VISION BASED REGISTRATION FOR AUGMENTED REALITY USING MULTI-PLANES IN ARBITRARY POSITION AND POSE BY MOVING UNCALIBRATED CAMERA

Yuko Uematsu and Hideo Saito

{yu-ko,saito}@ozawa.ics.keio.ac.jp
Keio University, Dept. of Information and Computer Science, Yokohama, Japan

ABSTRACT

This paper presents a novel AR registration method for overlaying a virtual object onto real world by multiple planes in arbitrary positions and poses. In this method, we do not require neither artificial markers nor sensors, but natural feature points on arbitrary multiple planes. We assign a 3D coordinate system for each plane independently. For each coordinate system, the projection matrix is computed in order to relate the coordinate system to the input images. Then, all the 3D coordinates are integrated into a projective 3D coordinate space that is defined by two reference images. Such integration can reduce errors in registration of a virtual object with the real world coordinate. For demonstrating the effectiveness of the proposed method, some experiments for overlay of virtual objects into image sequences taken with a handy video camera are performed. Accuracy evaluation by the number of planes via a synthesized image sequence is also performed, which indicates that the integration of a number of planes reduces the registration error. Additionally, comparison of proposed method with one of related work is performed. Our approach can contribute to image generation applicable to movie, sports casting, TV show, and so on.

1. INTRODUCTION

Augmented Reality (AR) / Mixed Reality (MR) allows users to see the real world with virtual objects superimposed onto the real world. Thus AR can provide the users with more effective view [1, 2]. In movies and TV programs, adding the virtual objects to images taken in advance can create new effect such as the images have been shot in the place which are hard to go to or cannot obtain the permission to shoot film. It is called "post-production" and often applied to media production recently.

One of the most important issues for AR is geometrical registration between the real and the virtual world. In order to achieve correct registration, accurate measurements of the camera rotations and translations (correspond user's view) are required. For the measurements, some kind of sensors such as magnetic or gyro sensors can be used. The registration by positioning sensors is stable against the change of light condition and effective especially when a camera moves rapidly. However, the rotations and translations obtained only from positioning sensors are not enough accurate to achieve perfect geometrical registration. Furthermore, the use of sensors has some limitations in practice: user's movable area, perturbation caused by the environment, and so on. Thus, when using such sensors, it is necessary to employ a hybrid system that combines vision-based method and sensors [3–5].

On the other hand, vision-based registration does not require any special devices expect cameras. Therefore AR system can be constructed easily. If the vision-based registration works well, the augmentation results are generally more accurate than the results obtained only from sensors. Vision-based registration relies on the identification of features in the images. Typically artificial markers placed in the real world [6–9], model-based [10–13], and / or natural features [14, 15] are used for the registration. Since markers are designed to be easily detectable, the registration based on the artificial markers generally works well. However, arranging the markers takes extra efforts. It also limits user's moving range, and gives bad visual appearance. For avoiding using the artificial markers, we may take model-based approach [10, 13], which can also provide high accuracy, but it is hard to construct the system simply because of the model generation. On the contrary, the registration based on natural features has few limitations, so it is suitable for using outdoor environments and holds grate promise in the future.The related works based on natural features have used various features: feature points [6, 14, 15], edges [16] or curves [17]. Neumann et al. [14] have applied optical-flow of natural features to estimation of the motion of the camera. Chia et al. [15] have proposed on-line AR registration. Their approach based on only epipolar geometry (fundamental matrices) between two reference cameras, which are calculated by tracked natural features, so construction of the AR system is very easy. However, it is commonly true that few features available for registration in the real world. As a result, the augmentation becomes unstable and generates tracking jitters. For reducing such instability, still more effective and stronger constraint should be employed.

