Computer Vision and Image Understanding, Vol. 113, No. 5, pp. 653-662, 2009.

Ball Route Estimation under Heavy Occlusion in Broadcast Soccer Video

Jun Miura [a,*] Takumi Shimawaki [a] Takuro Sakiyama [a]
Yoshiaki Shirai [b]

[a] *Department of Mechanical Engineering, Osaka University, Suita, Osaka 565-0871, Japan*

[b] *Department of Human and Computer Intelligence, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan*

Abstract

This paper deals with analysis of broadcast soccer video. Estimation of ball movements is necessary to recognize interesting scenes such as goal scenes. It is, however, sometimes difficult to detect a ball using a simple color-based or shape-based method when the ball overlaps with multiple players and lines. We therefore developed a method of estimating a ball route during such overlaps by considering spatio-temporal relationships among players, lines, and the ball. The method can deal with difficult cases such as one in which a ball disappears at a player and reappears from another player. Experimental results demonstrate the effectiveness of the method.

Key words: Ball tracking, Broadcast soccer video, Occlusion

1 Introduction

Summarizing broadcast soccer videos (and other sports videos) is increasingly demanded to produce digests of interesting scenes in a game (e.g. goal scenes) so that viewers can quickly survey the game. We are now developing a system for retrieving interesting and informative scenes based on scene understanding. It is necessary to know the movements of players and the ball to understand

* Corresponding author.
E-mail address: jun@ics.tut.ac.jp (J. Miura, currently with Toyohashi University of Technology)

various scenes in soccer games. This paper specifically examines ball detection and tracking.

Several ball detection methods exist, which use, for example, SVM [1] or a generalized Hough transform [2]. Most previous methods are, however, applicable only when a ball is sufficiently large and is not moving quickly in the image. In our case, a ball is usually small and sometimes moves fast. Moreover, it often overlaps with players and lines. Ball tracking in such a case is a challenging computer vision problem.

Several ball-tracking methods deal with occlusions or overlaps by applying statistical filters such as a Kalman filter [3] and a particle filter [4] or using ball-trajectory models [5,6]; these methods can handle only short-term occlusions or overlaps using motion continuity. In actual scenes, however, players and lines often mutually overlap; consequently, a ball which has been overlapped with (or occluded by) a player might appear from another player's region after a certain period of time. Because the ball motions before and after an overlap (or occlusion) period might not be continuous in such cases, it is necessary to examine possible *routes of the ball* (i.e., a sequence of objects which overlap with (or occlude) the ball) based on spatio-temporal relationships among players, lines, and the ball.

This paper describes a method of estimating the ball route when it overlaps with players and lines in a broadcast soccer video. We use images taken from the *center camera*, which has the widest scope of any camera on the field. We detect shots from that camera automatically and estimate the camera parameters during those shots. Using the estimated parameters, all image data are transformed into a fixed *image*- coordinate system, where all analyses are performed.

The remainder of the paper is organized as follows. Section 2 briefly describes shot detection and the parameter estimation. Section 3 briefly explains how to track players and how to track a ball when the ball does not overlap with players and lines for a long period. Section 4 presents a detailed description of the ball route estimation during overlaps. Section 5 summarizes important points of the paper and presents a discussion of future work.

2 Camera Parameter Estimation

The camera position is estimated once in advance by manually matching the lines and the frame of the goal in the image with those of a model of the field. The other three parameters (pan, tilt, and zoom) are estimated at every frame because they usually change from frame to frame. Our estimation method comprises the following three processes.

- Detection of a shot from the center camera.
- Estimation of initial camera parameters at the first frame of the shot.
- Estimation of camera parameters in subsequent frames [7].

This section briefly explains the first two parts. Refer to [8] for more details.

2.1 Detecting Shots from the Center Camera

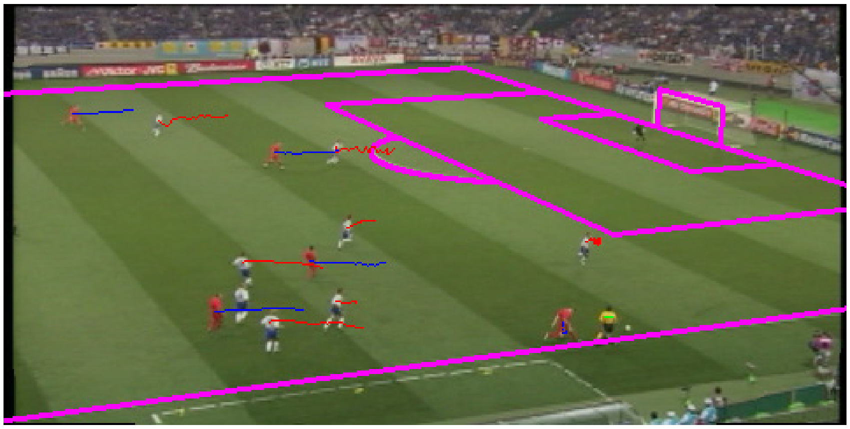
We use a color histogram of 48 bins (16 bins for each color (R, G, B)) for shot change detection; for each pixel of an image, the value of each color component is added to the corresponding bin. The current frame is considered to be the first frame of a new shot if the sum of the differences between the corresponding bins in the current and the previous frame exceeds a threshold. Using the size of the ground-colored (green) region, the size of the players, and the number of horizontal lines used, this first frame is analyzed to determine if the shot is taken by the center camera.

