AUTOMATIC BROADCAST VIDEO GENERATION FOR BALL SPORTS FROM MULTIPLE VIEWS

Kyuhyoung Choi, Sang Wook Lee and Yongduek Seo

School of Media, Sogang University,
1 Shinsu-dong Mapo-gu, Seoul, 121-742, Korea
{kyu, slee, yndk}@sogang.ac.kr

ABSTRACT

Generally a TV broadcast video of ball sports is composed from those of multiple cameras strategically mounted around a stadium under the supervision of a master director. The director decides which camera the current view should be from and how the camera work should be. In this paper, such a decision rule is based on the 3D location of ball which is the result of multi-view tracking. While current TV sports broadcast are accompanied with professional cameramen and expensive equipments, our system requires few video cameras and no cameraman. The resulted videos were stable and informative enough to convey the flow of a match.

1. INTRODUCTION

Soccer, basketball, baseball and tennis are so popular sports all over the world that sometimes make people stay awake in front of TV to cheer for their favorite teams or players. For this ball sports TV broadcast, major broadcasting companies have a special team which consists of a director, expert cameramen and high-quality equipment. Basically TV broadcast videos are generated by switching among the cameras under the direction of the producing director. Some experienced camera works and computer graphics are complementary elements. After all such an expert direction is all about how much he or she can make TV viewers understand and be immersed in the flow of a match. For ball sports, the key of the flow is the ball, that is, where is the ball, who possesses the ball [1]. The structure of player positions may be the underlying key for the players and coaching staffs. However, for TV viewers watching a sequence without the ball in it is very annoying and unimaginable thing. Thus this paper presents a system for automatically making stable and informative sports(soccer) broadcast video based on 3D ball tracking and 2D positions of players.

Surprisingly, there have been few researches about this kind of automatic broadcast video generation. The most similar to our research is Wang *et. al.*'s [9]. While our goal is to generate optimal broadcast video from common camcorder mounted by a general user, theirs is about how to make a computer take a role of the master director with setting of major broadcast company's level. The topic of

[1]The view in this paper do not mean close-up view such as of Figure 1(a) and 1(b) but normal view like 1(c)

(a) (b) (c)

Fig. 1. Typical broadcast soccer scenes of various resolutions.

Fig. 2. Camera plane of 2006 FIFA German Worldcup [2]

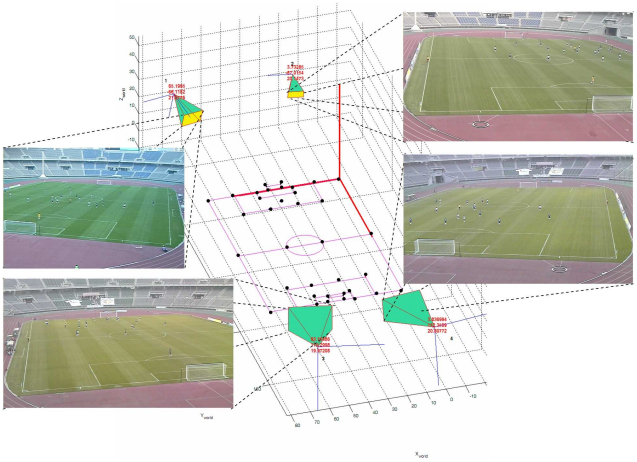


Fig. 3. Multiple cameras surrounding the pitch.

this paper is one of general modern sports broadcast issues such as slow-motion-replay, statistics of a match and so on which are examined in [9]. The criterion for camera selection in [9] is the clearness of a view while ours is the projected ball size. Though it is not about sports broadcast, the research in [7] is quite interesting and rigorous in a way that the instructions from professional cameramen are programmed to produce optimal broadcast of video lectures. The organization of this paper is as following. Section 2 is about the calibration of cameras mounted along the pitch, Section 3 deals with 3D ball tracking. The rules of camera view selection and subimage cropping are discussed in Section 4 and Section 5 respectively followed by experimental setting and results(Section 6) and conclusion(Section 7)

