

# 3D Tracking of a Soccer Ball Using Two Synchronized Cameras

Norihiro Ishii<sup>1</sup>, Itaru Kitahara<sup>2</sup>, Yoshinari Kameda<sup>2</sup>, and Yuichi Ohta<sup>2</sup>

<sup>1</sup> University of Tsukuba, Graduate School of System and Information Engineering,  
Tennodai 1-1-1, Tsukuba-shi, Ibaraki, 305-8577, Japan

ishii@image.esys.tsukuba.ac.jp,

WWW home page: <http://www.image.tsukuba.ac.jp/>

<sup>2</sup> {kitahara,kameda,ohita}@iit.tsukuba.ac.jp

**Abstract.** We propose an adaptive method that can estimate 3D position of a soccer ball by using two viewpoint videos. The 3D position of a ball is essential to realize a 3D free viewpoint browsing system and to analyze of soccer games. At an image processing step, our method detects the ball by selecting the best algorithm based on the ball states so as to minimize the chance to miss the ball and to reduce the computation cost. The 3D position of the ball is then estimated by the estimated 2D positions of the two camera images. When it is impossible to obtain the 3D position due to the loss of the ball in an image, we utilize the Kalman Filter to compensate the missing position information and predict the 3D ball position. We implemented a preliminary system and succeeded in tracking the ball in 3D at almost on-line speed.

## 1 Introduction

Live broadcasting of sports is considered as one of the promising applications in multimedia literature. Among the various kinds of sports, soccer is one of the most popular sports and it is worth broadcasting and making innovative multimedia content for that. While traditional live broadcasting system is designed to send a video stream that is just taken by cameras, new style of browsing the sports, live 3D video browsing, is going to be in reality. In the 3D video browsing, audience can virtually fly over the soccer field and see the game from any preferred viewpoint.

*Kanade et al.* have developed a pioneering system for 3D sport browsing, named Eye Vision System[1]. It can produce a video stream in which the camera virtually revolves around a player in focus though viewers cannot choose the player to see in the system. Some of the research works succeeded in providing a true 3D browsing system for games held in a soccer stadium[2][3].

The 3D position of the soccer ball is very important to enjoy watching soccer games, especially in 3D browsing system. This is because audience usually watches the soccer games by chasing the ball. In addition, its 3D position information can contribute to analyze sport scenes and to improve strategies of soccer team.

However, the 3D position estimation of a ball is not an easy task because it is usually very small in an image and it could be easily lost due to the occlusion caused by players. *D’Orazio et al.* have detected the ball by using the Circle Hough Transform[4]. *Shimawaki et al.* have extracted the candidate balls by using the size and color attributes and then selected the one as the true ball over some frames[5]. *Ren et al.* have tracked a soccer ball using the Kalman Filter and succeeded in obtaining the ball trajectory in a single camera image[6]. But, these methods cannot estimate 3D position of a ball. *Yan et al.* and *Misu et al.* have proposed a ball tracking algorithm by particle filter[7][8]. *Choi et al.* and *Saito et al.* have used the template matching to detect a ball[9][10]. Unfortunately, these methods need much calculation cost to realize the real time detection. *Iizuka et al.* have developed the real time 3D position measurement system[11]. But it needs the large object in a image. In this paper, we propose an efficient ball tracking method that can estimate the 3D position of the soccer ball at almost on-line speed. We use two video images captured simultaneously at two stable cameras. In order to reach the real-time processing speed, the search area of a ball in a video image is narrowed according to ball states and an appropriate image processing method is selected to detect the ball. When the 3D position cannot be estimated because of the lack of the two-view information of the ball, we exploit the Kalman Filter to compensate the missing information and predict the 3D position of the ball.

The rest of the paper is organized as follows. In section 2, we explain our ball detection method at an image processing stage. Then, we discuss the algorithm to narrow the search area of the ball that can contribute to reduce the computation cost in section 3. We exploit the Kalman Filter to estimate the 3D position of the ball even in the case where one of the cameras misses the ball in section 4. We conducted an experiment and showed the efficiency of our proposed method in section 5. Finally, we conclude our paper and discuss future works in section 6.

