25

Vision-based Augmented Reality Applications

Yuko Uematsu and Hideo Saito
Keio University
Japan

1. Introduction

Augmented Reality (AR) is a technique for overlaying virtual objects onto the real world. AR has recently been applied to many kinds of entertainment applications by using visionbased tracking technique, such as [Klein & Drummond, 2004, Henrysson et al., 2005, Haller et al. 2005, Schmalstieg & Wagner, 2007, Looser et al. 2007]. AR can provide users with immersive feeling by allowing the interaction between the real and virtual world.

In the AR entertainment applications, virtual objects (world) generated with Computer Graphics are overlaid onto the real world. This means that the real 3D world is captured by a camera, and then the virtual objects are superimposed onto the captured images. By seeing the real world through some sort of displays, the users find that the virtual world is mixed with the real world. In such AR applications, the users carry a camera and move around the real world in order to change their view points. Therefore the pose and the position of the moving user's camera should be obtained so that the virtual objects can be overlaid at correct position in the real world according to the camera motion. Such camera tracking should also be performed in real-time for interactive operations of the AR applications.

Vision-based camera tracking for AR is one of the popular research areas because the visionbased method does not require any special device except cameras, in contrast with sensorbased approaches. And also, marker-based approach is a quite easy solution to make the vision-based tracking robust and running in real-time.
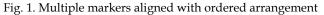
This chapter focuses on marker-based approach. Especially, "AR-Toolkit" [H. Kato & M. Billinghurst, 1999] is a very popular tool for implementing simple on-line AR applications. ARToolkit is a kind of planar square marker for the camera tracking and estimates the camera position and pose with respect to the marker. By using the camera position and pose, virtual objects are overlaid onto the images as if the objects exist in the real world where the marker is placed. Since the user only has to place the marker, this kind of marker-base registration is very easy to implement AR systems. If only one marker is utilized, however, the camera's movable area is limited so that the camera (user) can see the marker. Moreover, when the marker cannot be recognized properly because of a change in its visibility, the registration of the virtual objects is getting unstable. In order to solve such problems, using multiple markers is a popular way.

When multiple markers are utilized in a vision-based method, it is necessary to know the geometrical arrangement information of the marker such as their position and pose in advance. For example, the method in [Umlauf et al., 2002] requires the position and pose of a square marker. The method in [Kato et al., 2000] also needs the position of a point marker in advance. In [Genc et al., 2002], they proposed two-step approach; learning process and

marker-less registration method. In the learning process, the geometrical information of the markers is required for learning the markers. As shown in these methods, the task of measuring the marker arrangement information in advance is necessary for AR applications using a vision-based registration.

In most cases, multiple markers are usually aligned with ordered arrangement as shown in Fig. 1 because the users manually measure the geometrical arrangement of the markers. However such a measuring task is very time-consuming. Moreover, if the markers can not be distributed on the same plane, manual measuring becomes more difficult. Kotake et al. proposed a marker-calibration method which is a hybrid method combining the bundle adjustment method with some constraints on the marker arrangement obtained a priori (e.g. the multiple markers are located on a single plane) [Kotake et al., 2004]. Although a precise measurement of the markers is not required, a priori knowledge of the markers is necessary.

Fig. 1. Multiple markers aligned with ordered arrangement

In this chapter, a vision-based registration method using multiple planar markers placed at arbitrary positions and poses is introduced [Uematsu & Saito, 2005-b]. This method is extended from the method using multiple planar structures in the real world without geometrical arrangement information [Uematsu & Saito, 2005-a].

In the previous method using multiple markers, a projection matrix from the marker to the input image is computed from each marker and then all the matrices are merged by using the marker arrangement information. However, such prior knowledge about the arrangement is not utilized in this method. In order to merge the projection matrices computed from the markers, this method introduces "Projective Space" which is defined by projective reconstruction of two reference images. Since the Projective Space is defined by 2D coordinate systems of the images, the coordinate system of the Projective Space is independent of every marker's coordinate system. Therefore by estimating the marker arrangement through the Projective Space, the projection matrices computed from the marker are merged. This method allows the markers to be distributed at arbitrary positions and pose, since the marker arrangement can be estimated by captured images.

This chapter also introduces two AR applications for entertainment; AR Baseball Presentation System [Uematsu & Saito, 2006]; Interactive AR Bowling System [Uematsu & Saito, 2007]. These applications are used on the tabletop with a handheld monitor and a web-camera connected to a general PC. Since any special device is not required such as high-speed cameras, high-performance PC or physical sensors, these applications are available for home users.

In Section 2, the registration method using multiple planar markers with Projective Space is explained. In Section 3 and Section 4, AR Baseball Presentation System and Interactive AR Bowling System are introduced.

2. Multiple markers based AR via 3D projective space

2.1 Definition of coordinate systems

In this section, three kinds of coordinate systems and transformation matrices used in this method are explained. In the typical registration method, two coordinate systems are utilized, which represent the real world and the input image, respectively. On the other hand, this method uses three kinds of coordinate systems; marker's coordinate system defined for each marker, Projective Space and the input image as shown in Fig. 2.
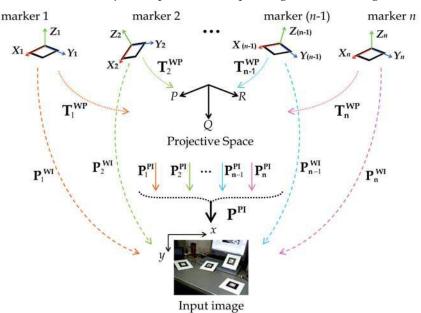

Fig. 2. Three kinds of coordinate systems in this method

$(X_i\text{-}Y_i\text{-}Z_i)$ is a 3D coordinate system independently assigned to each marker plane i. $(x\text{-}y)$ is a 2D coordinate system of the input image. Since the multiple markers are distributed in arbitrary positions and poses in the real world, the relationship among $(X_i\text{-}Y_i\text{-}Z_i)$ is unknown. In order to estimate the relationship, we introduce "Projective Space" ($P\text{-}Q\text{-}R$) which is a kind of 3D non-Euclidean coordinate system defined by projective reconstruction [Hartley & Zisserman, 2000] of two images called "reference images". The reference images are captured from two different viewpoints.

