Camera Pose Estimation of a Smartphone at a Field without Interest Points

Ruiko Miyano, Takuya Inoue, Takuya Minagawa,
Yuko Uematsu and Hideo Saito

Department of Information and Computer Science, Keio University
{rui,inoue,takuya,yu-ko,saito}@hvrl.ics.keio.ac.jp

Abstract. An Augmented Reality (AR) system on mobile phones has recently attracted attention because smartphones have increasingly been popular. For an AR system, we have to know a camera pose of a smartphone. A sensor-based method is one of the most popular ways to estimate the camera pose, but it cannot estimate an accurate pose. A vision-based method is another way to estimate the camera pose, but it is not suitable to a scene with few interest points such as a sports field. In this paper, we propose a novel method of a camera pose estimation for a scene without interest points by combining a sensor-based and a vision-based approach. In our proposed method, we use an acceleration and a magnetic sensor to roughly estimate a camera pose, then search the accurate pose by matching a captured image with a set of reference images. Our experiments show that our proposed method is accurate and fast enough to apply a real-time AR system.

1 Introduction

An Augmented Reality (AR) technology which projects virtual annotations onto a camera image has attracted attention in recent years. Especially an AR system which is using mobile devices has increased. Takacs et al. built an outdoor AR system which makes annotations of building information for mobile phones [1]. Yovcheva et al. explored recent researches in order to develop an AR system for tourism using smartphones [2]. As described in these papers, a system using mobile devices has advantages of being able to utilize devices such as a camera, a sensor, and a GPS. Moreover smartphones are suitable for an AR system used by ordinary people because they have become widespread in these days.

To project annotations onto the right position, a camera pose of a smartphone must be estimated. Some researchers tried to estimate a camera pose of a smartphone for an indoor AR navigation system [3, 4]. They achieved their goal by employing two approaches. The first approach is a sensor-based approach which uses sensors such as an acceleration and a magnetic, and the second approach is a vision-based approach which uses images captured with the camera. A vision-based approach is used to estimate the accurate camera pose by extracting local features. However a vision-based approach cannot be applied to a situation with few local features such as a sports field.

The contribution of this paper is to propose a method that estimates a camera pose of a smartphone at a field without interest points. We have achieved our goal by combining a sensor-based and a vision-based approach which does not use interest points. Our method has been experimented on a soccer field and evaluated regarding the processing time and the accuracy.

2 Related Research

Researches about a camera pose estimation are extremely important to develop an AR system.

Many AR services which use a sensor-based approach are developed in recent days [5–7]. 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 [8]. A sensor-based approach has advantage that a position and a pose of devices can be obtained without complex processing. However there is problem that smartphones are susceptible to noise.

On the other hand, many researchers have focused on a vision-based approach for a robust camera pose estimation. For example, Kato and Billinghurst proposed a marker based camera pose estimation method [9], and Klein and Murray proposed a local feature based camera localization techniques [10]. Klein and Murray also reduced computational cost of a camera pose estimation in order to apply it to a mobile application. They demonstrated an AR system on mobile phones by reducing the number of local features used to estimate a camera pose [11]. These methods sometimes do not work well when a sufficient number of feature points can not be detected to estimate the camera pose. Chen et al. proposed a framework for recognizing scenes using a panorama image [12]. In this research, input image sequences are matched to parts of a panorama image based on template matching. Since this method does not use interest points, we have tried to adopt the idea of using a panorama image in order to a camera pose estimation.

Furthermore Atzori et al. proposed an indoor navigation system based on both approaches [4]. An initial position of a camera is obtained from 2D barcode. A camera position is updated by sensor information. And an accurate pose is estimated by comparing SURF [13] between a current image and reference images. However this system requires barcode and textures with local features.

To address these problems, we propose to combine a sensor-based and a vision-based approach which does not use interest points.

