A Mobile AR System for Sports Spectators using Multiple Viewpoint Cameras

Ruiko Miyano, Takuya Inoue, Takuya Minagawa, Yuko Uematsu and Hideo Saito
Department of Information and Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Japan
{*rui, inoue, takuya, yu-ko, saito*}*@hvrl.ics.keio.ac.jp*

Keywords: Augmented Reality (AR), Sports Player Analysis, People Detection, Smartphone, Homography.

Abstract: In this paper, we aim to develop an AR system which supports spectators who are watching a sports game using smartphones in a spectators' stand. The final goal of this system is that a spectator can watch information of players through a smartphone and share experiences with other spectators. For this goal, we propose a system which consists of smartphones and fixed cameras. Fixed cameras are set to cover the whole sports field and used to analyze players. Smartphones held by spectators are used to estimate positions where they are looking on the sports field. We built an AR system which makes annotation of players' information onto a smartphone image. And we evaluated the accuracy and the processing time of our system and revealed its practicality.

1 INTRODUCTION

Recently there are diverse styles to watch sports. Some spectators watch a sports game in a spectators' stand of gyms and grounds. This style has advantages that they can directly enjoy an exciting game and share their emotions with other spectators. The other spectators do that on TV, Internet and mobile devices. This style has advantages that they can easily get additional information about the game. We believe that it is useful to develop a system which combines both merits. For instance, if spectators could obtain additional information in a spectators' stand, they enjoyed watching a sports game more.

Therefore we focus on an Augmented Reality (AR) technology which projects virtual annotations onto a camera image. Using an AR system for smartphones, spectators can watch information of players through a smartphone in a spectators' stand like figure 1. Furthermore it is expected that they can communicate with other spectators through smartphones.

In this paper, we develop an AR system which supports spectators who are watching a sports game using smartphones in a spectators' stand. For this goal, our system uses fixed cameras and smartphones. Fixed cameras are set to cover the whole sports field and used to analyze players. That is to say, they are used to detect players' positions and recognize players' identities. Smartphones held by spectators are used to estimate positions where they are looking on the sports field.

Figure 1: AR system for watching sports.

There are difficulties for achieving each part. We have to analyze players in real time. Therefore we simplify the state-of-the-art researches about player detection and recognition. On the other hand we have to estimate where a smartphone is capturing. However it is hard to estimate an accurate position on the sports field because images captured by a smartphone camera have few interest points in many cases. Therefore we propose a method which estimates a homography matrix between a smartphone input image and the field using images and sensor information.

The contribution of this paper is to develop a mobile AR system for watching sports in a spectators' stand. We have carried out experiments to evaluate the accuracy and the processing time of our system.

2 RELATED RESEARCH

We introduce a research about a mobile AR system for watching sports in section 2.1. Our system adopts two techniques. One is to detect and identify players on the sports field. The other is to estimate the position where a smartphone camera is capturing in the field. We explain researches about sports player analysis in section 2.2 and a mobile AR system in section 2.3.

2.1 Sports AR System on Mobile Phones

Lee et al. proposed a mobile AR system to support spectators who are watching a baseball game in a spectators' stand (Lee et al., 2011). They achieved their goal by estimating the field, detecting players from images captured by a mobile phone and searching information of players from an information server. However there are two problems that a camera of the mobile phone must capture the entire field to estimate it and players' identities cannot be obtained.

Therefore we develop a system uses fixed cameras and smartphones. Smartphones do not need to capture the entire field since fixed cameras do it instead. Players' positions and identities are obtained from images captured by fixed cameras.

2.2 Sports Player Analysis

There are many researches about sports videos. Player analysis is one of the most popular researches and used for applications such as tactics analysis, automatic summarization and video annotations.

Some researchers proposed a player detection and tracking method using broadcast cameras (Miura and Kubo, 2008; Liu et al., 2009). However not all players' positions are estimated by these methods because cameras are moving and not covering the entire field.

Since we have to obtain positions and identities of players on the whole sports field, we focused on researches using multiple fixed cameras.

There are two approaches to detect people using fixed cameras such as a bottom-up and a top-down approach. In a bottom-up approach, foreground objects are projected onto a field image using the precalculated homography matrix (Khan and Shah, 2009; Delannay et al., 2009). This approach is suitable for a real-time processing because it can instantly detect people. In a top-down approach, existence probabilities of people are estimated using an iteration algorithm based on a generative model (Fleuret et al., 2008; Shitrit et al., 2011). This approach can accurately detect people, however it requires long processing time for iterations.