The registration using planes [18–21] attracts attention re-

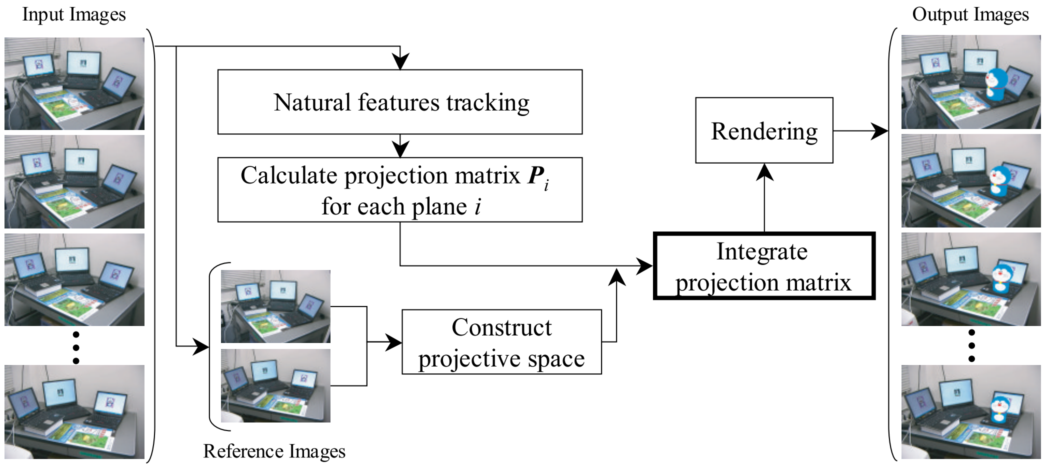


Figure 1: *Overview of the proposed method.*

cently. Using planar structures of the scene is very significant approach and gives effectively restricted conditions, because a lot of planes exist indoor or urban environment. Ferrari et al. [21] have tracked a planar region in order to overlay virtual textures onto the region. The textures automatically deform their shape with changing view point by affine transformation. This approach, however, can only overlay 2D textures. Simon et al. [18–20] have proposed related approaches. They constructed AR systems using multiple planes such as room's floor and walls or wall surfaces of buildings. In [19], they estimated projection matrix by multiple planes which are perpendicular to the reference plane, using uncalibrated camera . In [20], they estimated projection matrix using calibrated camera by multiple planes of arbitrary position and pose. In this method, the geometrical relationship between these planes and motion of the camera are calculated by bundle adjustment which carries out over all frames.

In this paper, we propose a method for vision-based registration using multiple planes in arbitrary position and pose by uncalibrated camera. Our approach does not require any information on physical relationship of the planes. We estimate the relationship of the multiple planes by constructing "projective space". The main contribution of our method is to construct the projective space with two reference images for estimation of geometrical relationship among the planes and camera. The constructed projective space provides geometrical relationship of the planes even if the planes are not perpendicular and intrinsic parameters are unknown. Furthermore our method can estimate the camera motion frame by frame as long as the projective space is defined.

2. OVERVIEW

This section explains the algorithm for our registration method using multiple planes in arbitrary position and pose. Actually a lot of planes exist indoors or urban environment, thus using planes is useful for AR registration which does not require any artificial markers. We assign a 3D coordinate system for each plane. Since the geometrical relationship among the planes is unknown, the 3D coordinate systems are independent to each other. We define a projective 3D space by two reference images. (This reference images are selected from input image sequence.) By using this projective 3D space, independent coordinate systems for each plane are related and the profit of multiple planes can be taken. Figure 1 describes an overview of the proposed method. Input image sequence is taken with an uncalibrated video camera. First, the natural features are tracked for the input image sequence in which n planes exist. The KLT-feature-tracker [22] is used for the tracking of the natural features. We assume that the extracted feature points are segmented into areas of planes. In this paper, we do not focus on the method for the segmentation, so we segment them by manual. Using the features on each real world plane, a homography that relates the plane to image plane is computed for each plane. Next, a projection matrix that relates a 3D coordinate on the plane to the image is computed by extending the homography to three dimensions from two dimensions. Hence, the projection matrices are computed for all planes. Then, these projection matrices are integrated through the projective space in order to reduce the error, which may be included in the matrices, and to relate the coordinate system of each plane. When the projection matrices are integrated, we can get the coordinates, which overlay a virtual object.

Finally, a virtual object is rendered onto output images.

3. REGISTRATION WITH MULTIPLE PLANES

3.1. Assign 3D coordinate systems

In this method, we first assign a 3D coordinate system for each plane in the 3D real world independently. As shown in figure 2, each coordinate system is defined as setting each plane to $Z=0$. This is for computing the projection matrix. The detail of computing will be described in the next section.

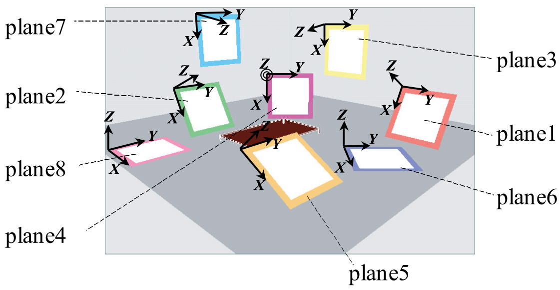


Figure 2: *Example of assigning 3D coordinate systems.*

3.2. Calculate projection matrix

A projection matrix $\boldsymbol{P}$ is the 3×4 matrix which relates 3D real world to 2D image plane, so that a 3D point $\boldsymbol{X}\simeq(X,Y,Z,1)$ is projected to a 2D point $\boldsymbol{x}\simeq(x,y,1)$ as follows.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \simeq \boldsymbol{P}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \simeq \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{1}$$

When $Z=0$, that is a 3D point exists on the plane which is setting to $Z=0$ and it has a form $\boldsymbol{X}\simeq(X,Y,0,1)$, equation (1) will be

$$\begin{aligned} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} &\simeq \boldsymbol{P}\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} \simeq \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} \\ &\simeq \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \simeq \hat{\boldsymbol{P}}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \end{aligned} \tag{2}$$

where $\hat{\boldsymbol{P}}$ is the 3×3 matrix, the thrid column of $\boldsymbol{P}$ has been deleted. Therefore, $\hat{\boldsymbol{P}}$ relates a 2D point to a 2D point and is equivalent to a planar homography $\boldsymbol{H}$. Then, we can write

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \simeq \hat{\boldsymbol{P}}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \simeq \boldsymbol{H}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{3}$$

$$\therefore\ \hat{\boldsymbol{P}} = \boldsymbol{H}$$

Thus, if the homography between the plane in 3D real world and 2D image is known, we can compute the projection matrix from $\boldsymbol{H}$. In the following, the process based on this theory will be described.