3 Proposed Method

In this paper, we define three coordinate systems: $C^W(X^W, Y^W, Z^W)$ for a target field, $C^C(X^C, Y^C, Z^C)$ for a camera and $C^I(u, v)$ for an image captured by a smartphone (Fig. 1). We assume that a camera is not translated, but only

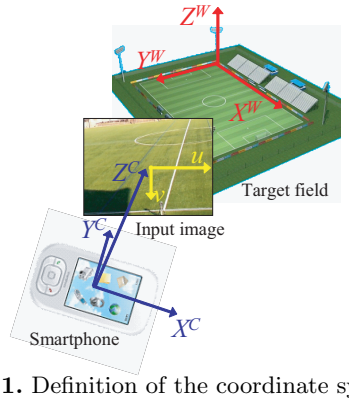


Fig. 1. Definition of the coordinate system

rotated. Because, for example, users in spectators' stands do not change their seat positions. Hence our final goal is to estimate the rotation from C^C to C^W.

Fig. 2 shows the overview of our proposed method. An initial estimate of the camera pose is provided from sensors such as an acceleration and a magnetic. Then the pose is refined by comparing a captured image with reference images created by panorama images like the research of Chen et al. [12].

In preprocessing, two types of information are obtained from a smartphone. The first information are images captured by a smartphone and the second information are camera angles calculated from sensors. Then panorama images are generated to create reference images in online processing.

In online processing, a smartphone captures an image and a camera angle every frame. However this camera angle is not suitable to be directly used because of noise. Therefore an accurate camera angle is refined by a vision-based approach. To do that, a captured image is compared with reference images gen-

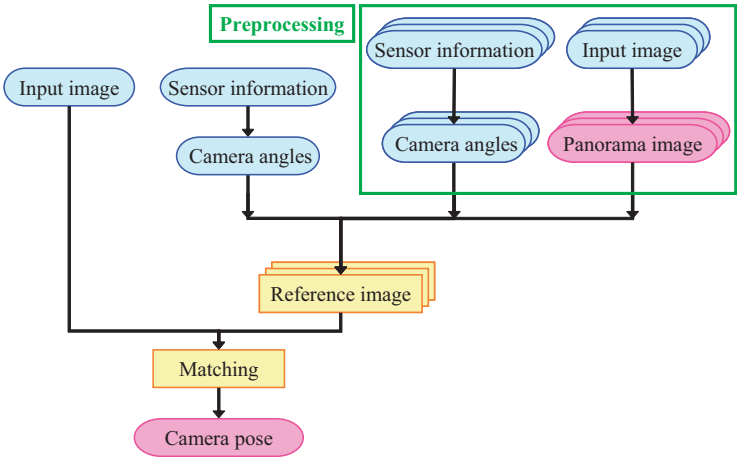
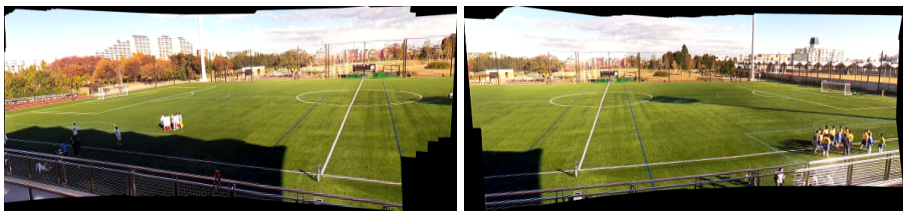


Fig. 2. Overview of the proposed method



(a) Left side (b) Right side

Fig. 3. Panorama images

erated from panorama images and camera angles. Finally an accurate camera pose is obtained from the most similar reference image.

3.1 Preprocessing

In preprocessing, panorama images are generated for image matching in online processing. The smartphone must be moved to capture images and sensor information of the whole target field. Three angles such as a pan, a tilt, and a roll can be obtained from sensor information. Pan, tilt, and roll angles of a camera mean the rotation around Y^C, X^C, and Z^C axes in Fig. 1, respectively.

If one panorama image is generated from all captured images, there will be distortion of the target field in the panorama image. Therefore captured images are classified into several groups according to pan angles. Then panorama images of every groups $G = \{P^1, P^2, \cdots P^i, \cdots\}$ are generated by image mosaicing [14, 15]. Fig. 3 shows examples of panorama images created by Image Composite Editor (ICE) [15].