As for transformation matrices, a transformation matrix $\mathbf{T}_i^{WP}$ which relates each marker i to the Projective Space is computed by corresponding points between each marker's coordinate system and the Projective Space. This matrix indirectly represents the geometrical relationship of the markers. A projection matrix $\mathbf{P}_i^{WI}$ which relates each marker i to the input image represents the camera's position and pose with respect to the marker. Therefore, it can be computed by marker tracking algorithm [H. Kato & M. Billinghurst, 1999] at every frame. Then a projection matrix $\mathbf{P}_i^{PI}$ which relates the Projective Space to the input image can be written as following equation.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \mathbf{P}_i^{\mathbf{WI}} \left(\mathbf{T}_i^{\mathbf{WP}} \right)^{-1} \begin{bmatrix} P \\ Q \\ R \\ 1 \end{bmatrix} \cong \mathbf{P}_i^{\mathbf{PI}} \begin{bmatrix} P \\ Q \\ R \\ 1 \end{bmatrix} \qquad (1)$$

By using these coordinate systems, the geometrical relationship of the multiple markers is estimated, so that virtual objects can be overlaid onto the input image even though specific marker is not continuously detected in the input image sequence.

2.2 Registration algorithm of virtual objects with 3D projective space

For registration of virtual objects onto the real world, the virtual objects need to be defined in the 3D coordinate system of the real world. By computing the projection matrix from the real world to the input image at every frame, then, the virtual objects are projected onto the input image. In this way, the virtual objects are overlaid onto the real world.

In this method, the 3D coordinate system representing the real world corresponds to a marker's coordinate system. Therefore we select one marker plane as a base plane to define the virtual objects in the base marker's coordinate system. Then a projection matrix which relates the base marker to the input image has to be computed at every frame for overlaying the virtual objects.
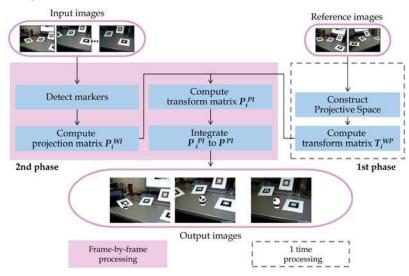

Fig. 3. Flowchart of this method

Fig. 3 shows the flowchart of this method. This method is divided into two phases. At the first phase, the projective space is constructed by two reference images. Then the transformation matrix $\mathbf{T}_i^{\mathbf{WP}}$ of each marker i is computed by using corresponding points between each marker's coordinate system and the Projective Space. The first phase is performed just one time.

At the second phase, the projection matrix $\mathbf{P}_i^{\mathbf{WI}}$ of each marker i is computed at every frame. Then the projection matrix $\mathbf{P}_i^{\mathbf{PI}}$ for each marker is computed by eq. (1). Since all the matrices

$\mathbf{P}_i^{PI}$ represent the relationship between the Projective Space and the input image, all $\mathbf{P}_i^{PI}$ should coincide with each other and can be easily merged into one matrix in order to include the camera's position and pose from all the markers. However each $\mathbf{P}_i^{PI}$ may include computation errors. Therefore these matrices are merged into one projection matrix $\mathbf{P}^{PI}$ by least-square-method. By using the merged matrix $\mathbf{P}^{PI}$ and the transformation matrix $\mathbf{T}_i^{WP}$ of the base marker plane, the projection matrix from the base marker, in which the virtual objects are defined, to the input image is computed as following equation.

$$\mathbf{P}_{base} = \mathbf{P}^{PI}\mathbf{T}_{base}^{WP} \tag{2}$$

Finally, the virtual objects are overlaid onto the input image by $\mathbf{P}_{base}$. This matrix $\mathbf{P}_{base}$ includes the information from all the markers detected in the input image. Therefore as long as one marker is detected in the input image, the virtual objects can be overlaid onto the input image.

2.3 Constructing 3D projective space

3D Projective Space is used to estimate the geometrical relationship among the multiple planar markers which are distributed in arbitrary positions and poses. The Projective Space is defined by projective reconstruction of two reference images.

As shown in Fig. 6, the object scene is captured from two viewpoints. These captured images are called the reference image A and B, respectively. By using 2D coordinates (u_A, v_A) and (u_B, v_B) on the reference images, a 3D coordinate system (P-Q-R) is defined so that it has the following projective relationships with each of the reference images. This 3D coordinate system is called 3D Projective Space.

$$[u_A,v_A,1]^\mathsf{T} \cong \mathbf{P}_A[P,Q,R,1]^\mathsf{T}, \qquad [u_B,v_B,1]^\mathsf{T} \cong \mathbf{P}_B[P,Q,R,1]^\mathsf{T} \tag{3}$$

$$\mathbf{P}_A = [\mathbf{I}\,|\,\mathbf{0}]; \qquad \mathbf{P}_B = \left[-\frac{[\mathbf{e}_B]_\times \mathbf{F}_{AB}}{\|\mathbf{e}_B\|^2}\,|\,\mathbf{e}_B \right] \tag{4}$$
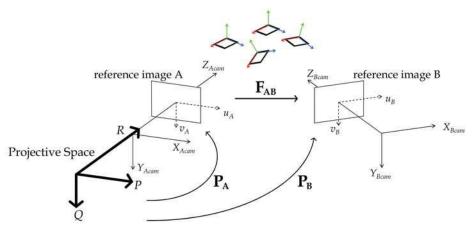

Fig. 4. Projective Space defined by two reference images

$\mathbf{F_{AB}}$ is a fundamental matrix from the image A to image B as shown in Fig. 4. $\mathbf{e_B}$ is an epipole on the image B, and $[\mathbf{e_B}]_\times$ is the skew-symmetric matrix of $\mathbf{e_B}$ [Hartley & Zisserman, 2000]. In this method, $\mathbf{F_{AB}}$ and $\mathbf{e_B}$ are computed by eight or more corresponding points between the reference images, which are detected by each marker. $\mathbf{P_A}$ and $\mathbf{P_B}$ are defined by eq. (4). Then 3D coordinates (P-Q-R) are computed by $\mathbf{P_A}$, $\mathbf{P_B}$ and 2D coordinates in the reference images when computing $\mathbf{T}_i^{\mathbf{WP}}$. The detail is described in the next section.

