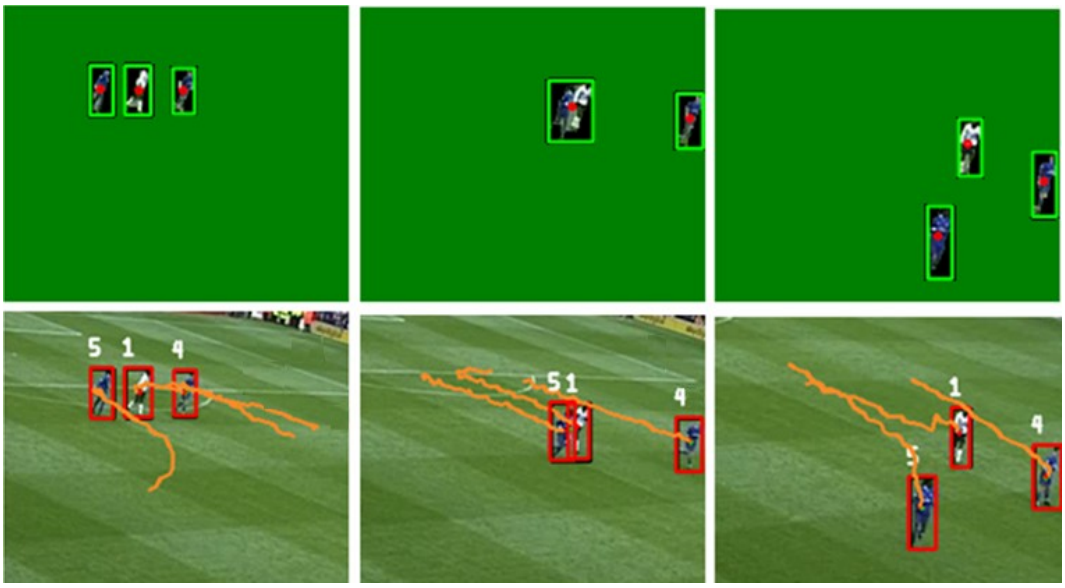


Figure 2. Player detection (top) and tracking (bottom) from a single camera.

After tracking of the players, an estimate of the category (color of uniform) is added to each measurement using a histogram-intersection method [23]. The result for each object is a seven-element vector $\mathbf{c}_j^i(k)$, indicating the likelihood that the object is a player in one of the five categories of uniform (two teams, two goalkeepers, and referees), the ball or a false alarm, where i, j and k are the indexes of the camera, the object and the frame, respectively.