## 2 Ball Detection in an Image

Ball detection on a video image is essential to estimate the 3D position of a ball. In this section, we propose a method that can extract a ball region effectively for the most of the time of a soccer game.

### 2.1 Detection of Moving Objects

In soccer games, a ball is moving for the most of the time. Therefore, we first extract moving objects by frame subtraction operation. Pixel-wise subtraction of two successive images can extract moving object regions. However, the pairwise subtraction of two successive frames ( $k - 1$  and  $k$ ) extracts not only moving objects in a current frame but also the one in the previous frame. Hence, we exploit the result of the next frame pair to eliminate the moving object region that belongs to the previous frame. Only the moving object region at the current

frame  $k$  can survive when we conduct logical AND operation for the subtraction frames of  $I_{k-1,k}$  and  $I_{k,k+1}$ .

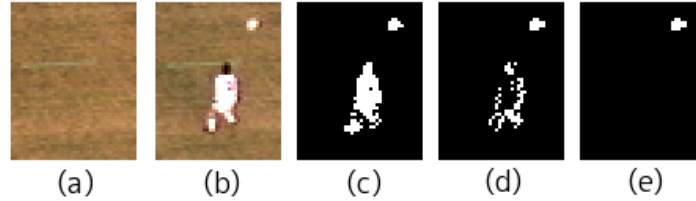
## 2.2 Elimination of Player Regions

The motion object regions detected by the method described in section 2.1 includes both the ball and players. Therefore, player regions must be removed from the regions so as to extract the ball region.

One of the most significant differences between a ball and a player on an image is their size. Therefore, we can eliminate the player regions by setting the threshold of area size appropriately. Unfortunately, however, the motion object regions may be split into small pieces because of the subtraction method. Therefore, we exploit background subtraction method. The background subtraction method can extract relatively large segments and we can obtain player segments by selecting the segments that are larger than a threshold value. Then, the motion object regions that overlap the player segments are eliminated because they are considered to be a part of player regions.

Note that the background subtraction method is useful only when we extract relatively large objects because wide range of intensity threshold can be accepted for that. Since the ball region is usually very small, it is hard to set the intensity threshold appropriately to preserve the ball region in the extracted foreground segments. Our hybrid method can extract the ball regions stably during the soccer game.

Fig.2 shows the image processing steps to extract ball regions. Fig.1(a) is a background image, (b) is an input image, and (c) is the foreground segments. (d) shows the motion object regions obtained by the method of section 2.1. By referring (c), we can remove player regions and obtain ball-like regions shown in (e). We call the extracted regions ball candidates.



**Fig. 1.** Extraction of ball region

## 2.3 Ball Detection

Since some of the ball candidates are false ball regions due to noise and/or small objects that can be found around the soccer field, we need one more step

to remove those false candidates. We count for the speed of the soccer ball to remove false candidates.

Suppose the maximum speed of soccer ball is 100km/h, the diameter of the ball is 22cm, and the frame rate of our video system is 30fps. Then, the ball goes for 90cm at one frame interval. That means the ball movement for one frame is within the several times of the apparent size of the ball in an image. We denote the maximum distance on an image as  $R$ .

The elimination algorithm is as follows. Suppose the current frame is  $k$ . First, the ball candidates of frame  $k$  that have at least one ball candidate of frame  $k-1$  within the maximum distance  $R$  are selected. If more than one ball candidates are selected, we then examine whether the selected ball candidates of frame  $k-1$  have at least one corresponding ball candidates at frame  $k-2$  or not within  $R$ . We repeat this process until only one ball candidate remains. The remained ball candidate is considered to be the true ball region.

### 3 Selection of Ball Extraction Methods

The disadvantage of the ball detection method described in section 2 is the computation cost because it requires both the frame subtraction process and background subtraction process. It also has the other disadvantage that it works well only when the ball is moving though the ball is moving for the most of the time in a game.

In order to diminish these two disadvantages, we propose to reduce the search area as much as possible and to exploit a template matching method if the ball is considered to move very slowly or stops. The selection algorithm of the ball extraction methods is shown in Fig.2. Suppose the current frame is  $k$ . If the ball is found at  $m_{k-1} = (u_{k-1}, v_{k-1})$  at frame  $k-1$ , it means the ball should be observed within the distance  $R$  from  $m_{k-1}$ . We choose the proposed method (A) if the ball is moving, otherwise we use a template matching method (T).