2.2 Estimating Camera Parameters in a Shot

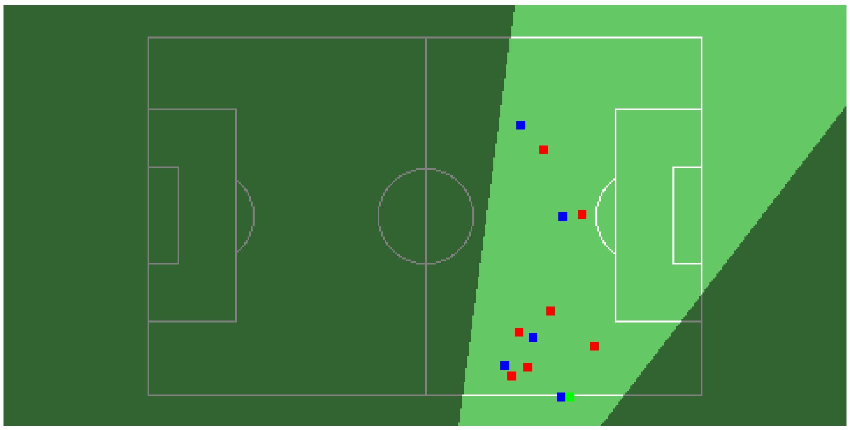
The initial values of the three camera parameters can be estimated from two pairs of intersections of lines in the image and those in the field model. We first generate a set of such pairs using the possible ranges of the gradients of the model lines *in the image* and then select the one which maximizes the degree of matching between the projected model lines and the line regions in the image.

Once the initial parameters are determined, the parameters are estimated continuously in subsequent frames using a local search-based method [7]. Figure 1 depicts a result of the camera parameter estimation. Figure 1(a) presents the result of projecting model lines onto the ground using the estimated parameters. In Fig. 1(b), the bright area corresponds to the field of view of the camera; red and blue points represent the players' positions.

Using the estimated parameters, all images in a shot are transformed into the coordinate system of the first one of the shot. We perform the following analyses in that coordinate system.



(a) projected lines.



(b) Field of view and player positions.

Fig. 1. Camera parameter estimation.

3 Player and Ball Tracking

3.1 Detecting and Tracking Players

We use a color-based player tracking method [7]. The colors of the uniform in the HSV space are registered in advance for each team. This color information and the sizes of regions are used for extracting player regions. We use a simple linear motion model for tracking. When two players overlap, we determine which one is occluded using their colors and vertical positions in the image. A snapshot of player tracking is presented in Fig. 1.

We use images of 480×240 pixels. The ball diameter in the image is 3–9 pixels, depending mainly on the zoom value of the camera. Because the ball is small and sometimes moves fast, it is difficult to detect reliably from a single image. We therefore extract ball candidate regions every frame and see if they form a continuous movement. A ball candidate region is a white region inside the ground whose size and aspect ratio are within some predetermined ranges. For each ball candidate region in a frame, we search its neighbor in the next frame for candidate regions. The presence of the ball is hypothesized if we find candidate regions in three consecutive frames.

For this hypothesis, we track it using a simple linear prediction where the predicted position $\boldsymbol{x}(t)$ at frame t is calculated using an averaged speed during the previous three frames as follows:

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-1) + \left(\boldsymbol{x}(t-1) - \boldsymbol{x}(t-3)\right)/2. \tag{1}$$

We continue tracking while repeating the prediction if no candidates are found around the predicted position. We consider that the ball has disappeared (occluded or overlapped) if we do not find ball candidates for five consecutive frames.

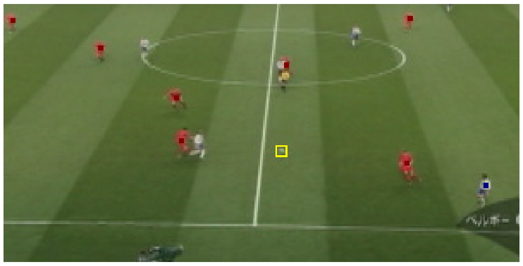
Figure 2 depicts a result of ball tracking with a short overlap. The yellow boxes indicate a tracked ball; green ones indicate predicted positions when a ball candidate is not extracted; blue ones indicate other ball candidates.

We use the resolution because it resembles that of usual videos. Although the use of high-resolution videos might simplify detection of the ball when it is not occluded, handling occlusions and overlapping of the ball is still necessary.

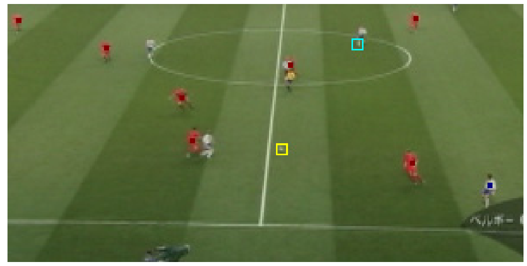
4 Ball Route Estimation during Overlaps

The simple ball detector fails to detect a ball when it overlaps with players (or referees), lines, or stands (see Fig. 3). As described above, a difficult situation is that a ball overlaps continuously with several players and lines. Figure 4 presents such a situation: (a) a ball is detected near two red players (R1 and R2); (b) the ball overlaps with a line (L13) and tracking is suspended; (c)–(d) a red player (R1) keeps the ball; (e) he kicks the ball; (f) the ball is detected again behind the white player (W1).

