
UNSPECIFIED.

This version is available at https://strathprints.strath.ac.uk/29423/

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (https://strathprints.strath.ac.uk/) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk

The Strathprints institutional repository (https://strathprints.strath.ac.uk) is a digital archive of University of Strathclyde research outputs. It has been developed to disseminate open access research outputs, expose data about those outputs, and enable the management and persistent access to Strathclyde's intellectual output.
Generating Ball Trajectory in Soccer Video Sequences

Jinchang Ren¹,², James Orwell³, Graeme A. Jones³

¹ School of Computers, Northwestern Polytechnic University, Xi’an, China
² Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K.
npurjc@nwpu.edu.cn j.ren@surrey.ac.uk
³ Digital Imaging Research Centre, Kingston University, Surrey, U. K.
{j.orwell, g.jones}@kingston.ac.uk

Abstract. This paper demonstrates innovative techniques for estimating the trajectory of a soccer ball, using fixed cameras with constant calibration parameters. In contrast with broadcast coverage, for fixed camera data, the ball is often rendered with poor resolution away from the image centre. The rapidly moving ball is subject to motion-blur, caused by finite shutter speeds and interlaced fields, resulting in variable shape, size and colour. The velocity estimated from Kalman tracking is used in both normalising ball size and filtering the ball from false alarms. Furthermore, occlusion-reasoning and tracking-back methods are utilized to estimate its position when it is occluded, and also to remove false alarms. Finally, temporal hysteresis based thresholding of the ball likelihood is applied for trajectory filtering to improve the robustness and continuity of the tracked ball. Promising experimental results from several long sequences are reported.

1 Introduction

The convergence of computer vision and multimedia technologies has led to opportunities to develop applications for automatic soccer video analysis, including content-based indexing, retrieval and visualization [1-3]. Through image and motion analysis, additional information can be extracted for better comprehension of video and sports contents, such as video content annotation, summarization, team strategy analysis and verification of referee decisions, as well as further 2-D/3-D reconstruction and visualization [4-9].

In a soccer match the ball is invariably the focus of attention. Although players can be successfully detected and tracked on the basis of colour and shape [1, 3, 6, 9], similar methods cannot be extended to ball detection and tracking for several reasons:

- The ball is very small and moves fast, and consequently exhibits it usually has irregular shape, various size and unstable colour when moving in different velocities (see examples in Fig 1);
- It is hard to identify a ball as it is frequently occluded or possessed by players;
- There are many false alarms similar to the ball, such as small regions near the field lines and regions of players’ bodies.
Generally, TV broadcast streams and video sequences from fixed cameras are the most common sources of soccer videos. In 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 colour and circular shape are employed to detect balls in image sequences. In Yu et al [8], candidate balls are first identified by size range, colour and shape, and further verified using motion information obtained from a Kalman filter. In Yow et al [2], ball detection is completed by template matching in each of the reference frames and then the ball is tracked between these frames. In Seo et al [6], template matching and Kalman filter are used to track balls after manual initialization. Since colour and shape varies considerably in soccer games (see Fig. 1), these methods seem unlikely to provide robust solutions.

Using multiple fixed cameras has the advantage that calibration is easier to establish and that accurate on-field positions can be extracted for visualization. Bebie and Bieri [4] and Matsumoto et al [5] used two and four cameras in their systems for soccer game reconstruction and optimized viewpoint determination, respectively. Ohno et al [9] adopted eight cameras arranged on both sides of the field to attain full view of the game. Although motion-based tracking models are introduced in [4] and [9], there is no given process to automatically identify the ball before tracking. In Matsumoto et al [5] and D’Orazio et al [7], template matching and a modified Hough transform are presented to detect balls in soccer videos respectively. Since irregular ball shapes are usually extracted in different velocities, these two methods are still insufficient.

In this paper, a comprehensive model based method is proposed for ball detection and tracking from real soccer sequences. The main highlights of our method can be summarized below. Firstly, ball classification is performed on the Kalman tracked segmented objects which allows velocity information to be employed in the classification stage. Secondly, the expected appearance of a moving ball is explicitly modelled to improve the ball classification process. Thirdly, occlusion-reasoning and tracking-back is employed to recover any ball merged with players as well as to remove false alarms. Finally, temporal hysteresis based thresholding of the ball likelihood is used to further improve the robustness and continuity of the tracked ball.