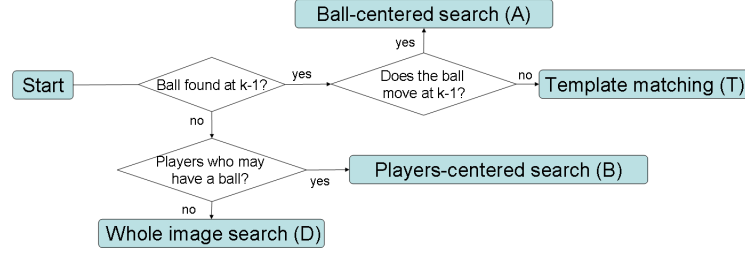
If the ball has been already lost at frame  $k-1$ , we find all the possible players who may hold the ball and set the search area around them (B). If there are no players around the location where the ball is observed for the last time, we conduct the whole image search (C) though it consumes computation cost.

#### (A) Ball-Centered Search

When the ball is detected in frame  $k-1$ , the search area in frame  $k$  can be narrowed within a circle the center of which is  $m_{k-1}$  and its radius is  $R$ .

#### (T) Template Matching

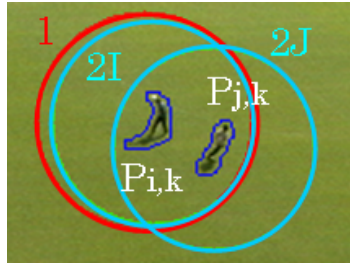
When the ball speed is considered to be very slow, the method proposed in section 2 may not succeed in finding the ball region because the frame subtraction may not extract the ball region as motion object regions. Therefore, in this case, the template matching is employed in the same search area of (A). The ball region extracted at frame  $k-1$  is used as the template.



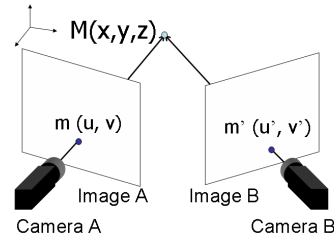
**Fig. 2.** Selection of Ball Extraction Methods

### (B) Players-Centered Search

When the ball is detected in frame  $k - 2$  (or before) and is lost in frame  $k - 1$ , we can say that one of the players (at  $p_{s,k-1}$ ) who were close to  $m_{k-2}$  should have the ball in frame  $k - 1$ , and then one of the players (at  $p_{t,k}$ ) who are close to one of  $p_{s,k-1}$  probably hides or keeps the ball. In this case, the search area in frame  $k$  can be narrowed to the area of multiple circles. The center of each circle is the location of a player who stands within the distance  $R$  from  $p_{s,k-1}$  and the radius of the circle is  $R$ . As there may be several players within  $R$  from  $p_{s,k-1}$ , the search area should include all the corresponding circles. Fig.3 shows the search area of players-centered search in frame  $k$ . Circle 1 is defined by the player (not shown in the image) who stood within the distance  $R$  from  $m_{k-2}$ . The center of the circle 1 can be denoted as  $p_{s,k-1}$ . Then the search area in frame  $k$  is the circles 2I and 2J that are defined by the players  $P_{i,k}$  and  $P_{j,k}$  because they are found within circle 1.



**Fig. 3.** Players-Centered Search



**Fig. 4.** Estimation of ball position in 3D

### (C) Whole Image Search

When there are no players around the location where the ball is observed for the last time, it means the ball has disappeared by some reasons. Hence the search area is the whole image in this case.

## 4 3D Tracking by the Kalman Filter

When the ball positions for the both cameras are estimated, its 3D position can be easily calculated by a wide baseline stereo method (Fig.4). Unfortunately, we cannot expect to detect the ball regions for the both cameras for every frame due to some occlusions or overlaps with players, and some noise also matters.