2.4 Computing $\mathbf{T}_i^{\mathbf{WP}}$

Since both of the marker's coordinate system (X_i-Y_i-Z_i) and the Projective Space (P-Q-R) are 3D coordinate systems, the transformation matrix $\mathbf{T}_i^{\mathbf{WP}}$ is a 4×4 matrix, which can be computed from five or more corresponding points between (X_i-Y_i-Z_i) and (P-Q-R).

We assume that a 3D point (X_i-Y_i-Z_i) is projected onto the reference image A and B as a 2D points (u_A, v_A) and (u_B, v_B), respectively. When these 2D points are projected into the Projective Space as a 3D point (P-Q-R), the relationship is as follows,

$$\begin{bmatrix} \mathbf{p_A^1} - u_A \mathbf{p_A^3} \\ \mathbf{p_A^2} - v_A \mathbf{p_A^3} \\ \mathbf{p_B^1} - u_B \mathbf{p_B^3} \\ \mathbf{p_B^2} - v_B \mathbf{p_B^3} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \\ 1 \end{bmatrix} = 0 \tag{5}$$

where, $\mathbf{p_A^k}$ and $\mathbf{p_B^k}$ are the kth row vectors of $\mathbf{P_A}$ and $\mathbf{P_B}$ in eq. (4). Then (P-Q-R) is computed by Singular Value Decomposition of the 4×4 matrix on the left-hand side of eq. (5). Therefore, the corresponding points (X_i-Y_i-Z_i) and (P-Q-R) are obtained through the 2D points (u_A, v_A) and (u_B, v_B) on the reference images.

As described before, five or more corresponding points are required to compute $\mathbf{T}_i^{\mathbf{WP}}$. In order to obtain the corresponding points, a cube is drawn on each marker in the reference image A and B as shown in Fig. 5. Since the size of the cube is already known, 3D coordinates (X_i-Y_i-Z_i) of the cube's vertices are known. By using (u_A, v_A) and (u_B, v_B) which are 2D coordinates of the vertices of the cube in the reference image A and B, eight corresponding 3D coordinates (P-Q-R) are computed by eq. (5).

Fig. 5. Projected cubes on the reference images to obtain corresponding points

After obtaining the corresponding points in the coordinate system of each marker i, $\mathbf{T}_i^{\mathbf{WP}}$ is computed. When corresponding points between (X_i-Y_i-Z_i) in the coordinate system of the marker i and (P-Q-R) are obtained, the relationship between each marker i and the Projective Space is expressed by the following equation.

$$[X_{i,j}, Y_{i,j}, Z_{i,j}, 1]^{\mathsf{T}} \cong \mathbf{T}_i^{\mathbf{WP}}[P_j, Q_j, R_j, 1]^{\mathsf{T}} \quad (j > 0) \tag{6}$$

$$\mathbf{T}_i^{\mathbf{WP}} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & 1 \end{bmatrix} \tag{7}$$

By the following equation, the elements of $\mathbf{T}_i^{\mathbf{WP}}$ are obtained.

$$\mathbf{M}_j \mathbf{t} = \mathbf{b}_j \tag{8}$$

where,

$$\mathbf{M}_j = \begin{bmatrix} X_{i,j} & Y_{i,j} & Z_{i,j} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -X_{i,j}P_j & -Y_{i,j}P_j \\ 0 & 0 & 0 & 0 & X_{i,j} & Y_{i,j} & Z_{i,j} & 1 & 0 & 0 & 0 & 0 & -X_{i,j}Q_j & -Y_{i,j}Q_j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_{i,j} & Y_{i,j} & Z_{i,j} & 1 & -X_{i,j}R_j & -Y_{i,j}R_j \end{bmatrix} \tag{9}$$

$$\mathbf{t} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{41} & t_{42} & t_{43} \end{bmatrix}^{\mathsf{T}}$$
$$\mathbf{b}_j = \begin{bmatrix} P_j & Q_j & R_j \end{bmatrix}^{\mathsf{T}}$$

If m corresponding points are obtained, following equation is derived by eq. (8).

$$\begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_m \end{bmatrix} \mathbf{t} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix} \tag{10}$$

Then $\mathbf{t}$ (the elements of $\mathbf{T}_i^{\mathbf{WP}}$) is obtained by least-square-method. In this method, the corresponding points are obtained by the cube's vertices. Therefore $m=8$.

