ICAT 2005
December 5-8, Christchurch, New Zealand

AR Registration by Merging Multiple Planar Markers at Arbitrary Positions and Poses via Projective Space

Yuko Uematsu and Hideo Saito
Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
{*yu-ko, saito*}*@ozawa.ics.keio.ac.jp*

Abstract

This paper presents a vision-based registration method for Augmented Reality by merging multiple planar markers, which are placed at arbitrary positions and poses. In our approach, we use planar markers like AR-Toolkit markers. We need no information about geometrical relationship of the markers while most of marker-based approaches require the measurement of the markers' 3D positions and poses. Thus we can implement an AR system very easily without time-consuming calibration of markers and a camera.

In the proposed registration method, we assign a 3D coordinate system for each marker independently, and compute a projection matrix between each marker's 3D coordinate system and the input image coordinate system. For merging all markers' 3D coordinate, we construct a projective 3D space defined by projective geometry between two reference images which are selected while the on-line process automatically. This merging process enables to use the multiple markers at arbitrary positions and poses, and also to reduce errors in registration. For demonstrating the effectiveness of the proposed method, we implement a real-time on-line AR system. Virtual objects are superimposed onto input image sequences that are captured with a handy camera. Since the multiple markers are placed in the wide area, the virtual objects can move around the real world widely. Moreover, the registration can be carried out stably, because the markers face to various directions and some of them can be always recognized from most view points.

Keywords: Augmented Reality, Mixed Reality, vision-based registration, projective space, multiple planar markers, AR-Toolkit

1. Introduction

Augmented Reality(AR)/Mixed Reality(MR) is a technique for overlaying virtual objects onto the real world. We can see the virtual objects as if they really exist in the real world, so AR can provide users more effective views [1, 2]. In AR, one of the most important issues is geometrical registration between the real world and the virtual world. For such registration, it is required to measure camera motions (corresponding to user's view) precisely.

The registration methods are divided into sensor-based approach and vision-based approach. By using some sensors (such as magnetic or gyro), the registration is stable against a various illumination conditions and rapid motion of cameras [21, 15]. However, the rotations and translations of the camera obtained only from the sensors are not accurate enough to achieve complete geometrical registration of virtual objects in AR. Furthermore, the use of sensors has some limitations in practice: user's movable area, perturbation caused by the using environment, and so on.

In contrast, vision-based registration does not require any special devices except cameras, thus a lot of methods have been proposed. The vision-based approaches are also divided into marker-based approach and using natural features approach. Using natural features, the augmentation is realized very naturally [16, 20, 5]. However, it is not easy to construct a robust on-line system, because exact recognition of natural features in real-time is hard task and registration jitters are often caused. It is also true that only few features are available for registration in the real world. Since construction of an on-line system is very important issue in AR, we focus on marker-based approach.

Marker-based approach is one of the most accurate methods. Especially, "AR-Toolkit" [4, 3] is very popular and highly-used tool for simple implementation of an on-line AR system [17, 7, 6]. Although such marker-based registration method is very easy way to realize an AR system, the camera's movable area is limited to the area where the camera (user) can see the marker in using only one marker. Moreover, when the marker cannot be recognized properly because of a change in its visibility, the registration of virtual objects is getting unstable.

In order to solve such problem, using multiple markers is a popular way. However, in order to use multiple markers, it is necessary to know their geometrical arrangement of the markers in advance such as their positions and poses [13, 11, 9, 10, 8]. In [13], they need the position and pose of a square marker and the position of a point marker in advance. In [11], they proposed markerless registration method by setting up a learning process. In the learning process, however, the markers' geometrical information is required for learning the markers. In most cases, the task for the measurement of such information is implemented manually. However this task is very time-consuming and not sufficiently accurate. Kotake et al. [14] proposed a marker-calibration method combining multiple planar markers with bundle adjustment. Although they does not require a precise measurement of the markers, they need a priori knowledge of the markers such as qualitative information to compute markers' geometrical information from a set of images by bundle adjustment, *e.g.* multiple markers are coplanar.

In this paper, we propose a vision-based registration method using multiple markers without their geometrical placement and implement a real-time on-line AR system. This means that the markers can be placed at arbitrary positions and poses. In order to realize such a system, we employ the algorithm of merging arbitrary multiple planes [22]. In this method [22], using multiple planar structures in the real world, a projection matrix that relates the real world with each plane is computed for each plane. Then all the projection matrices for all planes are merged into one transformation matrix via a projective 3D space.

Table 1 shows the comparison of characteristic of the pro-

Table 1: Comparison of characteristic of the proposed method and related works using multiple planar structures.

	arrangement of planes	online /off-line	visual features	frame by frame processing
[18]	perpendicular	on-line	natural	possible
[19]	arbitrary	off-line	natural	impossible
[14]	a priori knowledge	on-line	marker	possible
[22]	arbitrary	off-line	natural	possible
proposed method	arbitrary	on-line	marker	possible

posed method and related works. In [18], they use multiple natural planes in on-line system, however, there is a restriction that the planes need to be perpendicular to a reference plane. Although there is no restriction about arrangement of the planes in [19], the geometrical relationship of the planes is computed by bundle adjustment which should be carried out over all frames. In [14], a priori knowledge of the markers is required as mentioned previously. On the other hand, [22] does not require any geometrical placement information about the planes and can estimate the relationship between the real world and the input image frame by frame. However, automatic recognition of natural planes is very hard task, the experiments in [22] are just off-line process.