We model the ball motion model three dimensionally and exploit the Kalman Filter to compensate the status of the ball-missing frame. Since our method aims to build up on-line system that broadcasts the real soccer game in virtual fashion[2], we need to exploit a deterministic method that can uniquely estimate the ball position. Hence we use the Kalman-Filter which costs small calculation and is deterministic, and we avoid the Particle-Filter[9][10] for that reason. Note that we assume the cameras are calibrated in advance and they are synchronized.

### 4.1 The Prediction of 3D Ball Position

Let us denote the state vector  $\mathbf{X}_k$  as the 3D position, the velocity, and the acceleration of the ball at frame  $k$  and the measurement vector  $\mathbf{Y}_k$  as the 3D position calculated according to the two ball regions on the two camera images.  $\mathbf{X}_k$  and  $\mathbf{Y}_k$  are defined by:

$$\mathbf{X}_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k, \ddot{x}_k, \ddot{y}_k, \ddot{z}_k]^T \quad (1)$$

$$\mathbf{Y}_k = [p_k, q_k, r_k]^T \quad (2)$$

Where  $(x_k, y_k, z_k)$  and  $(p_k, q_k, r_k)$  are the 3D position vector in the soccer field. The transition equation and the observation equation are:

$$\mathbf{X}_{k+1} = A\mathbf{X}_k + \gamma \quad (3)$$

$$\mathbf{Y}_k = C\mathbf{X}_k + \omega \quad (4)$$

Where  $A$  is the state transition matrix,  $C$  is the measurement matrix, and  $\gamma$  and  $\omega$  are the process noise and measurement noise that are modeled by the Gaussian distribution. We assume horizontal velocity and the vertical acceleration as constant. Therefore,  $A$  and  $C$  are given by:

$$A = \begin{bmatrix} 1 & 0 & 0 & \delta_t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \delta_t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \delta_t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \delta_t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

Where  $\delta$  is the interval between two successive frames. When the 3D position of the ball cannot be obtained because of the failure of ball detection in any of the two cameras, we compensate it with the predicted vector by the Kalman Filter. If the prediction of vertical position component is negative, it is regarded as a bounce and the vertical component value is changed to satisfy the complete elastic collision on the ground.

## 5 Experiment

### 5.1 Image Acquisition

We set up video cameras for taking images of soccer scene in the national Kasumigaoka stadium of Japan. As shown Fig.5, half of the field can be captured by camera A and camera B. We use two NTSC cameras (SONY DXC-9000). The cameras are synchronized by GPS signals. The size of the captured image is 640x480 and the frame rate is 30 fps. We implement the proposed method and run it on the PC that has Intel(R) Core(TM)2 CPU and 2,048 MB memory with Vine Linux 4.1.

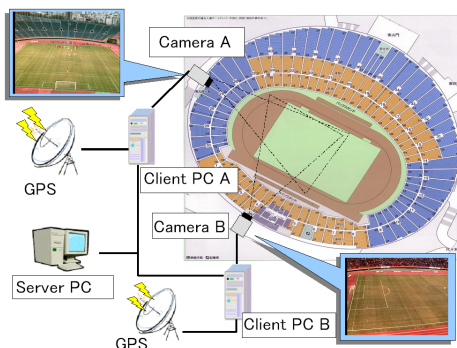


Fig. 5. Image Acquisition

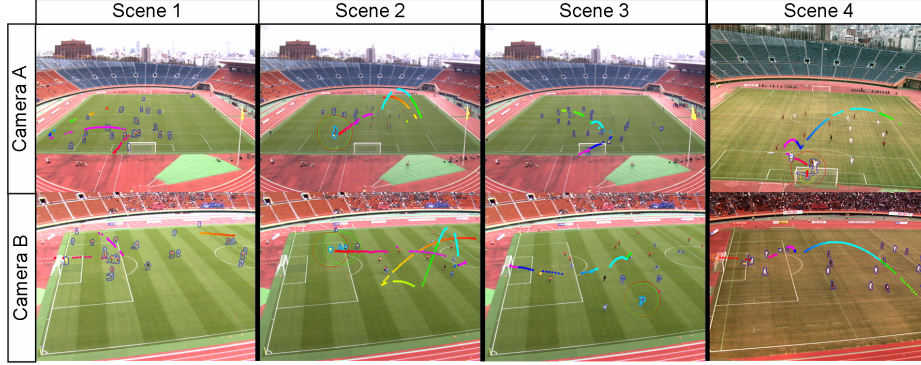


Fig. 6. Camera (SONY DXC-9000)

