Proc. 2008 ACM Int. Conf. on Image and Video Retrieval, pp. 505-514,
Niagara Falls, Canada, July 2008.

Tracking Players in Highly Complex Scenes in Broadcast Soccer Video Using a Constraint Satisfaction Approach

Jun Miura
Dept. of Information and Computer Sciences
Toyohashi University of Technology
Toyohashi, Aichi 441-8580, Japan
jun@ics.tut.ac.jp

Hiroyuki Kubo
Dept. of Mechanical Engineering
Osaka University
Suita, Osaka 565-0871, Japan
kubo@cv.mech.eng.osaka-u.ac.jp

ABSTRACT

This paper deals with player tracking in broadcast soccer video. In soccer games, players sometimes gather in a small area in the case of, for example, a corner kick. In such a case, due to a heavy occlusion, a simple detection-and-tracking method will certainly fail. We cope with such difficult cases using a constraint satisfaction approach. To integrate pieces of evidence at various places/frames, we construct a graph of player blobs representing possible player transitions. The view of each blob provides a constraint on the number of players in the blob. All such constraints are propagated through the graph to reduce the ambiguities in the numbers. The remaining ambiguities after the propagation is handled by a statistical approach in which a set of the most likely interpretations on the numbers is selected. Finally the players' trajectory are determined based on their smoothness. Experimental results show the effectiveness of the method.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*video analysis*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*tracking*

General Terms

Algorithms

Keywords

Player tracking, broadcast soccer video, constraint satisfaction

(a) An easy scene.

(b) A difficult scene.

Figure 1: An easy and a difficult scene.

1. INTRODUCTION

There is an increasing demand for summarizing broadcast soccer videos (or other sports videos) to make a digest 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. As viewers' interests spread and deepen, more various scenes should be recognized and detected; for example, not only goal scenes but also scenes with a specific player delivers a nice pass to a forward player. Some previous works use image and sound features directly [2, 9, 10] for recognition and summarization. To meet various and detailed requirements of viewers, however, it is essential to know the movements of players and a ball [6].

In a certain portion of a whole video, most of players exist in isolation in a video and are thus easily detected and tracked. In a cluttered scene like a corner kick and a free kick in front of the goal, however, tracking players become very difficult due to heavy occlusions between players. Fig. 1 shows examples of easy and difficult scenes.

Some previous methods deal with occlusions using filtering-based approaches based on motion continuity [5, 4, 12]. Such an approach will most probably fail in a heavy occluded scene as shown in Fig. 1(b) because several players of a team sometimes form one region, from which each player cannot be segemented, and because several occlusions between various players occur continuously without a period during which each player exists in isolation. In addition, players sometimes perform feint motions against opponent players, which violate the motion continuity assumption.

To cope with such heavily occluded scenes, a promising approach is to deliberately examine what transitions of occlusions would be possible by observing the whole sequence. We [7] took a similar approach in tracking a ball when the ball overlaps with a series of objects (players, lines, and the stands) without the period of isolation in the image. A key of this approach is to construct a graph representing possible transitions of overlaps.

Such a graph-based approach has been tackled by several previous works. Chia et al. [1] proposed a method of first constructing small-sized graphs which connect blobs of a player and then merging the graphs considered to come from the same player into a larger one using color and spatial similarities. This method may merge graphs of multiple players because it uses only local information for merging. Figueroa et al. [3] proposed to represent possible correspondence between player blobs in adjacent frames as a graph and to select the most probable correspondence using pieces of evidence obtained at various places and frames. Since this method assumes that each player is isolated at the first frame of the sequence, it is not applicable to the case where occlusions have already existed at the first frame of the shot. Sullivan and Carlsson [8] also proposed a graph-based approach. They used a wide-screen soccer video which captures the entire field with a high resolution for the entire game period. The use of the wide-screen video enables us to use strong information such as the *position* of each player, a long period of each player being isolated, and the maximum number of field players in the image (that is, ten). Such information cannot be always available in the case of broadcast soccer videos.

Our method also takes a graph-based approach with so-called *least-commitment strategy*. We gradually reduce possible interpretations of the scene using pieces of evidence which appear at various locations and frames; each piece of evidence may not be strong but if we integrate them, they will become strong. Fig. 2 shows the overview of the proposed method. We start by generating a graph, nodes and edges of which are the player blobs and their temporal correspondence, respectively. We then performs a constraint propagation through the graph. For an efficient propagation, we extract and organize subgraphs in a hierarchical way. This propagation greatly reduces the ambiguity but not completely. We then introduce the plausibility evaluation of possible assignments of players to nodes using probabilistic model of player size and sampling-based MAP-like estimation. After reducing the ambiguities as much as possible, we extract players' routes.

The rest of the paper is organized as follows. Sec. 2 briefly explains the image processing techniques used in the method. Sec. 3 describes the hierarchical structure of the graph and the constraints used. Sec. 4 describes a probabilistic player appearance model and its use as a unary constraints on each node. Sec. 5 explains the constraint propagation process and Sec. 6 explains the subsequent probabilistic sampling-based search for probable interpretations. Sec. 7 describes the route candidate generation and route extraction. Sec. 8 shows some results of our methods. Sec. 9 concludes the paper and discusses future work.

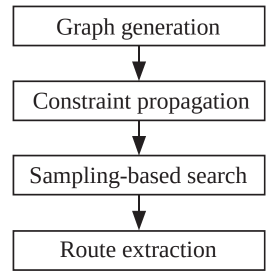


Figure 2: Overview of the proposed method.

2. IMAGE PROCESSING FOR SOCCER VIDEO ANALYSIS

This section briefly explains image processing procedures used for player tracking. See [7] for more details.

2.1 Shot detection

A soccer image sequence is divided into shots. We use a color histogram for shot change detection. If the difference between the histograms in the current and the previous frame exceeds a threshold, the current frame is considered to be the first frame of a new shot and is analyzed to determine which camera takes the shot. In this analysis, the size of the ground-colored (green) blob, the size of the players, and the number of horizontal lines are used.