3.2.1. *Computing homography*

Figure 3 shows the homography between the 3D world plane and the image plane. In order to compute a homography, more than four points that is on the plane are required. We use the natural features points tracked by KLT-feature-tracker [22].

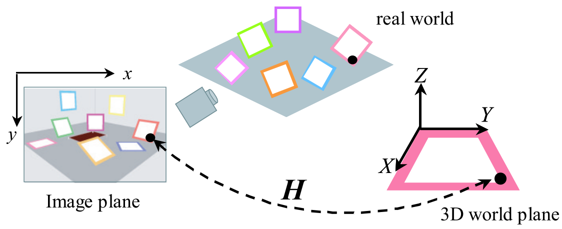


Figure 3: *Homography between the world plane and the image plane.*

3.2.2. *Estimate intrinsic parameters*

Since we take the input image sequence by an uncalibrated camera, we need to estimate the intrinsic parameters of the camera. We approximate by fixing skew to 0, aspect ratio to 1 and principal point to the center of the image, and the intrinsic parameters can be defined as follows.

$$\boldsymbol{A} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{array}{cl} (c_x, c_y) & : \text{principal point} \\ f & : \text{focal length} \end{array}$$

Then, we explain how to compute the focal length f. A projection matrix $\boldsymbol{P}$ is

$$\boldsymbol{P} = \boldsymbol{A}\left[\boldsymbol{R}\,|\,\boldsymbol{t}\right] = \boldsymbol{A}\left[\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{r_3}\ \boldsymbol{t}\right] \tag{4}$$

where $\boldsymbol{R}$ is a 3×3 rotation matrix, $\boldsymbol{t}$ is a translation vector of the camera, and $\boldsymbol{r_1}$, $\boldsymbol{r_2}$, $\boldsymbol{r_3}$ are column vectors of $\boldsymbol{R}$. Thus, we can write with equation (3)

$$\hat{\boldsymbol{P}} = \boldsymbol{A}\left[\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{t}\right] = \boldsymbol{H} \tag{5}$$

$$\left[\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{t}\right] = \boldsymbol{A}^{-1}\boldsymbol{H} \tag{6}$$

According to the property of $\boldsymbol{R}$, the inner product of $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ is set to 0. Therefore, we can obtain the focal length, if the homography $\boldsymbol{H}$ has this form,

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{7}$$

$$f^2 = \frac{(h_{11}-c_x h_{31})(h_{12}-c_x h_{32})+(h_{21}-c_y h_{31})(h_{22}-c_x h_{32})}{-h_{31}h_{32}} \tag{8}$$

3.2.3. Estimate extrinsic parameters

Extrinsic parameters of a camera consist of a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$. Since $\boldsymbol{r_1}$, $\boldsymbol{r_2}$ (the first and second column vectors of $\boldsymbol{R}$) and $\boldsymbol{t}$ are already known, we should estimate only $\boldsymbol{r_3}$. Then, according to the property of $\boldsymbol{R}$, we compute $\boldsymbol{r_3}$ since $\boldsymbol{r_3}$ is set to the cross product of $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$. Therefore, $\boldsymbol{R}$ is

$$\boldsymbol{R} = [\boldsymbol{r_1}\ \boldsymbol{r_2}\ (\boldsymbol{r_1} \times \boldsymbol{r_2})] \tag{9}$$

By performing such processing to all planes, projection matrices based on X-Y coordinate system assigned each plane are computed.

3.3. Integrate projection matrix

In the previous section, the projection matrix is computed for each plane. Although each matrix is reliable around each plane, as the position of a virtual object moves away from each plane, the accuracy becomes lower. Therefore, we will integrate the projection matrices of all planes and compute more accurate matrix at any place in the image and reduce registration errors. We construct a 3D projective space in order to integrate the projection matrices. Two reference images are picked up from the input image sequence (usually first image and last image). The projective space is defined these referece images and it is a common 3D coordinate system for the whole input images.

Figure 4 shows the relationship of the coordinate systems among the real world, the projective space and the image. $\boldsymbol{P_k}$ is the projection matrix from kth plane, $\boldsymbol{T_k^{WP}}$ is a 4×4 transformation matrix which relates the real world of kth plane to the projective space, and $\boldsymbol{T_k^{PI}}$ is a 3×4 transformation matrix which relates the projective space to the image. When we know $\boldsymbol{T_k^{WP}}$, we can compute $\boldsymbol{T_k^{PI}}$ using the projection matrix $\boldsymbol{P_k}$ as the following relationship.

$$\boldsymbol{T_k^{PI}} = \boldsymbol{P_k} \boldsymbol{T_k^{WP}}^{-1} \tag{10}$$

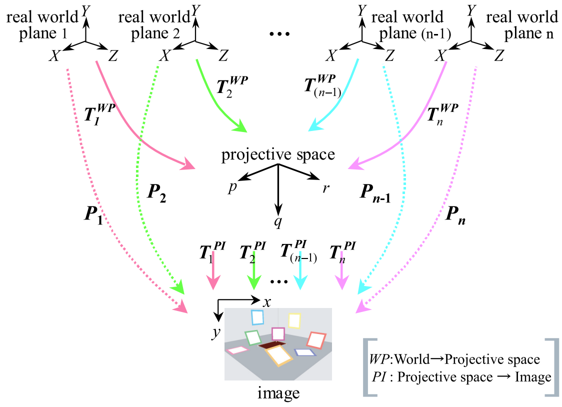


Figure 4: *Relationship among 3 coordinate sysmtes.*

$\boldsymbol{T_k^{PI}}$ is the transformation from the projective space to the image that is computed by kth plane, but it should theoretically be unique no matter which plane is used for computing it. Therefore, we integrate all $\boldsymbol{T_k^{PI}}$ computed by all planes for the unique $\boldsymbol{T^{PI}}$.