2.5 Computing $\mathbf{P}_i^{\mathbf{PI}}$

As described in Sec. 2.1, $\mathbf{P}_i^{\mathbf{PI}}$ is the projection matrix which relates the Projective Space to the input image and is computed by $\mathbf{T}_i^{\mathbf{WP}}$ and $\mathbf{P}_i^{\mathbf{WI}}$. $\mathbf{T}_i^{\mathbf{WP}}$ is computed in the first phase just one time. $\mathbf{P}_i^{\mathbf{WI}}$ is computed by marker tracking at every frame in the second phase. As shown in eq. (1), $\mathbf{P}_i^{\mathbf{PI}}$ is computed as follows.

$$\mathbf{P}_i^{\mathbf{PI}} = \mathbf{P}_i^{\mathbf{WI}} \cdot \left(\mathbf{T}_i^{\mathbf{WP}} \right)^{-1} \tag{11}$$

2.6 Merging $\mathbf{P}_i^{\mathbf{PI}}$ into $\mathbf{P}^{\mathbf{PI}}$

As described in Sec. 2.2, all $\mathbf{P}_i^{\mathbf{PI}}$ should coincide with each other because they represent a common geometrical projection between the Projective Space and the input image. Therefore, all $\mathbf{P}_i^{\mathbf{PI}}$ are merged into one matrix $\mathbf{P}^{\mathbf{PI}}$. The details are as follows.

When $\mathbf{P}_i^{PI}$ is computed from each marker i, corresponding points between (P-Q-R) in the Projective Space and (x-y) in the input image can be obtained by following equation.

$$[x, y, 1]^\top \cong \mathbf{P}_i^{PI}[P, Q, R, 1]^\top \tag{12}$$

In this method, by using (P_j-Q_j-R_j) obtained in computing $\mathbf{T}_i^{WP}$, corresponding points (x_j-y_j) in the input image is computed by eq. (12). Therefore m corresponding points are obtained for each marker.

If n markers are detected in the input image, the relationship between the corresponding points is as follows.

$$\begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_n \end{bmatrix} \mathbf{p} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} \tag{13}$$

where,

$$\mathbf{p} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & \cdots & p_{33} \end{bmatrix}^\top \quad \mathbf{P}^{PI} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & 1 \end{bmatrix} \tag{14}$$

$$\mathbf{M}_i = \begin{bmatrix} P_1 & Q_1 & R_1 & 1 & 0 & 0 & 0 & 0 & -P_1 x_1 & -Q_1 x_1 & -R_1 x_1 \\ 0 & 0 & 0 & 0 & P_1 & Q_1 & R_1 & 1 & -P_1 y_1 & -Q_1 y_1 & -R_1 y_1 \\ \vdots & & & & & & & & & & \\ P_m & Q_m & R_m & 1 & 0 & 0 & 0 & 0 & -P_m x_m & -Q_m x_m & -R_m x_m \\ 0 & 0 & 0 & 0 & P_m & Q_m & R_m & 1 & -P_m y_m & -Q_m y_m & -R_m y_m \end{bmatrix} \tag{15}$$

$$\mathbf{b}_i = \begin{bmatrix} x_1, & y_1, & \cdots & x_m, & y_m \end{bmatrix}^\top \tag{16}$$

Then $\mathbf{p}$ (the elements of $\mathbf{P}^{PI}$) is obtained by least-square-method.

By using this merged projection matrix $\mathbf{P}^{PI}$, the projection matrix from the base marker's coordinate system, in which the virtual objects are defined, to the input image is computed by eq. (2). Then, the virtual objects are overlaid onto the input image by $\mathbf{P}_{base}$.

2.7 Automatic selection of reference images

As described in Sec. 2.3, the Projective Space is defined by projective reconstruction of two reference images. In projective reconstruction, a fundamental matrix between the two reference images is used for defining the Projective Space. This means that it is very important issue to compute an accurate fundamental matrix between the reference images. In this method, therefore, a reasonable pair of images as the reference images should be selected.

This method introduces automatic selection algorithm of the reference images. The overview is shown in Fig. 6. First, the object scene where multiple markers are distributed is captured for a few seconds by a moving camera. This image sequence in which all the

markers should be included becomes the candidates of the reference image. From the candidate images, a pair of images is sequentially selected as temporal reference image A and B. Then a good pair is decided by evaluating the accuracy of the Projective Space which defined by the temporal reference images.
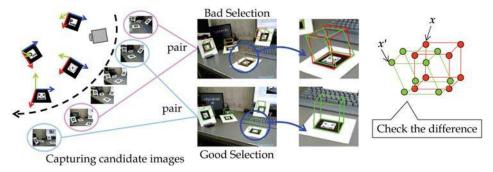

Fig. 6. Overview of automatic selection of reference images

For evaluating the temporal reference images, first, a fundamental matrix is obtained by projection matrices computed from every marker in the temporal reference images using ARToolkit algorithm [H. Kato & M. Billinghurst, 1999]. Where $\mathbf{P_{Ai}}$ and $\mathbf{P_{Bi}}$ are the projection matrix computed from marker i in the temporal reference image A and B, respectively. Then a fundamental matrix based on marker i is computed as following equation.

$$\mathbf{F_{ABi}} = \left[\mathbf{e_{Bi}}\right]_\times \mathbf{P_{Bi}} \mathbf{P_{Ai}^-} \tag{17}$$

where $\mathbf{P_{Ai}^-}$ represents the pseudo inverse matrix of $\mathbf{P_{Ai}}$ [Hartley & Zisserman, 2000]. By computing $\mathbf{F_{ABi}}$ based on every marker, one matrix which has the smallest projection error *error* is selected as FAB which is the fundamental matrix between the temporal reference images.

$$error = \begin{bmatrix} u_B & v_B & 1 \end{bmatrix} \mathbf{F_{ABi}} \begin{bmatrix} u_A, & v_A, & 1 \end{bmatrix}^\top \tag{18}$$

where, (u_A, v_A) and (u_B, v_B) are corresponding points in the temporal reference image A and B, respectively. In the same way as computing $\mathbf{T_i^{WP}}$ in Sec. 2.4, these corresponding points are obtained by vertices of a cube drawn on each marker as shown in Fig. 5.