To recognize players' identities, color information and uniform numbers provide useful clues. A team of a player is estimated from a color of a uniform (Kasuya et al., 2010). And a player is identified from the team and a uniform number (Shitrit et al., 2011).

In this paper, we adopt a simplified method to detect players based on the idea of Fleuret et al. (Fleuret et al., 2008). And a team of a player is recognized from color information.

2.3 Mobile AR System

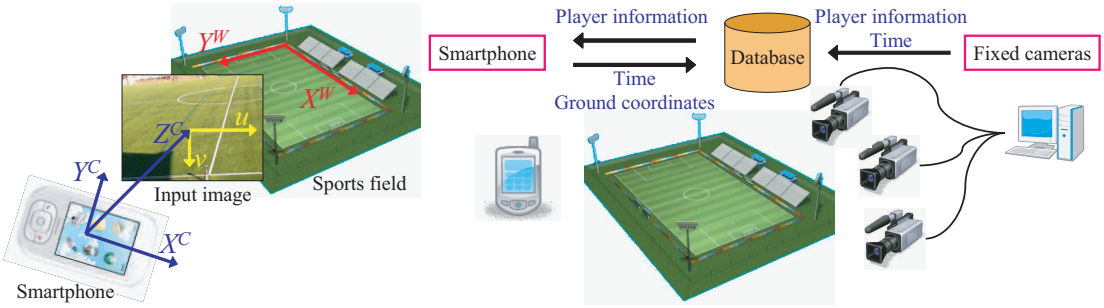
The position where a mobile device is capturing must be estimated for a mobile AR system. There are two approaches to estimate it: a sensor-based and a vision-based approach.

Many AR services adopted a sensor-based approach to roughly estimate a camera pose of a mobile device (Tonchidot, 2009). In these services, a GPS and an electronic compass are used to obtain a position and a camera pose of a mobile device. Tokusho and Feiner introduced an AR street view system using a GPS sensor and a digital pedometer (Tokusho and Feiner, 2009). A sensor-based approach has an advantage that a position and a pose of the device can be obtained without complex processing. However there is a problem that the sensors are susceptible to noise.

On the other hand many researchers focused on a vision-based approach for a robust AR system. Klein and Murray demonstrated an AR system on mobile phones using local features (Klein and Murray, 2009). A vision-based approach has an advantage that a position where a device is capturing can be accurately estimated by extracting local features. However it requires a lot of interest points.

If the user is in the spectator's stand, it can be assumed that the camera is not translated but only rotated. Therefore the capturing position can be estimated easier from panorama images with a vision-based approach. Oe et al. proposed to estimate a camera pose and position by comparing local features between a captured image and panorama images (Oe et al., 2005). This approach cannot be used for sports scene because there are few local features on the field. Chen et al. proposed to recognize scenes using template matching (Chen et al., 2004). In this approach, a template image is created from panorama images, however they consider only simple transformations such as a rotation and a translation.

To address each problem of the sensor-based and the vision-based approach, we proposed to combine



(a) Coordinate system (b) System architecture

Figure 2: Our proposed system.

both approaches (Miyano et al., 2012). In a vision-based approach, a transformed image created from a panorama image and sensor information is used to estimate a capturing position. Therefore we do not require the use of interest points, and can handle a complicate transformation.

3 PROPOSED SYSTEM

In this paper, we define three coordinate systems: $C^W(X^W, Y^W)$ for the sports field, $C^C(X^C, Y^C, Z^C)$ for a smartphone camera and $C^I(u, v)$ for an image captured by a smartphone camera (Figure 2(a)).

Coordinates of players in C^W and the homography matrix between C^W and C^I are required in real time to project annotations about players onto a smartphone image.

3.1 System Architecture

Figure 2(b) shows a system architecture. Our system consists of fixed cameras and a smartphone. Fixed cameras are used to estimate a position and a team of each player. The smartphone is used to find a position where the smartphone camera is capturing. Player information are exchanged through databases.

3.2 Player Analysis

We explain how to detect players and how to recognize a team of a player using multiple fixed cameras in this section. The purpose of player analysis is to find coordinates of players in C^W and to estimate a team of each detected player. These information are registered to the database every frame.