In our method, we define two projective space, "by projective grid" [23] and "by projective reconstruction". In the following, we descrive how to compute $\boldsymbol{T_k^{WP}}$ and $\boldsymbol{T_k^{PI}}$, define the two projective space, and compute a coordinate point in the space corresponding to one in the real world.

3.3.1. Calculate $\boldsymbol{T_k^{WP}}$

Consider $\boldsymbol{C_w} \simeq (X_j, Y_j, Z_j, 1)$ as a point on the kth plane in the real world, and $\boldsymbol{C_p} \simeq (p_j, q_j, r_j, 1)$ as a point in the projective space, we can write the relation of the two coordinate system by

$$\boldsymbol{T_k^{WP}} = \begin{bmatrix} \boldsymbol{t_1} \\ \boldsymbol{t_2} \\ \boldsymbol{t_3} \\ \boldsymbol{t_4} \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \tag{11}$$

$$\boldsymbol{C_p} \simeq \boldsymbol{T_k^{WP}} \boldsymbol{C_w} \tag{12}$$

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

$$\boldsymbol{M_j t} = \boldsymbol{b_j} \tag{13}$$

$$\boldsymbol{M_j} = \begin{bmatrix} \boldsymbol{C_{w_j}^\top} & \boldsymbol{0} & \boldsymbol{0} & -X_j p_j & -Y_j p_j & -Z_j p_j \\ \boldsymbol{0} & \boldsymbol{C_{w_j}^\top} & \boldsymbol{0} & -X_j q_j & -Y_j q_j & -Z_j q_j \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C_{w_j}^\top} & -X_j r_j & -Y_j r_j & -Z_j r_j \end{bmatrix} \tag{14}$$

$$\boldsymbol{t} = \begin{bmatrix} \boldsymbol{t_1} & \boldsymbol{t_2} & \boldsymbol{t_3} & t_{41} & t_{42} & t_{43} \end{bmatrix}^\top \tag{15}$$

$$\boldsymbol{b_j} = \begin{bmatrix} p_j \\ q_j \\ r_j \end{bmatrix} \tag{16}$$

Therefore, if the corresponding m points ($m \geq 5$) in the real world and projective space are given, equation (13) is

$$\begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix} \boldsymbol{t} = \begin{bmatrix} \boldsymbol{b_1} \\ \vdots \\ \boldsymbol{b_m} \end{bmatrix} \tag{17}$$

Thus, by least-square method, we can write

$$\boldsymbol{t} = \begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^{+} \begin{bmatrix} \boldsymbol{b_1} \\ \vdots \\ \boldsymbol{b_m} \end{bmatrix} \tag{18}$$

However, it is referred to as

$$\begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^{+} = \left(\begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{M_1} \\ \vdots \\ \boldsymbol{M_m} \end{bmatrix}^{\top} \tag{19}$$

If $m = 5$, any combination of three points in the five points must not be aligned on the same straight line. Any combination of four points must not also be placed on the same plane.

3.3.2. *Calculate $\boldsymbol{T_k^{PI}}$*

When $\boldsymbol{T_k^{WP}}$ is known, we can compute $\boldsymbol{T_k^{PI}}$ by equation (10). Thus, we compute $\boldsymbol{T_1^{PI}} \sim \boldsymbol{T_n^{PI}}$ for each plane. Those matrices $\boldsymbol{T_1^{PI}} \sim \boldsymbol{T_n^{PI}}$ should be same, if every measurement is completely accurate. However, they are slightly different each other because of errors in previous procedures, such as tracking feature points. The integration of those matrices will provide a more accurate projection matrix relating the projective space and the image plane. Because those matrices are almost the same, we assume that merging them can approximately provide the integrated matrix. In the merging computation, we take weights $w_1 \sim w_n$, which are determined according to the distance from the point to superimpose to 3D coordinate origin of each plane.

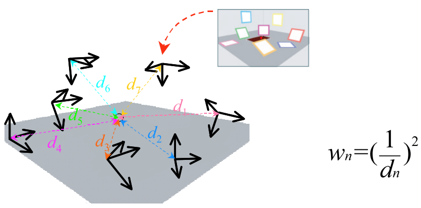


Figure 5: *Weight for each 3D coordinate system.*

$$\boldsymbol{T_k^{PI}} = \begin{bmatrix} t_{k_{11}} & t_{k_{12}} & t_{k_{13}} & t_{k_{14}} \\ t_{k_{21}} & t_{k_{22}} & t_{k_{23}} & t_{k_{24}} \\ t_{k_{31}} & t_{k_{32}} & t_{k_{33}} & t_{k_{34}} \end{bmatrix} \tag{20}$$

$$\boldsymbol{T^{PI}} = \frac{1}{n} \sum_{k=1}^{n} w_k \begin{bmatrix} t_{k_{11}} & t_{k_{12}} & t_{k_{13}} & t_{k_{14}} \\ t_{k_{21}} & t_{k_{22}} & t_{k_{23}} & t_{k_{24}} \\ t_{k_{31}} & t_{k_{32}} & t_{k_{33}} & t_{k_{34}} \end{bmatrix} \tag{21}$$

Such integration of the matrices can be regarded as integration of the tracked planes.