After selecting the best fundamental matrix as $\mathbf{F_{AB}}$, a Projective Space is defined by eq. (4). using $\mathbf{F_{AB}}$. $\mathbf{T_i^{WP}}$ and $\mathbf{P_i^{PI}}$ are also computed and merged into $\mathbf{P^{PI}}$. Then two projected coordinates (x_i, y_i) and (x'_i, y'_i) are compared as follows,

$$[x_i, y_i, 1]^\top \cong \mathbf{P_{Bi}}[X, Y, Z, 1]^\top , \qquad [x'_i, y'_i, 1]^\top \cong \left(\mathbf{P^{PI}T_i^{WP}}\right)[X, Y, Z, 1]^\top \tag{19}$$

(x_i, y_i) and (x'_i, y'_i) are corresponding to a red cube and a green cube in Fig. 6, respectively. Although these cubes should coincide with each other, these matrices in eq. may include computational error of computing $\mathbf{F_{AB}}$. Therefore, the temporal pair of images which has smaller difference between two cubes than threshold value is selected as a good pair of reference images. If no temporal pair of images has smaller error than threshold value, the candidate image sequence is captured again. In the experiment described in the next section, threshold value is defined as 3 pixels.

2.8 Demonstrations

In this section, registration of a virtual object is demonstrated by using our method. In the real world, 4 planar markers distributed at arbitrary positions and poses as shown in Fig. 7. Of course, the relationship between the markers is unknown. This method is implemented by a PC (OS: Windows XP, CPU: Intel Pentium IV 3.6GHz) and a web camera (ELECOM UCAM-E130HWH). The captured image's resolution is 640x480 pixels and graphical views of virtual objects are rendered using OpenGL.

Fig. 7. Object scenes where multiple markers are distributed at arbitrary positions and poses

First, a user captures the object scene for 100 frames and waits about 1 minute for the automatic selection of the reference images from the captured sequence. After that, a virtual object is overlaid onto the input images as if the virtual object exists in the real world. The user can move the camera around the object scene for watching the virtual object from favorite view point.

The resulting images are shown in Fig. 8~11. In Fig. 8, a virtual object is overlaid onto one position in the real world. Even though specific marker is not continuously detected in all over the input image sequence, the virtual object is stably overlaid onto the same position. This result shows that the relationship of the markers can be estimated successfully. And also, the moving camera's position and pose with respect to the virtual object can be computed by introducing the detected markers via the Projective Space. Therefore, the registration of virtual object is achieved without prior knowledge about the geometrical relationship of the markers.

Fig. 8. Static object is overlaid on the real world.

In Fig. 9, the virtual object is walking on the tabletop. The user's camera moves according to the virtual object. Therefore, it is impossible that specific marker is always detected in the input image. Moreover, a planar marker is not detected depending on the camera's angle with respect to the marker plane. In Fig. 10, even though the marker indicated by the red circle is
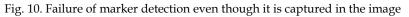
captured in the image, it cannot be detected. Therefore, if all the markers are placed in the same plane, no marker is detected depending on the camera's angle. In that case, it fails in the registration. On the other hand, since this method allows the markers to be distributed at arbitrary positions and poses, the registration of the virtual object can be continued.

Fig. 9. Moving object is overlaid on the real world.

Fig. 10. Failure of marker detection even though it is captured in the image

In Fig. 11, the human-sized virtual object is overlaid onto the large space instead of the tabletop. For applying multiple markers based registration in such a large space, it becomes more difficult to manually measure the arrangement of the markers. Therefore this method is useful because of estimating the arrangement only by capturing the object scene.

Fig. 11. Moving object as large as a human is overlaid on the real world.

3. AR baseball presentation system

AR Baseball Presentation System is an observation system of a virtual baseball game. The overview of the system is shown in Fig. 12. Users place a real baseball field model on the tabletop and input a baseball game history (scorebook) that they want to watch into the system. Then they can watch the game by replaying with 3D virtual baseball players on the field model in front of them. On the field model, 2D markers are placed for registration of the virtual players. Therefore the users can watch the game from their favorite viewpoints around the field.

This sytem visually replays the baseball game which was previously played in the other place. In contrast with the usual way to know the game history, such as watching the captured video or reading the recorded scorebook, this AR system can provide the users with much realistic sensation as an entertainment application.
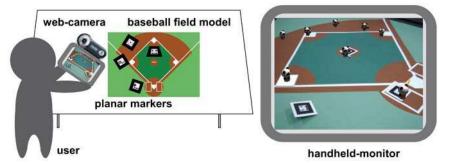
This sytem visually replays the baseball game which was previously played in the other place. In contrast with the usual way to know the game history, such as watching the captured video or reading the recorded scorebook, this AR system can provide the users with much realistic sensation as an entertainment application.

Fig. 12. AR Baseball Presentation System

3.1 Input scorebook data file

This system visually replays the baseball game which was previously played in the other place by a scorebook data, in which the game history they want to know is described. The input data file is called "Scorebook Data File" (SDF).

As shown in Fig. 13, the game history is described play-by-play in the SDF. "1 play" means the actions of the players and the ball from the time the pitcher throws the ball until the ball
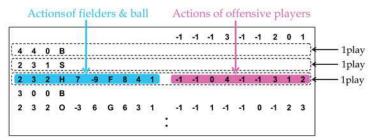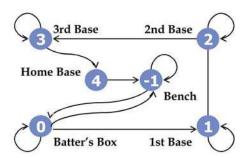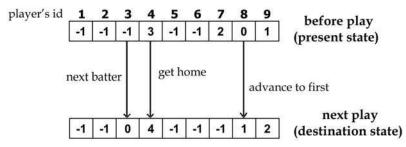

Fig. 13. Scorebook Data File (SDF)

returns to the pitcher again. It is about for 15 to 30 seconds. The actions of the players and the ball in 1 play are described on one line in the SDF. The former part of the line represents the actions of the fielders and the ball, while the latter part describes the actions of the offensive players. This file is loaded in starting the system and is sequentially read out lineby- line at every 1 play.