We examine the frames between the disappearance and the re-appearance of

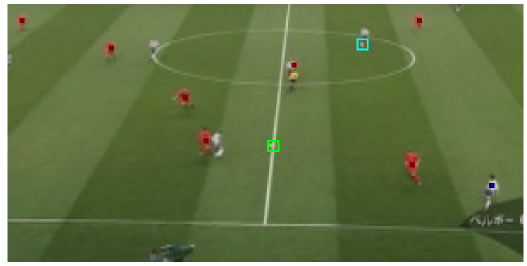


(a) 65th frame



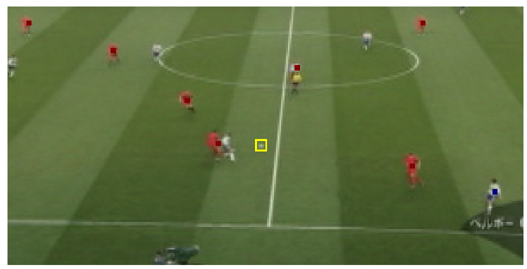
(b) 66th frame

(c) 67th frame



(d) 68th frame

(e) 69th frame



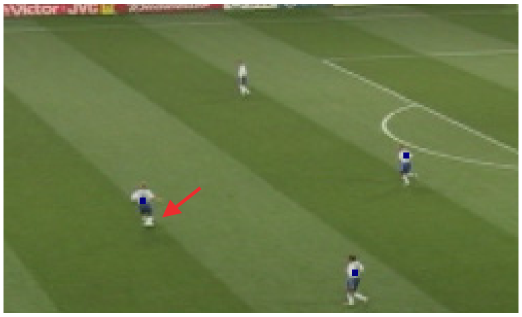
(f) 70th frame

Fig. 2. Result of ball tracking.

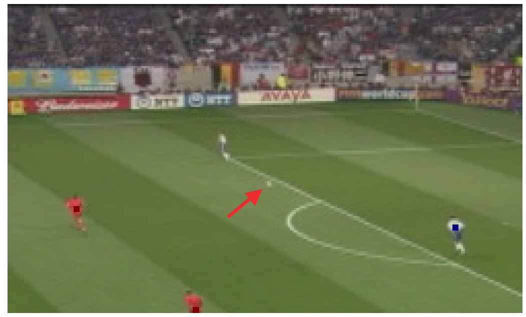
a ball to estimate the ball route in the following steps.

(1) Enumerate possible transitions of the ball between objects (players, lines, or the stands) that overlap with the ball.
(2) Generate ball route candidates considering spatio-temporal relationships between the objects and the ball.
(3) Generate a rough ball trajectory for each ball route candidate, if possible, by considering constraints on ball movements.
(4) Evaluate the trajectories based on the detection of ball-like regions; select the best trajectory and, therefore, the best ball route.

The following subsections explain these steps in detail.



(a) player



(b) line

(c) the stands

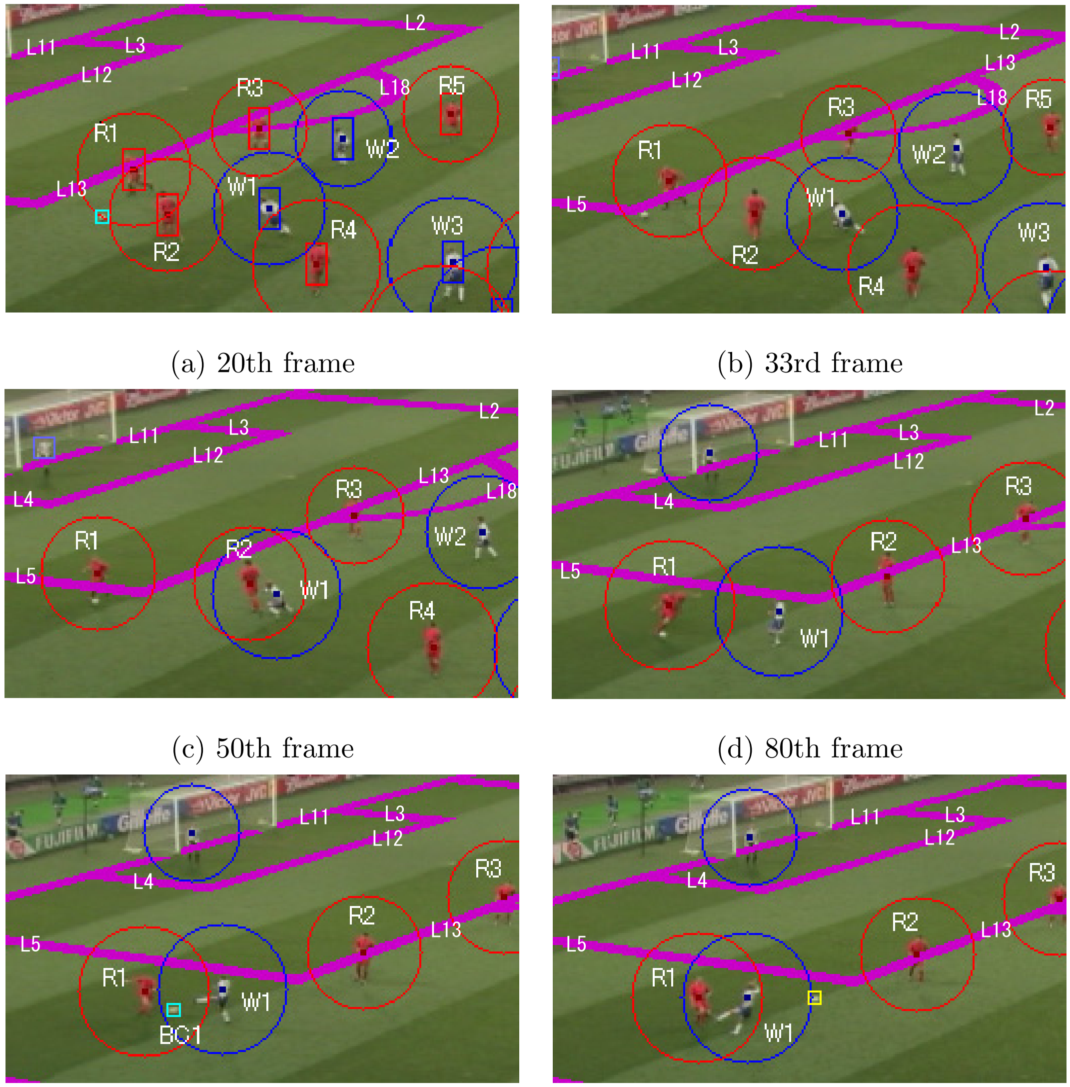
Fig. 3. Exemplary cases in which a simple ball detection fails.