2. CAMERA CALIBRATION

General TV soccer broadcast accompanies with 10 to 25 cameras along the stadium according to the importance of the match (See Figure 2). Sometimes, for such games enjoying global attention as FIFA Worldcup, specialized organization manages the broadcasting [2]. Except such special places as goal post, cameraman is taking care of each camera which is supposed to be able to pan, tilt and zoom. However, as shown in Figure 3 the number of cameras used in this paper is much smaller(four) and they are static with no cameraman. Therefore, ideally one person is needed to mount all the cameras. Even for the case of built-in cameras, no one is required. In some sense, generating quality broadcast video with minimum number of cameras and people can be an advantage of the proposed system. Camera calibration is essential to extract 3D information of objects(ball or players). In this paper, Tsai's camera model [8] is used to model 3D to 2D perspective projection, that is, a function P_c which relates a 3D world position $\mathbf{X}$ to an image point $\mathbf{x_c}$ of the camera c

$$\mathbf{x}_c = P_c\left(\mathbf{X}\right) \tag{1}$$

For this, matching between 3D pitch landmarks(mostly intersections of white lines) and its corresponding 2D image (corner)points should be done somehow as many as possible such as in Figure 4.[2] In most of related papers, this is manually done by a user. However, the proposed system find correspondences hence camera parameters automatically as following.

Input: edge map of camera view (Figure 5(a)) and line segment image via Hough transform [6]
Output: camera parameters

1. Find the projected half circle by ellipse detection (Figure 5(a)).
   In [10], a pair of pixels is tested for how likely the two pixels are the ends of long axis of an ellipse. This method is known to be fast and we made it even faster by resizing the edge image by half or quarter and restricting the pixels within the pitch image blob obtained by using mean shift image segmentation [4].

2. Find and refine the half line w
   Since the detected ellipse O is supposed to be the projection of the half circle, there must be a segment of the half line w inside O. The Hough line with the most votes inside O is selected as a part of w. However, extrapolation of this line segment to both direction is not likely to give the exact w since the segment is just fitting for its small part clipped by O. Iterative weighted least square on edge pixels with the line segment as the initial converges to the refined w . The two intersections(A' and C') of O and w are computed. Then the center of A' and C' is assumed to be a tentative pitch center(B').

3. Find the vanishing point and its supporting lines (Figure 5(b))
   Since Hough line segments mostly come from the white lines drawn on the pitch, they can be clustered into two groups, that is, one from lines parallel and the other(m) from lines perpendicular to the half line. Lines of each group are supposed to meet at a vanishing point. After k-means clustering($k = 2$), RANSAC [5] with lines in m gives a vanishing point q.

4. Compute the intersections(D') of half and side lines using cross ratio.
   As shown Figure 5(c), the cross ratio u is invariant under perspective transform from a set of four points (A, B, C and D) on the half line to the projected (A', B', C' and D').

$$u = \frac{\overline{AB}\,\overline{BD}}{\overline{AD}\,\overline{BC}} = \frac{9.15\text{m} \times 34\text{m}}{(9.15\text{m} + 34\text{m}) \times 9.15\text{m}} = \frac{\overline{A'B'}\,\overline{B'D'}}{\overline{A'D'}\,\overline{B'C'}} \tag{2}$$

   Knowing the cross ratio u from the real pitch size [3], the (tentative) pitch center(B') and two intersections(A'

[2] If the real pitch size(in meters) of interest can be known, that can be used to compute the 3D coordinates of a corner point. Otherwise, international standard size can be used instead [1].

[3] 9.15 and 34 meter are the mandatory and recommended standard of half circle radius and pitch half width respectively

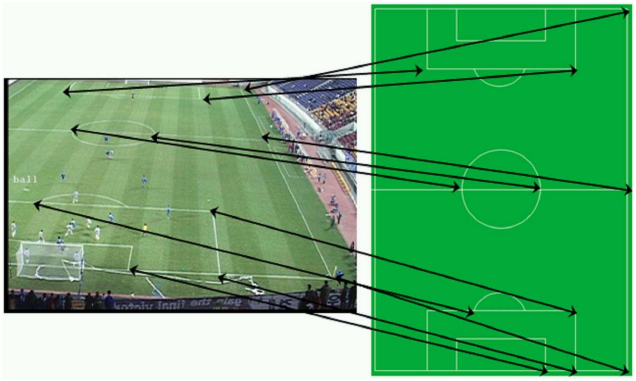


Fig. 4. Correspondences between points in camera view and those in 3D pitch coordinates.

and C'), we can compute the expected position(D') of the intersection of a side line and the half line. Due to the inaccuracy of B', D' does not coincide with the actual intersection. So the nearest among intersections between the half line and lines in m is chosen to be D'.