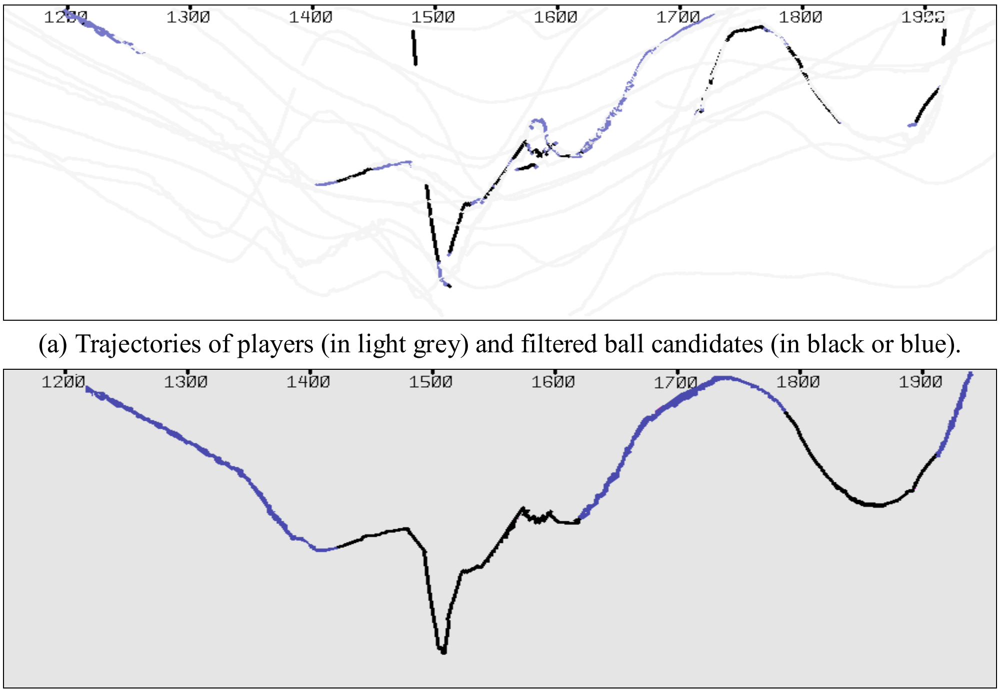
3.2 Filtering of the ball

In our ball filtering process, *velocity* and *longevity* features, along with *size* and *color*, are employed to discriminate the ball from other objects. Each segmented object o_i is assigned a likelihood of being the ball by an operator $D(o_i) \in [0,1]$ using the object's absolute velocity $|\mathbf{v}_i|$ and longevity n_i (in tracked frames) as below:

$$D(o_i) = D_1(o_i) + \frac{1}{2}\frac{|\mathbf{v}_i|}{v_{\max}}(1 - e^{-n_i T_0}) \qquad (7)$$

where $v_{\max}$ is the maximum absolute velocity of all the objects (including the ball and non-ball objects), and T_0 is a constant. The bias $D_1(o_i)$ is defined as 0.5 if the size (width, height and area) and color of the candidate target are all within certain predefined ranges for a ball. Otherwise, the bias is set as 0. In fact, this bias constrains to extract a nearly circular ball candidate of its size and color within valid ranges and appearance. Figure 3(a) represents trajectories of both players (in light grey) and filtered ball candidates in 1100 frames, in which blue trajectories are those of ball likelihoods between 0.5 and 0.65, whilst black ones are those with likelihoods higher than 0.65. Moreover, in some frames we may find several black and/or blue trajectories. This happens due to occlusion or false alarms and will be resolved below.

To resolve the uncertainties within the filtered ball trajectories, occlusion reasoning is applied on the basis of tracking states obtained from Kalman filters. Assume a candidate ball B_i is merged with a non-ball object P_j, thus B_i can be moving or possessed. It may be defined as moving if a new ball candidate can be detected near P_j within n_0 frames since it was merged. Otherwise, it should be defined as possessed by P_j. Therefore, a buffer is introduced to store the tracking states of P_j before finally determining the real ball trajectory. If we find it is still merged at frame $\#(n+\Delta n)$, where $\Delta n > n_0$, then tracking-back is employed to reclassify the state between frame $\#n$ and frame $\#(n+\Delta n)$ as possessed.



(a) Trajectories of players (in light grey) and filtered ball candidates (in black or blue).

(b) Filtered ball trajectory after occlusion-reasoning and tracking-back.

Figure 3. Examples of thirty-two seconds of tracking data, in which time t moves from left to right, and the horizontal image co-ordinates of the object centroids, $c_{0,}$ are plotted up the y-axis (the vertical image co-ordinate is omitted from this diagram).

Furthermore, tracking-back also contains temporal *hysteresis*-based thresholding [24] of the ball likelihood along the trajectory. Here, we have three thresholds, h_1, h_2, h_3, 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 'non-ball' (*i.e.* false alarms). Candidate objects with likelihood l satisfying $h_1 > l > h_2$ are relabeled as a 'ball' if the object as been labeled as a ball in a neighboring frame (along the tracking history). Similarly, objects with likelihood l satisfying $h_2 > l > h_3$ are labeled as 'not ball' if the object has that label in a neighboring frame (along the tracking history). The result of this temporal hysteresis tracking-back procedure is a considerable improvement in the robustness of detection and continuity of trajectories as shown in Figure 3(b). At the same time, false alarms are dramatically reduced.

