Robust Vision-Based Registration Utilizing Bird's-Eye View with User's View

Kiyohide Satoh, Shinji Uchiyama, Hiroyuki Yamamoto, and Hideyuki Tamura[†]
MR Systems Laboratory, Canon Inc.
2-2-1 Nakane, Meguro-ku, Tokyo 152-0031, Japan
{sato.kiyohide, uchiyama.shinji, yamamoto.hiroyuki125}@canon.co.jp, HideyTamura@acm.org

Abstract

This paper describes new vision-based registration methods utilizing not only cameras on a user's head-mounted display but also a bird's-eye view camera that observes the user from an objective viewpoint. Two new methods, the Line Constraint Method (LCM) and Global Error Minimization Method (GEM), are proposed. The former method reduces the number of unknown parameters concerning the user's viewpoint by restricting it to be on the line of sight from the bird's-eye view. The other method minimizes the sum of errors, which is the sum of the distance between the fiducials on the view and the calculated positions of them based on the current viewing parameters, for both the user's view and the bird's-eye view. The methods proposed here reduce the number of points that should be observed from the user's viewpoint for registration, thus improving the stability. In addition to theoretical discussions, this paper demonstrates the effectiveness of our methods by experiments in comparison with methods that use only a user's view camera or a bird's-eye view camera.

1. Introduction

Registration between virtual and physical spaces is one of the most important technologies in augmented reality (AR). In the case of video see-through AR, the problem is equivalent to the pose estimation of a camera that observes the physical space. Early AR systems used a physical sensor such as a magnetic sensor to measure the position and orientation of the camera. However, with the dramatic increase in computing performance in recent years, the trend is shifting toward vision-based registration, i.e., the pose of the camera is estimated by using images taken by the camera itself.

The pose of the camera can be calculated based on the image coordinates of multiple feature points or fiducials on the photographed image as long as the world coordinates of these points are known. The method for finding the solution has long been known in the field of photogrammetry [1]. Although the registration has the same goal, AR places some restrictions on this kind of method, that is, all processing must be done automatically, and video rate processing is desirable. Hence, there is much discussion on the question of how these processes can be performed robustly within a certain time frame.

For example, some have proposed artificial markers [2][3] that would simplify the detection and identification. Others have suggested a robust estimation method addressing how the camera pose can be accurately obtained even when the input data contain errors [4][5]. In addition, hybrid registration, which combines inertial sensors with a vision-based method, was found to be effective in reducing the instability of vision [6][7]. Some values, particularly the orientation values, measured with inertial sensors can be trusted without adjustment and the data can be used as is, eliminating the degrees of freedom that the vision method should calculate [8].

In the field of motion capture, the 3D position of a target object is measured by multiple bird's-eye view cameras set up around the space. These cameras track retroreflective markers attached on the object. If more than three markers are installed on the rigid object, not only the position but also its orientation can be determined. Such a system is already in use as a tracker for virtual reality [9].

In this paper, we propose vision-based registration methods that use both the user's view camera attached on

[†] Current Affiliation: Faculty of Science and Engineering, Ritsumeikan University

an HMD and a bird's-eye view camera which takes images of the user from an objective viewpoint. The methods have both the advantages of the bird's-eye view and the user's view: the former contributes to stability and robustness of the registration process because changes in the user's viewpoint, such as caused by rotation of the head, do not affect the observation significantly, whereas the latter affects the accuracy of registration since virtual objects are superimposed directly on the user's view. These advantages of the proposed methods lead to robust, high-accuracy registration. In addition to theoretical discussions, this paper demonstrates the effectiveness of our methods by experiments in comparison with methods using only a user's view camera or a bird's-eye view camera.

2. Our approach

2.1. Basic configuration

The concept of the methods proposed in this paper is vision-based registration using both a camera from an objective viewpoint and a camera from the user's viewpoint[‡], as shown in Figure 1.

One bird's-eye view camera is set up at a position such that it can always observe the head of the user experiencing AR. We assume that the intrinsic parameters of this bird's-eye view camera are known and that its set-up position has already been measured. The HMD worn by the user has a marker (or markers) that is to be detected by this bird's-eye view camera (hereafter referred to as "head markers"). The number of head markers depends on which method will be used, among those listed later.

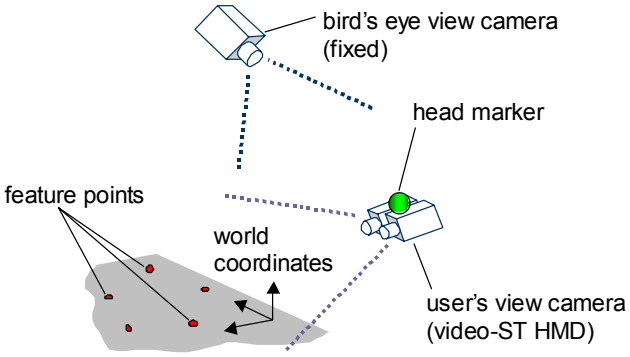


Figure 1. Concept of our approach

In the scene, feature points are set up to be observed and detected by the user's view camera installed in the HMD. These feature points can be either artificial markers (fiducials) or natural features automatically selected by an interest operator, but we assume that their coordinates in the world coordinate system are known.