3.3.3. *Calculate coordinate points in projective space*

In order to calculate $\boldsymbol{T_k^{WP}}$, the corresponding m points ($m \geq 5$) in the real world and projective space are required. We explain the definition of the projective space and how to calculate coordinate points in the projective space by real world and image coordinate.

By projective grid

A projective space by projective grid is called a Projective Grid Space (PGS) [23]. PGS is a 3D space that is constructed by two reference images captured by two reference cameras shown in figure 6(a). The three axes of the space are expressed by $\boldsymbol{P}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$, which are the horizontal and vertical axes ($\boldsymbol{U_A}$, $\boldsymbol{V_A}$) of the reference image A, and the horizontal axis ($\boldsymbol{U_B}$) of the reference image B, respectively.

Consider $\boldsymbol{C_P} \simeq (p, q, r, 1)^{\top}$ as a 3D point in PGS, $\boldsymbol{C_A} \simeq (p, q, 1)^{\top}$ and $\boldsymbol{C_B} \simeq (r, v_B, 1)^{\top}$ as a 2D point on the reference image A, B respectively shown in figure 6(b). $\boldsymbol{C_A}$ is back-projected into the space as a 3D line and a vertical line going through $\boldsymbol{C_B}$ is also back-projected as a 3D line. In fact, $\boldsymbol{C_P}$ is the intersection $(u_A, v_A, u_B, 1)$ of the line and plane in PGS. Thus, five (or more) points are computed in this projective space.

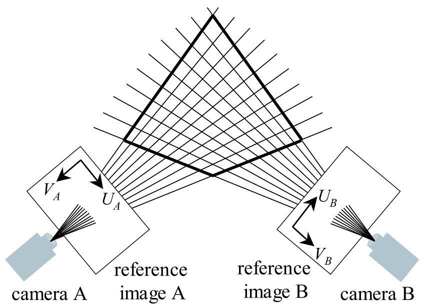


Figure 6: *Projective grid space (PGS).*

By projective reconstruction

A projective space by projective reconstruction is constructed by two reference images captured by two reference cameras shown in figure 7. When epipolar geometry between the reference images (cameras) is known, the relationship between the projective space and reference images is respectively

$$\begin{cases} \boldsymbol{P_A} = [\mathbf{I} \mid \mathbf{0}] \\ \boldsymbol{P_B} = [\boldsymbol{M e_B}] \end{cases} \tag{22}$$

$$\boldsymbol{M} = -\frac{[\boldsymbol{e_B}]_\times \boldsymbol{F_{AB}}}{\|\boldsymbol{e_B}\|^2} \tag{23}$$

where $\boldsymbol{F_{AB}}$ is a fundamental matrix of image A to B, and $\boldsymbol{e_B}$ is an epipole on the image B. Consider $\boldsymbol{C_p} \simeq [p, q, r, 1]^\top$ as a point in the projective space,
$\boldsymbol{C_A} \simeq [u_A, v_A, 1]^\top$ as on the image A, $\boldsymbol{C_B} \simeq [u_B, v_B, 1]^\top$ as on the image B, we can write

$$\boldsymbol{Q C_P} = \mathbf{0} \tag{24}$$

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{P_A^1} - u_A \boldsymbol{P_A^3} \\ \boldsymbol{P_A^2} - v_A \boldsymbol{P_A^3} \\ \boldsymbol{P_B^1} - u_B \boldsymbol{P_B^3} \\ \boldsymbol{P_B^2} - v_B \boldsymbol{P_B^3} \end{bmatrix} \tag{25}$$

$\boldsymbol{P^i}$ is the ith column vector of $\boldsymbol{P}$. Then, we obtain $\boldsymbol{C_p} \simeq [p, q, r, 1]^\top$ by the singular value decomposition of $\boldsymbol{Q}$. Thus, five (or more) points are computed in this projective space.