4. Multi-view processing

4.1 Multi-view tracking of players

For tracking of players in the world coordinate, we assume that they touch the ground plane. Each player has multiple measurements, each projected from an individual image-plane to the ground-plane, which include the ground plane positions $\mathbf{z}_j = [x_w, y_w]^T$, position covariance matrices $\mathbf{R}_j$ and category measurements $\mathbf{c}_j$ ($j \in [1,7]$). An association matrix $\beta^{(i)}$ is used to represent the association of the set of target players with the measurements from the i-th camera. Each element $\beta_{jt}^{(i)}$ is 1 for association between the t-th target and j-th measurement or 0 otherwise. The association matrix is decided according to the nearest Mahalanobis distance between the measurement and the target prediction. The individual camera measurements assigned to the t-th target in (8) are weighted by measurement uncertainties and integrated into an overall measurement, where *tr*() represents the trace of a matrix.

$$\begin{cases} \mathbf{R} = \left[\sum_i \sum_j \beta_{jt}^{(i)} \left(\mathbf{R}_j^{(i)} \right)^{-1} \right] \\ \mathbf{z} = \mathbf{R} \left[\sum_i \sum_j \beta_{jt}^{(i)} \left(\mathbf{R}_j^{(i)} \right)^{-1} \mathbf{z}_j^{(i)} \right] \\ \mathbf{c} = \sum_i w^{(i)} \sum_j \beta_{jt}^{(i)} \mathbf{c}_j^{(i)} \\ w^{(i)} = \dfrac{\sum_j \beta_{jt}^{(i)} / tr\left(\mathbf{R}_j^{(i)} \right)}{\sum_i \sum_j \beta_{jt}^{(i)} / tr\left(\mathbf{R}_j^{(i)} \right)} \end{cases} \tag{8}$$

The fused measurements are then incorporated into a Kalman filter to estimate the t-th player's state $\mathbf{x} = [x_w, y_w, \dot{x}_w, \dot{y}_w]^T$, state covariance $\mathbf{P}$ and category estimate $\mathbf{e}$ iteratively. The ground-plane process evolution and measurement equations are:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_w \mathbf{x}_k + \mathbf{w}_{w,k} \\ \mathbf{z}_k = \mathbf{H}_w \mathbf{x}_k + \mathbf{v}_{w,k} \end{cases} \tag{9}$$

where $\mathbf{w}_w$ and $\mathbf{v}_w$ are the ground-plane process noise and measurement noise, respectively. The state transition and measurement matrices are:

$$\mathbf{A}_w = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{H}_w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

Each target with measurement from at least one camera is then updated using the integrated measurement:

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_w^{\mathrm{T}} \left[\mathbf{H}_w \mathbf{P}_k^- \mathbf{H}_w^{\mathrm{T}} + \mathbf{R}_k \right]^{-1} \\ \hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\mathbf{z}_k - \mathbf{H}_w \hat{\mathbf{x}}_k^- \right] \\ \mathbf{P}_k^+ = \left[\mathbf{I} - \mathbf{K}_k \mathbf{H}_w \right] \mathbf{P}_k^- \\ \hat{\mathbf{e}}_k = (1-\eta)\, \hat{\mathbf{e}}_{k-1} + \eta\, \mathbf{c}_k \end{cases} \tag{11}$$

where $0 < \eta < 1$. After fusion of multiple views, the ground plane accuracy has been improved, as shown in Figure 6, when compared with the measurement covariance.

If no measurement is available for an existing target, then the state estimate is updated using its prior estimate. In this case, the state covariance increases linearly with time. Once a target has no measurement over a certain number of frames, the state covariance reflecting uncertainty will be larger than a tolerance. The tracking of this target will be automatically terminated and this target will be re-tracked as a new one when the measurement flow is resumed.

Moreover, if there are more than 25 target players in the model (ten outfield players and one goalkeeper per team plus one referee and two sidemen), then the most likely 25 tracks are selected as the estimated positions of the players. This likelihood is decided by its longevity, category estimate and duration of occlusion with other players. A sub-optimal search method gives reasonable results (see Figure 7).

4.2 Estimating 3-D ball positions

During a soccer game, the ball is moving regularly from one place to another. We assume that a ball's trajectory will form a curve in a virtual plane. In a special case, the curve will become a line when the ball is rolling on the ground. In each vertical plane π, the ball will generate a single trajectory curve. Then, the complete ball trajectory is modeled as a sequence of adjacent planar curve segments.