To know relative relation between a captured image and a panorama image P^i in online processing, a camera coordinate system C^{Ci}_{center} is required (Fig. 4). $C^{Ci}_{center}(X^{Ci}_{center}, Y^{Ci}_{center}, Z^{Ci}_{center})$ is defined so that the camera is pointing to the center of the 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 that have been used to generate P^i: the medium angle between the minimum and the maximum one in these angles.

3.2 Creating Reference Images

In online processing, an accurate camera pose is estimated. Reference images are created from panorama images. Then an accurate camera pose is selected based on image matching between a captured image and reference images. In this section, we explain how to create reference images.

Reference images are created through three steps. First, smartphone gets an image and a camera angle. Second, variation ranges are added to the angle from sensors to handle errors of sensors. 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 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

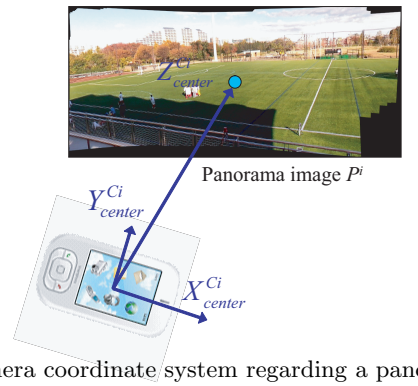


Fig. 4. Camera coordinate system regarding a panorama image

nearest to the current pose is selected according to a pan angle $p_{current}$ in order to create reference images. A rotation angle from camera coordinate system C^{Ci}_{center} to the current pose is calculated by Eq. 1.

$$(\theta, \phi, \psi) = (p_{current} - p^i_{center}, t_{current} - t^i_{center}, r_{current} - r^i_{center}) \quad (1)$$

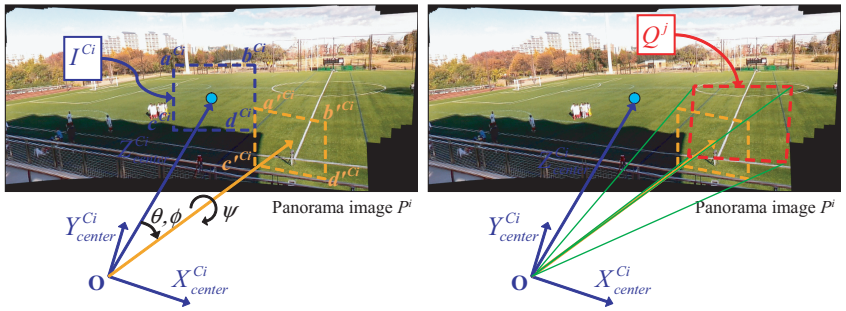
The accurate pose is assumed to be around the pose obtained from sensors. Therefore we define three variation ranges: Δp as a pan angle, Δt as a tilt angle and Δr as a roll angle. And they are added to Eq. 1 like Eq. 2. Multiple reference images are created based on Eq. 2. We define Q^j and $D = \{Q^1,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

$$(\theta, \phi, \psi)_j = (p_{current} - p^i_{center} + \Delta p, t_{current} - t^i_{center} + \Delta t, r_{current} - r^i_{center} + \Delta r) \quad (2)$$

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

The coordinates of the image plane Q^j in C^{Ci}_{center} are calculated by rotating an image plane I^{Ci} in Fig. 5(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 $a^{Ci} = (-\frac{w}{2}, -\frac{h}{2}, f)$, $\boldsymbol{b^{Ci}} = (\frac{w}{2}, -\frac{h}{2}, f)$, $\boldsymbol{c^{Ci}} = (\frac{w}{2}, \frac{h}{2}, f)$ and $\boldsymbol{d^{Ci}} = (-\frac{w}{2}, \frac{h}{2}, f)$ in C^{Ci}_{center}. Variable w and h mean width and height of a captured image.

