TRACKING THE BALL AND PLAYERS FROM MULTIPLE FOOTBALL VIDEOS *

KYUHYOUNG CHOI, BOOIL PARK†, SUBIN LEE and YONGDUEK SEO
Dept. of Media Technology, Sogang University, Mapo-Gu
121-742, Seoul, Korea

Received (to be inserted by publisher)

This paper deals with tracking players and the ball in multiple soccer video sequences taken from fixed cameras located around the stadium. One of the major obstacle in player tracking occurs when occlusion between people results in not enough observation information. If a player blob is not visible in some of the input videos due to occlusion, one solution is to check whether he is visible in others. This check is helped by the homography transformation induced by the play-ground because the players' positions can be represented on a virtual planar soccer field and can be mapped into any video image through a corresponding homography between the virtual field and the video image. Therefore, the position estimate of the player is exploited to predict and check the visibility in each of the video images. Measurements are taken from those visible images. On the other hand, before initiating tracking, we do a color correction in order to enhance the measurement and matching process because each of the video produces different RGB values for the same scene object due to different camera position and color sensor characteristics. After tracking players, ball tracking is carried out by eliminating the image blobs of players and accumulating the ideally ball-only images. We implemented our tracking algorithm, and experiments for sets of multiple real videos showed promising results.

Keywords: Multi-object tracking, multi-view tracking, sports video analysis

1. Introduction

Soccer has been one of the most popular sport in the world, and it will help to improve the efficiency in games, if we can analyze matches by building up a database of players' trajectories. Tracking players in this paper means that a computer follows the locations of players during a given sequence. Afterwards, we expect that a reconstruction of a game can be made from the recorded trajectories for intensive strategy analysis for instance.

Recently, tracking soccer players and/or ball has been investigated. In [Seo et al., 1997], the ball is tracked by template matching and Kalman filter, when there is no player close enough to the ball. When the ball is occluded by

*A major part of this work has also been presented in [Park et al., 2005] at International Conference on Intelligent Computing 2005
†She is now with LG Electronics.
nearby players, they are searched for the ball in a bounding box around the players until a good detection was obtained. Homography is used to map the image into a domain of soccer pitch.

Fig. 1: Our four-camera configuration.

[Yan et al., 2004] introduced a system which enables users to enjoy a 3D reconstructed soccer match from various points of view based on the tracking results of players and ball, even though the tracking problem does not seem to be fully investigated yet. The most difficult problem in player tracking is caused by occlusions between players due to the nature of ball sports. Sometimes even human eyes can hardly track them when a pretty long occlusion occurs between players in the same uniform. To address this, [Ok et al., 2002] used particle filters with repulsion principle between particles of different players. In [Choi & Seo, 2004], a robust ball trajectory tracking, one of the main issues in football video analysis, was possible using the result of players tracking. Several recent papers have tried to use images from multiple cameras to solve the occlusion. In [Koyama et al., 2003], a camera was installed on the roof of a stadium so as to make the tracking problem easy. The work of [Kang et al., 2003] utilized homography transformation to find non-occluded blobs from multiple views for soccer player tracking. In [Figueroa et al., 2004], four cameras are used and the image blob is split into pieces based on the color information on occasion of occlusion. The player trajectories are represented by graph. In [Iwase & Saito, 2004], tracking results from 18 cameras are displayed on virtual ground image using homography so that users can see where the players are in the real ground.

The input sequences in our study are taken by a system of eight cameras installed in a stadium. Four cameras that cover the right half of the pitch are illustrated in Figure 1. Figure 2 shows images of a real game taken from the four cameras. As one can see, the four images have different colors even at the same scene location due to different sensor characteristics of the cameras. We apply color correction before initiating the tracking algorithm, explained in Section 2.1. Our basic idea is given with the help of Figure 3. While players 2 and 3 appear in the same blob in the images of camera 1 and 2, they appear in separate blobs from camera 3 and 4. Therefore the positions of player 2 occluded by player 3 from camera views 1 and 2 can be estimated using those of camera 3 and 4, computed by plane-to-plane homography transformation. The ball tracking method shown in [Choi & Seo, 2004] is used in our system to estimate the ball trajectory. Full explanation of tracking is given in Section 2. Section 3 shows a result of our tracking and finally Section 4 concludes this paper. Followings are the symbols used throughout this paper.