In preprocessing, rectangles which approximate player's standing area at the sports field C^W are generated. And color histograms are computed to recognize a team.

In online processing, players are detected using foreground masks and approximate rectangles. And a team of each detected player is recognized from pre-calculated color histograms.

3.2.1 Preprocessing

In preprocessing, information for player analysis are generated.

At first, we define *grids* to quantize the sports field C^W. Grids are defined so that each grid describes an area occupied by a standing player like figure 3.

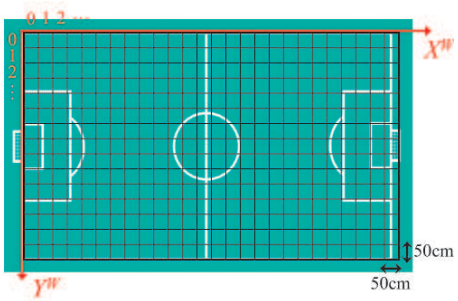


Figure 3: Grids on the sports field.

Then an *approximate rectangle* $A^c(k)$ is generated to detect and recognize players. This is defined so that $A^c(k)$ approximates a player who is standing at a grid k from a camera c like figure 4(a).

These rectangles are generated from the homography matrices between the sports field C^W and an image captured by a fixed camera.

To calculate homography matrices, images in which people are standing on intersection points of lines are required (Figure 4(b)). We define these images as *calibration images*. 4 or more corresponding points between calibration images and an image of the sports field are manually selected.

From these corresponding points, two homography matrices are calculated. One is the homography matrix $\mathbf{H_{g \to h}}$ from the sports field to a head plane captured by a fixed camera. The other is the homography

(a) Approximate rectangle (b) Calibration image (c) Points of a head plane and a foot plane

Figure 4: An image used for calculating the homography matrices.

(a) Background image (b) Foreground mask (c) Existence score of player

Figure 5: Player detection.

matrix $\mathbf{H_{g\to f}}$ from the field to a foot plane captured by a fixed camera. Red and green points in figure 4(c) represent points on a head plane and a foot plane.

To recognize a team of each player, color histograms are obtained by images of players' uniforms. Upper bodies of players are manually clipped from images captured by fixed cameras because they are regarded as players' uniforms.

We define a color histogram as 64 dimensions. RGB brightnesses of images are divided into 4 parts. Therefore color histograms have $4 \times 4 \times 4 = 64$ bins.

The approximate rectangles of players, the homography matrices, and the color histograms are independently calculated regarding each fixed camera.

3.2.2 Player Detection

In online processing, players are detected through three steps. First, a background image of each camera is created. Then an existence score of a player at each grid is estimated from a foreground mask and an approximate rectangle of a player. Finally existence scores from all cameras are combined.

A background images is created by adopting the medium values of previous images (Figure 5(a)). 10 images are selected, and a background image is updated every 100 frame.

As shown in Figure 5(b), a foreground mask is generated by subtracting a background image from an input image. Existence scores of players at all grids are calculated from Eq. 1. $p^c(k)$ is an existence score from the camera c which a player is standing at the grid k. Variable w and h denote a width and a height of an approximate rectangle $A^c(k)$. $B^c(k)$ is the number of pixels about foreground objects which is covering $A^c(k)$. A yellow area in figure 5(c) represents $B^c(k)$. $rate$ is defined as 0.7 because about 70% of approximate rectangles is assumed to be occupied by foreground objects.

$$p^c(k) = 1 - \frac{|B^c(k) - w \times h \times rate|}{w \times h \times rate} \tag{1}$$

Finally existence scores from all cameras are averaged using Eq. 2. Note that $v^i(k)$ is set to 1 when the i th camera is capturing the grid k, and to 0 otherwise. If a combined score $p(k)$ is larger than pre-defined threshold value, there is a player at grid k.

$$p(k) = \frac{\sum_{i=1}^{C} v^i(k) p^i(k)}{\sum_{i=1}^{C} v^i(k)} \tag{2}$$

3.2.3 Team Recognition

In online processing, a team of each detected player is recognized through three steps. First, an image of an upper body is clipped from an approximate rectangle of a detected player. Then a team is recognized by comparing to color histograms which are obtained in preprocessing. Finally the results of all cameras are combined.

An upper body of a player is clipped to recognize

Figure 6: Upper bodies clipped from detected players.

a team. This is clipped from a detected rectangle at the rate like figure 6.