Assume the ball is observed from N cameras ($N \geq 2$), then we have N 3D lines from each the projected ground-plane ball position to its corresponding 3D camera position. If $N = 2$, the 3D ball position $\mathbf{B} = (x, y, z)^{\mathrm{T}}$ is defined as the middle point of the shortest possible line, a common perpendicular, between these two lines. Otherwise, the least square analysis is utilized to estimate $\mathbf{B}$ as a point which minimizes the overall distance to all the N 3D lines [15].

If we have two estimated 3-D ball positions, r and s , a vertical plane π can be uniquely decided as follows. Firstly, locate points r' and s' on the ground plane β with $rr' \perp \beta$ and $ss' \perp \beta$, then we have a line $r's'$ on β . Finally, π is determined as a plane through $r's'$ and perpendicular to β .

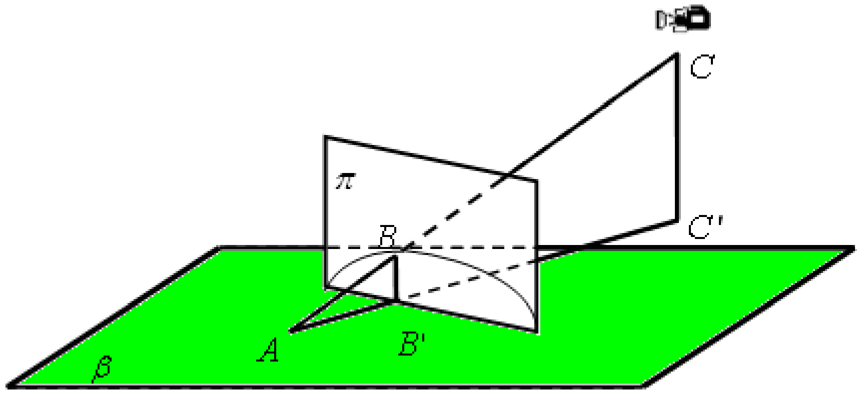


Figure 4. Estimate 3-D ball position B with known vertical plane π , projected single view ball position A and camera position C , β is the ground plane.

With the assistance of this virtual vertical plane π, 3D ball position B on π can even be recovered from only single camera observation. Let A be a projected single-view ball position from known camera position C , and the points B' and C' are the projected positions of B and C on ground plane β (see Figure 4). Firstly, we can find B' defined as the intersection of $\overline{AC'}$ and π . Then, calculate B using the fact that triangles $\Delta ACC'$ and $\Delta ABB'$ are similar, and the prior is known completely. Let us define $X_p = (x_p, y_p, z_p)$ as the world coordinates of any point p , then we have:

$$z_B = \frac{\|X_A - X_{B'}\|}{\|X_A - X_{C'}\|} \cdot z_C \tag{12}$$

Furthermore, a parabola curve of the ball trajectory can also be determined with only two-known 3-D ball positions, p and q, by using the gravity acceleration g. Let (x_p, y_p, z_p) and (x_q, y_q, z_q) denote the 3-D co-ordinates of p and q, and t_p and t_q are the corresponding time moments, respectively. Then, the corresponding trajectory is given by

$$x(t) = x_p + \frac{x_q - x_p}{t_q - t_p}(t - t_p) \tag{13}$$

$$y(t) = y_p + \frac{y_q - y_p}{t_q - t_p}(t - t_p) \tag{14}$$

$$z(t) = \begin{cases} 0 & for \quad 2D - moving \\ \left(\frac{z_p t_q - z_q t_p}{t_q - t_p} \right) + \left(\frac{z_q - z_p}{t_q - t_p} \right) t - \frac{g}{2}(t - t_p)(t - t_q) & otherwise \end{cases} \tag{15}$$

5. Results

In our system, the soccer videos are captured by eight stationary cameras positioned around a football stadium, and all the cameras are manually calibrated before tracking. To evaluate the tracking accuracy of the ball, a group of ground truth (GT) positions of the ball are manually derived from about every 25th frame in the image sequences. For the frames between two GT frames, an estimated GT position is obtained by using linear interpolation. The distance between the estimated positions and ground truth positions are obtained.