- $M_n$: image sequence taken from $n$th camera
- $I_n(t)$: $t$th image of $M_n$
- $B_n$: background image of $M_n$
- $I_g$: playground (image) of real soccer field size
2. Proposed Algorithm

Our aim is to track the blobs of soccer players in image sequences taken from multiple fixed cameras. Low level image processing is done according to the algorithm in [Choi & Seo, 2004], but not explained in this paper due to space limit.

2.1. Color correction

Before running the tracking algorithm, we apply a color transformation so that the four images may have the same color at the corresponding pixels. First, we choose one of the four as the reference image in which a small window is selected manually. Then its average color vector \( c_R \) is computed. The average vector \( c \) for the same scene area in one of the rest images is computed, too. The transformation parameters consist of a rotation \( R \) and positive scale \( \lambda \), which are then found to be \( Rc = c_R \) and \( \lambda \|c\| = \|c_R\| \).

The rotation angle \( \theta \) and axis \( v \) of \( R \) are respectively given by:

\[
\begin{align*}
\theta &= \cos^{-1} \left( \frac{c \cdot c_R}{\|c\| \|c_R\|} \right) \\
v &= c \times c_R / \|c \times c_R\|
\end{align*}
\]

(1)

(2)

Finally, all the color pixels are transformed using the computed transformation. Figure 3 shows the output of color correction applied to the four views in Figure 2.

2.2. Initiating the tracker

In Figure 3, extracted player blobs are bounding boxed with numbers. Note that some players in some views are marked with one rectangle because they are extracted as a merged image blob due to their proximity. Considering that players in 2D images stand on the 2D playground, we choose the foot positions of players as the player positions (target state). This means that there is a 3 x 3 homography matrix between the image and the actual playground. Therefore, if we map their foot positions in the image into those in the playground \( I_g \), we can locate the players. The mapped points (for a player) from multiple cameras are not guaranteed to coincide in \( I_g \) because (1) there are noises in preliminary image processing and (2) some players may be extracted as a merged blob. To address this problem, following procedure is used to automatically group the mapped points (In [Iwase & Saito, 2004] required is a manual intervention).

1. \( I_g \) is partitioned into grids (See Figure 4) to build up a two dimensional histogram. Here, a grid size of 5 x 5 in pixels is used.
2. For \( I_n(0), n = 1, 2, 3, 4 \), player blobs are extracted and their representative positions are transformed into \( I_g \) by the homographies, which are calculated using the patterns on the playground. Let \( p_{n,g} \) be a coordinate vector in \( I_g \).
3. After finding out the grid index in \( I_g \) for each \( p_{n,g} \), we increase the number of corresponding pixels and colors. (See Figure 4)
4. Done are steps 2 and 3 for each of the blobs and the images to have counts of pixels and colors for histogram grids in \( I_g \). The color of a grid is finally assigned to be the color of maximum count.
5. CCL(connected component labeling) applied to \( I_g \) to find blobs. Among the grids of a blob, the one with the maximum pixel count is be considered as the position of the player. Note that if there are two colors in a blob, it is taken as two split blobs.
6. For the grid location with maximum pixel count, computed are the positions in each of the images through the homographies.
7. Image matching (observation) is done based on the back-transformed coordinates given in 6. The mean location \( \left( \frac{1}{N} \sum_{n=1}^{N} p_{n,g} \right) \) is used as the position of the player, where \( N \)
is the number of cameras from which the corresponding player is detected.

Figure 4 shows an example of this process applied to the four images in Figure 3. In the counting of Step 3, we used a small window for a robustness.

Matching rule for getting observations

Matching between measured player blobs from an input image and players in the tracking list is done in the following way.

1. The difference between the predicted position and the measured should be small. Given an appropriate threshold $T_h$, The ID of $k$th blob is determined by:

$$i_k = \arg\min_i \left( \| p_{n(t)}^{pred}(i) - p_n(k) \| < T_h \right),$$

where $i$ denotes an ID in the player list, $p_{n(t)}^{pred}(i)$ is the predicted position of $i$th player in the $n$th image at time $t$ and $p_n(k)$ is the foot position of $k$th blob.