Then an image of an upper body is represented as a 64 dimensional histogram. The similarity of every team is calculated using histogram intersection, that is Eq. 3 (Swain and Ballard, 1991). I, M and n denote the histogram of a detected player, the pre-calculated color histogram and the dimension of histogram, respectively.

$$hi_t^c(k) = \frac{\sum_{i=1}^{n} min(I_i, M_i)}{\sum_{i=1}^{n} M_i} \tag{3}$$

Finally histogram intersections from all cameras are averaged using Eq. 4. The team of a player standing at grid k is determined as t when $hi_t(k)$ is the largest value of all teams, and is larger than a predefined threshold value.

$$hi_t(k) = \frac{\sum_{i=1}^{C} v^i(k) hi_t^i(k)}{\sum_{i=1}^{C} v^i(k)} \tag{4}$$

3.3 Homography Estimation

The relation between the sports field C^W and a smartphone image C^I is required to project annotations of player information. Hence we need to calculate the homography matrix between C^W and C^I.

There are two approaches to estimate a position where the smartphone is capturing: a sensor-based and a vision-based. A sensor-based approach cannot estimate an accurate position because of a noise, and a vision-based approach cannot be suitable for the situation without interest points. Therefore we propose to combine a sensor-based and a vision-based approach which does not use interest points.

In a vision-based approach, it is effective to use panorama images for estimating a capturing position. However interest points and a general template matching are not adequate to apply to our situation. Therefore we utilize sensor information as parameters of a template matching. Reference images are created from pre-created panorama images and sensor information captured by an acceleration and a magnetic sensor. Then the position is refine by comparing a captured image with reference images.

In preprocessing, panorama images are generated. And in online processing, reference images are created from panorama images and sensor information. Finally an accurate matrix $\mathbf{H_{g\rightarrow i}}$ between C^W and C^I is obtained from the most similar reference image.

3.3.1 Preprocessing

In preprocessing, panorama images P^i are generated for image matching, and the homography matrices $\mathbf{H_{g\rightarrow p^i}}$ from the sports field to panorama images are calculated.

The smartphone must be moved to capture images and sensor information of the whole sports field by a user. We assumed that the smartphone is not translated but only rotated because a spectator does not change his seat. Therefore three angles such as a pan, a tilt and a roll obtained from sensor information should be considered. A pan, a tilt and a roll angle of a camera mean rotation around Y^C, X^C and Z^C axis in figure 2(a), respectively.

If one panorama image were generated from all captured images, there would be distortion of the sports field in the panorama image. Therefore captured images are classified into several groups according to pan angles. Then panorama images $G = \{P^1, P^2, \cdots\}$ are generated by image mosaicing from the every group. Figure 7 shows the example of panorama images generated by Image Composite Editor (Microsoft-Research, 2011).

(a) Left side (b) Right side

Figure 7: Panorama images.

To know relative relation between an input image and a panorama image P^i in online processing, a camera coordinate system $C^{Ci}_{center}(X^{Ci}_{center}, Y^{Ci}_{center}, Z^{Ci}_{center})$ is required (Figure 8). This is defined so that the camera is pointing to the center of panorama image P^i. The rotation angle $(p^i_{center}, t^i_{center}, r^i_{center})$ of C^{Ci}_{center} is calculated from the angles of captured images that have been used to generate P^i: the medium angle between the minimum and the maximum one in these angles.

The homography matrices $\mathbf{H_{g\rightarrow p^i}}$ between the sports field and each panorama image are calculated by manually inputting corresponding points like figure 9.

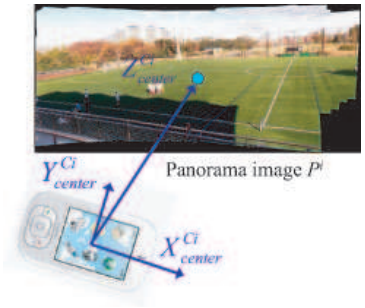


Figure 8: Camera coordinate system regarding a panorama image P^i.

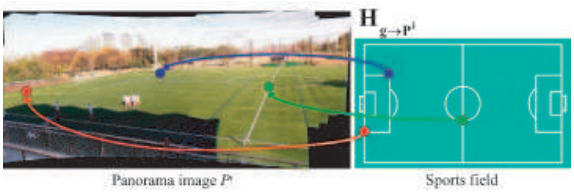


Figure 9: Corresponding points to calculate the homography matrices $\mathbf{H}_{\mathbf{g}\to\mathbf{P^i}}$.