5. Refine the center position B' of the pitch
With A', C' and D', the pitch center B' can be re-estimated again using cross ratio. If allowed, we can re-estimate the center from both pitch sides, namely, B'_L and B'_R. Then B' is taken as the mean.

$$B' = \frac{\overline{B'_L D'_L}}{\overline{B'_L D'_L} + \overline{B'_R D'_R}} B'_L + \frac{\overline{B'_R D'_R}}{\overline{B'_L D'_L} + \overline{B'_R D'_R}} B'_R \quad (3)$$

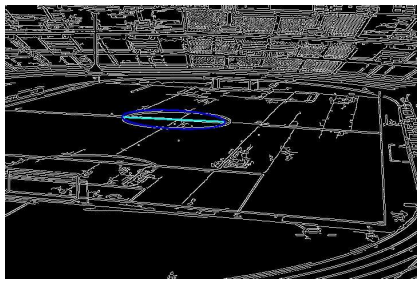
6. Compute center positions of goal lines by cross ratio. If we draw a line e passing though q and B', it is a projection of a line which is perpendicular to the half line $P^{-1}(w)$ and dividing the pitch into equal halves. We can find the intersections of e and both goal lines in the same way as finding D' with cross ratio(Step 4).

7. Find other correspondences as many as possible (Figure 5(d)) With the corner points found so far, we can estimate the projected four corners of the pitch. The rest of visible intersections of pitch lines can be also estimated by cross ratio likewise.

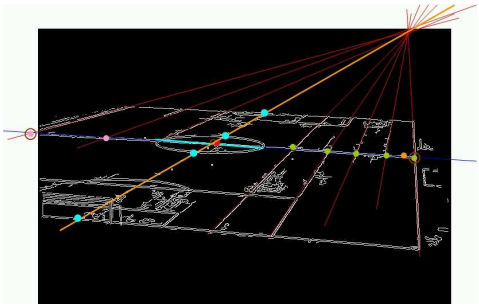
8. Compute camera parameters by Tsai's method with the input as the found correspondences (Figure 5(e))

3. 3D BALL TRAJECTORY

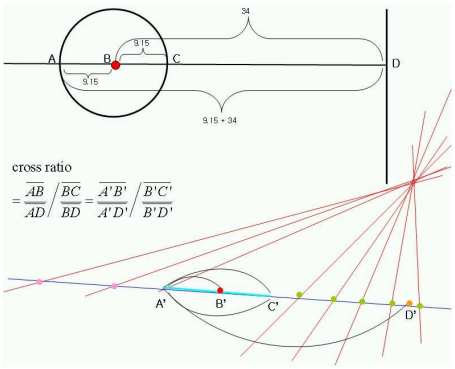
Though it would be better if information about players were also extracted and used, the most important element in decision of camera view selection is the ball, that is, the 3D position (and velocity) of ball. This implies 3D tracking of ball for which a method of [3] is used. The standard computer vision algorithm for 3D reconstruction is triangulation using



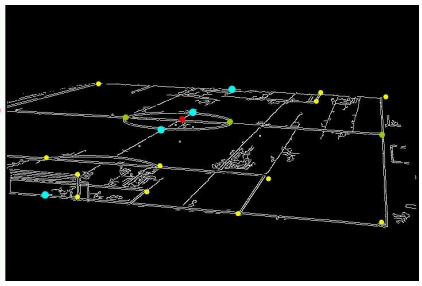
(a) The pitch half circle is detected as an ellipse along with half line inside



(b) A vanishing point as an intersection of pitch lines which are not parallel to the half line



(c) Cross ratio invariant under transform from the bird-eye(upper) to camera(lower) view



(d) Found landmark point correspondences

(e) Each re-projection error in pixels and the mean(bottom)

Fig. 5. Automatic camera calibration by ellipse and vanishing point detection as well as cross ratio.

stereo vision where the 3D coordinates of an (object)point are computed by the corresponding projections onto two (or more) camera views. Though a ball as a sphere is ideally projected as a circle which is a primitive figure with few parameters, its actual image in broadcast video is like ellipse due to motion blur and imaging process noise. Decisively ball tracking is easily failed when there are noises and clutters due to it's relatively small size in a camera view.