2.2 Camera parameter estimation in a shot

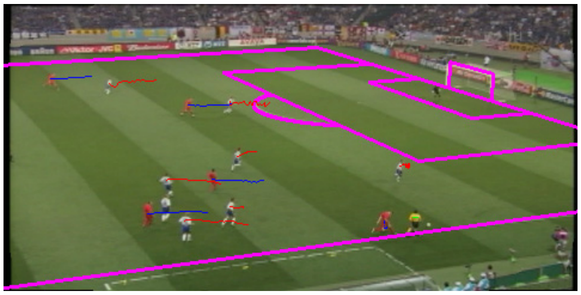
Several cameras are used in broadcasting soccer video. We estimate the camera positions once in advance by manually matching the lines and the frame of the goal in the image and those of a model of the field. Each camera has the other there parameters (pan, tilt, and zoom). We estimate the parameters at every frame since they usually change from frame to frame.

The initial values of the 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.

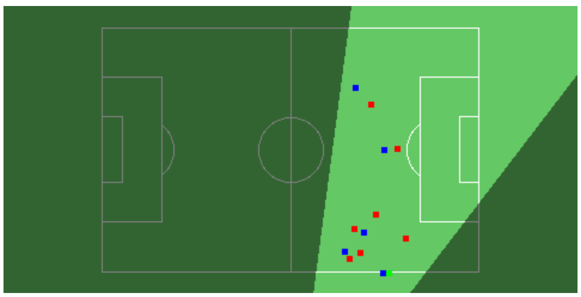
Once the initial parameters are determined, the parameters are continuously estimated in subsequent frames using a local search-based method [11]. Fig. 3 shows a result of the camera parameter estimation. Fig. 3(a) indicates the result of projecting model lines onto the ground using the estimated parameters. In Fig. 3(b), the bright area corresponds to the field of view of the camera, and red and blue points indicate the positions of players.

2.3 Player blob detection

We detect player blobs using colors of shirts and pants for each team. Fig. 4 shows an example of player blob detection. Fig. 4(a) is an input image from World Cup 2002, Japan vs. Belgium and Fig. 4(b) shows the detected blobs; the blue, red, and white blobs indicate Japan (uniform is blue and white), Belgium (red) players, and a goalkeeper, respectively.



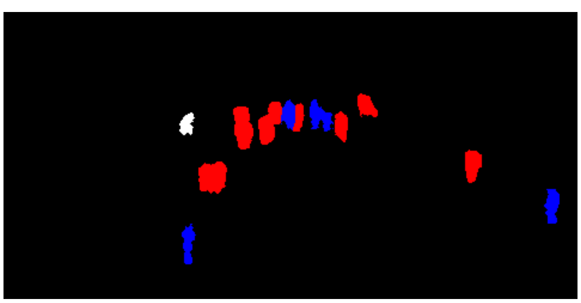
(a) projected lines.



(b) Field of view and player positions.

Figure 3: Camera parameter estimation.

(a) An input image.



(b) Extracted player blobs.

Figure 4: Extraction of player blobs using color.

3. PLAYER TRANSITION GRAPH AND CONSTRAINTS

Integration of many pieces of evidence, which arise at various places in the image and at various frames, is necessary for recognizing a difficult scene. For such an integration, we use a graph representation called *player transition graph*. A node of the graph is a player blob and an edge is a possible transition between player blobs at two consecutive frames. We consider the transition is possible between the player blobs if they overlap with each other in the image coordinates.

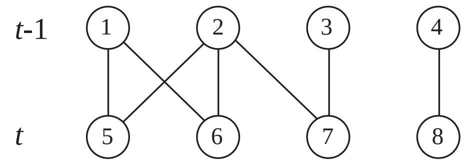


Figure 5: An example player transition graph.

Each node has, as the attribute, a pair of the minimum and the maximum number of players included in the corresponding blob. We call the attribute the *range of the number of players* or *RNP*. Each RNP is represented in the form of (min, max). We propagate the RNP information of all nodes through the graph for obtaining tighter RNP's.

In a highly complex scene with many occlusions, a player (or players) of one team may exist behind a player (or players) of the other team. We therefore prepare two RNP's for each blob. The team corresponding to the color of a blob is called the *front team* of the blob and the other the *behind team*. A graphs is generated for each team and processed independently.

In one shot, players sometimes forms mutually-independent clusters, each of which can be analyzed separately. This corresponds to the case where the entire graph can be divided into several connected graphs. In addition, if we focus on a short period among the whole sequence of the shot, we can find smaller independent subgraphs. Considering such subgraphs makes it efficient to propagate RNP's. For finding subgraphs, we use a prominent feature of the graph that there are no edges between nodes at a frame.

3.1 Constraints to be used

Two classes of constraints are used. One class is for unary constraints on nodes obtained from the size and the shape of each player blob in the image *at each frame*. The other is for n-ary constraints between nodes connected by edges.

We use the following three types of n-ary constraints in the RNP propagation: *equality constraint*, *inequality constraint*, and *summation constraint*. The first two constraints come from the fact that no player disappear or appear during the transition. The last one comes from the fact that the total number of players of a team is limited (or constant) at each frame. We explain these constraints using Fig. 5. Let n_i be the number of players of node-i. Then, the following constraints exist:

- Equality constraints: $n_1 + n_2 + n_3 = n_5 + n_6 + n_7$, $n_4 = n_8$.
- Inequality constraints: $n_1 \leq n_5+n_6$, $n_2 \leq n_5+n_6+n_7$, $n_3 \leq n_7$, $n_5 \leq n_1 + n_2$, $n_6 \leq n_1 + n_2$, $n_7 \leq n_2 + n_3$.
- Summation constraints: $n_1 + n_2 + n_3 + n_4 \leq N$, $n_5 + n_6 + n_7 + n_8 \leq N$. N is the total number of players of one team (i.e., 10) when the number of players in a shot is not known. If that number is known, it is used and the above inequality becomes the equality.