Therefore, we apply [22] algorithm to the multiple markers and construct a real-time on-line system in this paper. Using markers instead of natural planes, we can recognize the planar regions automatically and easily estimate the extrinsic parameters of a camera at every frame. Of course the markers can be arranged at any positions and poses, so our system can be implemented easily without advance preparation.

2. Overview

In this section, we introduce the algorithm of merging arbitrary multiple planes [22] and how to apply it to the proposed system. Fig. 1 describes an overview of our proposed approach. We assume that there are multiple planar markers without any information on their positions and poses in the object scene. First, a 3D coordinate system is assigned for each plane appeared in input images independently. Then a projection matrix $\boldsymbol{P_i}$ that relates each plane's 3D coordinate system i to the input image is computed by a planar homography for each plane. The subscript i is a plane's id number $(1 \leq i \leq n)$. Since those projection matrices are computed from the coordinate systems that are defined by the planes without geometrical relationship each other, we need to find the relationship among all the coordinate systems for merging them. For estimating the relationships, a projective 3D space is defined only by the projective geometry of two reference images, so it does not depend on any plane's 3D coordinate system. The relationship among the three coordinate systems, the real world, the projective space, and the input image, is shown in fig. 2.

After the projective space is constructed, a transformation matrix $\boldsymbol{T_i^{WP}}$ between each plane's coordinate system i and the projective space can be computed. Using $\boldsymbol{P_i}$ and $\boldsymbol{T_i^{WP}}$, another transformation $\boldsymbol{T_i^{PI}}$ is computed, which relates the projective space to the input images respectively. These transformations relate between the same two coordinate systems (projective space → input images), so it is possible to merge $\boldsymbol{T_i^{WP}}$. After merging process, virtual objects are aligned at accurate positions and poses by the merged $\boldsymbol{T^{PI}}$.

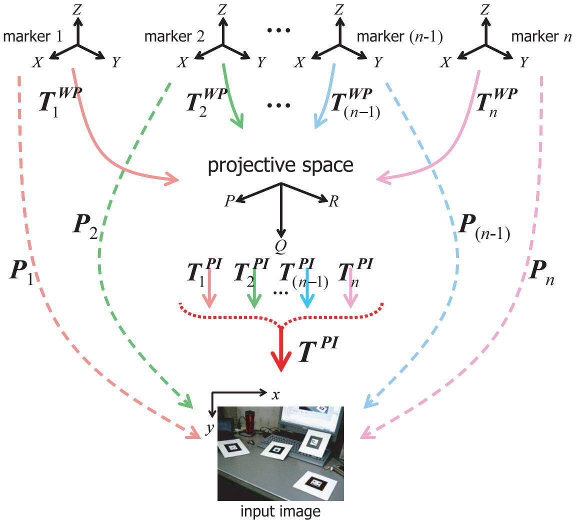


Figure 2: Relationship among three coordinate systems.

In this paper, the input image sequence is taken by a handy camera with pre-calibrated intrinsic parameters. We assign a 3D coordinate system for each markers, and a projection matrix is computed from each marker independently, which relates each marker's 3D coordinate system i to the input image's 2D coordinate system. Then we select two reference images automatically in order to construct a projective 3D space. Since this projective space is defined by the projective geometry of these reference images, selection of the reference images is very important process to construct a good projective space. We also achieve this process on-line. Once the projective space is constructed, $\boldsymbol{T_i^{WP}}$ and $\boldsymbol{T_i^{PI}}$ are computed. Then $\boldsymbol{T_i^{PI}}$ are merged into one transformation matrix $\boldsymbol{T^{PI}}$. Then virtual objects are superimposed onto the input images and rendered as the output images. In sec. 3, we describe a detailed theory of the registration method used in this paper.

3. Registration Method

In this paper, registration means to estimate the extrinsic parameters of a camera at every frame, and align virtual objects according to the parameters.

3.1. Construction of Projective Space

The projective space used for merging process is based on the concept of "projective reconstruction" as shown in fig. 3. This space is constructed by two reference images.