In [3], ball tracking is a process of filtering the most likely sequence of 3D positions, $Q^*_{1:T}$ out of a sequence of noisy multi-view 2D observation sets, G.

$$\begin{aligned} Q^*_{1:K} &= \{\mathbf{X}^*_k\}^K_{k=1} && (4)\\ S &= \{Q_{k_1:k_2} | 1 \le k_1 \le k_2 \le K\} && (5)\\ U &= \left\{\{\mathbf{X}^i_k\}^{I_k}_{i=1}\right\}^K_{k=1} && (6)\\ G &= \left\{\left\{\{\mathbf{x}^c_k(j)\}^{J_c}_{j=1}\right\}^C_{c=1}\right\}^K_{k=1} && (7)\end{aligned}$$

where C is the number of cameras and J is the number of observations. S is a set of 3D trajectory segments and U is a set of 3D ball candidates. As in reverse order, U is built from G, then S is from U as well as $Q^*_{1:K}$ is extracted from S. To build U from G, for each time k, 3D ball candidates are generated from all the possible pairs $\left(\frac{C!}{2(C-2)!}\right)$ of synchronized C views. Given the camera parameters, a point, for example the one $\mathbf{X}^g$ on the pitch ground, on a ray from the camera center, $\mathbf{X}^c$ are projected on a point on the image. Ideally if there exists a 3D object(as a point) and a pair of camera project it, the rays from each camera center to the projected point meet at the 3D point. However, due to some noise, the rays may not meet each other posing some distance. If the distance is tolerable, the mid-point between the rays is taken as a 3D ball candidate. The mid-point $\mathbf{X}^{mid}$ between two rays passing through two points $\mathbf{X}^c_1$ and $\mathbf{X}^g_1$, and $\mathbf{X}^c_2$ and $\mathbf{X}^g_2$ respectively is computed as following.

$$\mathbf{X}^{mid}_{1,2} = \frac{\mathbf{X}^{closest}_1 + \mathbf{X}^{closest}_2}{2} \quad (8)$$

$$\begin{bmatrix}\mathbf{X}^{closest}_1 \\ \mathbf{X}^{closest}_2\end{bmatrix} = A^{-1}B \quad (9)$$

where

$$A = \begin{bmatrix} \mathbf{X}^c_1(z) & 0 & \mu^{g,c}_1(z) & 0 & 0 & 0 \\ 0 & \mathbf{X}^c_1(z) & \mu^{g,c}_1(z) & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{X}^c_2(z) & 0 & \mu^{g,c}_2(x) \\ 0 & 0 & 0 & 0 & \mathbf{X}^c_2(z) & \mu^{g,c}_2(y) \\ \mu^{c,g}_1(x) & \mu^{c,g}_1(y) & \mu^{c,g}_1(z) & \mu^{g,c}_1(x) & \mu^{g,c}_1(y) & \mu^{g,c}_1(z) \\ \mu^{c,g}_2(x) & \mu^{c,g}_2(y) & \mu^{c,g}_2(z) & \mu^{g,c}_2(x) & \mu^{g,c}_2(y) & \mu^{g,c}_2(z) \end{bmatrix} \quad (10)$$

$$B = \left(\mathbf{X}^g_1(x)\mathbf{X}^c_1(z), \mathbf{X}^g_1(y)\mathbf{X}^c_1(z), \mathbf{X}^g_2(x)\mathbf{X}^c_2(z), \mathbf{X}^g_2(y)\mathbf{X}^c_2(z), 0, 0\right)^T \quad (11)$$

$\mathbf{X}^{closest}_1$ and $\mathbf{X}^{closest}_2$ are the points on the two rays closest to each other and $\mu^{a,b}_i = \mathbf{X}^a_i - \mathbf{X}^b_i$. Then U is the set of mid-points :

$$U = \left\{\left\{\mathbf{X}^{mid}_{\mathrm{m}_C(c),\mathrm{m}_C(c+1)}\left(p^i_{\mathrm{m}_C(c)}(k), p^j_{\mathrm{m}_C(c+1)}(k)\right)\right\}^C_{c=1}\right\}^K_{k=1} \quad (12)$$

where $1 \le i \le J^k_{\mathrm{m}_C(c)}, 1 \le j \le J^k_{\mathrm{m}_C(c+1)}$ and $\mathrm{m}_C(c) = MAX(\mathrm{mod}(c, C+1), 1)$.