3.2 Hierarchy in player transition graph

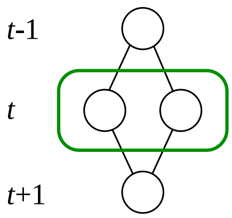


Figure 6: Related nodes.

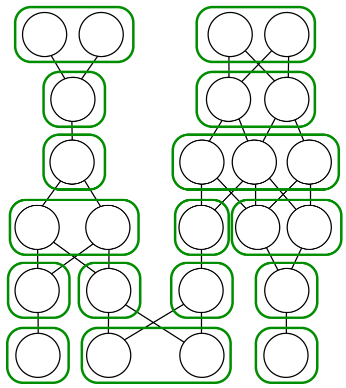


Figure 7: Basic blocks.

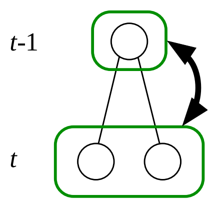


Figure 8: Basic blocks with the same number of players.

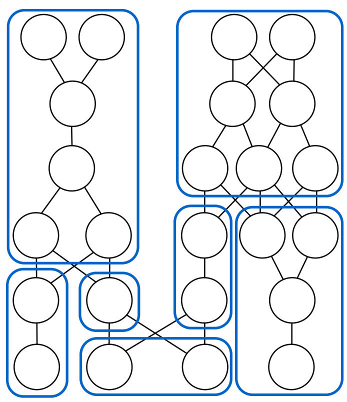


Figure 9: Maximum constant blocks.

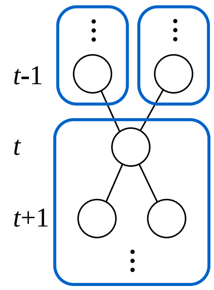


Figure 10: A set of MCB's with the equality constraint.

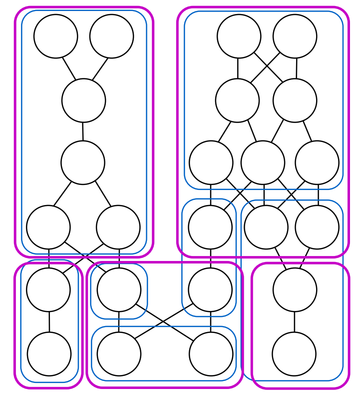


Figure 11: Maximum constant subgraphs.

We generate the following hierarchy of subgraphs. Each subgraph also has the RNP for an efficient constraint propagation.

3.2.1 Basic block

When two nodes at frame t are both connected to a node at frame $t-1$ *and* both connected to a node at frame $t+1$, such two nodes are called to be *related* (see Fig. 6). A set of related nodes at a frame forms a *basic block*. A node not having any such relations with others also forms a basic block by itself. Fig. 7 shows examples of generated basic blocks.

3.2.2 Maximum constant block

We then collect basic blocks in consecutive frames whose numbers of players are equal. This is for applying the equality constraints to a set of basic blocks. Fig. 8 shows an example pair of basic blocks satisfying this constraint. We extract locally-longest such sequences of basic blocks; each resultant collection of basic blocks is called a *maximum constant block* or *MCB*. Fig. 9 shows the generated MCB's from the basic blocks shown in Fig. 7.

3.2.3 Maximum constant subgraph

We next consider the constraints between maximum constant blocks (MCB's) having edges between them. We collect such MCB's so that the equality constraint holds within the collected set. Fig. 10 shows an example; the number of players of the lower MCB is equal to the sum of those of the upper MCB's. To use the summation constraint on such a set of MCB's, we generate such sets so that they have the same temporal boundaries. Each generated set is called a *maximum constant subgraph* or *MCS*.

There are in general multiple ways to construct MCS's for a graph. We currently use a greedy algorithm. Other algorithms can be used but they should divide all nodes into sets of MCS's exhaustively and exclusively. Fig. 11 shows the generated maximum constant subgraphs (drawn in purple) from the MCS's (in blue) shown in Fig. 9. Note that an MCB may belong to several MCS's.

3.2.4 Top-level connected graph

The top-level of the hierarchy is the whole connected graph. There may be several top-level, connected graphs for one team. Each connected graph also has an RNP and can be analyzed independently of the other connected graphs of the same team. The summation constraint based the total number of the players in the scene is applied to a set of connected graphs.

3.3 Players going out to or coming from the outside of the field of view

Some players may go out to or come from the outside of the field of view (FOV) during a shot. We deal with such cases by using a special node for players outside the FOV. This special node also contains the range of number of players, as others do.

4. PLAYER APPEARANCE MODEL AND UNARY CONSTRAINTS ON NODES

The shape and the size of a player blob provide evidence for the possible number of players of the blob. If the feet of a player are detected in a player blob, we can determine confidently that only one player exists there. In that case, we set $(1, 1)$ for the front team and $(0, 0)$ for the behind team, respectively.

Concerning the evidence from the size, a qualitative observation is that the larger the size is, the more players probably exist. We, however, need some quantitative evidence to be used for constraining RNP's. We therefore manually examined the actual numbers of players for over one thou-

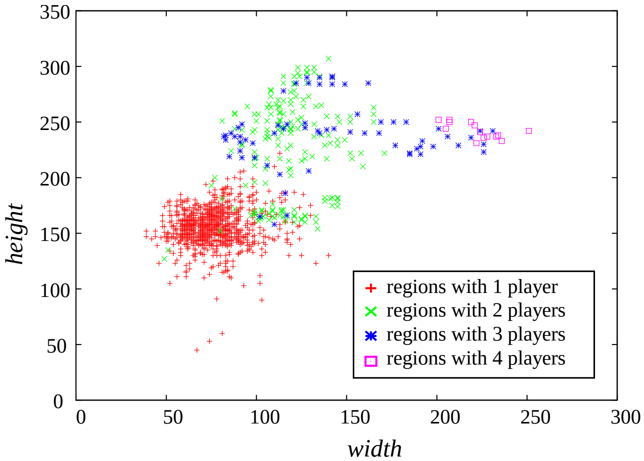


Figure 12: Distributions of width and height for four numbers of players.