4.1 Enumerate Ball Transitions

We construct a graph called a *transition graph*, which enumerates possible transitions of a ball between objects. Nodes of the graph include objects that might overlap with the ball: players, lines, and the stands. A *ball candidate*, which is an isolated ball candidate, is also represented as a node. Links consist of possible transitions between the nodes. Lines on the ground are divided into straight lines and curved ones; the stands are divided into four regions. Figure 5 shows the nodes of lines and the stands. We consider the following transitions:

(1) player $\longleftrightarrow$ player, ball candidate, line.
(2) ball candidate $\longleftrightarrow$ line.
(3) line or the stands $\longleftrightarrow$ line or the stands.

The transitions, including players and ball candidates, are *temporary* and effective only while two nodes are sufficiently close. In Fig. 4, the circle drawn around each player shows the range within which the ball can move in the next frame. The ball might make a transition to one of them if another player's centroid or a line exist in this circle. Transitions between lines and the stands are *fixed*; transitions between adjacent nodes in Fig. 5 are possible.



(a) 20th frame (b) 33rd frame

(c) 50th frame (d) 80th frame

(e) 87th frame (f) 93rd frame

Fig. 4. A scene in which a ball has not been detected for a long period.

Figure 6 shows some transitions generated from Fig. 4. Labels of nodes, W*, R*, L*, and BC* respectively indicate white players, red players, lines, and ball candidates. For example, because a white player (W1) exists near a red player (R1) during frames 90–93 (see Fig. 4(f)), transition R1→W1 for that period is generated; also, because R1 is near a line (L5) during frames 35–93 (see Figs. 4(c)– 4(e)), transition R1→L5 is generated.

4.2 Generate Ball Route Candidates

We generate ball route candidates by searching the transition graph for possible routes connecting the node where a ball disappears and the node where the

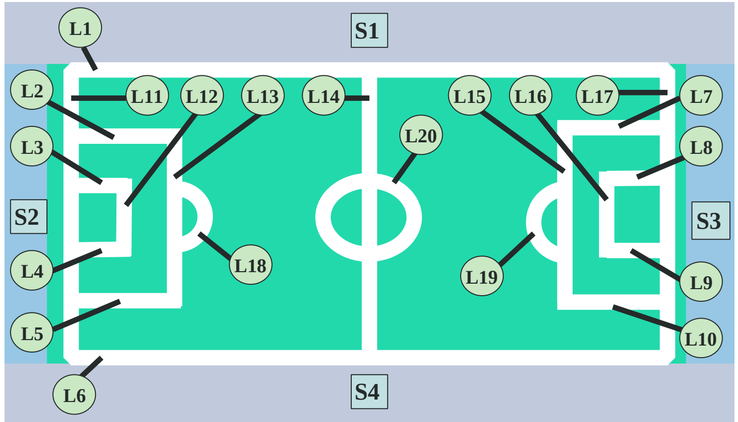


Fig. 5. Division of lines and the stands.

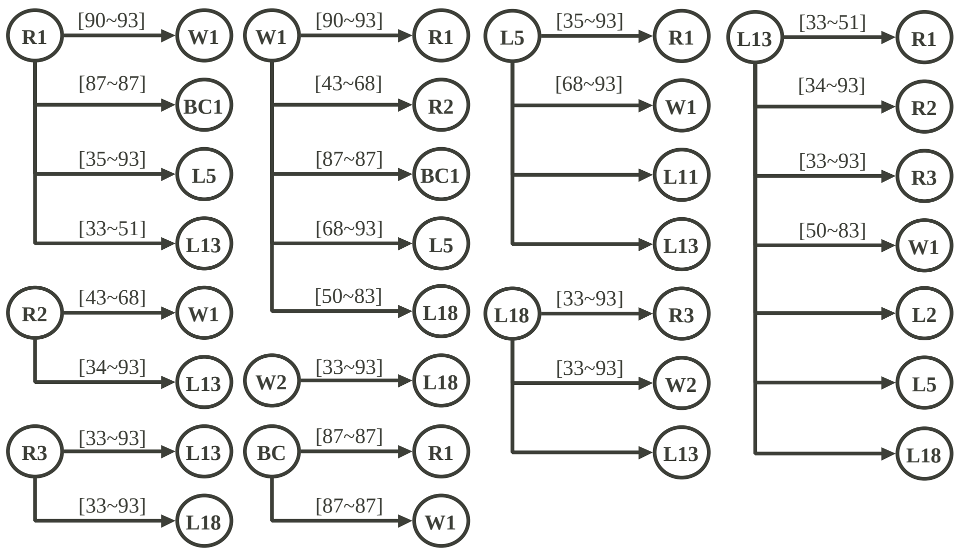


Fig. 6. Part of transitions between nodes from the sequence shown in Fig. 4.

ball reappears. In this candidate generation process, we consider the temporal consistency of transitions. That is, the earliest frame of the transition that gets into a node is expected to be earlier than the latest time of the transition that gets out of the node. We also use the following rules to avoid generation of unrealistic transitions.