3.3.2 Creating Reference Images

In online processing, an accurate homography matrix is estimated. Reference images are created from panorama images and sensor information to estimate a capturing position.

Reference images are created through three steps. First, the smartphone gets an input image and a camera angle. Then variation ranges are added to the angle in order to handle the error of sensor information. Finally reference images are created by clipping parts of a panorama image at the angles which include variation ranges. An image similar to a captured image is expected to be in these reference images.

A smartphone captures an input image and angles such as a pan $p_{current}$, a tilt $t_{current}$ and a roll $r_{current}$ every frame. The panorama image P^i that is the nearest to the current angle is selected to create reference images. This is done according to a pan angle $p_{current}$. Rotation angles from the camera coordinate system C^{Ci}_{center} to the current angle is calculated by Eq. 5.

$$(\theta,\phi,\psi) = (p_{current} - p^i_{center}, \\ t_{current} - t^i_{center}, r_{current} - r^i_{center}) \quad (5)$$

The accurate angle is assumed to be around the angle obtained from sensors. Therefore we define three types of variation ranges: Δp as a pan angle, Δt as a tilt angle and Δr as a roll angle. And they are added to Eq. 5 like Eq. 6. Multiple reference images are created based on Eq. 6. We define Q^j and $D = \{Q^1, Q^2, \cdots, Q^j, \cdots, Q^N\}$ as a reference image and a group of reference images, respectively. Note that variable N means the total number of reference images. That is to say, if the number of Δp, Δt and Δr are n_p, n_t and n_r, variable N is $n_p \times n_t \times n_r$.

$$(\theta,\phi,\psi)_j = (p_{current} - p^i_{center} + \Delta p, \\ t_{current} - t^i_{center} + \Delta t, r_{current} - r^i_{center} + \Delta r) \quad (6)$$

To create a reference image Q^j from the panorama image P^i, the coordinates of Q^j's corners in P^i should be computed. First, the coordinates of Q^j in C^{Ci}_{center} are calculated from the camera angle $(\theta,\phi,\psi)_j$. Then the coordinates of Q^j in P^i are calculated by projecting the coordinates in C^{Ci}_{center}. Finally Q^j is clipped from P^i and the homography matrix $\mathbf{H}_{\mathbf{P^i}\to\mathbf{Q^j}}$ from P^i to Q^j is calculated.

The coordinates of the image plane Q^j in C^{Ci}_{center} are calculated by rotating I^{Ci} in figure 10(a). I^{Ci} means an image when the camera is capturing the center of the panorama image P^i. Coordinates which represent four corners of I^{Ci} are described as $a^{Ci} = (-\frac{w}{2}, -\frac{h}{2}, f)$, $b^{Ci} = (\frac{w}{2}, -\frac{h}{2}, f)$, $c^{Ci} = (\frac{w}{2}, \frac{h}{2}, f)$ and $d^{Ci} = (-\frac{w}{2}, \frac{h}{2}, f)$ in C^{Ci}_{center}. Variable w and h mean a width and a height of an input image of the camera.

To rotate I^{Ci}, the vector from the origin of C^{Ci}_{center} to the center point of I^{Ci} should be calculated. This vector is represented as $(0,0,f)$ in C^{Ci}_{center}. Note that variable f means a focal length of the camera. This vector is rotated using θ and ϕ at first (Eq. 7). Then it is used as an axis to rotate four corners by ψ (Eq. 8). A coordinate a'^{Ci}, b'^{Ci}, c'^{Ci} and d'^{Ci} in figure 10(a) denote the coordinates of Q^j in C^{Ci}_{center}.