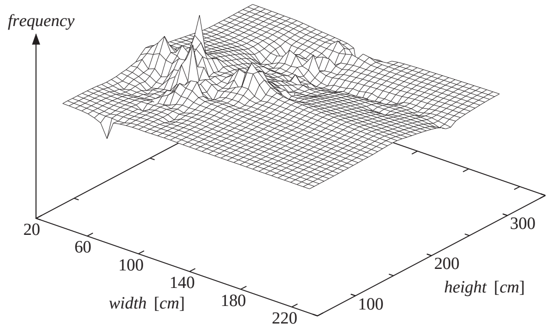


Figure 13: Distribution of width and height for the behind team with no players.

sand player blobs. The examination results are summarized in Fig. 12; four distributions in the *width-height* space corresponding to respective numbers of players are shown in different colors. We then approximated each distribution with a two-dimensional normal distribution:

$$p(\boldsymbol{s}|n) = \frac{1}{2\pi|\Sigma_n|^{1/2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{s}-\boldsymbol{\mu}_n\right)^T \Sigma_n^{-1}\left(\boldsymbol{s}-\boldsymbol{\mu}_n\right)\right\}, \quad (1)$$

where $\boldsymbol{s} = (w, h)$ is the size vector, $\boldsymbol{\mu}_n$ and Σ_n are the mean vector and the covariance matrix of the distribution for the blobs with n players, respectively.

For the behind team, whose colors do not exist in the player blob under consideration, there is a large possibility that the players of that team do not exist there. The probability of player existence for the behind team will decrease as the size of the blob decreases. We, again, need some quantitative evidence measure. We performed the above-mentioned examination for the cases where the number of players is zero for the behind team. Fig. 13 shows the obtained size distribution for zero-player blobs.

We chose the following sigmoid function as a model and fit

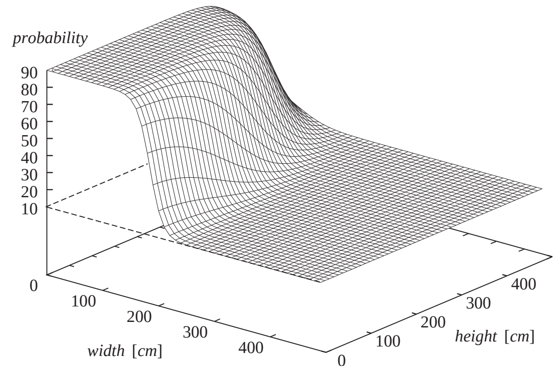


Figure 14: The approximate probability of the number of behind team players being zero.

it to the data so that the mean squared error is minimized:

$$f(x) = \frac{1}{1+e^{-a(x-b)}} \quad (2)$$

$$x = \sqrt{w^2 + (kh)^2}, \quad (3)$$

where a, b, and k are the parameters to be adjusted. The value of this function ranges in $[0.0, 1.0]$. Since the existence probability should not be very low or very high considering the uncertainty in probability estimation, we narrow the range to $100\,p_0$ percentage of the whole scale by using the following function as the probability:

$$P(x) = f(x) \cdot p_0 + \frac{1-p_0}{2}. \quad (4)$$

Currently we set $p_0 = 0.8$. Fig. 14 shows the estimated $P(x)$ from the data shown in Fig. 13.

5. CONSTRAINT PROPAGATION THROUGH THE GRAPH HIERARCHY

The first step in determining the number of players of each blob is to propagate RNP information through the entire graph to obtain tighter RNP's with adopting the equality, the inequality, and the summation constraints.

This step is composed of three parts: initialization, propagation using equality and summation constraints in the tree structure, and propagation using inequality constraints among nodes. The second and the third part are alternately executed until no change of RNP's is obtained by either of those parts. Fig. 15 shows the process of the proposed constraint propagation.

5.1 Initialize the ranges of the number of players

The initial RNP of a node is determined from the width and the height of the corresponding blob using the probabilistic distributions of eq. (1); the minimum and the maximum number of players is obtained from the distributions with some margins added. The minimum number of behind team is set to zero.

5.2 Constraint propagation using equality and summation constraints

The graph hierarchy has the tree structure; from the root to leaves, we have a top-level connected graph, maximum

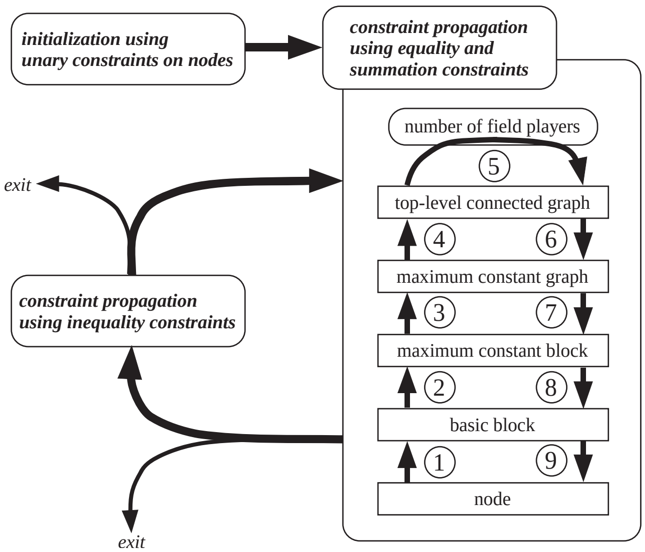


Figure 15: Outline of constraint propagation.

constant subgraphs (MCS's), maximum constant blocks (MCB's), basic blocks, and nodes. The constraint propagation starts at the nodes and goes upward to the top-level, and then it comes back downward to the nodes (see the right-hand part of Fig. 15).

We first describe the propagation between basic blocks and a maximum constant block (green marks in Fig. 7 and blue ones in Fig. 9, respectively). Let (B_{min}, B_{max}) be the RNP of an MCB and (C^t_{min}, C^t_{max}) be the RNP of the basic block at frame t included in the MCB. For the upward propagation from basic blocks, we perform the following calculations:

$$B_{max} = \min_t C^t_{max}, \tag{5}$$

$$B_{min} = \max_t C^t_{min}. \tag{6}$$

For the downward propagation from the MCB to basic blocks, we perform the following for all frames included in the MCB:

$$C^t_{max} = B_{max}, \tag{7}$$

$$C^t_{min} = B_{min}. \tag{8}$$

This two-way calculation basically applies the equality constraint but at the same time introduces information from the other parts of the graph.