### 5.2 The Ball Detection Result

We took short videos in 4 scenes during the games of the inter-university championship of Japan. Fig.7 shows the estimated trajectories of the ball. The result of image processing step is shown in Table.1.  $f_{total}$  indicates the total frames in each scene.  $f_{visible}$  indicates frames in which the ball is visible. Our method successfully detected the ball for  $f_{success}$  frames. The result shows our method detects the ball for more than 80% of frames.

The method failed to detect the ball for some situations. The method missed the ball for  $f_{failure1}$  frames though it is visible, and it incorrectly estimated



**Fig. 7.** Trajectory of detected ball

**Table 1.** Ball detection result

	Camera A				Camera B			
Scene	Scene 1	Scene 2	Scene 3	Scene 4	Scene 1	Scene 2	Scene 3	Scene 4
$f_{total}$	350	430	300	217	350	430	300	217
$f_{visible}$	74	195	79	101	67	253	79	116
$f_{success}$	73	171	76	101	54	224	71	116
$f_{failure1}$	0	24	3	0	13	29	7	0
$f_{failure2}$	1	0	0	0	0	0	1	0
$f_{failure3}$	0	1	2	0	0	1	1	0

the position of the ball in the images for  $f_{failure2}$  frames. it also incorrectly reported the detection of ball for  $f_{failure3}$  frames in which the ball is invisible. Our method can not detect the ball when a part of player is recognized as a ball. It also fails if its speed is very slow when the ball comes to be visible.

When the ball was invisible or involved in player regions, the system tried to detect the ball with the (C) players-centered search because several players were often found around the ball. Note that the ball detection is impossible when the ball visibility is no, if any of the cameras was in this status, we need the Kalman Filter step to compensate the 3D position of the ball.

The average time of the ball detection methods are shown in Table.2. Note that we succeeded in speeding up the ball detection for 8.5 (A) and 2.1 (B) times faster than the normal whole image search (C). As the frame interval for NTSC video is 33.3 [m-sec], we can say that our method can sufficiently achieve the real-time speed.

### 5.3 3D Position Estimation of the Ball

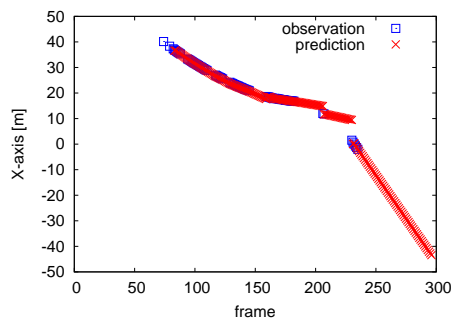
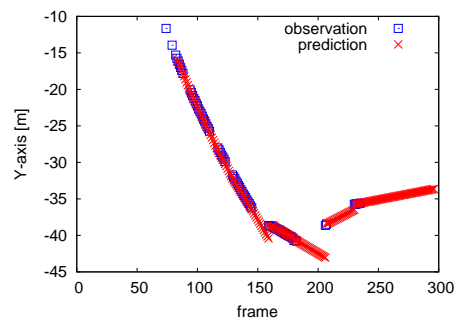
We calculated the 3D ball position from the above results. The blue small squares in Fig.8, Fig.9, and Fig.10 show the calculation result of the 3D position in



**Table 2.** Average time and frequency of the ball detection methods

The method	Average time [m-sec]	Frequency [%]
(A) Ball-Centered Search	3.08	29.6
(T) Template Matching	10.49	2.1
(B) Players-Centered Search	12.49	35.1
(C) Whole Image Search	26.17	33.2

scene 4. Each figure shows the result of 3D position in X, Y (horizontal), and Z (vertical) axis respectively. We utilized the Kalman Filter to estimate the 3D position of the ball for the frames at which the ball position cannot be directly calculated because of the failure of the ball detection in the two cameras. The results are shown in red crosses in the same figures. The estimation was acceptable for the most of the video sequence. However, in the sequence from frame 200 to frame 230, the prediction was failed due to the insufficient number of frames for observation. Fig.11 shows the 3D plot of the ball trajectory. The white ball indicates the directly calculated positions and the red wire ball indicates the estimated position given by the Kalman filter.