Fig. 1. Ball samples in different size, shape and colours from same image sequences: (a) to (d) are balls of static, slow, medium and fast moving, respectively.
2 Detecting and Tracking of Moving Objects

Image differencing is utilized for moving object detection followed by Kalman filter based tracking in the field of view (FOV) of each individual camera. For robustness, a two-stage adaptive background model is applied. In the first stage, a per-pixel Gaussians mixture model \[14, 17\], \((\mu_k^{(l)}, \sigma_k^{(l)}, a_k^{(l)})\), is used to estimate an initial background, where \(\mu_k^{(l)}\), \(\sigma_k^{(l)}\) and \(a_k^{(l)}\) are the mean, root of the trace of covariance matrix, and weight of the \(l\)-th Gaussian distribution at frame \(k\). For a new pixel observation, a matched distribution is updated with increasing weight as follows:

\[
\begin{align*}
\mu_k &= (1 - \rho)\mu_{k-1} + \rho I_k \\
\sigma_k^2 &= (1 - \rho)\sigma_{k-1}^2 + \rho (I_k - \mu_k)^T (I_k - \mu_k)
\end{align*}
\] (1)

where \(\rho\) 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.

In the second stage, this initial background image is continuously updated using a faster running average algorithm for efficiency \[15\]:

\[
\begin{align*}
\mu_k &= [\alpha_r I_k + (1 - \alpha_r)\mu_{k-1}]F_k + [\alpha_u I_k + (1 - \alpha_u)\mu_{k-1}]F_k \\
\end{align*}
\] (2)

where \(0 < \alpha_r \ll \alpha_u << 1\). This method helps to slowly update the background image even in foreground regions.

Given the input image \(I_k\), we can decide the foreground binary mask \(F_k\) by comparing \(\|I_k - \mu_{k-1}\|\) 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. An image-plane Kalman tracker is used to filter noisy measurements and split merged objects, in which the state \(x_j\) and measurement \(z_j\) are given by:

\[
\begin{align*}
x_j &= [r_0, c_0, \dot{r}_0, \dot{c}_0, \Delta r_1, \Delta c_1, \Delta r_2, \Delta c_2]^T \\
z_j &= [r_0, c_0, r_1, c_1, r_2, c_2]^T
\end{align*}
\] (3-1-2)

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 \((r_0, c_0)\) to \((r_1, c_1)\) and \((r_2, c_2)\).

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

\[
x_j(k + 1) = A_j x_j(k) + w_j(k)
\] (4)
where \( w_i \) and \( v_i \) are the image plane process noise and measurement noise, and \( A_i \) and \( H_i \) are the state transition matrix and measurement matrix, respectively. Given \( \Delta T \) as the time interval between two successive frames (for image formation), we have \( A_i \) and \( H_i \) defined as

\[
A_i = \begin{bmatrix}
1 & 0 & \Delta T & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \Delta T & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}, \quad H_i = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

The Mahalanobis distance is used to associate each observation to (at most) one tracked object, in which several states are defined to identify different cases. These states include new, normal, merged, missing and terminated, which are defined in the following process: For each existing tracked object, if it has corresponding observation matched, we define it in normal state. Otherwise, it is marked as missing and updated by predicted state estimation. If an object has been missing for more than \( M \) frames, it is terminated. All unmatched observations are identified as new objects in tracking. If different objects share common regions in their bounding boxes, they are merged. Moreover, we define age as the frames that an object has been tracked, and a new object should have its age as 1. Detail on splitting objects when their bounding boxes are merged together can be found in [18].

Fig. 2. Examples of forty seconds of tracking data, in which time \( t \) moves from left to right, and the horizontal image co-ordinates of the object centroids, \( c_0 \), is plotted up the \( y \)-axis. (The vertical image co-ordinate is omitted from this diagram).
Fig. 2 plots the trajectories of multiple objects from frame 1000 to 2100 in camera sequence 1. Using the Tsai’s algorithm for camera calibration [17], the measurements are transformed into world co-ordinates, using the provisional assumption that the object is located on the ground plane. A measurement in world co-ordinates is defined as \( \mathbf{m}_j = [w \ h \ a]^T \), where \( w, h \) and \( a \) are object’s width, height and area calculated by again assuming it is touching the ground plane.

Except for the ball and players, Fig 2 also contains trajectories from false alarms caused by field line noise and partial body of players, etc (see Fig 3 below). In the next Section, we will discuss the process to filter the ball from players and other false alarms.