We next explain the propagation between a maximum constant subgraph and maximum constant blocks (purple and blue marks in Fig. 11, respectively). Let (G_{min}, G_{max}) be the RNP of the MCS and $(B^{i,t}_{min}, B^{i,t}_{max})$ be the RNP of the ith MCB at frame t. For the upward propagation from MCB's to MCS, we perform the following calculations:

$$G_{max} = \min_t \left(\sum_i B^{i,t}_{max} \right), \tag{9}$$

$$G_{min} = \max_t \left(\sum_i B^{i,t}_{min} \right). \tag{10}$$

For the downward propagation from MCS to MCB's, we perform the following:

$$B^{i,t}_{max} = \min \left(B^{i,t}_{max}, G_{max} - \sum_{j \neq i} M^{j,t}_{min} \right), \tag{11}$$

$$B^{i,t}_{min} = \max \left(B^{i,t}_{min}, G_{min} - \sum_{j \neq i} M^{j,t}_{max} \right). \tag{12}$$

Similar propagation operations are also performed for nodes and a basic block, maximum constant subgraphs and the top-level connected graph, and between connected graphs.

5.3 Constraint propagation using inequality constraints

Constraint propagation using inequality constraints are applied to only connected nodes. Although similar constraints can also be applied to connected MCS's (see Fig. 11, for example), we do not do so because the ambiguity in the number of players will be large in upper levels of the graph hierarchy and inequality constraints are expected to be not very effective when the ambiguity is larger.

6. SAMPLING-BASED SEARCH FOR DETERMINING PLAUSIBLE NUMBERS OF PLAYERS

The constraint propagation explained in the previous section is the propagation of *hard* constraints; that is, the necessary conditions are checked which should certainly be satisfied. It is, however, difficult to completely determine the numbers of players of the nodes at that stage and, therefore, ambiguities on the numbers still remain. To cope with such remaining ambiguities and to obtain a plausible interpretation (i.e., a combination of the numbers of players of all nodes), we take a probabilistic approach.

We have constructed the probabilistic distributions of blob size for each specific number of players. Using this knowledge, we would like to obtain the maximum likelihood interpretation. Since the remaining interpretations satisfy all the constrains (equality, inequality, and summation), the likelihood of an interpretation is calculated as the product of the likelihood values of the number of players of every ambiguous node in the interpretation. Calculating the maximum likelihood interpretation is, however, very hard because the set of possible interpretation is huge. We therefore use a sampling-based method. In addition, to avoid the case where a not-so-good interpretation happens to gain the maximum likelihood, we keep a certain number of good interpretations and merge them to determine the RNP's of all nodes.

6.1 Normalized distribution of numbers of players

In the sampling-based method, for an *ambiguous node* whose RNP has not collapsed into a single number (i.e., the minimum number is equal to the maximum one) yet, we select one number in the RNP probabilistically using the *normalized* probabilistic distribution of the numbers.

The normalized distribution for a node is calculated by normalizing the set of likelihood values obtained for the remaining numbers in the RNP and the probability values for the size of the node. Whenever the RNP is reduced, this normalized distribution is updated. Fig. 16 shows a schematic view of calculating the normalized distribution.

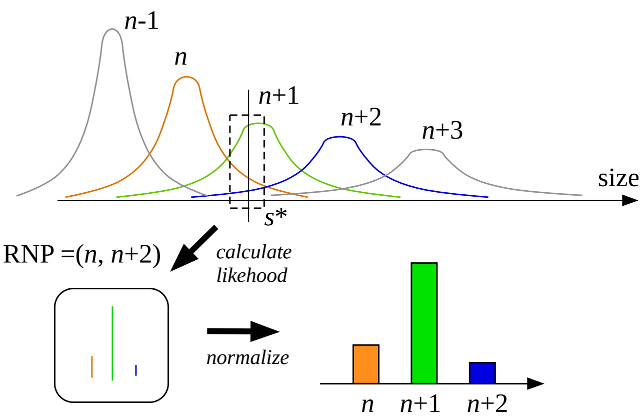


Figure 16: Calculation of normalized distribution. Size s is actually a pair of the width and the height.

6.2 Algorithm

We use the following sampling-based algorithm for obtaining a set of plausible interpretation.

1. Repeat the following hypothesis generation and evaluation for N times.
   (a) (hypothesis generation) Repeat the following steps until all ambiguities are resolved.
      i. Select an ambiguous node at random and select one number in the current RNP based on the normalized probabilistic distribution.
      ii. After fixing the number of players of that node, we perform the constraint propagation described in Sec. 5.
   (b) (evaluation) Calculate the likelihood of the hypothesis.
2. Reduce all RNP's using the top n hypotheses; that is, keep only the numbers of players which match with at least one of the hypotheses.

Currently we use $N = 1000$ and $n = 50$.

6.3 Using hypotheses on the total number of players

The constraint on the total number of players in the image is strong for resolving the ambiguities of RNP's. It is not, however, always the case where all players are in a shot and the constraint that the total number of field players is ten can be applied. We therefore hypothesize the number of players in a shot in several ways, and use it for the sampling-based search. We sample data for possible numbers of the total number in the image and select the best interpretation. Fig. 17 shows the samples for a connected graph of Japan team; the horizontal axis indicates the log-likelihood of an interpretation and the vertical one indicates the sum of the errors in the number of players of all nodes compared with the correct numbers. In this case, the samples for the case where the total number is hypothesized as six are best and the actual total number is also six. Note that inappropriate hypotheses are automatically eliminated using the constraint propagation steps if they are actually incorrect.