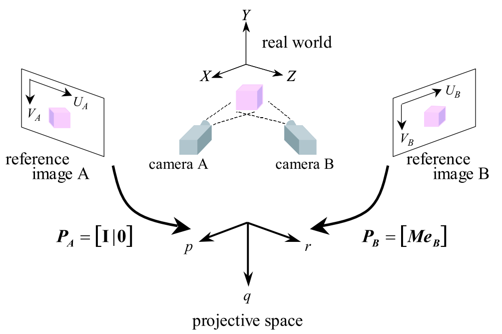


Figure 7: *Projective space by projective reconstruction.*

4. EXPERIMENTAL RESULTS

We implement the augmented reality system based on our method using only a PC (OS:Windows 2000, CPU:Intel Pentium Ⅳ 3.20GHz) and a CCD camera (SONY DCR-TRV900). The images in all the experiments are 720×480 pixels, and graphical views of a virtual object are rendered with OpenGL library.
The process of implementation is described below. First, the camera image sequence is captured by a hand-held uncalibrated camera. The first and last images of the sequence are selected for construction of the projective space (the reference images). Secondly, natural features in the scene are tracked by KLT. Then the extracted feature points are segmented into groups, so that the feature points in a group are located on a same plane. Next, a 3D coordinate system is independently assigned for each plane. Then homographic matrix between the 3D coordinate on each plane and the image plane is computed by the extracted feature points on the plane. At the same time, we also compute fundamental matrices between the reference images for constructing the projective space based projective reconstruction by correspondence of the extracted features points. Additionally, projection matrices are computed by the homographies of each planes, and integrated using the projective space constructed by the reference images. Then, a virtual object is rendered and overlaid onto each frame of the image sequence.
The overlaid AR camera images produced by the augmentation are shown in figure 9, 10. As figure 8 shows, figure 9 uses 4 planes, and figure 10 uses 3 planes. Both result images are generated using the projective space based on projective reconstruction. Our approach can superimpose a virtual object onto the image sequence successfully.

Next, in order to evaluate the registration accuracy in our method, we perform the same implementation for the synthesized image sequence rendered with OpenGL (figure 8 right column). Since we have to know the exact position and pose of a camera to evaluate accuracy, we use the synthesized images. The overlaid images are shown in figure 11. A virtual object overlaid is a cube.

Figure 11 and 12 show the results of the evaluation experiment. Figure 11 shows the images in result sequences which are overlaid by 1 plane, 8 planes (PGS) and 8 planes (projective reconstruction) respectively, and figure 12 shows the comparison of x-y coordinates between the theoretical values and the results of 1 plane and 8 planes (projective reconstruction).

In figure 11, the back cube (theoretical cube) is in correct position. As compared with the result of 1 plane, the result of 8 planes by projective reconstruction is much more precise and indicates the effectiveness of our proposed method. As for the result by the PGS, although the position superimposed is almost exact, projected shape of the cube is a little distorted. This is because that the transformation between the PGS and 3D coordinate system on each plane cannot strictly be represented with linear projective relationship. However, computation of the geometrical transformation is faster than the case of projective reconstruction. Thus, registration with the PGS may be useful if the approximation of linearity in the transformation can be satisfied.

Figure 12 also shows that the result by 8 planes has less registration errors and jitters than using only 1 plane, in spite of no information about the relationship of the planes.

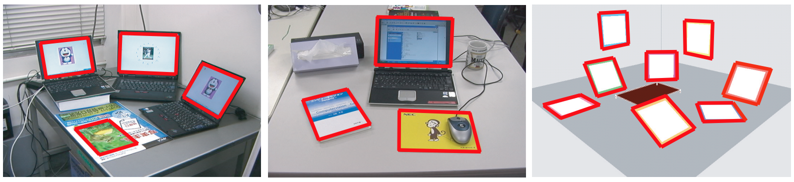
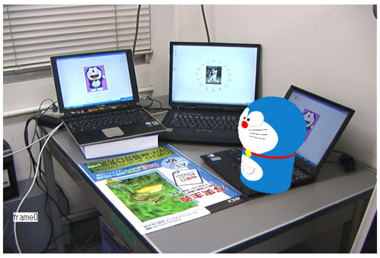
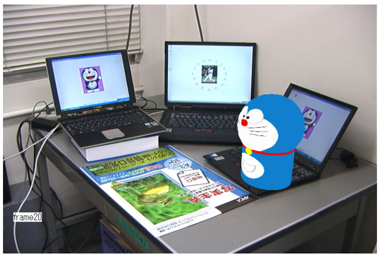


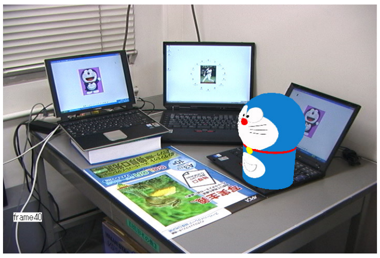
Figure 8: *Planes used for each experiment.*