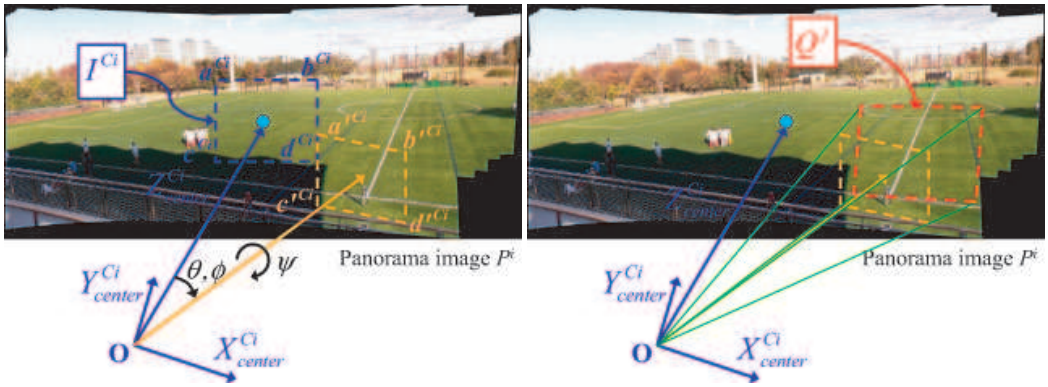
$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ f \end{pmatrix} \quad (7)$$

$$\mathbf{R} = \cos\psi \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (1-\cos\psi) \begin{pmatrix} X'^2 & X'Y' & Z'X' \\ X'Y' & Y'^2 & Y'Z' \\ Z'X' & Y'Z' & Y'^2 \end{pmatrix} \\ + \sin\psi \begin{pmatrix} 0 & -Z' & Y' \\ Z' & 0 & -X' \\ -Y' & X' & 0 \end{pmatrix} \quad (8)$$

Four corners of Q^j in P^i are calculated from these four corners in C^{Ci}_{center} like figure 10(b). For example, if coordinates in C^{Ci}_{center} is (x,y,z), coordinate (u,v) on P^i is calculated using Eq. 9.

$$(u,v) = (x\frac{f}{z}, y\frac{f}{z}) \quad (9)$$

Thus four corners in P^i are obtained, the reference image Q^j is created by clipping this region, and the homography matrix $\mathbf{H}_{\mathbf{P^i}\to\mathbf{Q^j}}$ can be calculated.



(a) Rotating I^{Ci} to Q^j in C^{Ci}_{center} (b) Projecting Q^j in C^{Ci}_{center} to Q^j in P^j

Figure 10: Calculating coordinates of Q^j in P^j.

3.3.3 Image Matching

An accurate homography matrix is estimated using reference images created in section 3.3.2.

As shown in figure 11, a reference image Q^i is linked to a camera angle $(\theta,\phi,\psi)_j$ and the homography matrix $\mathbf{H_{P^i \to Q^j}}$.

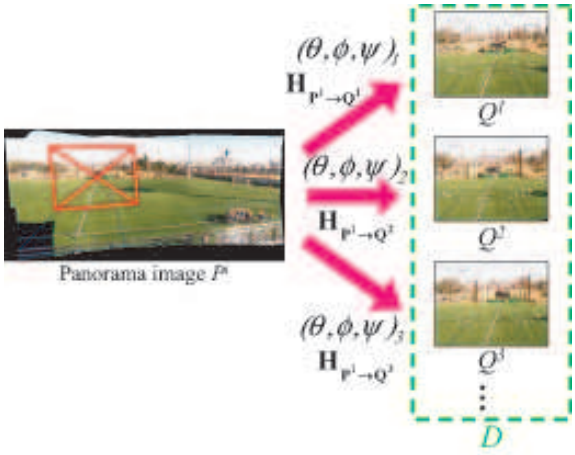


Figure 11: Reference images.

The most similar image Q^g is determined by comparing SSD (Sum of Squared Differences) between an input image and each reference image. $\mathbf{H_{P^i \to Q^g}}$ is same as the correct homography matrix $\mathbf{H_{P^i \to i}}$ from P^i to an input image. Therefore the homography matrix $\mathbf{H_{g \to i}}$ from the sports field to the one in a smartphone captured image is calculated from Eq. 10.

$$\mathbf{H_{g \to i}} = \mathbf{H_{P^i \to i}} \cdot \mathbf{H_{g \to P^i}} \tag{10}$$

4 EXPERIMENTAL RESULTS

We have done two experiments in order to evaluate our proposed system. The accuracies of all processes are evaluated in 4.1. The processing time is measured in section 4.2.

Here is our experiment environment.