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}. Variable f means a focal length of the camera. This vector is rotated using θ and ϕ at first (Eq. 3). Then it is used as an axis to rotate four corners by ψ (Eq. 4). A coordinate $\boldsymbol{a'^{Ci}}$, $\boldsymbol{b'^{Ci}}$, $\boldsymbol{c'^{Ci}}$ and $\boldsymbol{d'^{Ci}}$ in Fig. 5(a) denote the coordinates of Q^j in C^{Ci}_{center}.



(a) Rotation axis (b) Reference image

Fig. 5. Rotation about camera coordinate system C^{Ci}_{center}

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ f \end{pmatrix} \tag{3}$$

$$\mathbf{R} = \cos\psi \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (1 - \cos\psi) \begin{pmatrix} X'^2 & X'Y' & Z'X' \\ X'Y' & Y'^2 & Y'Z' \\ Z'X' & Y'Z' & Y'^2 \end{pmatrix} + \sin\psi \begin{pmatrix} 0 & -Z' & Y' \\ Z' & 0 & -X' \\ -Y' & X' & 0 \end{pmatrix} \tag{4}$$

Four corners in P^i can be calculated by projecting four corners in C^{Ci}_{center} like Fig. 5(b). For example, if coordinates in C^{Ci}_{center} is (X, Y, Z), coordinate (x, y) in P^i is calculated using Eq. 5.

$$(x, y) = (X\frac{f}{Z}, Y\frac{f}{Z}) \tag{5}$$

Thus four corners in P^i are obtained, the reference image Q^j can be created by clipping this region.

3.3 Image Matching

The camera pose is estimated using reference images created in section 3.2.

Every reference image Q^j is linked to a camera angle $(\theta, \phi, \psi)_j$ like Fig. 6. This angle is represented by Eq. 2.

A captured image is compared with each reference image using SSD (Sum of Squared Differences). Then the most similar image Q^g which has the highest score of SSD is selected from all reference images. A camera angles $(\theta, \phi, \psi)_g$ corresponding to the image Q^g is the accurate camera pose.

4 Experimental Results

We carried out two experiments to evaluate our proposed method. One is an experiment which measures the processing time for estimating the camera pose

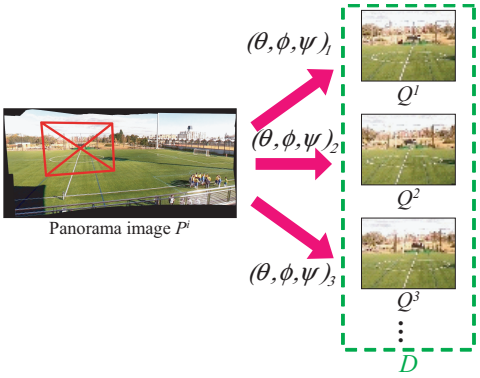


Fig. 6. Reference images

(Section 4.1). The other is an experiment which evaluates the accuracy of a camera pose estimation (Section 4.2).

We developed a camera pose estimation system using a smartphone and a server PC. The smartphone captured both images and sensor information, and the server PC estimated a camera pose of the smartphone. Captured data were transferred via TCP/IP connection over wireless LAN.

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 (pixel): 320 240
- Panorama images
  - Number: 3
  - Resolution (pixel): 1076×485, 1397×560 and 1171×545
- Variation ranges
  - $\Delta p \in \{-4.0, -3.0, \cdots, 3.0, 4.0\}$
  - $\Delta t \in \{-1.0, 0.0, 1.0\}$
  - $\Delta r \in \{-2.0, -1.5, \cdots, 1.5, 2.0\}$

The number of Δp, Δt, and Δr were $n_p = 9$, $n_t = 3$, and $n_r = 9$, respectively. The variable N in section 3.2, the total number of reference images, was $n_p \times n_t \times n_r = 243$.

4.1 Processing Time

In this experiment, we show the processing time to estimate the camera pose. The processing time includes time to select a panorama image, to create reference images, to calculate SSD and to obtain the camera pose of the smartphone. Table 1 shows an average and a variance of the processing time using 1307 frames.