From U, S is built by extending all the possible triplets, three consecutive 3D ball candidates, as long as possible to give trajectory segment candidates. A sequence of three ball candidates in U is qualified to be a triplet if their acceleration and velocities show that of ballistic motion under gravity :

$$S = \left\{\{\mathbf{X}_k\}^{k_2}_{k=k_1} | \beta_k\delta_k\delta_{k+1}\varphi_k > 0, \forall k : k_1 \le k \le k_2 - 2\right\} \quad (13)$$

where

$$\beta_t = \begin{cases} 1 & T^l_\beta < \mathbf{X}_{k+2}(z) - 2\mathbf{X}_{k+1}(z) + q_t(z) < T^u_\beta \\ 0 & otherwise \end{cases} \quad (14)$$

$$\delta_t = \begin{cases} 1 & T^l_\delta < \|\mathbf{X}_{k+1} - \mathbf{X}_k\|_2 < T^u_\delta \\ 0 & otherwise \end{cases} \quad (15)$$

$$\varphi_t = \begin{cases} 1 & \cos^{-1}\left(\frac{(\mathbf{X}_{k+1}-\mathbf{X}_k)\cdot(\mathbf{X}_{k+2}-\mathbf{X}_{k+1})}{\|\mathbf{X}_{k+1}-\mathbf{X}_k\|_2\|\mathbf{X}_{k+2}-\mathbf{X}_{k+1}\|_2}\right)_2 < T_\varphi \\ 0 & otherwise \end{cases} \quad (16)$$

To get $Q^*_{1:K}$ from S, the longest one among the segment candidates is chosen and fitted to a parabolic curve parameterized by Θ.

$$\begin{aligned}\Theta = \{a, b, c, d, e, f, g | \mathbf{X}_k(x) = ak + b, \mathbf{X}_k(y) = c\mathbf{X}(x) + d,\\ \mathbf{X}_k(z) = e\mathbf{X}(x)^2 + f\mathbf{X}(x) + g\}\end{aligned} \quad (17)$$

The nearest segments to the both ends of the longest are merged into the longest if its fitness to the curve is tolerable, then the curve is updated considering the new support. After iterations as shown in Algorithm 1, a parabolic representative of the ball motion for a certain period is estimated. At the starting and ending points of the period, the 3D curve takes off and lands on the pitch ground respectively. Since the coefficients of 3D parabolic curve equation is estimated $Q^*_{1:T}$ is a function of time.

$$\begin{aligned} Q^*_{1:K} &= \{\mathbf{X}^*_k\}^K_{k=1} && (18)\\ S &= \{Q_{k_1:k_2} | 1 \le k_1 \le k_2 \le K\} && (19)\\ U &= \left\{\{\mathbf{X}^i_k\}^{I_k}_{i=1}\right\}^K_{k=1} && (20)\\ G &= \left\{\left\{\{\mathbf{x}^c_k(j)\}^{J_c}_{j=1}\right\}^C_{c=1}\right\}^K_{k=1} && (21)\end{aligned}$$

where C is the number of cameras and J is the number of observations. S is a set of 3D trajectory segments and U is a set of 3D ball candidates. As in reverse order, U is built from G, then S is from U as well as $Q^*_{1:T}$ is extracted from S.

Input: $\{q\}_S = \arg \max_{\{q_t\}_{t=i}^{j} \in S} (j-i)$
while $isChanged = T$ and $isOutOfRange = F$ **do**
$isChanged := F$
$isOutOfRange := F$
foreach $i, \{q\}^i \in S$ **do**
if $\{q\}^i$ and $\{q\}_S$ are temporally overlapped and qualified to be parts of the same sequence **then**
$\{q\}_S := \{q\}_S \cup \{q\}^i$
$\Theta_S := \arg\max_{\Theta} p(\{q\}_S|\Theta)$
$isChanged := T$
if $S = \emptyset$ *or* $q_1^S(z) \leq 0$ *and* $q_{|\{q\}^S|}^S(z) \leq 0$ **then**
$isOutOfRange := T$
end
break
end
end
end

Algorithm 1: Growing a sequence of 3D points supporting a 3D parabolic curve

4. CAMERA SELECTION

Basically, which camera view should be selected for current image depends on from which camera TV audience can see the largest ball, that is, from which camera the area of the projected ball will be the largest.

$$c* = \arg\max_{c} \|P_c(\mathbf{X} + \Delta\mathbf{X}) - P_c(\mathbf{X})\|_2 \qquad (22)$$

However, as the ball kicked by players moves back and forth and left and right over the pitch, some scene [4] might be too short if it is only generated by Equation 22.