- Smartphone
  - OS: Android 2.3
  - CPU: Samsung Exynos 4210 Orion Dual-core 1.2GHz
  - RAM: 1.00 GB
- Server PC
  - OS: Windows 7 Professional 64 bit
  - CPU: Intel Xeon 2.67GHz
  - RAM: 4.00 GB
- Smartphone camera
  - Resolution: 320 240
- Fixed cameras
  - Resolution: 1920 1080
  - Number: 3
- Panorama images
  - Number: 3
- Variation ranges
  - $\Delta p \in \{-4.0,-3.0,\cdots,3.0,4.0\}$
  - $\Delta t \in \{-1.0,0.0,1.0\}$
  - $\Delta r \in \{-2.0,-1.5,\cdots,1.5,2.0\}$
- Databases
  - PostgreSQL

Figure 12 shows the examples of images captured by fixed cameras used in our experiments.

(a) Camera 1 (b) Camera 2 (c) Camera 3

Figure 12: Images captured by fixed cameras.

Figure 13: The results of player detection.

Figure 14: The results of player recognition.

4.1 Accuracy

We have done experiments to evaluate the accuracy of our proposed system. We explain an accuracy of a player analysis in section 4.1.1, an accuracy of a homography estimation of a smartphone in section 4.1.2 and an accuracy of the whole system in section 4.1.3.

4.1.1 Player Analysis

In this section, the accuracy of a player detection and a recognition are evaluated.

To evaluate a player detection, we calculate the MODA (Multiple Object Detection Accuracy) value from Eq. 11 (Kasturi et al., 2009). Variable N^f means the number of frames. n_m^t, n_f^t and n_g^t mean the number of miss detections, false detections and ground truths in frame t. Table 1 shows the MODA values of the result which uses camera 1 and the result which combines camera 1, 2 and 3. Figure 13 shows the result images which combines 3 cameras.

Table 1: MODA.

	Camera 1	Cameras 1-3
MODA	0.33	0.32

$$MODA = \frac{\sum_{t=1}^{N^f}(n_m^t + n_f^t)}{\sum_{i=1}^{N^f} n_g^t} \quad (11)$$

From these results, it seems that the our method can accurately detect players. However the MODA value when 3 cameras were not much different from it when 1 camera was used. We have to consider how to combine the results of multiple cameras.

To evaluate the accuracy of a team recognition, we define a recognition accuracy as a proportion of the number of correct results to the number of detected players. Table 2 shows the recognition accuracies of the result which uses camera 1 and the result which combines camera 1, 2 and 3. Figure 14 shows the result images when 3 cameras are combined.

Table 2: Recognition accuracy.

	Camera 1	Cameras 1-3
Accuracy (%)	77.6	72.6

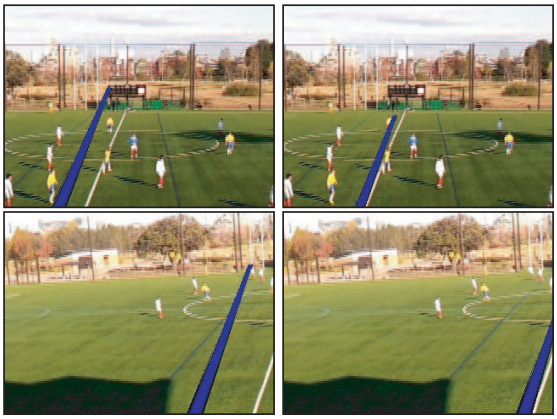
From these results, it seems that the our method can accurately recognize a team.

A player analysis in the current system is a time-independent process hence the correct detection and

the recognition in a certain frame are not taken over to the next frame. The error of player detection occurred by a noise and the error of player recognition occurred by an illumination change are expected to be decreased by tracking players.

4.1.2 Camera Pose Estimation of a Smartphone

We project the center line of the sports field using homography matrix $\mathbf{H_{g\rightarrow i}}$ in section 3.3.3. $\mathbf{H_{g\rightarrow i}}$ is calculated according to two methods, which are using only sensor information and using sensor information and an input image together. Figure 15 shows the result images.



(a) Only sensor (b) Sensor & image

Figure 15: Comparison between two methods.

Additionally we calculated the projection errors. 5 points on the center line are projected onto a smartphone image. These points are compared with ground truth points. We define these distance as projection errors. Table 3 shows an average of the projection errors of 50 frames.

Table 3: Projection error.

	Only sensor	Sensor & image
Projection error (pixel)	19.97	8.69

From these results, it seems that the combination of sensor information and an image is effective to estimate the position where the smartphone is capturing.

4.1.3 Whole System

To evaluate the accuracy of our whole system, we calculate projection errors.