By epipolar geometry between the two reference images (cameras), each relationship between the projective space and the reference images is represented as follows respec-

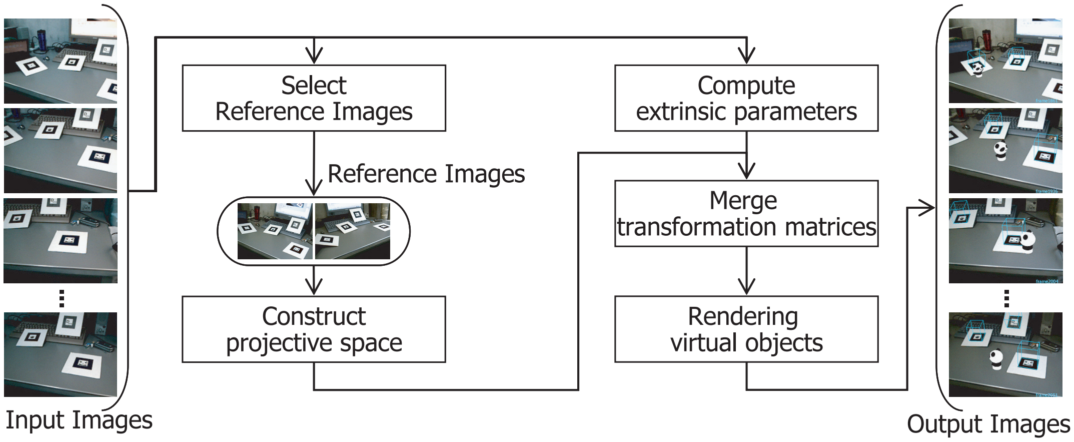


Figure 1: Overview of our approach.

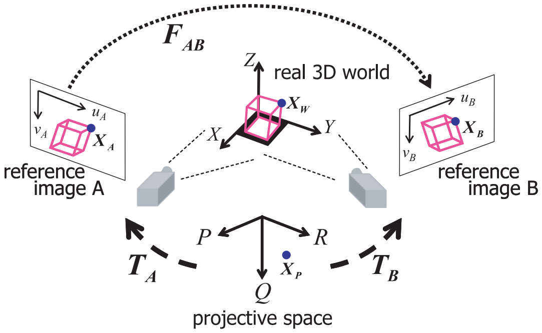


Figure 3: Projective space by projective reconstruction.

tively,

$$\boldsymbol{T_A} = [\mathbf{I} \,|\, \mathbf{0}]\,, \qquad \boldsymbol{T_B} = \left[-\frac{[\boldsymbol{e_B}]_\times \boldsymbol{F_{AB}}}{\|\boldsymbol{e_B}\|^2} \,\middle|\, \boldsymbol{e_B} \right] \tag{1}$$

where, $\boldsymbol{F_{AB}}$ is a fundamental matrix from the image A to B, $\boldsymbol{e_B}$ is an epipole on the image B, and $[\boldsymbol{e_B}]_\times$ is the skew-symmetric matrix [12]. How to select the two reference images and compute $\boldsymbol{F_{AB}}$ will be explained in sec. 3.6.

If $\boldsymbol{X_W}$ is a point in the real 3D world, this point $\boldsymbol{X_W}$ is projected to the reference images as $\boldsymbol{X_A}(u_A, v_A)$ and $\boldsymbol{X_B}(u_B, v_B)$ respectively. Then $\boldsymbol{X_A}$ and $\boldsymbol{X_B}$ are transformed to a point $\boldsymbol{X_P}(p, q, r)$ in the projective space by next equation,

$$\begin{bmatrix} \boldsymbol{t_A^1} - u_A \boldsymbol{t_A^3} \\ \boldsymbol{t_A^2} - v_A \boldsymbol{t_A^3} \\ \boldsymbol{t_B^1} - u_B \boldsymbol{t_B^3} \\ \boldsymbol{t_B^2} - v_B \boldsymbol{t_B^3} \end{bmatrix} \boldsymbol{X_P} = \mathbf{0} \tag{2}$$

where, $\boldsymbol{t^j}$ is the j th row vector of the matrix $\boldsymbol{T_A}$ and $\boldsymbol{T_B}$. $\boldsymbol{X_P}$ is obtained by the singular value decomposition of this 4×4 matrix. The projective space coordinate system is related to the real 3D world coordinate system through the two reference image coordinate systems in this way.

3.2. Computation of Projection Matrix $\boldsymbol{P_i}$

First, we assign a 3D coordinate system for each marker independently, as shown in fig. 4. From the image sequence captured by a calibrated camera, we compute the extrinsic parameters of the camera for each marker's 3D coordinate system frame by frame. For computing the parameters, we employ the algorithm of AR-Toolkit [4, 3]. Since we assume that the intrinsic parameters are already known, when the extrinsic parameters are obtained in a frame, we can compute a projection matrix $\boldsymbol{P_i}$ for each marker i, which relates each 3D coordinate system to the frame.

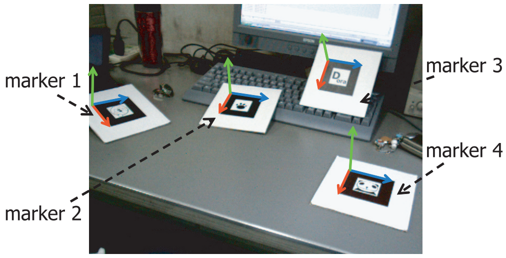


Figure 4: Example of markers 3D coordinate systems.