3.2 Actions of offensive players

Offensive players indicate a batter, runners, and players who are waiting in the bench. Each player belongs to one of the six states as shown in Fig. 14(a). The batter is in the batter's box, so its state is "0", third runner is "3", and the waiting players are "-1". In SDF, the destination state to which every player changes in each play is sequentially recorded. When one line of the file is read out, the destination of each player is decided according to the data like Fig. 14(b). Then the game scene that 3D players are moving from the present state to the destination state while 1 play is created with CG.

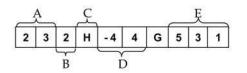(a) State transition of offensive players

(b) Example of Scorebook Data File for offensive players

Fig. 14. Actions of the offensive players

3.3 Actions of fielders and ball

In contrast to the offensive players who are just moving from present state to destination while 1 play, the fielders are doing some actions while 1 play, such as moving around the field and throwing and catching the ball, etc. Therefore only the action of the ball is described in the SDF. Fielders move to catch the ball according to the action of the ball. The action of the ball while 1 play is described as shown in Fig. 15.

A: pitching position of strike zone
B: through / swing out / hit
C: ball / strike / out / hit
D: flying position on field model
E: fielders' No. who catch the ball

Fig. 15. Scorebook Data File of the fielders and the ball

In this system, the fielders basically stay own positions. First, the ball is thrown by the pitcher and hit to the position which is described in part D of Fig. 15. Then the player whose position number is described in the fist of part E moves to the position of part D to catch the ball. After catching the ball, the player throws the ball to the next player whose position number is described next. The next player moves to the nearest base and catches the ball. After the same iterations, finally, the ball is thrown to the pitcher.

3.4 Demonstrations

This section shows the demonstration results of this system. This demonstration is performed with a web-camera (ELECOM UCAM-E130HWH) attached to a handheld monitor connected a PC (OS: Windows XP, CPU: Intel Pentium IV 3.2GHz). The resolution of the captured image is 640x480 pixels. Multiple planar markers are distributed inside and outside the field model. In this system, one of the markers must be put on one of the bases in order to determine relationship between the field model and the markers. The other markers can be placed at arbitrary positions and poses. In this demonstration, 4 markers are utilized and one of them is placed on the third base. Fig. 16 shows the actual experimental environment of this system. A Scorebook Data File of a baseball game is manually prepared in accordance with Sec. 3.1. 3D models of virtual objects, such as players and a ball, are rendered with OpenGL.

Fig. 16. Experimental environment of AR Baseball Presentation System

Fig. 17 shows an example scene of a baseball game: team RED vs. team WHITE. In this situation, team WHITE is in the field and team RED is at bat. The bases are loaded and 4th batter of team RED is in the batter's box (frame 0~15). The pitcher throws the ball (frame

15~29). The batter hits safely to left (frame 29~35), and then all runners move up a base (frame 50~89). In the result, team RED gets a score. In this experiment, frame rate of AR presentation is about 30 fps. The user can see the baseball game at video-rate.

In Fig. 18, the angle of the camera with respect to the tabletop is too small to detect the markers lying on the tabletop plane as same as Fig. 10. Therefore one marker placed near the home base cannot be detected. Since another marker is placed at different pose from the ground plane, the registration of the virtual players can be continued. This is quite useful for observation system so that the user can watch the object scene from various view points.

(a) frame 0   (b) frame 15   (c) frame 20   (d) frame 29

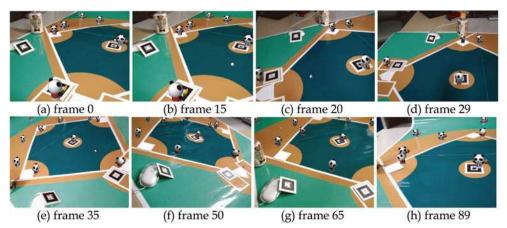(e) frame 35   (f) frame 50   (g) frame 65   (h) frame 89

Fig. 17. Example scene of play: 4th batter of team RED sends a hit to left with the bases loaded, and then 3rd runner gets home
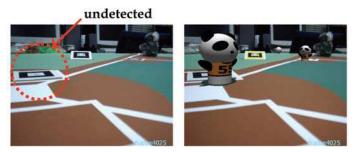

Fig. 18. Failure of marker detection even though it is captured in the image

Fig. 19 shows the resulting images which are watched from various view points. By using this system, the user can watch the baseball game from favorite view points, such as catcher's view point. For changing view points from (a) to (b), from (b) to (c), from (c) to (d), 15 users spend an average of about 7 seconds in this system. In contrast with using a typical CG viewer in which a keyboard is used for changing view points, the users spend an average of about 43 seconds. This is because the users only have to move around the field model for changing the view points in this system. Such simple way for watching the Cgrepresented event using the AR system is very useful and can provide immersive feeling as an entertainment tool.
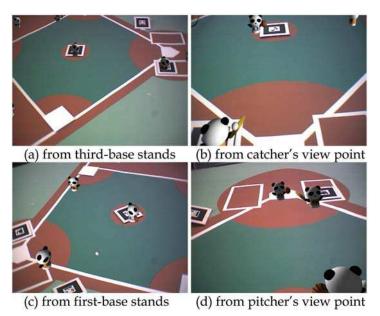
(a) from third-base stands   (b) from catcher's view point

(c) from first-base stands   (d) from pitcher's view point

Fig. 19. Example resulting images watched from various view points

4. Interactive AR bowling system

With Interactive AR Bowling System, users can enjoy the bowling game by rolling a real ball down a real bowling lane model placed on a tabletop in the real world. Fig. 20 shows the overview of this system. On the lane model, there are virtual pins generated with CG. They knock down the virtual pins by rolling the real ball.