A histogram is used to show the distance between the automatic estimate and the manually recorded ground truth of the ball (see Figure 5), which represents the probability of estimate being within given distance of the ground truth. In eight test sequences of more than 5000 frames each, we have 3D ball positions estimated in about 3700 frames. Regarding the ball is out of play in about 1100 frames, over 85% in-play balls can be recovered in our system, with over 90% of them lie within 3m of the ground truth. The maximum discrepancy between the ground planes of each calibrated camera is of the same order of magnitude, which puts this result in context.

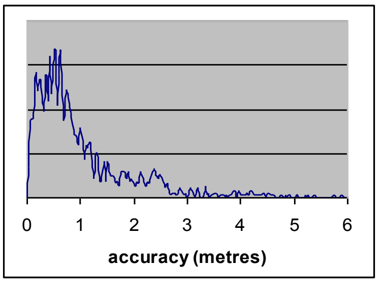


Figure 5: Tracking accuracy of the ball compared with manual ground truth.

As for tracking players, Figure 6 uses an example to show how tracking accuracy has been improved by fusion of multi-view data. According to the camera arrangements in Figure (1), each player within the overlapping fields-of-view of multiple cameras is often at different distances to those cameras. Usually a larger distance means a higher uncertainty in corresponding ground-plane measurement, and this uncertainty is properly represented by the measurement covariance indicated by the size of the ellipses. By weighting the inverse of measurement covariance, the fused measurement and thereafter the estimate for each player are automatically biased to the most accurate measurement from the closest camera. As a result, improved accuracy is obtained by fusion with smaller covariance than that from any individual camera as shown in Figure 6(c).

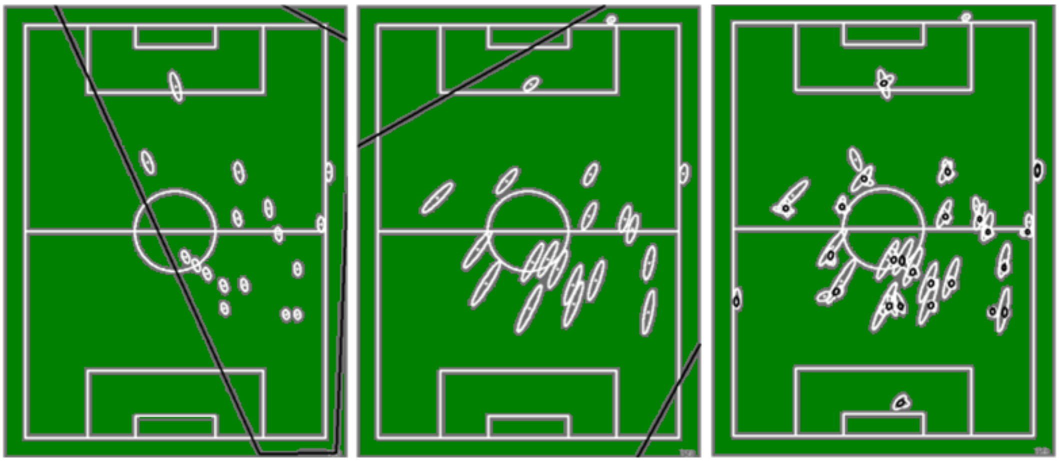


Figure 6. Improved ground-plane accuracy compared by the covariance in ellipses: The left two images are from single cameras with dark line denoting extent of field of view; the right one represents the fused measurements from all eight cameras (black).

Furthermore, the performance on tracking players is also measured using the number of tracked target before a final selection step. Two sequences of 5000 frames are used for this evaluation and the mean μ and standard deviation σ of target number are found as (26.63, 1.55) and (26.12, 1.56), respectively. It is not surprising that the mean of target number is slightly larger than the expected 25, as the algorithm encourages those targets with missing measurement to be maintained for some time. It is also noted that the relative tracking error rates ($1.65\sigma/\mu$) are low enough (within ±10%) in 90% of the tested frames. The remaining 10% of the tested frames with larger tracking errors corresponds to some over-crowded situations, where players are tightly packed in pairs, e.g. during a corner kick, or where camera calibration errors and measurement errors are comparable to the small distances between players.