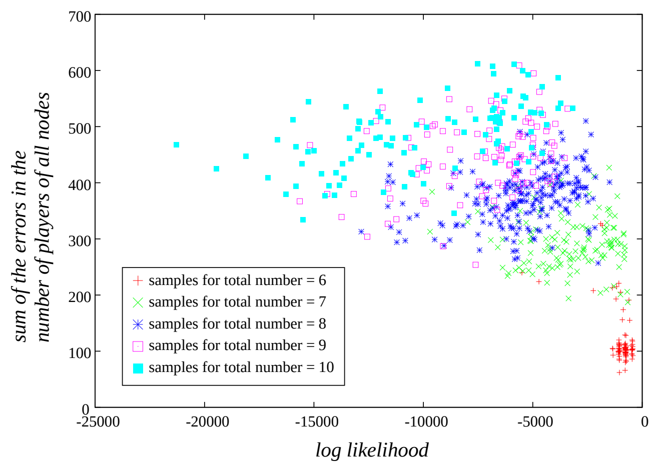


Figure 17: Sampling with hypothesized total number of players

7. EXTRACTING PLAYER ROUTES

Determining the number of players of the nodes is not equivalent to determining the routes (a sequence of nodes) of players; several combinations of player routes may be possible for a single interpretation of the numbers of players of the nodes. In addition, there usually remain multiple interpretations after the constraint propagation and the sampling-based search. The number of possible player routes, therefore, may still be large.

If we analyze a zoomed-up image sequences, players appear in large sizes in the image and, therefore, visual cues such as faces and uniform numbers may be effective in identifying players. In the case of video sequences as shown in Fig. 1, however, each player is too small to use such visual cues. We need to use other cues such as the smoothness and/or continuity of the route.

It is costly to generate the possible routes directly at the node level. So we first generate them at the maximum constant block (MCB) level. For each route at the MCB level, we then generate a route at the node level.

7.1 Generating possible combinations of player routes at MCB level

At the MCB level, a route of a player is a sequence of MCB's on which the player moves. We analyze the graph of MCB's (such as the one shown in Fig. 9) to enumerate all possible routes on that graph. For each route, we determine the RNP from the RNP's of all MCB's on the route. Next we generate combinations of the routes by enumerating possible combinations and then checking their compatibility with the RNP's of the related nodes.

7.2 Extracting a combination of player routes at node level

Given a combination of MCB-level routes, we then determine which sequence of nodes each of the MCB-level route passes. A certain portion of the node-level route can be determined without ambiguity from a MCB-level route; that is, we can determine many nodes on which the route certainly passes. We then interpolate the route segement between two such nodes. Fig. 18 schematically explains the process of

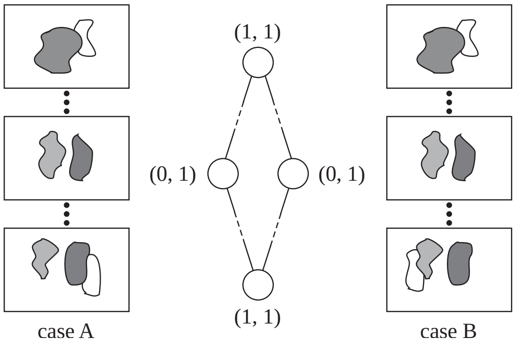
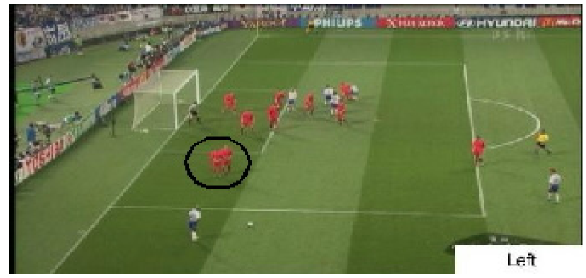
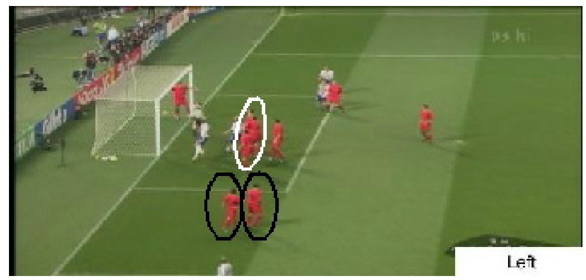


Figure 18: Route ambiguity.



(a) frame 5502.



(b) frame 5598.

Figure 19: An example scene.

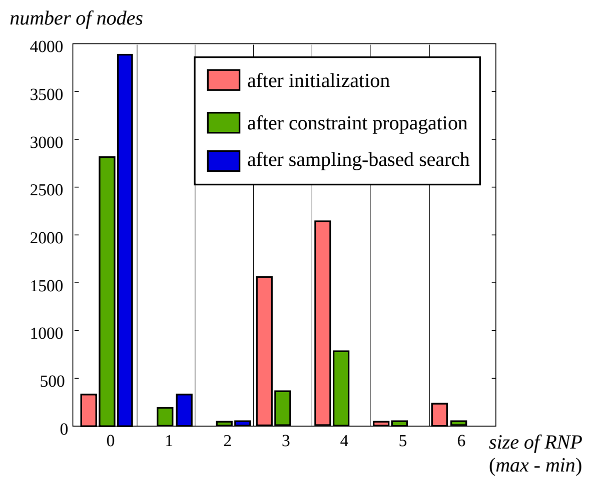


Figure 20: Ambiguity reduction by constraint propagation and sampling-based search.

interpolation.

The figure shows two cases of player movements and one graph with RNP's corresponding to both cases. There are three players, one (white) from a team and two (light and dark gray) from the other. There is an ambiguity in the middle where a white player is behind one of the two gray players in both cases. By considering the smoothness of the player movement, the left node is selected in the middle for case A and the right node for case B. This is a very simple case; it can probably be handled by the existing methods which uses the smoothness constraint (or the motion continuity). The important difference of the proposed method in the usage of smoothness constraint from the existing ones is that we adopt the constraint *after greatly reducing the possibility of routes by a constraint satisfaction approach.*

After generating combinations of player routes at the node level, we choose the best combination which maximizes the smoothness measure of the routes. For each player, the smoothness of motion is defined as the negation of the sum of the abosolute values of velocity changes calculated from the positions at every k (currently, 10) frames. The total of such smoothness values for all players is used for evaluation.