(1) One player node can appear only once in a route. Thereby, the system avoids ball movements for which the ball moves back and forth between the same players.
(2) The maximum number of successive line nodes is two. In addition, the shapes of the two successive lines are expected to match a possible ball movement physically. In Fig. 4(b), for example, the transition R1→L13→L18→W2 is possible because lines L13 and L18 can be approximated using a parabola in the image and the ball movement can sometimes be parabolic.
(3) The maximum number of successive stand nodes is two. Because the

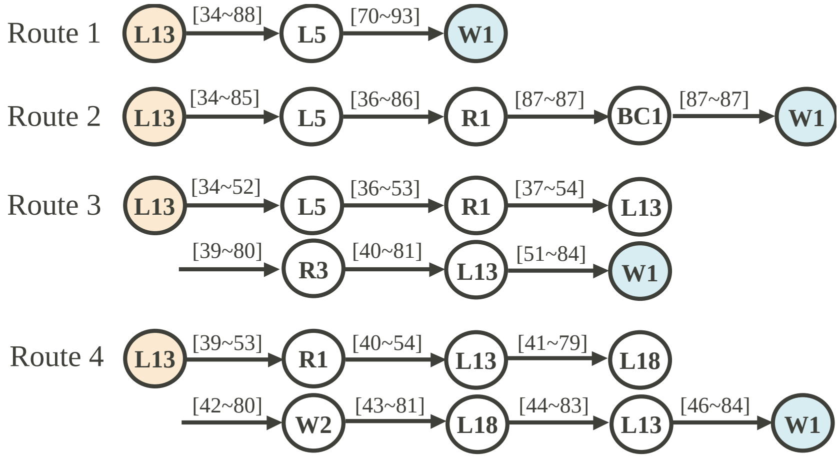


Fig. 7. Some generated ball route candidates from Fig. 6.

stand regions are divided into four subregions, as shown in Fig. 5, it is possible that the ball passes two adjacent stand regions in the image.

Figure 7 depicts some generated ball route candidates from Fig. 6.

4.3 Generate and Evaluate Rough Ball Trajectories

It is difficult to ensure that the ball is reliably detected when it is overlapping with one of the other elements described above. Nevertheless, we would like some evidence to be used to rank ball route candidates. We therefore search for ball-like regions around each route candidate and assume that the more such regions exist, the more probable the route candidate is. We also consider another condition in which the ball-like regions are expected to satisfy the constraints on ball movement by seeing if a feasible ball trajectory can be generated using the extracted ball-like regions. Detailed steps for ranking are explained in the following subsections.

4.3.1 Extraction of Ball-Like Regions

We search the region determined by each ball route candidate for ball-like regions using a separability filter [9]. A separability filter responds to concentric circular patterns as in Fig. 8. Such a filter is often used for detecting eyes in face recognition. It outputs a separability value η defined as:

$$\eta = \begin{cases} \eta' & : \overline{I_1} \geq \overline{I_2} \\ -\eta' & : otherwise \end{cases} \tag{2}$$

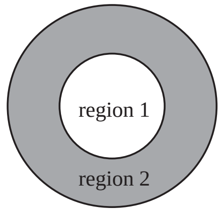


Fig. 8. Separability filter.

$$\eta' = \frac{n_1(\overline{I_1} - \overline{I_m})^2 + n_2(\overline{I_2} - \overline{I_m})^2}{\sum_{i=1}^{N}(I_i - \overline{I_m})^2}, \tag{3}$$

where n_1 and n_2 are the quantities of pixels in region 1 and region 2, respectively; also, $N = n_1 + n_2$; I_i is the brightness of pixel i; $\overline{I_1}$, $\overline{I_2}$, and $\overline{I_m}$ are the averaged brightness of region 1, region 2, and the whole region, respectively. We adaptively change the filter size depending on the estimated zoom value and resolution of the camera.

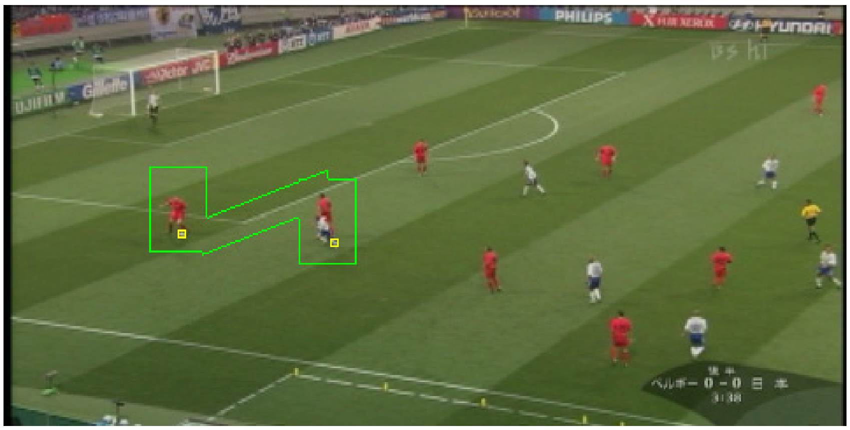
The regions that have higher response to the separability filter than some threshold are extracted. Because regions of players' white shirts and white socks might output high responses, we remove shirt regions using the result of player tracking. We also remove the socks regions by examining their shape; if the ratio of the longer principal axis to the shorter one of a region is larger than some threshold, the region is considered a socks region. Figure 9(a) depicts the filter output after the removal of shirts and shorts regions. Figure 9(b) shows ball-like regions inside a part of the search region. Because several white regions other than the ball region might remain as shown in Fig. 9(b), we use motion continuity to filter out such regions, as described below.

4.3.2 Generation of Rough Ball Trajectories

We first generate sequences of ball-like regions (called *segments*). We perform a simple clustering of the regions; if two regions are within a certain distance *in space and time*, they are put into a cluster. Clusters with fewer than three regions are deleted. We then fit a line to each cluster to generate a segment. Figure 10 presents a result of clustering and segment generation for Route 3 in Fig. 7. Triangles are extracted for ball-like regions and their colors indicate cluster IDs; lines indicate generated segments. The sum of the outputs of the separability filter for the regions in a segment is called the *score* of the segment; it is used for ranking the ball route candidates.

Using the segments, we generate a set of rough ball trajectories during the overlapping period. We first enumerate all possible combinations of segments such that no more than one segment exists at a time. We then attempt to generate a trajectory for each combination. A trajectory comprises segments and additional straight lines for connecting the segments, and is expected to

(a) filter output



(b) extraction result

Fig. 9. Extraction of ball-like regions.