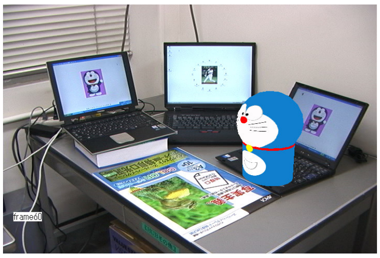
(a) *Frame0*



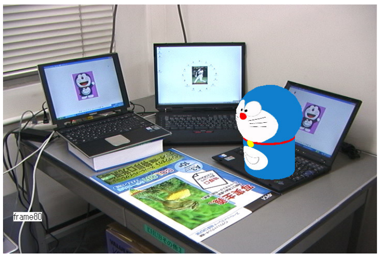
(b) *Frame20*



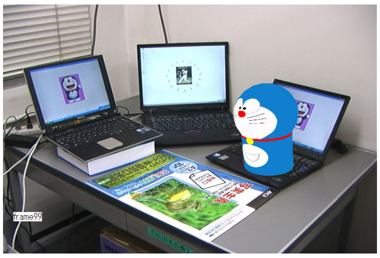
(c) *Frame40*



(d) *Frame60*

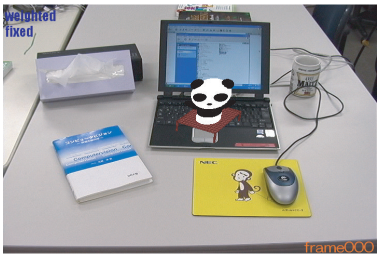


(e) *Frame80*

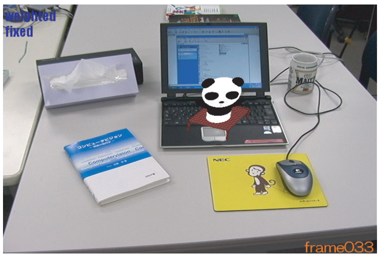


(f) *Frame100*

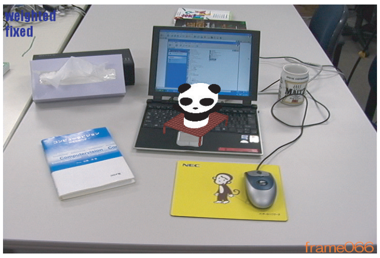
Figure 9: *Overlaid image sequence of a virtual object.*



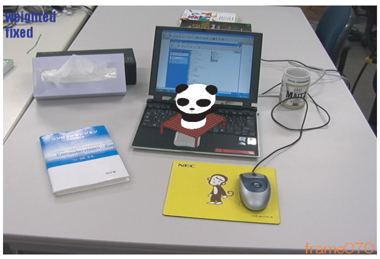
(a) *Frame0*



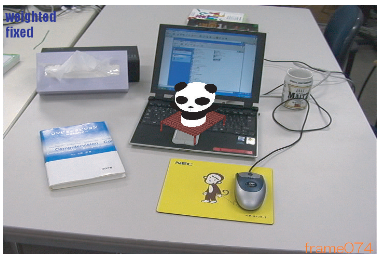
(b) *Frame33*



(c) *Frame66*



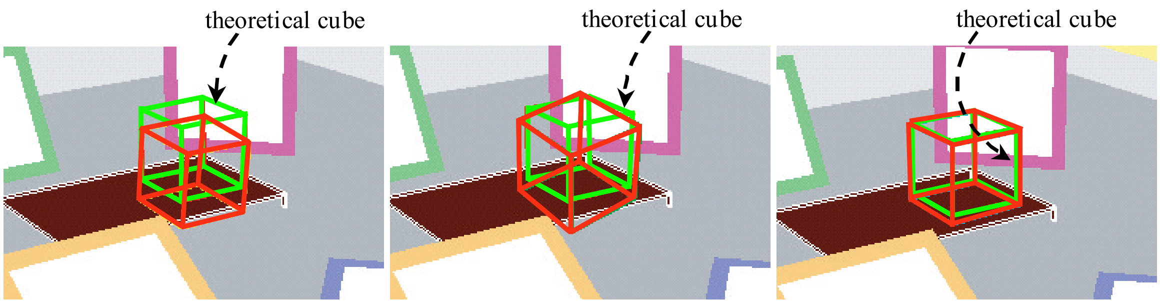
(d) *Frame70*



(e) *Frame74*

(f) *Frame99*

Figure 10: *Overlaid image sequence of a virtual object.*



(a) *1 plane* (b) *8 planes (PGS)* (c) *8 planes (Projective Reconstruction)*

Figure 11: *Overlaid images with the theoretical cube.*

This suggests that increasing the number of planar structure in the scene can improve the registration accuracy.

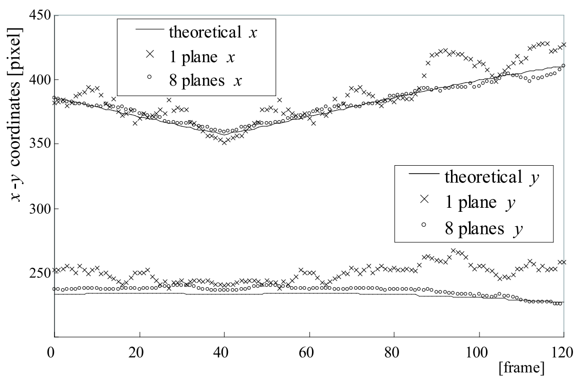


Figure 12: *Comparison of x-y coordinates between 1 plane and 8 planes with theoretical values.*

We also evaluate the proposed method by comparing with one of related works, Simon et al.'s method [19], in which multiple planes need to be perpendicular to the reference plane (that is one of multi-planes). For the comparison, we apply the image sequence (shown in figure 13) to this method and our method, and evaluate the registration accuracy. The result of the evaluation is shown in figure 14. Even though our method does not require any geometrical information of the plane, our method achieves almost same accuracy with Simon's method, in which the planes need to be perpendicular to the reference plane.

5. CONCLUSION

A geometrical registration method for AR system with uncalibrated camera based on multiple planes has been proposed in this paper. In our approach, the planes do not need to be perpendicular to each other. This means that any planes in arbitrary position and pose can be used for registration. Furthermore the registration can be performed frame by frame without using all frames in input image sequence. Thus we can construct the AR system easily, and overlay a virtual object onto the image sequence correctly.

Automatic segmentation of features into planes is currently under study. If the processing is automated, the system will become much more useful, furthermore, will be possible to perform on-line. This will be further interesting work.