3.3. Computation of $\boldsymbol{T_i^{WP}}$

As previously mentioned in sec. 3.1, a point $\boldsymbol{X_W}(X, Y, Z)$ in the real world is transformed to a point $\boldsymbol{X_P}(p, q, r)$ in the projective space by eq. (2). Since a transformation matrix $\boldsymbol{T_i^{WP}}$, which relates each marker's coordinate to the projective space, is 4×4 matrix, we can compute it from five or more corresponding points. If a cube is drawn on each marker by a projection matrix $\boldsymbol{P_i}$ like fig. 3, its vertices can be transformed from the marker's 3D coordinate to the projective space via the reference images. Therefore, eight correspoinding points can be obtained, so $\boldsymbol{T_i^{WP}}$ is computed. The computation process is as follows.

$$\boldsymbol{X_P} \simeq \boldsymbol{T_i^{WP}} \boldsymbol{X_W} \tag{3}$$

$$\boldsymbol{T_i^{WP}} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \tag{4}$$

where we let $t_{44} = 1$,

$$\boldsymbol{Mt} = \boldsymbol{b} \tag{5}$$

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{X_W^\top} & \boldsymbol{0} & \boldsymbol{0} & -Xp & -Yp & -Zp \\ \boldsymbol{0} & \boldsymbol{X_W^\top} & \boldsymbol{0} & -Xq & -Yq & -Zq \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{X_W^\top} & -Xr & -Yr & -Zr \end{bmatrix} \tag{6}$$

$$\boldsymbol{t} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{42} & t_{43} \end{bmatrix}^\top \tag{7}$$

$$\boldsymbol{b} = \begin{bmatrix} p & q & r \end{bmatrix}^\top \tag{8}$$

Using m corresponding points ($m \geq 5$), eq. (5) is extended as follows,

$$\begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix} \boldsymbol{t} = \begin{bmatrix} \boldsymbol{b_1} \\ \vdots \\ \boldsymbol{b_m} \end{bmatrix} \tag{9}$$

By the least-square method,

$$\boldsymbol{t} = \begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^+ \begin{bmatrix} \boldsymbol{b_1} \\ \vdots \\ \boldsymbol{b_m} \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^+ = \left(\begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^\top$$

3.4. Computation of $\boldsymbol{T_i^{PI}}$

As shown in fig. 2, the transformation matrix $\boldsymbol{T_i^{PI}}$ relates the projective space to the input images. When $\boldsymbol{P_i}$ and $\boldsymbol{T_i^{WP}}$ have already been computed, we can obtain $\boldsymbol{T_i^{PI}}$ by those two matrices.

$$\boldsymbol{T_i^{PI}} = \boldsymbol{P}_i \left(\boldsymbol{T_i^{WP}} \right)^{-1} \tag{11}$$

Since each $\boldsymbol{T_i^{PI}}$ is the transformation between the same two coordinate systems, which is from the projective space to the input images, we can merge them simply. The merging process is described in next section.

3.5. Merging of $\boldsymbol{T_i^{PI}}$

In order to use the information from multiple markers and to reduce the registration errors than using only one marker, we merge $\boldsymbol{T_i^{PI}}$ computed from each visible marker i at the frame. When $\boldsymbol{T_i^{PI}}$ is computed, some corresponding points between the projective space and the input images are computed by eq. (12), *e.g.* eight vertices of a cube like the computation of $\boldsymbol{T_i^{WP}}$ in sec. 3.3.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \simeq \boldsymbol{T_i^{PI}} \begin{pmatrix} p \\ q \\ r \\ 1 \end{pmatrix} \tag{12}$$

These corresponding points are computed for each marker. If n markers are visible at the frame, $8n$ corresponding points are obtained. Then, using the $8n$ corresponding points, we can merge $\boldsymbol{T_i^{PI}}$ into $\boldsymbol{T^{PI}}$ by eq. (14), like the computation of $\boldsymbol{T_i^{WP}}$ by eq. (5).

$$\begin{bmatrix} \boldsymbol{X_{P_1}^\top} & \boldsymbol{0} & \boldsymbol{0} & \text{-}p_1x_1 & \text{-}q_1x_1 & \text{-}r_1x_1 \\ \boldsymbol{0} & \boldsymbol{X_{P_1}^\top} & \boldsymbol{0} & \text{-}p_1y_1 & \text{-}q_1y_1 & \text{-}r_1y_1 \\ & & \vdots & & & \\ \boldsymbol{X_{P_{8n}}^\top} & \boldsymbol{0} & \boldsymbol{0} & \text{-}p_{8n}x_{8n} & \text{-}q_{8n}x_{8n} & \text{-}r_{8n}x_{8n} \\ \boldsymbol{0} & \boldsymbol{X_{P_{8n}}^\top} & \boldsymbol{0} & \text{-}p_{8n}y_{8n} & \text{-}q_{8n}y_{8n} & \text{-}r_{8n}y_{8n} \end{bmatrix} \begin{bmatrix} t_{11} \\ t_{12} \\ \vdots \\ t_{32} \\ t_{33} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_{8n} \\ y_{8n} \end{bmatrix}$$

$$\Leftrightarrow \boldsymbol{Mt} = \boldsymbol{b} \tag{13}$$

By the least-square method,

$$\boldsymbol{t} = \left(\boldsymbol{M}^\top \boldsymbol{M} \right)^{-1} \boldsymbol{M}^\top \boldsymbol{b} \tag{14}$$

In this way, one merged transformation matrix $\boldsymbol{T^{PI}}$ that relates the projective space and the input images is obtained. Then, we can draw virtual objects by eq. (15) whose coordinate systems $\boldsymbol{X_V}$ are defined in the projective space's coordinate system.

$$\boldsymbol{x} = \boldsymbol{T^{PI}} \boldsymbol{X_V} \tag{15}$$

3.6. Selection of reference images

The projective space is defined by the projective geometry of two reference images. In order to construct the accurate projective space, selection of the reference images is very important. We can automatically select them while on-line process. The detail is shown in fig. 5 and as follows.