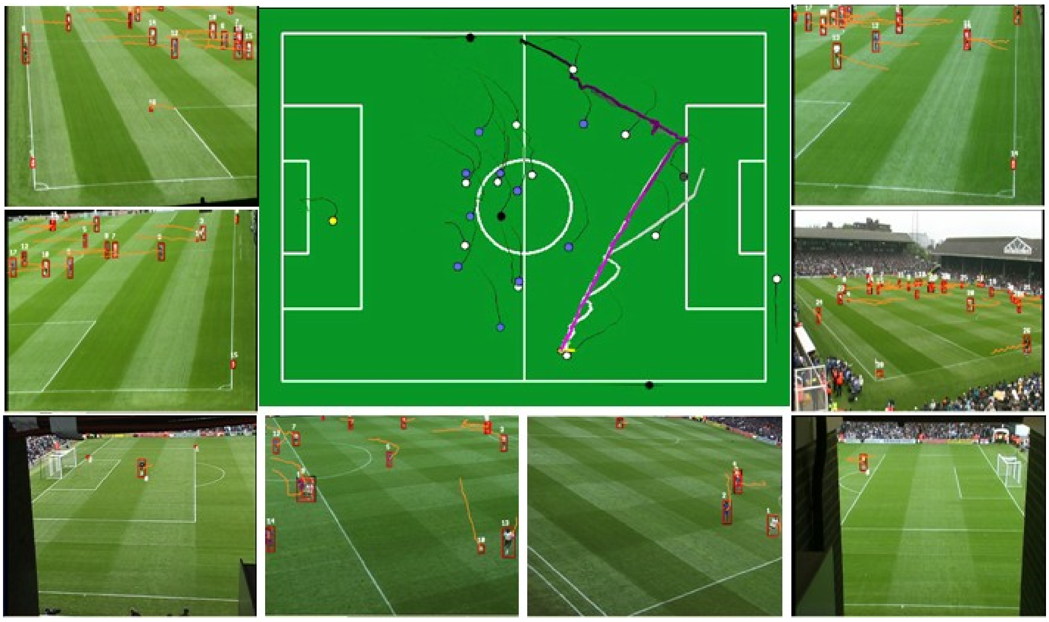


Figure 7: Multiview and single-view tracking at one frame. The surrounding images (from top-left to top-right) correspond to cameras C4, C3, C8, C2, C1, C7, C6 and C5.

A comprehensive example of the tracked ball and selected players is illustrated in Figure 7 along with the single-view detection and tracking results at the same frame. In the virtual pitch, the game is visualized using the projected positions on the ground plane. The projected 3-D ball trajectory is represented in magenta with associated 2-D trajectories shown in grey. At the same time, the ball filtered from the corresponding cameras can be also found in the single view images from both camera C3 and C2. Besides, there are 26 players in the pitch among which we can easily recognize the two goal-keepers, one referee and two sidemen. Except one player running along the outer border on right-side of the pitch (which can be clearly located in the single view image from camera C6), ten players of each team can be also clearly identified.

6. Conclusions

Techniques and system implementation of an application on tracking of soccer players and the ball with multiple cameras have been presented. The ball is effectively filtered by using velocity and longevity as well as occlusion-reasoning and tracking back. With partial observations, players are accurately tracked in single view processing. Through fusion of multi-view data, 3-D ball positions are estimated using simple geometric constraints, and each player is biased to the most accurate measurement of more accuracy and robustness than those from any individual camera.

Acknowledgements This work forms part of the INMOVE project, supported by the European Commission IST 2001-37422.