Table 1
Scores of routes in Fig. 7.

ball route candidate	score	sum of filter outputs	trajectory length
Route 1	0.00	0	75
Route 2	22.24	3403	153
Route 3	4.00	1204	301
Route 4	4.12	1832	445

satisfy the following two conditions:

(1) A trajectory is expected to pass all nodes of the ball route candidate under consideration and include at least one segment.
(2) The added lines are expected to satisfy the constraint of the maximum ball speed.

Figure 11 shows two rough ball trajectories generated from the sequence shown in Fig. 4 for Routes 2 and 3 in Fig. 7. Each trajectory is generated so that

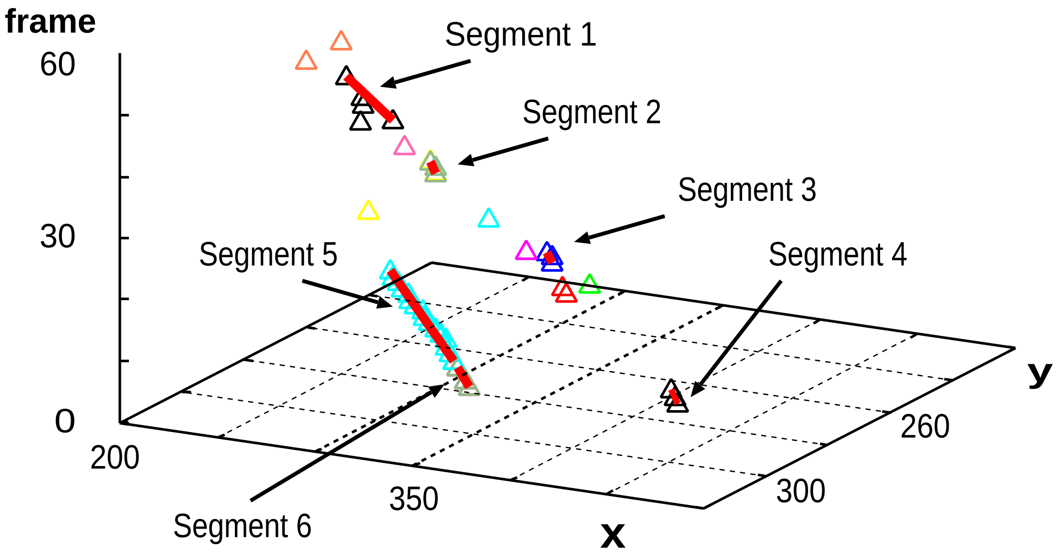


Fig. 10. Clustering of ball-like regions and segment generation.

it passes the nodes (players, lines, and ball candidates) in the corresponding routes. Red bold lines and green ones respectively represent the segments (i.e., sequences of extracted ball-like regions) and the added lines. Players and lines in space-time are also shown. The generated segments are different for the two cases because the corresponding route candidates are different; the different areas in the image are thus searched for ball-like regions. Figure 12 illustrates how the ball moves in the mosaicked image for Routes 2 and 3.

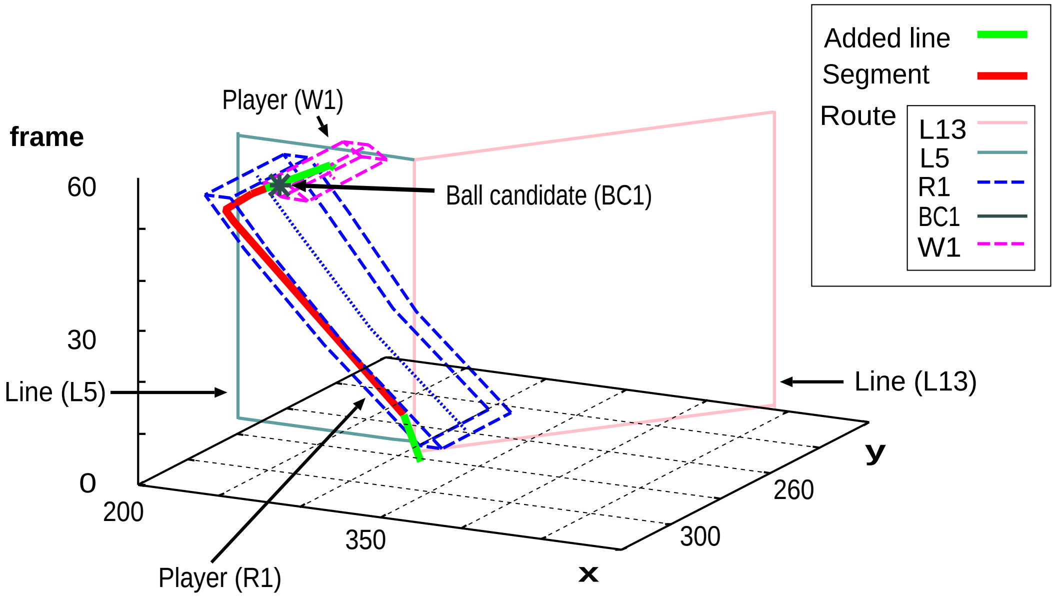
4.3.3 Selection of Most Probable Ball Route

The score of a feasible trajectory is the sum of the scores of its segments normalized by the total length of the trajectory. For each ball route candidate, we select the trajectory among its possible feasible trajectories which has the highest score; this score is the score of the ball route candidate. We finally select the ball route with the highest score. Table 1 summarizes the scores of the four routes in Fig. 7; Route 2 (L13→L5→R1→BC1→W1) is finally selected.