When capturing the object scene starts, each projection matrix $\boldsymbol{P_i}$ for the marker i is computed every frame. Then we capture some images around the target scene and select two frames from the sequence, we call them reference image A and B. The result is n pairs of projection matrices, one pair ($\boldsymbol{P_{A_i}}$ and $\boldsymbol{P_{B_i}}$) for each marker ($1 \leq i \leq n$). Using each pair of projection matrices, we can compute a fundamental matrix for marker i as follows,

$$\boldsymbol{F_{AB_i}} = [\boldsymbol{e_{B_i}}]_\times \boldsymbol{P_{B_i}} \boldsymbol{P_{A_i}^-} \tag{16}$$

where, $[\boldsymbol{e_{B_i}}]_\times$ is also the skew-symmetric matrix and $\boldsymbol{P_{A_i}^-}$ represents the pseudo inverse matrix of $\boldsymbol{P_{A_i}}$ [12]. When n fundamental matrices are computed, we select the one with the smallest value of the computation result:

$$result = \boldsymbol{m_B}^\top \boldsymbol{F_{AB_i}} \boldsymbol{m_A} \tag{17}$$

where, $\boldsymbol{m_A}$ and $\boldsymbol{m_B}$ is a corresponding point in each reference image respectively. Decided one fundamental matrix, we can construct a projective space temporarily by eq. (1), (2), and then $\boldsymbol{T_i^{WP}}$ and $\boldsymbol{T_i^{PI}}$ are computed for each maker respectively via the temporal projective space. Additionally, $\boldsymbol{T_i^{PI}}$ for each marker is merged into one $\boldsymbol{T^{PI}}$ by eq(14). Then we compare these two coordinates $\boldsymbol{x}, \boldsymbol{x'}$:

$$\boldsymbol{x_i} = \boldsymbol{P}_i \boldsymbol{X_W} \tag{18}$$

$$\boldsymbol{x'_i} = \left(\boldsymbol{T^{PI}} \boldsymbol{T_i^{WP}} \right) \boldsymbol{X_W} \tag{19}$$

Although these two coordinates should be equal, if the combination of the two reference images is not reasonable, they will be different. In such case, we return to the phase of selecting a pair of the reference images. We iterate these processes until all differences of $\boldsymbol{x_i}$ and $\boldsymbol{x'_i}$ for the markers become smaller than 3 pixels.

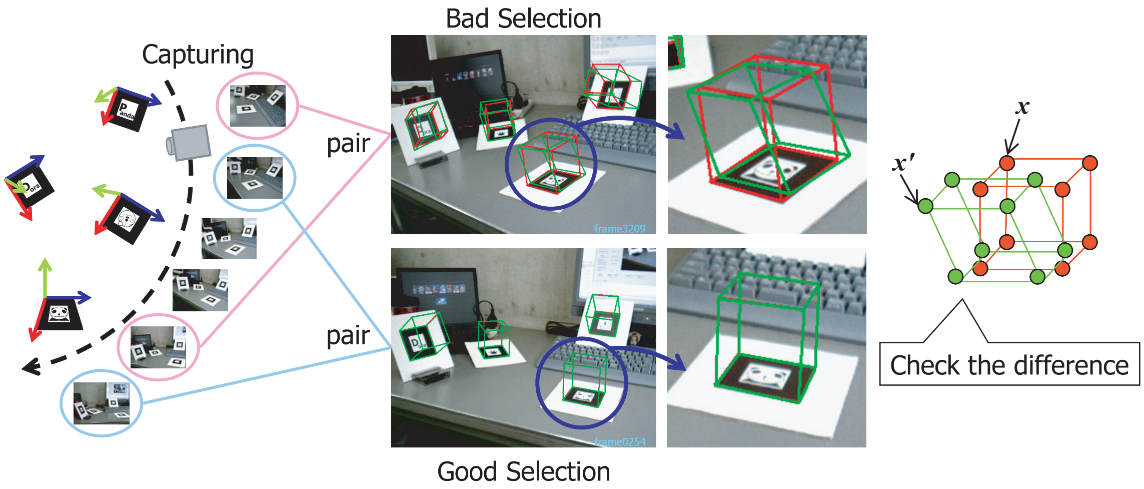


Figure 5: Selection of the reference images.