References

1. Gong, Y., Lim, T.-S., Chua, H.C., Zhang, H. J., Sakauchi, M.: Automatic parsing of TV soccer programs. In: Proc. Multimedia Computing and Systems, pp. 167-174. (1995)

2. Yow, D., Yeo, B. L., Yeung, M., Liu, B.: Analysis and presentation of soccer highlights from digital video. In: Proc. ACCV, pp. 499-503. (1995)

3. Ekin A., Tekalp M., Mehrotra R.: Automatic soccer video analysis and summarization. IEEE Trans. on Image Processing. **12**(7), 796-807 (2003)

4. Bebie, T., Bieri, H.: SoccerMan – reconstructing soccer game from video sequence. In: Proc. ICIP, pp. 898-902. (2000)

5. Matsumoto, K., Sudo, S., Saito, H., Ozawa, S.: Optimized camera viewpoint determination system for soccer game broadcasting. In: Proc. IAPR Workshop on Machine Vision Applications, pp. 115-118. Tokyo (2000)

6. Seo, Y, Choi, S., Kim, H., Hong K. S.: Where are the ball and players?: soccer game analysis with color based tracking and image mosaick. In: Proc. ICIAP, pp. 196-203. (1997)

7. D'Orazio, T., Guaragnella C., Leo M., Distante A.: A new algorithm for ball recognition using circle Hough transform and neural classifier. Pattern Recognition. 37(3), 393-408 (2004)

8. Yu, X., Xu, C., Leong, H. W., Tian, Q., Tang, Q., Wan, K. W.: Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In: Proc. ACM Multimedia, pp. 11-20. (2003)

9. Ohno, Y., Miura, J., Shirai, Y.: Tracking players and estimation of the 3D position of a ball in soccer games. In: Proc. ICPR, pp. 145-148. (2000)

10. Intille, S. S., Bobick, A. F.: Closed-world tracking. In: Proc. ICCV, pp. 672-678. (1995)

11. Needham, C., Boyle, R.: Tracking multiple sports players through occlusion, congestion and scale. In: Proc. BMVC, pp. 93-102. (2001)

12. Cai, Q., Aggarwal, J. K.: Tracking human motion using multiple cameras. In: Proc. ICPR, pp. 68-72. (1996)

13. Javed, S. G., Rasheed, Z., Shah, M.: Human tracking in multiple cameras. In: Proc. ICCV, pp 331-336. (2001)

14. Stein G.: Tracking from multiple view points: self-calibration of space and time. In: Proc. DARPA IU Workshop, pp. 1037-1042. (1998)

15. Black, J., Ellis, T., Rosin, P.: Multi view image surveillance and tracking. In: Proc. IEEE Workshop on Motion and Video Computing, pp. 169-174. (2002)

16. Xu, M., Orwell, J., Lowey, L., Thirde, D. J.: Architecture and algorithms for tracking football players with multiple cameras. IEE Proceedings - Vision, Image and Signal Processing. 152(2), 232-241 (2005)

17. Ren, J.-C., Orwell, J., Jones, G. A., Xu, M.: Real-time Modeling of 3D Soccer Ball Trajectories from Multiple Fixed Cameras. IEEE Trans. Circuits Systems for Video Technology, 18(3), pp. 350-362 (2008)

18. Kim, T. Seo, Y., Hong, K. S.: Physics-based 3D position analysis of a soccer ball from monocular image sequences. In: Proc. ICCV, pp. 721-726. (1998)

19 Reid, I., North, A.: 3D trajectories from a single viewpoint using shadows. In: Proc. BMVC, pp. 863-872. (1998)

20. Stauffer, C., Grimson, W. E. L.: Adaptive background mixture models for real-time tracking. In: Proc. CVPR, pp. 246-252. (1999)

21. Tsai, R. Y.: An efficient and accurate camera calibration technique for 3D machine vision. In: Proc. CVPR, pp. 364-374. (1986)

22. Xu, M., Ellis, T.: Partial observation vs. blind tracking through occlusion. In: Proc. BMVC, pp. 777-786. (2002)

23. Kawashima, T., Yoshino, K., Aoki Y.: Qualitative image analysis of group behaviour. In: Proc. CVPR, pp. 690-693. (1994)

24. Canny, J.: A computational approach to edge detection. IEEE T-PAMI. **8(6)**, 679-698 (1986)