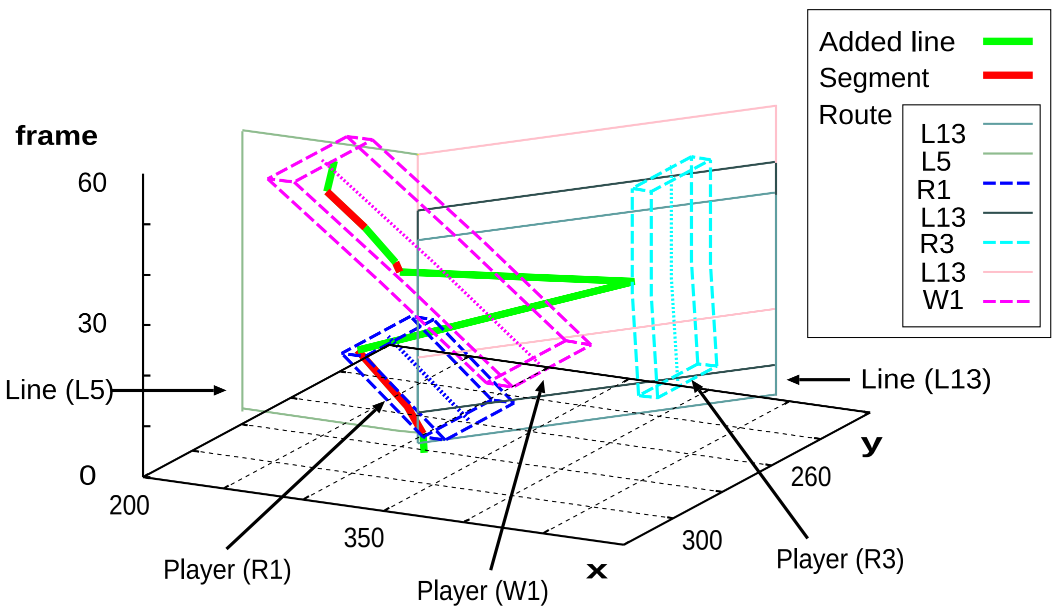
4.4 Results for Other Sequences

Figure 13 shows another sequence for which the proposed method can estimate the ball route correctly. Blue arrows indicate the nodes which are estimated to have the ball at each frame. The estimated route includes three red players and one white player.

We also tested the method for a longer soccer video of about nine and a half



(a) Trajectory for Route 2 in Fig. 7.

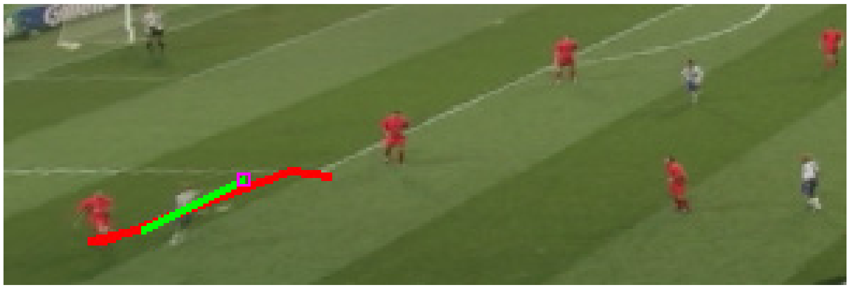


(b) Trajectory for Route 3 in Fig. 7.

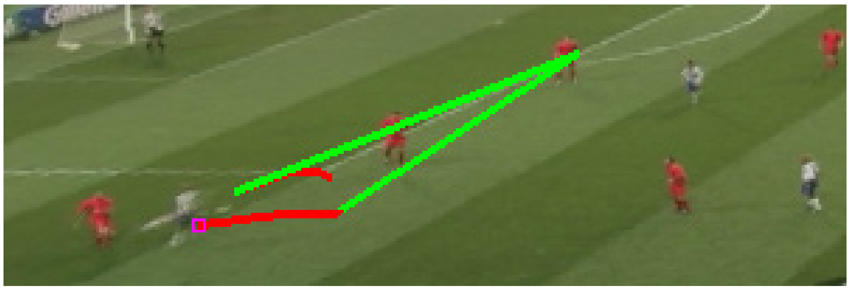
Fig. 11. Examples of generated trajectories.

minutes. The number of frames in the video is 17,279. Our shot extraction method correctly detected shots from the center camera; the total number of frames of the shots is 8,469 (49%).

As described in Sec. 3.2, if we do not find ball candidates for five consecutive frames using a simple detection method, we consider that the ball has disappeared. We manually selected five shots, which include at least one longer (i.e., more than one second) sequence of disappearance for testing. In all, the



(a) Ball movement for Fig. 11(a).



(b) Ball movement for Fig. 11(b).

Fig. 12. Ball movements projected onto the mosaicked image.

Table 2
Breakdowns of success and failure cases.

	# of frames				# of nodes			# of segments		
	< 10	$10 - -30$	$31 - -60$	$60 >$	< 4	$4 - -10$	$10 >$	< 4	$4 - -10$	$10 >$
success	11	10	6	1	16	10	2	22	4	2
failure	0	3	1	2	2	2	2	4	1	1

frames of the selected shots are 2,289; 34 such sequences are included in the shots. We examined the outputs of the proposed method for these sequences and found that the estimated ball route is correct for 28 sequences. Breakdowns of the success and the failure cases in terms of the number of frames, of nodes in the transition graph, and of generated segments are presented in Table 2.

As the table shows, the method tends to fail in longer sequences; however, in terms of the quantities of nodes and segments, which are roughly related to the complexity of situation, there is no such strong tendency. We further analyzed the failure cases and found that these are mostly attributable to the failure in detecting ball-like regions, especially when a ball goes off the field into the stands immediately after a player kicks or heads it and comes back after a long period. Detection of ball-like regions must be improved to cope with such cases.