![Fig. 3. Enlarged images of detected moving objects in different colour boxes with the ball (in white), players (in blue), and false alarms (in red or yellow).](image)

### 3 Filtering of the Ball Trajectory

Domain knowledge including colour, ball shape and pitch geometry are widely adopted in soccer and other sports video analysis systems [1-3, 5, 7-9, 13]. Using the closed captions, audio, slow-motion replays and special zoom, more specific models could be explored in shot detection and semantic indexing of broadcast data [3, 10-13]. However, for analyzing real soccer sequences we need to explore some new spatial-temporal constraints for filtering the ball.

#### 3.1 Forward Filtering

After tracking, in each frame every tracked object is assigned with a tracking state, an age and an estimated velocity vector. In our ball filtering process, *velocity* and *longevity* features, along with *size* and *colour*, are employed to discriminate the ball from other objects. Owning to motion effect, a moving ball usually appears larger than a static one. Let us suppose the ball has a constant velocity during image formation, the detected width \( w \) and height \( h \) in frame \( n \) satisfy
\[
\begin{align*}
    w(n) &= w_0 + v_x(n)\Delta T \\
    h(n) &= h_0 + v_y(n)\Delta T
\end{align*}
\]

where \(w_0\) and \(h_0\) are the stationary width and height of the detected ball, and \(\Delta T\) is the temporal aperture.

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 \(v_i\) and longevity \(n_i\) as below:

\[
D(o_i) = D_1(o_i) + \frac{1}{2} \frac{|v_i|}{v_{max}} (1 - e^{-nT_0})
\]

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. Typically the moving ball moves more quickly than players and is often the fastest moving object. The bias \(D_1(o_i)\) is defined as

\[
D_1(o_i) = \begin{cases} 
0.5 & \text{if } w_0 \in R(w), h_0 \in R(h), a \in R(A) \\
0 & \text{otherwise}
\end{cases}
\]

\[
\begin{align*}
&c \in R(c), \frac{2}{3} h_0 < w_0 < \frac{3}{2} h_0
\end{align*}
\]

where \(R(x) = [x_1,x_2]\) specifies an allowed range of \(x\). Furthermore, \(a\) and \(c\) are area and ball colour percentage, respectively.

\[\text{Fig. 4. Filtered results of the data in Fig 2, in which green and blue part of trajectories are results of higher or lower ball likelihood, respectively.}\]

In fact, equation (10) is used to extract a near circular ball candidate of its size and colour within valid ranges. The results of filtering the trajectories shown in Figure 2
using equations (9) and (10) are presented in Fig. 4. Blue trajectories are those of trajectories whose likelihoods are between 0.5 and 0.65, while green ones trajectories are those with likelihoods higher than 0.65. Moreover, in one certain frame we may find several green or blue trajectories. This happens due to occlusion or false alarms and will be resolved in Section 3.2.

3.2 Occlusion Reasoning and Tracking-back

To resolve the uncertainties within the filtered ball trajectories, occlusion reasoning is applied based on tracking states obtained from Kalman filter. Assume a candidate ball $B_i$ is occluded at frame $n$ (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 $n_{th}$ 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 before finally determining the real ball trajectory. If we find it is still merged at frame $(n + \Delta n)$, where $\Delta n > n_{th}$, then tracking-back is employed to reclassify the state of frame $(n + \Delta n)$ (and those after frame $n$) as possessed.

In addition, each new ball candidate is also examined by tracking-back process, as we assume there should be at least one player $P_j$ accompanied this ball nearby. Thus a new ball should always come from a player who possessed it; otherwise it must be a false alarm. Moreover, if a candidate ball has an age less than a given threshold, say 4 frames, it is also considered as a short-lived false alarm (caused by inaccurate foreground detection, see examples in Fig 3).

Finally, tracking-back also applies temporal hysteresis-based thresholding [16] 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 desig-
nated a ‘ball’ label; and candidates with a likelihood below $h_1$ are unequivocally classified as ‘non-ball’ (i.e. false alarms). Candidate objects with likelihood $l$ satisfying $h_1 > l > h_2$ are relabelled as a ‘ball’ if the object as been labelled as a ball in a neighbouring frame (along the tracking history). Similarly, objects with likelihood $l$ satisfying $h_2 > l > h_3$ are labelled as ‘not ball’ if the object has that label as a ball in a neighbouring 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. Comparing the final ball trajectory in Fig 5 with those trajectories of both players and the ball in Fig 2 clearly demonstrates that tracking-back has effectively recovered the ball positions even when it is occluded (being possessed with players). At the same time, false alarms are dramatically reduced.