4. Experiment

4.1. Implementation

We implement an Augmented Reality system based on the proposed method and perform experiments to prove the efficacy of our method. This system is constructed only by a PC (OS:Windows XP, CPU:Intel Pentium Ⅳ 3.6GHz) and a web camera (ELECOM UCAM-E1D30MSV). The captured image's resolution is 640×480 pixels and graphical views of virtual objects are rendered using OpenGL. The actual operation process of our system is as follows.

Operation Process

1. **Arrangement**
   Place multiple markers at arbitrary positions and poses;
2. **Capturing**
   Capture around the object scene as candidates for two reference images;
3. **Seletion**
   Select two reference images automatically from the captured image sequence by using the method described in sec. 3.6;
4. **Overlay**
   Overlay virtual objects onto the images captured the object scene at free view points by merging the visible markers as sec. 3.5;

Firstly, some markers are arranged at arbitrary positions and poses in the object scene. Next, we start capturing the object scene and computing $\boldsymbol{P}_i^{\boldsymbol{WI}}$ for each marker frame by frame. Then, we make some frames of the captured sequence candidates for two reference images. In this experiment, we use 100 frames as the candidates. In selection process, one reasonable combination of the reference images are selected from the candidate images automatically as shown in sec. 3.6. Using the two reference images, a projective space is constructed and each $\boldsymbol{T}_i^{\boldsymbol{WP}}$ is computed by the algorithm in sec. 3.3. When the projective space is prepared, the system goes to overlay process. At every frame, $\boldsymbol{T}_i^{\boldsymbol{PI}}$ is computed for each visible marker and is merged via the projective space, as described in sec. 3.5. Then virtual objects are aligned at accurate positions and poses by using the merged $\boldsymbol{T}^{\boldsymbol{PI}}$.

4.2. Results

The overlaid result images of first experiment are shown in fig. 6. In this sequence, 4 markers are placed at arbitrary positions and poses, a virtual object (a figure of cartoon panda) moves around the markers. In the first experiment, the multiple markers are almost coplanar. In the next experiment, the multiple markers are arranged to be more various poses. The result images are shown in fig. 7.

As seen in these fig. 6 and 7, the virtual object can successfully be aligned at accurate position and pose, although different markers are visible according to the camera motion. This means that the consistency of the geometrical relationship between the camera and the virtual object is kept properly, since the multiple markers' 3D coordinate systems can be merged successfully even though the geometrical arrangement among them is unknown.

In particular, the notable results are shown in the bottom line of fig. 7. In fig. 7(m), (n), and (o), the angle of the camera relative to the desktop is too small to detect the markers on the desktop plane. In such a case, the markers on the plane cannot be recognized. If all the markers exist on almost the same plane like the first experiment, it even fails recognition for most of the markers. In the second experiment, however, some markers can always be recognized properly because the markers face to various directions. Therefore the registration can be continued stably by selecting only those markers. In this way, it becomes possible to arrange the markers at various positions and poses without measuring their geometry by using our method.

Furthermore, the process of selecting the reference images and constructing the projective space takes only less than 30 seconds, and frame rate of overlay process is about

$10 \sim 12$ fps. Therefore, we can say that our system is applicable without extra efforts and realizes a real-time on-line AR system easily.

5. Conclusion

We presented a real-time on-line AR system based on multiple-marker-based registration. The multiple markers used in our system can be placed at arbitrary positions and poses, so that we do not need to measure their geometrical arrangement in advance, because of the merging process via a projective space. Therefore this system can be constructed very easily, unlike most marker-based approaches which require a priori knowledge about the multiple markers. Furthermore we can move a camera freely according to the widely moving virtual object, and the registration can be continued independently of the camera's view point. Placing the markers at any positions enables to achieve the registration in the wide area. Making the markers face to various directions also allows keeping the accurate registration stably. Thus our system can highly contribute as an effective AR system.

References

[1] R. T. Azuma. A survey of augmented reality. *Presence*, pages 355–385, 1997.

[2] R. T. Azuma. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, Nov-Dec 2001.

[3] M. Billinghurst, S. Cambell, I. Poupyrev, K. Takahashi, H. Kato, W. Chinthammit, and D. Hendrickson. Magic book: Exploring transitions in collaborative ar interfaces. *Proc. of SIGGRAPH 2000*, page 87, 2000.

[4] M. Billinghurst and H. Kato. Collaborative mixed reality. In *Proc. of ISMR*, pages 261–284, 1999.

[5] K. W. Chia, A. Cheok, and S. J. D. Prince. Online 6 DOF augmented reality registration from natural features. In *Proc. of the ISMAR*, pages 305–313, 2002.

[6] D. Claus and A. W. Fizgibbon. Reliable automatic calibration of a marker-based position tracking system. In *Proc. of the WACV*, pages 300–305, 2005.

[7] E.J.Umlauf, H. Piringer, G. Reitmayr, and D. Schmalstieg. ARLib: The augmented library. In *Proc. of the ART02*, pages TR–188–2–2002–10, 2002.

[8] E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington. Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision. In *Proc. of the ISMAR*, pages 212–221, 2004.