2. The color of ID satisfying condition 1 is the same as that of the matched blob.

Occlusion reasoning

We need choose measurements from multiple views when the position of a player blob is to be updated during the tracking. We transfer the bounding boxes of all the players into each of the multiple views; their image locations are calculated using the homography matrices. If the bounding box of a player in a view is not overlapped with any others, it is categorized as a valid measurement. We collect only those valid measurements during the state update. When the bounding box is overlapped with others, which means no observation from the view for the player, we record the player ID for the blob. Hence, blobs of occlusion will have a list of overlapped or occluded player IDs after applying the procedure to each of the players. The record is used later in the tracking to resolve whether a detected blob is from a player in the tracking list or from a new player. Following is the summary of this procedure:

1. Compute $p$, the position of a player $k$ in view $I_n$.
2. Find out blob(s) where $k$ belongs in $I_n$.
3. When there are more than one blob from step 2, choose the one with the biggest size, because the player’s position is assumed to be occluded by other.
4. Mark the chosen blob to indicate that $k$ is included in it.

Because it is known whether a blob in a view is a mixture blob or not, we may easily collect the blob of only one player from each of the views for the player. However, there can be no such a blob if a player is appeared as a mixture blob for every camera. In this case, the position prediction is not carried out for him.
Fig. 6: An instance of four view tracking of multiple players.

Fig. 7: Estimated trajectories of all 21 people. For better illustration, we show only a portion of the whole sequence.

2.5. **Ball tracking**

Even though ball tracking belongs to single object tracking while player tracking falls within multi-object tracking, ball tracking is not easier than players tracking due to following aspects. Usually ball blobs in images are very small, which makes it difficult to derive features from and to be characterized. Sudden changes in its motion is another factor to make it challenging. In addition, occlusion and overlapping with players causes a severe problem in tracking the ball continuously. The ball becomes invisible and appears at places where a continuous prediction could not reach. As shown in [Choi & Seo, 2004], we can achieve much higher signal to noise ratio by accumulating images without players and things outside the court. The image blobs of players are eliminated by filtering with respect to size. Accumulating those images for some frames gives an image with a long string of ball blobs and rather small or short blobs of noise. After morphological and shape filtering, we can obtain an image approximately showing the ball trajectory as in Figure 8 and Figure 9. On that image of ball trajectory, we apply a particle filter which is non-linear and non-Gaussian estimator. The string of ball blobs takes a role of a proposal density in sequential Monte Carlo framework by limiting the range of particle(hypothesis) generation.

3. **Experimental Results**

The four set of input video sequences have 403 frames long and were taken from four camcorders with rate of 30 frames per second using SONY VX2000 camera. Figure 6 shows a shot of the four views and bounding boxes of players under tracking. The result is given in the movie file. Figure 7 shows the trajectories of all 21 people through the sequences. Because the trajectories overlap each other, the figure shows only a part of the whole duration. Figure 10 shows a comparison of tracking results; the first column is the ground truth obtained manually, the second is a tracking result using only one camera, and the last is with all four cameras. Trajectories of players at every 100th frame are shown Figure 11 with estimated positions from each view. Because of our algorithm using the multiple sensor information, multi camera tracking gives better performance for the case of occlusion between players of the same uniform or a swarm of players.

4. **Conclusion**

In this paper, a better way of player tracking is proposed for soccer video sequences simultaneously taken from four different cameras. To address the difficulty arising from occlusion between players of the same team, information from multiple camera images are selectively exploited so that our algorithm may yield stable tracking results. For efficient ball tracking, the results of player tracking are used to give images of accumulated ball blobs which shows the ball
trajectory.

Acknowledgments

This work was supported by the Korea Research Foundation Grant. (KRF-2004-003-D00356).

References

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

Fig. 8: Ball-accumulated images from four cameras at 60th frame.

Fig. 9: Ball-accumulated images from four cameras at 120th frame.
Fig. 10: A performance comparison. Tracking results of single and four camera tracking in terms of player 9.
Fig. 11: Trajectories on the pitch plane and tracked players in colored rectangles on four camera views at first (top), 100 th, 200 th, 300 th and 400 th (bottom) frames.