In broadcast soccer video, the camera parameters change from frame to frame. So we use the scene position (i.e., player position on the ground) for this interpolation and smoothness calculation.

8. EXPERIMENTAL RESULTS

We used two video sequences from World Cup 2002, Japan vs. Belgium. One of sequences, which has 181 frames is shown in Fig. 21

One of the important parts of the proposed method is the constraint propagation-based reduction of RNP's. Fig. 19 illustrates a typical case where such a reduction is effective. We here focus on red-colored players (Belgium). The black-marked blob at frame 5502 has the initial RNP of $(1, 4)$; the other eight blobs have the similar initial RNP's. By using the knowledge that the total number of field players is ten, the ambiguities are reduced to $(1, 2)$. This means that the information from only one frame is not sufficient. By examining the subsequent frames, however, the black-marked blob is known to break into two blobs at frame 5598, each of which has $(1, 1)$, and this leads to a conclusion that the black-marked blob at frame 5502 has $(2, 2)$ and the others have $(1, 1)$. Moreover, since we know that the numbers of players of all blobs at frame 5502 are fixed, the number of players of the white-marked blob at frame 5598 is determined as three, although this number cannot be determined only from the information at that frame.

The results explained above is a *lucky* case where the ambiguities are resolved only by the constraint propagation step. Actually, the total number of white players in this sequence is less than ten. In usual cases, each step of the method gradually reduces the ambiguities. Fig. 20 shows how the ambiguities are reduced as the constraint propagation and the sampling-based search are performed. For the abovbe sequence, the total number of nodes is 2158 with 181 frames. The ratio of the number of nodes with no ambiguity (i.e., min = max in RNP) after the initialization using unary constraints, that after the constraint propagation, and that

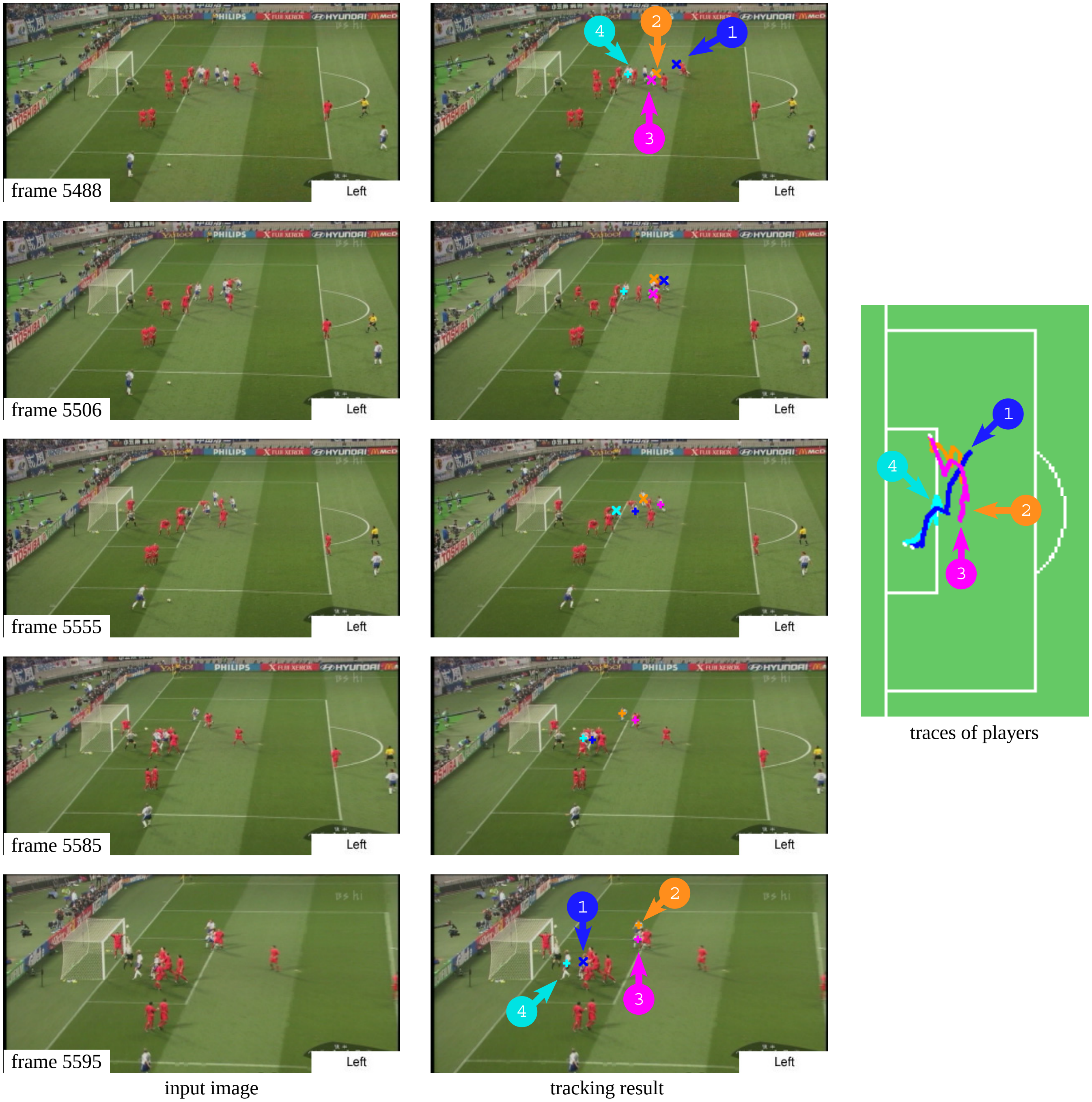


Figure 21: Snapshots of tracking four white players in a highly complex scene and their traces.

after the sampling-based search are 8.1%, 66.8%, and 90.7%, respectively. The computation times for graph generation, constraint propagation, sample-based search, and route extraction are 133.1 [*sec*], 0.05 [*sec*], 3070 [*sec*], and 125.3 [*sec*], respectively.