4 Results and Discussions

The proposed algorithm has been tested on several sequences captured from fixed cameras around a football stadium. To quantitatively evaluate our method, detected ball positions are compared with manual ground truth (GT) data, which includes image-plane bounding box of the ball and the centroid. In the whole sequence, those frames containing the ball are labelled, and whether the ball is isolated from players are also marked. For three sequences of 4800 frames each, we have totally about 6700 ball positions defined in the GT data. Fig. 6 gives examples of two ball trajectories tracked from two different sequences.

When there is no buffering and tracking-back, we can only detect about 48.1%, 46.1% and 46.6% ball positions from the three sequences, respectively. However, in each sequence more than 80% of the isolated balls can be successfully identified during the tracking. When the tracking-back is introduced with a buffer of 25 frames, the detection rate improves by 28% or more for each sequence, in which about 37% additional merged ball can be successfully recovered. When the buffer size increases to 50 frames, nearly 80% of the ball positions are localised.

The tracked rates under different buffer size are compared in detail in Table 1, and it seems that 50 frames of buffering is a good trade-off between correct tracking rates and the need for short latencies in a live stream. At each frame, we define tracked rate $R$ is then defined as

$$R(b_n) = \frac{\text{Area}(b_n) \cap \text{Area}(g_n)}{\text{Area}(b_n)}$$ (11)

where $b_n$ is the ball under tracking and $g_n$ is the corresponding ground truth at frame #n.
Fig. 6. Examples of two image-plane ball trajectories from Seq #2 (top) and Seq #3 (bottom), respectively.

Table 1. Tracked rate of the ball under various buffer size for tracking-back

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Buffer size</th>
<th>None buffering</th>
<th>25 frames</th>
<th>50 frames</th>
<th>75 frames</th>
<th>100 frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Separated ball</td>
<td>Merged ball</td>
<td>Overall</td>
<td>Separated ball</td>
<td>Merged ball</td>
</tr>
<tr>
<td>Seq #1</td>
<td>None</td>
<td>82.5%</td>
<td>26.6%</td>
<td>48.1%</td>
<td>81.4%</td>
<td>22.9%</td>
</tr>
<tr>
<td></td>
<td>25 frames</td>
<td>86.8%</td>
<td>62.9%</td>
<td>76.3%</td>
<td>84.2%</td>
<td>60.5%</td>
</tr>
<tr>
<td></td>
<td>50 frames</td>
<td>88.0%</td>
<td>68.4%</td>
<td>79.1%</td>
<td>85.3%</td>
<td>64.8%</td>
</tr>
<tr>
<td></td>
<td>75 frames</td>
<td>88.4%</td>
<td>70.7%</td>
<td>79.4%</td>
<td>85.7%</td>
<td>65.7%</td>
</tr>
<tr>
<td></td>
<td>100 frames</td>
<td>88.6%</td>
<td>71.1%</td>
<td>79.6%</td>
<td>85.9%</td>
<td>66.3%</td>
</tr>
</tbody>
</table>

Let us suppose there are totally $M$ ball positions in a given sequence amongst which we have $M_1$ frames in which the ball appears separately while in the other frames the ball is merged. We define the overall tracking accuracy as the tracked rates of both separated ball and merged ball weighted by their frequency of occurrence. Taking the distance between the centroid of the detected ball and the centroid of the
bounding box in GT data as an error measurement, we can further evaluate the accuracy of the tracked ball. Fig. 6 illustrates two ball trajectories (a) with average tracking accuracy (b), from which we can see that about 91% of the ball positions are tracked within 6 pixels spatial deviation.

![Graph showing error distribution](image1)

**Fig. 7.** Accumulative probability of average tracking accuracy (left) and its probability density function (right)

### 5 Conclusions

We have proposed a novel method for soccer ball detection and tracking from real video sequences. We found that using motion information and modelling the expected appearance of a moving ball significantly improves the detection accuracy. The application of an occlusion-reasoning process based on tracking-back and temporal hysteresis-based thresholding of ball likelihoods is essential to improve the tracking robustness and continuity of the ball trajectory. The effectiveness of the tracking-back approach is dependent on the size of the buffering. By comparing results for different buffer sizes and appropriate trade-off between the accuracy and latency is also suggested. Future work will investigate more accurate modelling and recognition of occluded balls as well as 3-D positioning and soccer event understanding.

### Acknowledgement

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