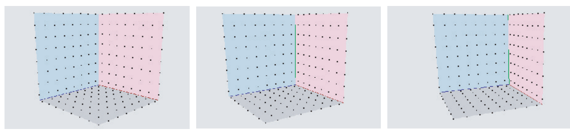


Figure 13: *Input images for comparing evaluation.*

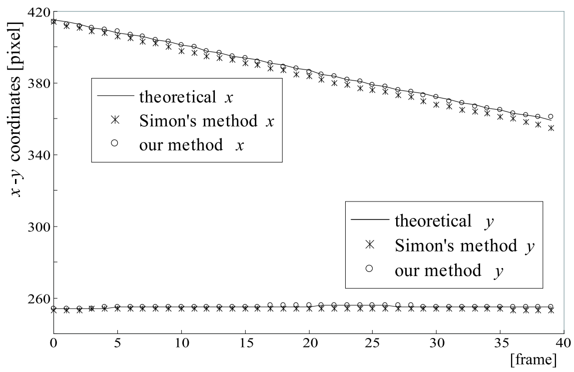


Figure 14: *Comparison of x-y coordinates between Simon's method and our method with theoretical values.*

6. REFERENCES

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

[2] R. T. Azuma, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, Nov-Dec 2001.

[3] J. P. Rolland, L. D. Davis, and Y. Baillot, "A survey of tracking technologies for virtual environments," *Fundamentals of Wearable Computers and Augmented Reality*, pp. 67–112, 2001.

[4] 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 IEEE and ACM ISMAR*, 2002, pp. 246–253.

[5] 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*, pp. 54–63, Nov-Dec 2002.

[6] A. State, G. Hirota, D. Chen, W. Garett, and M. Livingston, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," *Computer Graphics (Proceedings Siggraph New Orleans)*, pp. 429–438, 1996.

[7] Y. Seo and K. S. Hong, "Calibration-free augmented reality in perspective," *IEEE Trans. on Visualization and Computer Graphics*, vol. 6, no. 4, pp. 346–359, 2000.

[8] M. Billinghurst, S. Cambell, I. Poupyrev, K. Takahashi, H. Kato, W. Chinthammit, and D. Hendrickson, "Magic book: Exploring transitions in collaborative ar interfaces," *Computer Graphics Annual Conference Series, SIGGRAPH*, p. 87, 2000.

[9] K. Satoh, S. Uchiyama, H. Yamamoto, and H. Tamura, "Robust vision-based registration utilizing bird's-eye with user's view," in *Proc. of the IEEE and ACM ISMAR*, 2003, pp. 46–55.

[10] T. Drummond and R. Cipolla, "Real-time tracking of complex structures with on-line camera calibration," in *Proc. of the BMVC*, 1999, pp. 574–583.

[11] A. I. Comport and F. Marchand, E. Chaumette, "A real-time tracker for markerless augmented reality," in *Proc. of the IEEE and ACM ISMAR*, 2003, pp. 36–45.

[12] V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, "Fully automated and stable registration for augmented reality applications," in *Proc. of the IEEE and ACM ISMAR*, 2003, pp. 93–102.

[13] K. Klein and T. Drummond, "Sensor fusion and occlusion refinement for tablet-based ar," in *Proc. of the IEEE and ACM ISMAR*, 2004, pp. 38–47.

[14] U. Neumann and S. You, "Natural feature tracking for augmented reality," *IEEE Trans. on Multimadia*, vol. 1, no. 1, pp. 53–64, 1999.

[15] K. W. Chia, A. Cheok, and S. J. D. Prince, "Online 6 dof augmented reality registration from natural features," in *Proc. of the IEEE and ACM ISMAR*, 2002, pp. 305–313.

[16] D. Gennery, "Visual tracking of known three dimensional objects," *International Journal of Computer Vision*, vol. 7, no. 3, pp. 243–270, 1992.

[17] G. Simon and M. Berger, "A two-stage robust statistical method for temporal registration from features of various type," in *Proc. of 6th International Conference on Computer Vision*, 1998, pp. 261–266.

[18] G. Simon, A. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," in *Proc. of the IEEE and ACM ISAR*, 2000, pp. 120–128.

[19] G. Simon and M. Berger, "Reconstructing while registering: a novel approach for markerless augmented reality," in *Proc. of the IEEE and ACM ISMAR*, 2002, pp. 285–294.

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

[21] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Makerless augmented reality with a real-time affine region tracker," in *Proc. of the IEEE and ACM ISAR*, 2001, pp. 87–96.

[22] J. Shi and C. Tomasi, "Good features to track," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[23] H. Saito and T. Kanade, "Shape reconstruction in projective grid space from large number of images," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 49–54, 1999.

[24] 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 IEEE and ACM ISMAR*, 2002, pp. 7–13.

[25] M. Kanbara, N. Yokoya, and H. Takemura, "Registration for stereo vision-based augmented reality based on extendible tracking of markers and natural features," in *Proc. of 16th International Conference on Pettern Recognition*, 2002, pp. 1045–1048.

[26] M. Kanbara, N. Yokoya, and H. Takemura, "A stereo vision-based augmented reality system with marker and natural feature tracking," in *Proc. of 7th International Conference on Visual Systems and MultiMedia*, 2001, pp. 445–461.

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

[28] S. You, U. Neumann, and R. Azuma, "Hybrid inertial and vision tracking for augmented reality registration," in *Proc. of the IEEE Virtual Reality*, 1999, pp. 260–267.