For example, it is a common scenario for the ball kicked by a goalkeeper to come down on unexpected position after a player head and change the direction of the ball. Let's say we have a result sequence of [long kick (camera 1, 100 frames)] → [heading (camera 2, 5 frames)] → [falling and bouncing (camera 3, 80 frames)] according to Equation 22. If this video is broadcast, TV viewers will feel annoyed by the flash-like short scene, namely, five frames from camera 2.

Therefore, such a scene whose frame length is under a given threshold should be eliminated and filled up by adjacent scenes on both ends. In other words, either or both portions of 1 and 3 ae increased in order to vanish 2.

5. SUBIMAGE CROPPING

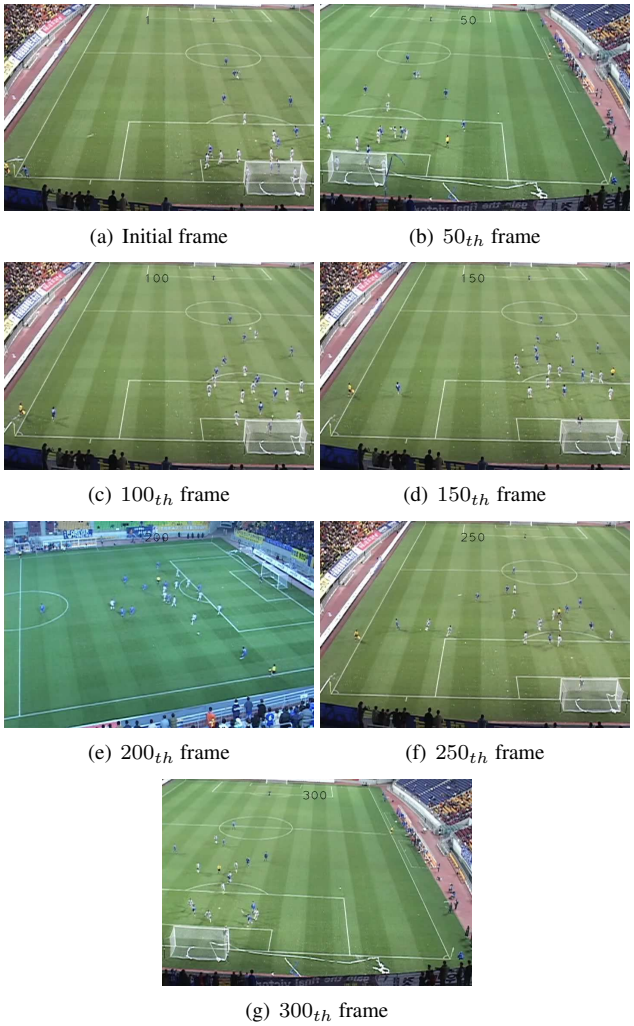
When the image size of broadcast video standard is different from those of camera views, we have to define the proper size of which a subimage is cropped from the original view [5]. The criteria for this is the position of ball in the view, that is, the subimage is cropped in a way that the ball is as central in the view as possible.

6. EXPERIMENTAL RESULTS

We tested our system on two sequences $SEQ1$ and $SEQ2$ of soccer match and the results are shown in Figure 6 and 7 respectively.

Four cameras are located around a goal-mouth on the half of the pitch and the scene captures the moments of goal-in for $SEQ1$. All camera views have the same size of 720 × 480.

For $SEQ2$, also four cameras are mounted almost at each corner top of the pitch. Three of them are size of 1280 × 720 and one is of 720 × 480. So the broadcast video is generated by cropping subimages in the size of 720 × 480 as in Section 5.