Table 1. Processing time

	Average	Variance
Processing Time (ms)	127.4	1.8

From this result, it seems that our proposed method can be used in a real-time processing.

4.2 Accuracy

In this experiment, we projected a center line of a soccer field onto a smartphone image to evaluate the accuracy of our proposed method.

The homography matrix between a soccer field (X^W, Y^W) and a smartphone image (u, v) is computed from a camera pose of a smartphone. Therefore we evaluate this matrix in order to evaluate the camera pose. For the accuracy evaluation of this matrix, the center line is projected and the projection error is calculated.

To project the center line onto a smartphone image, a homography matrix $\mathbf{H_{t\to i}}$ between the soccer field and a smartphone image is required. The homography matrix between a panorama image and each reference image can be calculated in section 3.2. Then the homography matrix $\mathbf{H_{P^i\to i}}$ between a panorama image and a smartphone image is selected in section 3.3. The homography matrix

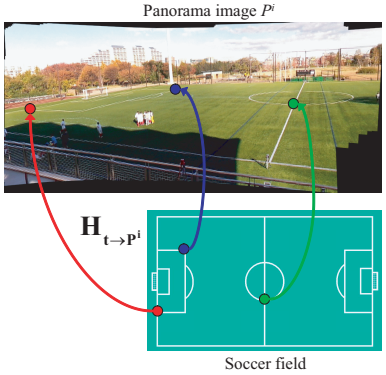


Fig. 7. Corresponding points between panorama image and soccer field

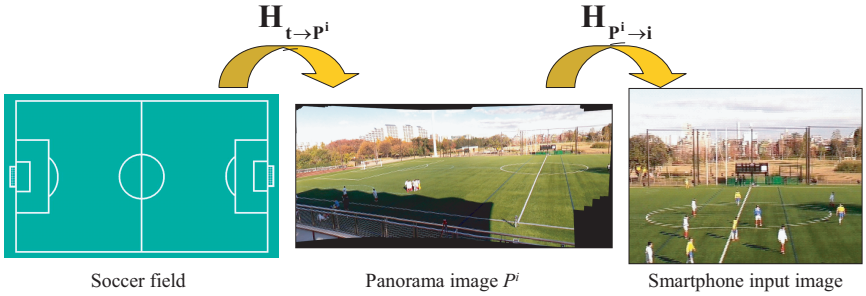


Fig. 8. Homography matrices that relates smartphone image and soccer field

$\mathbf{H}_{\mathbf{t}\to\mathbf{P}^i}$ between the soccer field and a panorama image is calculated in preprocessing by manually inputting corresponding points like Fig. 7. Therefore $\mathbf{H}_{\mathbf{t}\to\mathbf{i}}$ can be calculated by Eq. 6 like Fig. 8.

$$\mathbf{H}_{\mathbf{t}\to\mathbf{i}} = \mathbf{H}_{\mathbf{p}\to\mathbf{i}} \cdot \mathbf{H}_{\mathbf{t}\to\mathbf{p}} \tag{6}$$

We project the center line using homography matrix $\mathbf{H}_{\mathbf{t}\to\mathbf{i}}$. Fig. 9 shows result images. From these results, it seems our proposed method is efficient no matter where the camera is capturing.

We also calculated the projection error to compare two methods, which are using only sensor information and using both sensor information and a captured image. 5 points on the center line are projected onto a smartphone image. These points are manually compared with ground truth points. We define these distance as projection errors. Table 2 shows an average of the projection errors of 50 frames. Fig. 10 shows comparison of result images between two methods. It seems that the combination of sensor information and an image is effective to estimate the camera pose. It costs about 10 fps to project the center line to a smartphone image via TCP/IP connection, which is enough to apply to a real-time processing.

Fig. 9. Result images

Table 2. Projection error

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

(a) Only sensor (b) Sensor & image (c) Only sensor (d) Sensor & image

Fig. 10. Comparison between two methods

5 Conclusion and Future Work

We proposed a camera pose estimation method for a smartphone without interest points.