Touching and rolling the real ball provide a sort of tangible feeling in this system. It is wellknown that a tangible interface enhances the reality of communication [Ishii et al., 1999, Zigelbaum et al. 2006]. Moreover, because of placing some planar markers on the lane model, the users can watch the lane and pins from free view points. For registration of virtual objects such as the virtual players or the virtual pins, the motion of the user's camera is estimated by multiple 2D markers.
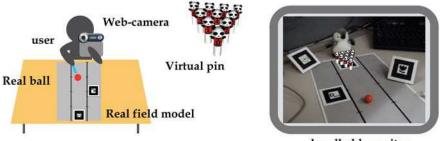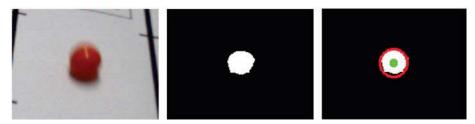

Fig. 20. Interactive AR Bowling System

4.1 Ball tracking

In this system, we assume that the color of the ball should be quite different from the lane model. In this paper, we use a red ball on a gray lane model as shown in Fig. 21(a). For detection of the ball, first, red regions are detected from the input image by dividing it into R, G, B channel images. Fig. 21(b) shows the image after dilation and erosion a few times. Finding the minimal circumscribed circle (contour) for the detected region, the center of the circle is considered as the 2D ball's position in the input image as shown in Fig. 21(c).

(a) Original image (b) Ball's region (c) Detected ball

Fig. 21. Ball detection

4.2 Transformation to top view image

Using homography **H** computed at the marker tracking process, the ball's position on the input image is transformed onto the top view image that provides a geometrical relationship between the ball and the pins on the lane model. As shown in Fig. 22, the trajectory of the ball can be obtained. This trajectory is used to detect the collision between the ball and the pins, and compute the directions which the pins are knocked down.
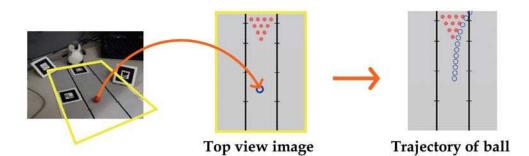

Fig. 22. Transform the ball's position to top-view image

4.3 Collision detection of ball and pins

We assume that radii of the ball and the pins are r_b and r_p, respectively, and define the distance between the ball and each pin as dis. For detecting a collision between the ball and the pins, the distance *dis* is computed from the top view image at every frame. The collision is detected by comparing distance and radius as following equations and Fig. 23.

$$(r_b + r_p) \leq dis \Rightarrow \text{Not hit}; \qquad (r_b + r_p) > dis \Rightarrow \text{Hit} \qquad (20)$$

Fig. 23. Collision detection by trajectory of the ball

4.4 Overlay virtual pins

After the collision detection, the pins are generated with CG and overlaid onto the image. If the collision is detected, the pins are gradually inclined and knocked down. The direction of knocking down is defined by trajectory of the ball. As shown in Fig. 24, the direction is computed by a motion vector of the ball, which is decided by ball's positions in previous and current frames, and a vector from the ball to each pin. The generated pins are superimposed onto the image by the extrinsic parameters computed by 2D markers. The user can see the virtual pins according to the motion of the camera and the rolling ball.

4.5 Demonstration

This section shows the demonstration results of this system. This demonstration is also performed with a web-camera (ELECOM UCAM-E130HWH) attached to a handheld monitor connected a PC (OS: Windows XP, CPU: Intel Pentium IV 3.2GHz). The resolution of the captured image is 640x480 pixels. On the table top, a real bowling lane model is placed and 4 planar markers are distributed inside and outside the lane model. In order to determine relationship between the lane and the markers, one marker is placed between the two lines of the lane. Fig. 25 shows the actual experimental environment of this system.

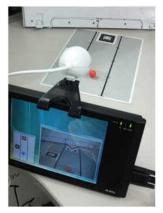Fig. 24. Direction of knocking down of pins

Fig. 25. Experimental environment of Interactive AR Bowling System

Fig. 26 shows the detected lane and ball's trajectory. Both of the lane and the ball can be correctly detected and tracked all over the frames according to the camera motion. The ball's position is also successfully transformed onto the top view image by the homography computed by the markers. Since a real ball is used in this system, the marker placed on the lane can be occluded by the ball. However, this method can estimate the relationship of the markers and compute the homography successfully, even if some of markers are occluded.

Fig. 26. Detected lane and ball, and trajectory of ball

Fig. 27 shows example scenes of playing this system. The virtual pins are overlaid on the lane model according to the camera motion. As described before, the virtual pins can b overlaid onto the lane model, even when the ball is rolling over the marker. The collision of the real ball and the virtual pins is successfully detected. Therefore some pins are knocking down by the real ball. The pins existing behind the hit pins are also knocked down as a chain reaction of the front pins by computing the direction of knocking down from the trajectory. In this experiment, this system also runs about 30 fps. The advantage of this system is that the user can physically touch and roll the real ball without special hardware such as physical sensors or special gloves in contrast with the previous application in [Matysczok et al. 2004]. Therefore this system achieves a real bowling style. If the ball and lane are also generated by CG, the user cannot freely control the ball. On the other hand, there are various ways to roll the ball in this system, the game is not too simple to complete; ex. inclining the lane, changing the material of the lane, or rolling the ball by a pen instead of a hand. Therefore this system can be challenging for the users.

Fig. 27. Example scene of playing Interactive AR Bowling System

5. Conclusion

In this chapter, we introduced a vision-based registration method using multiple planar markers placed at arbitrary positions and poses is introduced. We also demonstrate the performance of the proposed method by applying it to two AR applications. In contrast with the previous methods using multiple markers, this method requires no time-consuming measurement of the marker arrangement. Since the marker arrangement can be estimated by the Projective Space, the markers can be distributed at arbitrary positions and poses without a prior knowledge. This method can be applied not only for on the tabletop but also inside the room.