Positions and team names of players are searched from databases and they are projected onto a smartphone image using $\mathbf{H_{g\rightarrow i}}$. There are two teams: "Teikyo0" for players wearing white uniforms and "Teikyo1" for players wearing yellow uniforms. Figure 16 shows the result images.

Projection errors are calculated by comparing projected points with ground truth points.

The average of projection errors is 11.28 pixel. The error of projecting the center line is 8.69 pixel (Table 3). Therefore the error occurred by player analysis is 11.28 - 8.69 = 2.59 pixel. Our system might be more robust by improving a homography estimation because its error is larger than the error of a player analysis.

4.2 Processing Time

In this experiment, we show the processing time. We measured the time to analyze player (1), to register player information to databases (2), to calculate the homography matrix (3) and to search player information from databases (4).

Table 4 shows the average of the processing times.

Table 4: Processing time.

	(1)	(2)	(3)	(4)
Processing time (ms)	122.67	2.05	99.95	1.49

From this result, it seems that our proposed system is fast enough for a real-time application.

5 CONCLUSIONS

We proposed an AR system to support spectators who are watching a sports using smartphones in a spectators' stands.

We have achieved our goal by developing a system which consists of smartphones and fixed cameras. Player information are obtained from fixed cameras. A position where the smartphone is capturing is estimated from a camera and sensors of the smartphone. Because fixed cameras cover the entire field, spectators can easily retrieve information wherever a smartphone camera is directed toward in the field. Our system also could be used to share annotations and comments among spectators.

We have carried out experiments to evaluate the accuracy and the processing time of our proposed system. And we have demonstrated its practicality.

However there are several issues to consider.

- Improve player detection and team recognition.
- Recognize uniform number.
- Track player.

Figure 16: Projecting the teams of players.

- Improve a homography estimation.
- Evaluate our proposed system online.

Our system can be expanded to the other situations such as a concert, a car race and a horse race. To do that, the player recognition should be replaced by recognition of actors, cars and horses. As a future work, we plan to apply our system to the other situations.

ACKNOWLEDGEMENTS

This research is supported by National Institute of Information and Communications Technology, Japan.

REFERENCES

Chen, C.-S., Hsieh, W.-T., and Chen, J.-H. (2004). Panoramic appearance-based recognition of video contents using matching graphs. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):179–199.

Delannay, D., Danhier, N., and Vleeschouwer, C. D. (2009). Detection and recognition of sports(wo)men from multiple views. In *Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*.

Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2008). Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):267–282.

Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., and Zhang, J. (2009). Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):319–336.

Kasuya, N., Kitahara, I., Kameda, Y., and Ohta, Y. (2010). Real-time soccer player tracking method by utilizing shadow regions. In *18th International Conference on Multimedia*.

Khan, S. M. and Shah, M. (2009). A multiview approach to tracking people in crowded scenes using a planar homography constraint. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 31(3):505–519.

Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

Lee, S.-O., Ahn, S. C., Hwang, J.-I., and Kim, H.-G. (2011). A vision-based mobile augmented reality system for baseball games. In *International Conference, Virtual and Mixed Reality*.

Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., Wang, H., Yang, B., Sun, L., and Yang, S. (2009). Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30(2):103–113.

Microsoft-Research (2011). *Image Composite Editor*. http://research.microsoft.com/en-us/um/redmond/groups/ivm/ice/.

Miura, J. and Kubo, H. (2008). Tracking players in highly complex scenes in broadcast soccer video using a constraint satisfaction approach. In *International Conference on Content-based Image and Video Retrieval (CIVR)*.

Miyano, R., Inoue, T., Minagawa, T., Uematsu, Y., and Saito, H. (2012). Camera pose estimation of a smartphone at a field without interest points. In *ACCV Workshop on Intelligent Mobile Vision (IMV)*.

Oe, M., Sato, T., and Yokoya, N. (2005). Estimating camera position and posture by using feature landmark database. In *14th Scandinavian Conference on Image Analysis (SCIA2005)*, pages 171–181.

Shitrit, H. B., Berclaz, J., Fleuret, F., and Fua, P. (2011). Tracking multiple people under global appearance constraints. In *International Conference on Computer Vision (ICCV)*.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11–32.

Tokusho, Y. and Feiner, S. (2009). Prototyping an outdoor mobile augmented reality street view application. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

Tonchidot (2009). *Sekai Camera*. http://sekaicamera.com/.