(a) Initial frame (b) 50_{th} frame

(c) 100_{th} frame (d) 150_{th} frame

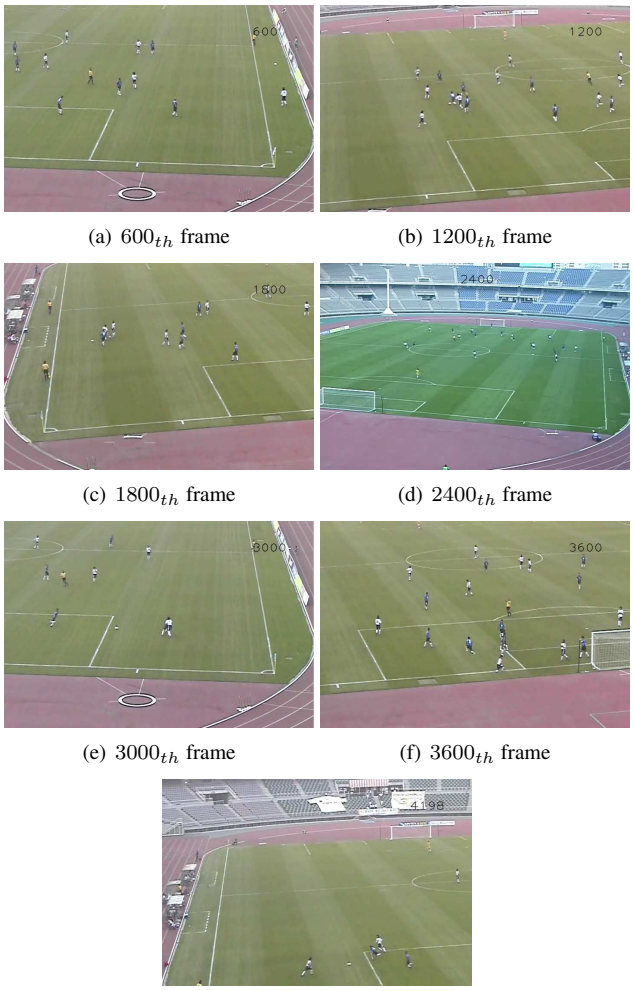
(e) 200_{th} frame (f) 250_{th} frame

(g) 300_{th} frame

Fig. 6. Sample images of a resulted broadcast video

[4] Here a scene is a sequence of continuous frames from the same camera

[5] Though the standard size can be either bigger or smaller than the camera view size, the latter is assumed here which means subimage cropping. In a way, this sequential cropping may take an effect of panning and tilting of a virtual subcamera.



(a) 600_{th} frame (b) 1200_{th} frame

(c) 1800_{th} frame (d) 2400_{th} frame

(e) 3000_{th} frame (f) 3600_{th} frame

(g) 4200_{th} frame

Fig. 7. Sample images of a resulted broadcast video. Note the view 7(c) and 7(g) are from the same camera, but cropped differently according to the ball position.

7. CONCLUSION

We proposed a vision-based broadcast video composition system whose camera selection and subimage cropping rules are based on the 3D position of ball which is the result of multi-view tracking.

While current TV sports broadcast are accompanied with professional cameramen and expensive equipments, our system requires few video cameras and no cameraman. The resulted videos were stable and informative enough to convey the flow of a match.

8. ACKNOWLEDGEMENTS

This research was accomplished as the result of the research project for Culture Contents Technology Development supported by KOCCA.

This work was also supported by the IT R&D program of MKE/IITA. [2008-F-030-01, Development of Full 3D Reconstruction Technology for Broadcasting Communication Fusion]

9. REFERENCES

[1] The field of play. URL http://en.wikipedia.org/wiki/Football_pitch

[2] Host boradcast services. URL http://www.hbs.tv/2006fwc-9.html

[3] Choi, K., Seo, Y.: Parabolic curve fitting as a 3d trajectory estimation of the soccer ball. In: Joint Conference on Communication Information, PyeongChang, Rep. of Korea, pp. 331–335 (2007)

[4] D. Comaniciu, P.M.: Robust analysis of feature spaces: color image segmentation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, p. 750 (1997)

[5] Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)

[6] Hough, P.V.C.: Machine analysis of bubble chamber pictures. In: International Conference on High Energy Accelerators and Instrumentation, pp. 554–556 (1959)

[7] Rui, Y., Gupta, A., Grudin, J., He, L.: Automating lecture capture and broadcast: technology and videography. Multimedia Systems **10**(1), 3–15 (2004). Journal Article

[8] Tsai, R.: An efficient an accurate camera calibration technique for 3D machine. In: Proceedings of CVPR'86, pp. 364–374 (1986)

[9] Wang, J., Xu, C., Chng, E., Lu, H., Tian, Q.: Automatic composition of broadcast sports video. Multimedia Systems (2008)

[10] Xie, Y., Ji, Q.: A new efficient ellipse detection method. In: International Conference on Pattern Recognition, vol. II (2002)