Both of the baseball system and bowling system can be enjoyed on the tabletop in the real world only with a web-camera and a handheld monitor connected a general PC. It is a big advantage for home users that these applications do not require any special device. Users can interactively change their view points by moving around the tabletop because of this registration method with multiple markers. In contrast with usual CG viewers in which a mouse or a keyboard is used for changing view points, changing view points by moving of the users is very intuitive and easy. By visualising 3D objects in front of the users, these applications will be future-oriented 3D game.

6. References

Y. Genc; S. Riedel; F. Souvannavong; C. Akinlar & N. Navab (2002). Marker-less Tracking for AR: A learning-Based Approach, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2002)*, pp. 295–304, ISBN: 0-7695-1781-1.

M. Haller; M. Billinghurst; J. Leithinger; D. Leitner & T. Seifried (2005). Coeno-enhancing face-to-face collaboration. *Proceedings of International Conference on Augmented Tele-existence (ICAT2005),* pp. 40–47, ISBN:0-473-10657-4.

R. Hartley & A. Zisserman (2000). In: *Multiple View Geometry in Computer Vision,* CAMBRIDGE UNIVERSITY PRESS, ISBN: 0521540518.

A. Henrysson; M. Billinghurst & M. Ollila (2005). Virtual Object Manipulation Using a Mobile Phone, *Proceedings of International Conference on Augmented Tele-existence (ICAT2005)*, pp. 164–171, ISBN:0-473-10657-4.

A. Henrysson; M. Billinghurst & M. Ollila (2006). AR Tennis, *Proceedings of ACM SIGGRAPH Emerging technologies*, Article No. 1, ISBN: 1-59593-364-6, Boston, Aug.

H. Ishii; C. Wisneski; J. Orbanes; B. Chun & J. Paradiso (1999). curlybot: designing a new class of computational toys. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 129–136, ISBN: 1-58113-216-6.

H. Kato & M. Billinghurst (1999). Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System, *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99)*, pp. 85-94, ISBN: 0-7695-0359-4.

H. Kato; M. Billinghurst; I. Poupyrev; K. Imamoto & K. Tachibana (2000). Virtual object manipulation on a table-top AR environment, *Proceedings of IEEE and ACM International Symposium on Augmented Reality (ISAR2000),* pp. 111–119, ISBN: 0- 7695-0846-4.

K. Klein & T. Drummond (2004). Sensor fusion and occlusion refinement for tablet-based AR, *Proceedings of Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2004)*, pp. 38–47, ISBN: 0-7695-2191-6.

D. Kotake; S. Uchiyama & H. Yamamoto (2004). A Marker Calibration Method Utilizing A Priori Knowledge on Marker Arrangement, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2004)*, pp. 89–98, ISBN: 0-7695-2191-6.

J. Looser; R. Grasset & M. Billinghurst (2007). A 3D Flexible and Tangible Magic Lens in Augmented Reality, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2007)*, pp. 51–54, ISBN: 978-1-4244-1749-0.

C. Matyszok; R. Radkowski & H. Nixdorf (2004). AR-bowling: immersive and realistic game play in real environments using augmented reality. *Proceedings of the ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE2004)*, pp. 269–274, ISBN: 1-58113-882-2.

D. Schmalstieg & D. Wagner (2007) Experiences with Handheld Augmented Reality, *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2007)*, pp. 3–18, ISBN: 0-7695-1781-1.

Y. Uematsu & H. Saito (2005). Vision-based Registration for Augmented Reality with Integration of Arbitrary Multiple planes, *Proceedings of International Conference on Image Analysis and Processing (ICIAP2005), LNCS 3617*, pp. 155–162, ISBN: 978-3-540- 28869-5.

Y. Uematsu & H. Saito (2005). AR Registration by Merging Multiple Planar Markers at Arbitrary Positions and Poses via Projective Space, *Proceedings of International Conference on Augmented Tele-existence (ICAT2005)*, pp. 48–55, ISBN: 0-473-10657-4.

Y. Uematsu & H. Saito (2006). AR Baseball Presentation System with Integrating Multiple *Planar Markers, Proceedings of International Conference on Augmented Tele-existence (ICAT2006), LNCS 4282, pp. 163-174, ISBN: 978-3-540-49776-9.*

Y. Uematsu & H. Saito (2007). Interactive AR Bowling System by Vision-based Tracking, *Proceedings of the international conference on Advances in computer entertainment technology(ACE2007)*, pp. 236-237, ISBN: 978-1-59593-640-0.

E. J. Umlauf; H. Piringer; G. Reitmayr & D. Schmalstieg (2002). ARLib: The Augmented Library, *Proceedings of International Workshop on ARToolkit (ART02).*

J. Zigelbaum; A. Millner; B. Desai & H. Ishii (2006). Bodybeats: Wholebody, Musical Interfaces for Children. *Proceedings of Conference on Human Factors in Computing Systems (CHI '06), pp. 1595–1600, ISBN: 1-59593-298-4.*

Computer Vision

Edited by Xiong Zhihui

This book presents research trends on computer vision, especially on application of robotics, and on advanced approachs for computer vision (such as omnidirectional vision). Among them, research on RFID technology integrating stereo vision to localize an indoor mobile robot is included in this book. Besides, this book includes many research on omnidirectional vision, and the combination of omnidirectional vision with robotics. This book features representative work on the computer vision, and it puts more focus on robotics vision and omnidirectioal vision. The intended audience is anyone who wishes to become familiar with the latest research work on computer vision, especially its applications on robots. The contents of this book allow the reader to know more technical aspects and applications of computer vision. Researchers and instructors will benefit from this book.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yuko Uematsu and Hideo Saito (2008). Vision-based Augmented Reality Applications, Computer Vision, Xiong Zhihui (Ed.), ISBN: 978-953-7619-21-3, InTech, Available from:
http://www.intechopen.com/books/computer_vision/vision-based_augmented_reality_applications

INTECH
open science | open minds