For the sequence shown in Fig. 21, we have 41 routes for a player and 116 combinations of routes at MCB level for Japan team (white) and 82 and 290 for Belgium (red). From these combinations, the most probable combination of player routes at node level is extracted for for each team.

The figure shows the tracking results of four white players, who forms the most complicated connect graph for their team. The traces of the players are also shown at the right. The + marks indicate that the player position is determined for the node with only one player, while the × marks indicate that the player position is determined using interpolation (or extrapolation) for the node with multiple players. Although they interact complicatedly not only with each other but also with red players, we confirmed that they are tracked correctly *at node level*. The position on the ground

Table 1: Success rates of tracking at node level.

	sequence 1		sequence 2	
	Japan	Belgium	Japan	Belgium
# of players	4	7	6	3
# of frames	181		101	
# of correctly tracked node	724	1255	604	303
success rate [%]	100	99.1	99.7	100

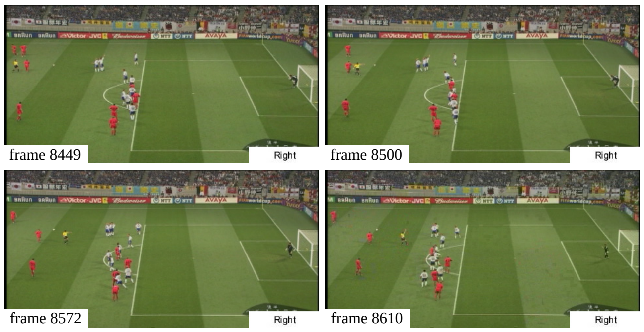


Figure 22: Not successfully tracked sequence.

is, however, sometimes deviated from the correct one due to interpolation or extrapolation.

Table 1 summarizes the success rates of tracking at node level for four connected graphs in two sequences. For each combination of sequence and team, we choose the most complicated connected graph for evaluation; the success rates for the other connected graphs are all 100 %. The result for Japan in sequence 1 corresponds to the one shown in Fig. 21. The success rates are very high for all four cases.

Fig. 22 shows another sequence for which our method failed to produce satisfactory results. This sequence is the one before Belgium team performs a free kick and the movements of most players are very scarce. As a result, for some players, there are no frames which can be used for determinig routes with interpolation. This is not, however, a serious problem because such a shots does not provide useful information for analyzing the game.

9. CONCLUSIONS AND DISCUSSION

This paper has described a novel player tracking method in a heavy occluded scene. The method first generates a graph of player blobs representing temporal transitions of players. Each node has the range of the number of players (RNP) as the attribute and the method gradually reduces the range by first performing a constraint propagation and then conducting a sampling-based search for the high-likelihood interpretations. After reducing as much ambiguity as possible, the route of each player is extracted based on the smoothness criterion. The method has been successfully applied to a very complex scene which the previous works are hard to cope with. We are now testing the method on various scenes for verification of its robustness.

The current implementation, especially the sampling-based search step, is costly. A possible way of reducing the cost is to use other kinds of knowledge on players' possible motions which could give more constraints on the possible numbers of players of the nodes. The use of other shots including zoomed-up scene and replays would also be beneficial.

The presented work is a part of our efforts to construct a versatile soccer video summarization and retrieval system. Combining the proposed method with other necessary functions such as a graph-based ball tracking [7] and shot categorization is necessary to construct an actual working system.

10. REFERENCES

[1] A.Y.S. Chia, W. Huang, and L. Li. Multiple Objects Tracking with Multiple Hypotheses Graph Representation. In *Proceedings of the 18th Int. Conf. on Pattern Recognition*, pp. 638–641, 2006.

[2] A. Ekin, A.M. Tekalp, and R. Mehrotra. Automatic Soccer Video Analysis and Summarization. *IEEE Trans. on Image Processing*, Vol. 12, No. 7, pp. 796–807, 2003.

[3] P.J. Figueroa, N.J. Leite, and R.M.L. Barros. Tracking Soccer Players Aiming Their Kinematical Motion Analysis. *Computer Vision and Image Understanding*, Vol. 101, No. 2, pp. 122–135, 2006.

[4] T. Misu, M. Naemura, W. Zheng, Y. Izumi, and K. Fukui. Robust Tracking of Soccer Players Based on Data Fusion. In *Proceedings of the 16th Int. Conf. on Pattern Recognition*, pp. 10556–10561, 2002.

[5] Y. Ohno, J. Miura, and Y. Shirai. Tracking Players and Estimation of the 3D Position of a Ball in Soccer Games. In *Proceedings of the 15th Int. Conf. of Pattern Recognition*, pp. 145–148, Barcelona, Spain, September 2000.

[6] Y. Seo, S. Choi, H. Kim, and K.-S. Hong. Where are the Ball and Players? Soccer Game Analysis with Color Based Tracking and Image Mosaick. In *Proceedings of 9th Int. Conf. on Image Analysis and Processing*, Vol. 2, pp. 196–203, 1997.

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

[8] J. Sullivan and S. Carlsson. Tracking and Labelling of Interacting Multiple Targets. In *Proceedings of 9th European Conf. on Computer Vision*, pp. 619–632, 2006.

[9] D. Tjondronegoro, Y.-P.P. Chen, and B. Pham. Integrating Highlights for More Complete Sports Video Summarization. *IEEE Multimedia*, Vol. 11, No. 4, pp. 22–37, 2004.

[10] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun. Structure Analysis of Soccer Video with Domain Knowledge and Hidden Markov Models. *Pattern Recognition Letters*, Vol. 25, No. 7, pp. 767–775, 2004.

[11] A. Yamada, Y. Shirai, and J. Miura. Tracking Players and a Ball in Video Image Sequence and Estimating Camera Parameters for 3D Interpretation of Soccer Games. In *Proceedings of the 16th Int. Conf. of Pattern Recognition*, pp. 303–306, 2002.

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