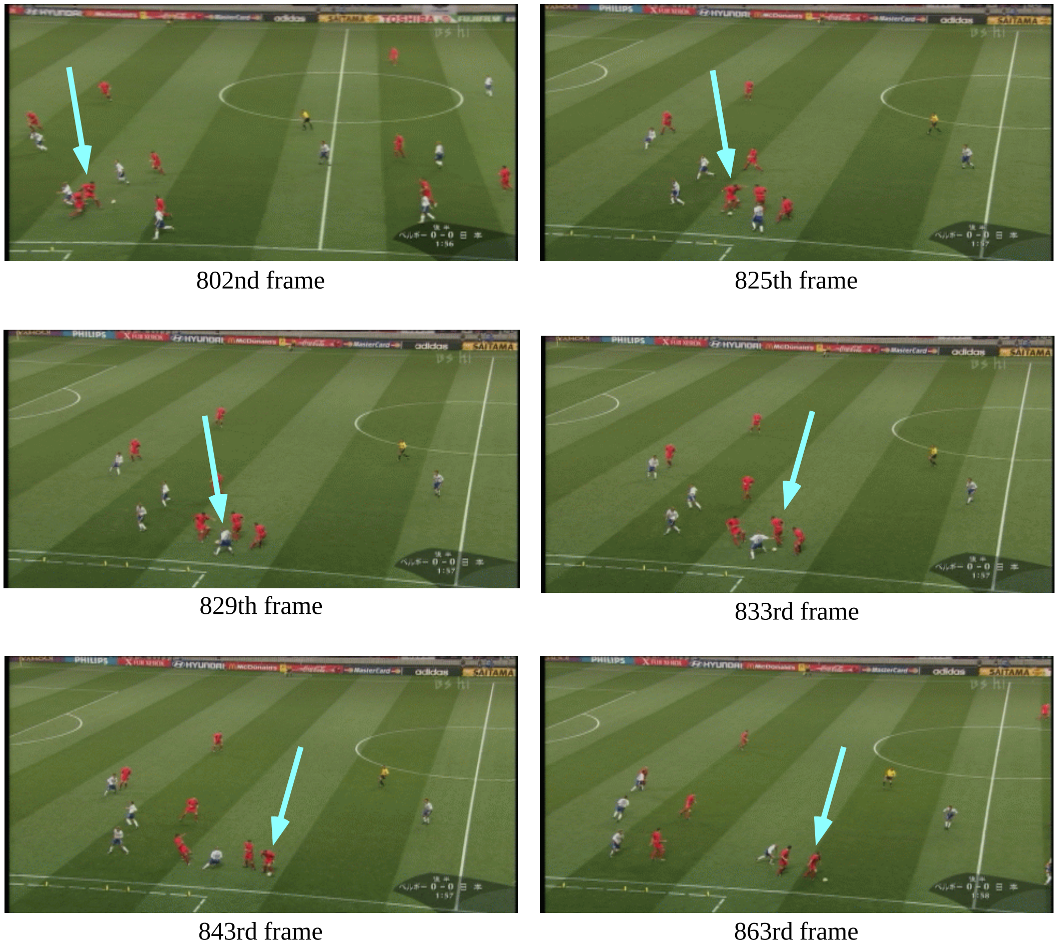


Fig. 13. Six still images from another sequence, showing the estimated ball route. Arrows indicate the node (i.e., player, in this case) estimated to have the ball at each frame.

5 Conclusions and Discussion

This paper has presented a method of estimating a ball route in a soccer broadcast video when a ball continuously overlaps with players and lines. We first generate a transition graph representing possible transitions of the ball between overlapping objects, based on their spatio-temporal relationships. We then enumerate ball route candidates from the graph and select the best one by searching for evidence for ball existence near each route candidate. Using this two-stage approach, we can greatly reduce the region that must be examined in the image. The method exhibits good performance for a set of difficult scenes. The proposed approach is potentially applicable to other broadcast sports videos such as American football and ice hockey, where a ball or a puck moves rapidly with frequent overlaps with (or occlusions by) players and other objects.

The current method estimates a ball route in a two-dimensional image coordinate system; when a ball passes a player, for example, the method does not tell whether or not the player actually touches the ball. To make such a judgment, we require additional inference related to the actual three-dimensional trajectory of the ball and the positions of the players on the ground. This inference is left as a subject for future work. Another element of future work is to apply the method, with necessary improvements, to shots other than those from the center camera. This additional feature is necessary for developing a scene retrieval and summarization system.

References

[1] N. Ancona, G. Cicirelli, E. Stella, and A. Distante. Ball Detection in Static Images with Support Vector Machines for Classification. *Image and Vision Computing*, Vol. 21, pp. 675–692, 2003.

[2] 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, pp. 393–408, 2004.

[3] X. Yu, C. Xu, Q. Tian, and H. W. Leong. A Ball Tracking Framework for Broadcast Soccer Video. *Proc. of Int. Conf. on Multimedia and Expo*, Vol. 2, pp. 273–276, 2003.

[4] K. Choi and Y. Seo. Probabilistic Tracking of the Soccer Ball. *Proc. of ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, pp. 50–60, 2004.

[5] Y. Ohno, J. Miura, and Y. Shirai. Tracking Players and Estimation of the 3D Position of a Ball in Soccer Games. *Proc. 15th Int. Conf. on Pattern Recognition*, pp. 145-148, 2000.

[6] J. Ren, J. Orwell, G.A. Jones, and M. Xu. A General Framework for 3D Soccer Ball Estimation and Tracking. *Proc. IEEE Int. Conf. on Image Processing*, 2004.

[7] 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. *ICPR*, pp. 303–306, 2002.

[8] T. Shimawaki, J. Miura, T. Sakiyama, and Y. Shirai. Ball Route Estimation in Broadcast Soccer Video. *Proc. ECCV-2006 Workshop on Computer Vision Based Analysis in Sport Environments*, pp. 26–37, 2006.

[9] K. Fukui and O. Yamaguchi. Facial Feature Point Extraction Method Based on Combination of Shape Extraction and Pattern Matching. *Systems and Computers in Japan*, Vol. 29, No. 6, pp. 696–710, 1999.