We achieved our goal by combining a sensor-based and a vision-based approach which does not use interest points. A rough camera pose is estimated using sensors, then an accurate camera pose is calculated by matching a captured image with a set of reference images.

Two experiments were carried out to validate our proposed method regarding the processing time and the accuracy. We confirmed that our proposed method can accurately estimate the camera pose and it is fast enough to apply a real-time AR system.

However our method is effective only when the smartphone is not translated, but only rotated. As future works, we expand our proposed method to deal with any camera motion using the pedometer.

Moreover we plan to develop a mobile AR system using our proposed method, for example an AR system for watching sports like Fig. 11. There are many researches about sports player detection, recognition and tracking [16–18]. Players' positions and identities can be analyzed using these researches. Annotations of players are projected by the homography matrix between a smartphone image and the sports field. Thus we plan to apply our method to a sports AR system.

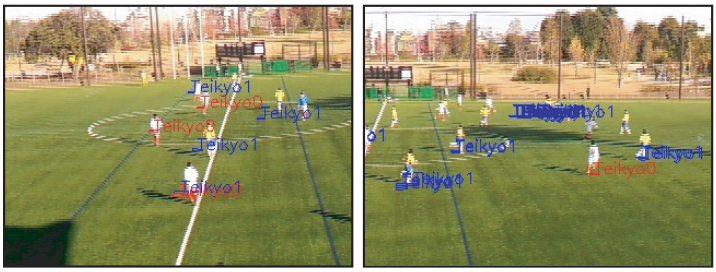


Fig. 11. Examples of a sports AR system

Acknowledgement

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

References

1. Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.C., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., Girod, B.: Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In: ACM International Conference on Multimedia Retrieval (ICMR). (2008)
2. Yovcheva, Z., Buhalis, D., Gatzidis, C.: Overview of smartphone augmented reality applications for tourism. e-Review of Tourism Research (eRTR) **10** (2012) 63–66
3. Kang, J.: Technique of tangible user interfaces for smartphone. In: International Conference on Information and Computer Applications (ICICA). (2012)
4. Atzori, L., Dessi, T., Popescu, V.: Indoor navigation system using image and sensor data processing on a smartphone. In: International Conference on Optimization of Electrical and Electronic Equipment (OPTIM). (2012)
5. Tonchidot: Sekai camera. http://sekaicamera.com/ (2009)
6. Layar: Layar. http://www.layar.com/ (2009)
7. mTrip: mtrip. http://www.mtrip.com/ (2009)
8. Tokusho, Y., Feiner, S.: Prototyping an outdoor mobile augmented reality street view application. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR). (2009)
9. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: International Workshop on Augmented Reality (IWAR). (1999)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR). (2007)
11. Klein, G., Murray, D.: Parallel tracking and mapping on a camera phone. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR). (2009)
12. Chen, C.S., Hsieh, W.T., Chen, J.H.: Panoramic appearance-based recognition of video contents using matching graphs. IEEE Transactions on Systems, Man, and Cybernetics **34** (2004) 179–199
13. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. Computer Vision and Image Understanding (CVIU) **110** (2008) 346–359
14. Levin, A., Zomet, A., Peleg, S., Weiss, Y.: Seamless image stitching in the gradient domain. In: European Conference on Computer Vision (ECCV). (2003)
15. Microsoft-Research: Image composite editor. http://research.microsoft.com/en-us/um/redmond/groups/ivm/ice/ (2011)
16. Delannay, D., Danhier, N., Vleeschouwer, C.D.: Detection and recognition of sports(wo)men from multiple views. In: Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC). (2009)
17. Kasuya, N., Kitahara, I., Kameda, Y., Ohta, Y.: Real-time soccer player tracking method by utilizing shadow regions. In: 18th International Conference on Multimedia. (2010)
18. Shitrit, H.B., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple people under global appearance constraints. In: International Conference on Computer Vision (ICCV). (2011)