[9] E. Foxlin and L. Naimark. Miniaturization, calibration & accuracy evaluation of a hybrid self-tracker. In *Proc. of the ISMAR*, pages 151–160, 2003.

[10] E. Foxlin and L. Naimark. Vis-traker: A wearable vision-inertial self-tracker. In *Proc. of Virtual Reality*, pages 199–206, 2003.

[11] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. navab. Marker-less tracking for ar: A learning-based approach. In *Proc. of the ISMAR*, pages 295–304, 2002.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision.* CAMBRIDGE UNIVERSITY PRESS, 2000.

[13] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proc. of the ISAR*, pages 111–119, 2000.

[14] D. Kotake, S. Uchiyama, and H. Yamamoto. A marker calibration method utilizing a priori knowledge on marker arrangement. In *Proc. of the ISMAR*, pages 89–98, 2004.

[15] R. Miguel, P. Lang, H. Ganster, M. Brandner, C. Stock, and A. Pinz. Hybrid tracking for outdoor augmented reality applications. *IEEE Computer Graphics and Applications*, pages 54–63, Nov-Dec 2002.

[16] U. Neumann and S. You. Natural feature tracking for augmented reality. *IEEE Trans. on Multimadia*, 1(1):53–64, 1999.

[17] S. Prince, A. D. Cheok, F. Farbiz, T. Williamson, N . Johnson, M. Billinghurst, and H. Kato. 3d live: Real time captured content for mixed reality. In *Proc. of the ISMAR*, pages 7–13, 2002.

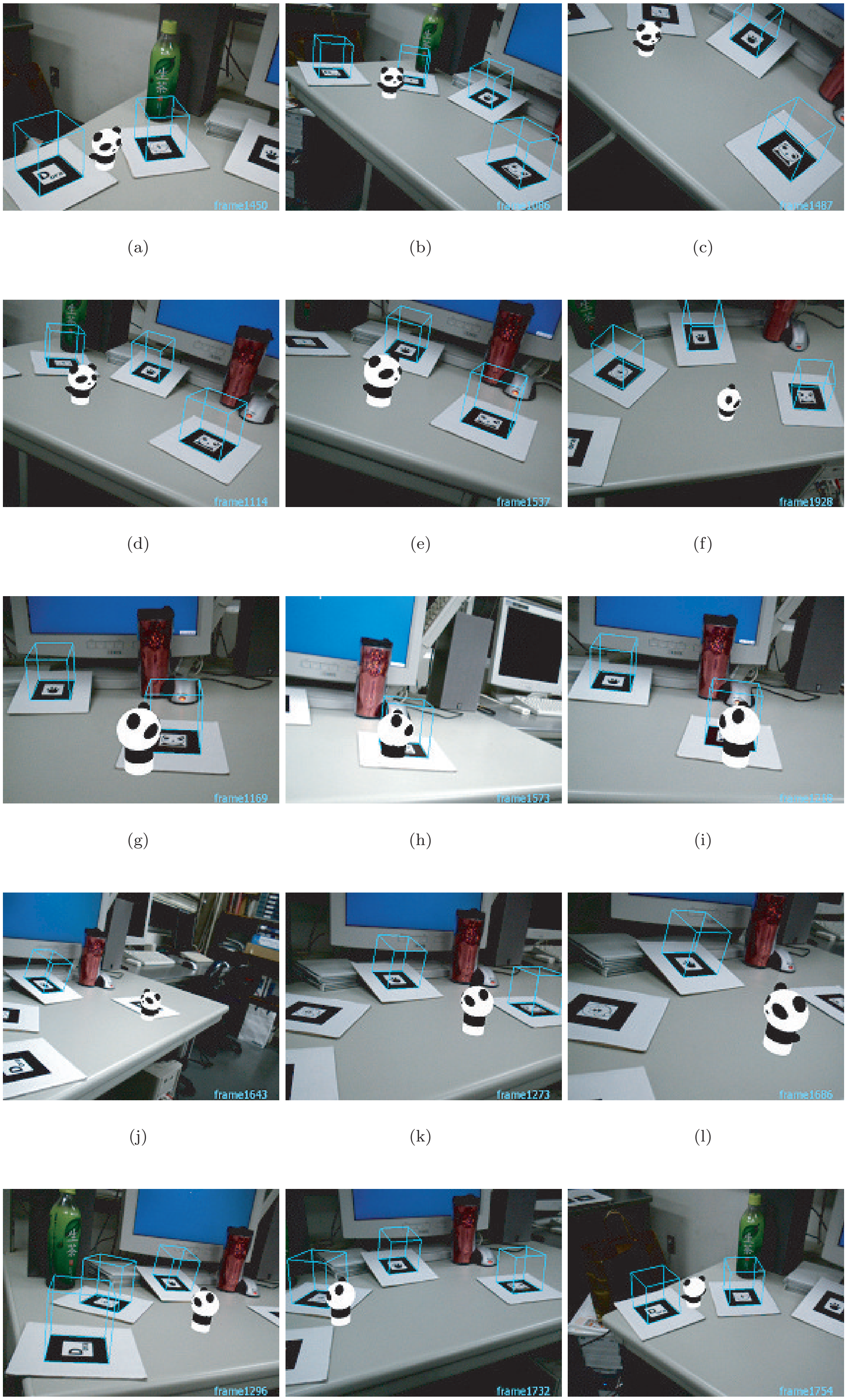
[18] G. Simon and M. Berger. Reconstructing while registering: a novel approach for markerless augmented reality. In *Proc. of the ISMAR*, pages 285–294, 2002.