**Fig. 8.** X-axis**Fig. 9.** Y-axis

## 6 Conclusion

We propose an adaptive method that can estimate 3D position of a soccer ball by using two viewpoint videos. At an image processing step, our method can detect the ball by selecting the best algorithm based on the ball states so as to minimize the chance to miss the ball. The 3D position of the ball is then calculated with the estimated 2D positions of the two camera images. We utilize the Kalman Filter to estimate the 3D ball position even when the ball was lost on the image processing step.

Since the ball cannot be detected when the ball is in front of the players in the proposed method, the ball detection method could be improved. Experiments for long video sequence are also needed to verify our proposed method.

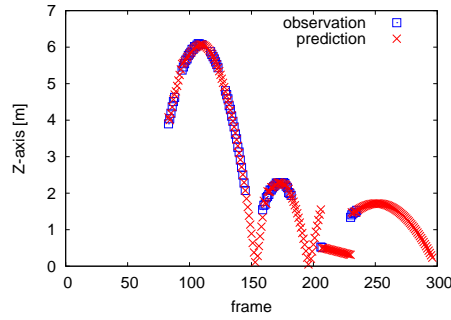


Fig. 10. Z-axis

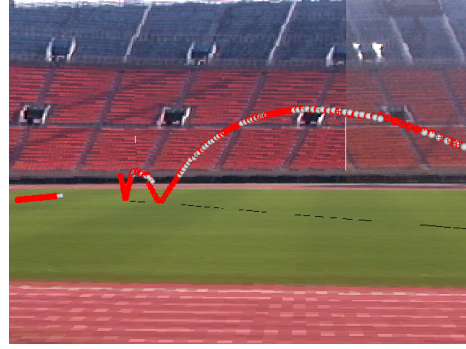


Fig. 11. Plot of 3D position of the ball

## References

1. <http://www.ri.cmu.edu/events/sb35/tksuperbowl.html>
2. T. Koyama, I. Kitahara, and Y. Ohta, "Live Mixed-Reality 3D Video in Soccer Stadium," ISMAR, pp.178-187, 2003.
3. K. Hayashi, and H. Saito, "Synthesizing Free-Viewpoint Images from Multiple View Videos in Soccer Stadium," IEEE International Conference on Computer Graphics, Imaging and Visualisation 26-28, pp.220-225, 2006.
4. T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti, "A Ball Detection Algorithm for Real Soccer Image Sequences," 16th International Conference on Pattern Recognition - Volume 1, 2002.
5. 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, Graz, Austria, 2006.
6. J. Ren, J. Orwell, and G. Jones, "Generating Ball Trajectory in Soccer Video Sequences," ECCV on Computer Vision Based Analysis in Sport Environments, 2006.
7. F. Yan and W. Christmas and J. Kittler, "A Tennis Ball Tracking Algorithm for Automatic Annotation of Tennis Match," Proceedings of the British Machine Vision Conference BMVC2005, pp.619-628, 2005.
8. T. Misu, A. Matsui, M. Naemura, M. Fujii, and N. Yahi, "Distributed Particle Filtering For Muticocular Soccer-ball Tracking," ICASSP, Distributed and Cooperative Adaptive Processing, 2007.
9. S. Choi, Y. Seo, H. Kim, and K. Hong, "Where are the ball and players?: Soccer Games Analysis with Color-based Tracking and Image Mosaik," Proc. of ICIAP, pp.196-203, 1997.
10. K. Matsumoto, S. Sudo, H. Saito, and S. Ozawa "Optimized Camera Viewpoint Determination System for Soccer Game Broadcasting," IAPR Workshop on Machine Vision Applications, pp.115-118, 2000.
11. T. Iizuka, T. Nakamura, and T. Wada, "Real Time 3-D Position Measurement System Using a Stereo Camera," MIRU2004 Proceedings I, pp.111-112, 2004. (Japan)