[19] G. Simon and M. O Berger. Real time registration known or recovered multi-planar structures: application to ar. In *Proc. of the BMVC*, pages 567–576, 2002.

[20] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. of the ISAR*, pages 120–128, 2000.

[21] S. Uchiyama, K. Takemoto, H. Yamamoto, and H. Tamura. MR platform: A basic body on which mixed reality applications are built. In *Proc. of the ISMAR*, pages 246–253, 2002.

[22] Y. Uematsu and H. Saito. Vision based registration for augmented reality using multi-planes in arbitrary position and pose by moving uncalibrated camera. In *Proc. of Mirage2005*, pages 111–119, 2005.



(a) (b) (c)

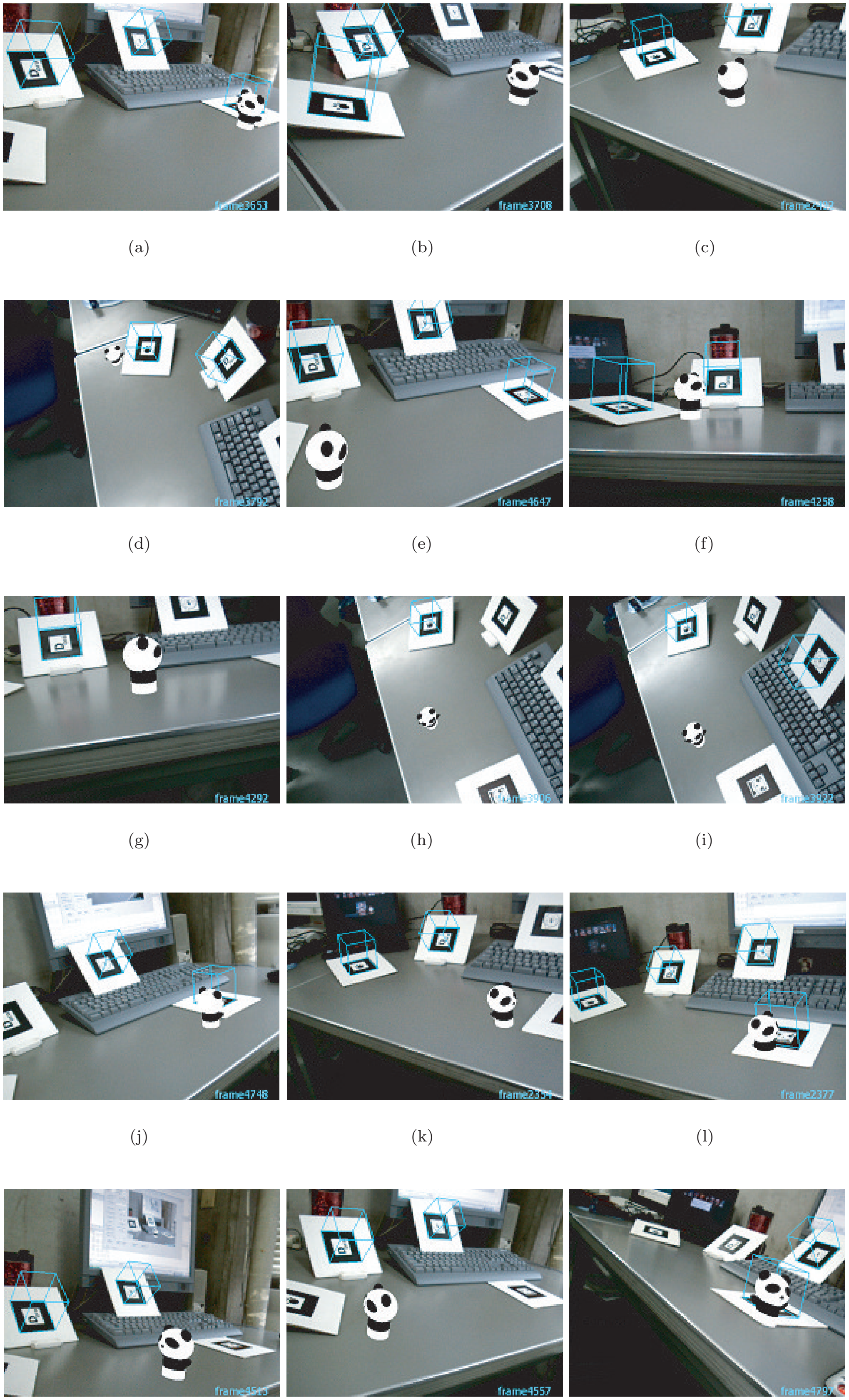
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 6: Overlaid image sequence of a virtual object.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 7: Overlaid image sequence of a virtual object.