The image coordinates of the head marker(s) are detected from the image taken by the bird's-eye view camera, and the image coordinates of the feature points are detected from the image taken by the user's view camera. Based on these results, the pose of the user's view camera is calculated as described below.

2.2. Overview of the registration methods

When the bird's-eye view camera detects one head marker, the world coordinates of this marker can be constrained on a straight line. We hereby propose two methods using this information for registration.

■ Line Constraint Method

The first method (hereafter referred to as the *Line Constraint Method* (*LCM*)) utilizes a step-by-step procedure, separating the vision process based on the bird's-eye view images and the vision process based on the user's view images. In this method, information from the bird's-eye view camera is used to reduce the number of unknown parameters that need to be determined using the user's view images.

First, the bird's-eye view camera detects one head marker so that the degrees of freedom of the camera position are reduced from three to one. Then, using user's view images, the four remaining unknown parameters (three parameters for the orientation and one for the position) are estimated. Because there are four unknown parameters, the number of feature points that need to be detected on the user's view image for a solution is two.

The *Line Constraint Method* relies entirely on the information obtained by the bird's-eye view camera. If this bird's-eye view camera is calibrated correctly, the position accuracy should be higher than with registration using only a user's view camera.

[‡] Similar concept using a head-mounted camera and a fixed optical sensor (not a camera) to track a target object was shown by Hoff [14]. Their main issue is how to integrate two 6DOF poses individually obtained form each sensor.

■ Global Error Minimization Method

The second method (hereafter referred to as the *Global Error Minimization Method* (*GEM*)) utilizes the user's view image and bird's-eye view image simultaneously. In this method, the registration error in the image coordinates of head markers detected by the bird's-eye view and the error in the image coordinates of feature points detected by the user's view are determined, and their sum is minimized in the registration process. In this method, a solution can be obtained as long as the sum of the number of detected head markers and the number of detected feature points is at least three.

2.3. Advantage of utilizing bird's-eye view

A direct advantage of using an additional camera from an objective viewpoint is that the number of feature points that need to be measured by the user's view camera is fewer compared to methods using only the user's view camera. Consequently, not only is the minimum number of feature points reduced, but also the solution obtained this way is more stable, given the same number of feature points. Hence, a second benefit is the reduction in the number of fiducial points set up in the environment.

As mentioned in Section 1 above, vision-based registration can be supported through inertial sensors. As far as the orientation is concerned, it is possible to obtain an acceptable level of accuracy even with the sensors alone. This is different with the position; it is still difficult to determine the position by integrating the measured acceleration, unless the period involved is very short. Eventually the determination must depend on the vision. The use of a bird's-eye view camera has the characteristic that it significantly reduces the degrees of freedom concerning the position by detecting only one marker. Because of this characteristic, it can provide additional information to vision-based registration, different from information given by inertial sensors.

Registration using only a bird's-eye view camera(s) is suitable for applications where the orientation of the head rapidly changes and/or the user looks around his surroundings, for this type of registration can be performed as long as the field of view contains multiple head markers. However, if a large area needs to be measured, the positional resolution of the bird's-eye view camera becomes too coarse, causing significant errors, particularly in the orientation. The registration method proposed here, on the other hand, has the property that, similar to hybrid registration using both a physical 6DOF sensor and vision, it is capable of performing closed-loop registration.

3. Registration by error-minimization

In the field of AR, a registration framework, which is to detect feature points on an image and to compute the pose of a user's view camera numerically by minimizing the error between these detected coordinates and their calculated coordinates, is generally used. Our registration methods also use a framework similar to this. In this section, we explain a general framework with which the camera pose is numerically computed by minimizing the error [10].[§]

3.1. Framework of error-minimization

Below, we denote the unknown parameters we are trying to determine by m-dimensional vector $\mathbf{s}$. Also, suppose that the coordinate transformation from the world to the image coordinate systems is given by function $f_{\mathbf{s}}$ determined by $\mathbf{s}$. We further assume that a number of feature points Q_i (where i is the index of each feature point) are set up in the world coordinate system and that their coordinates in this system are given by $\mathbf{x}_{\mathbf{WQ}_i}$ ($=[x_{WQ_i} \quad y_{WQ_i} \quad z_{WQ_i}]^T$).

The principle is as follows: the n feature points Q_i ($1 \le i \le n$) are detected on the user's view image, and the detected image coordinates $\mathbf{u}_{\mathbf{CQ}_i}$ are compared with the theoretical values

$$\tilde{\mathbf{u}}_{\mathbf{CQ}_i} = f_{\mathbf{s}}\left(\mathbf{x}_{\mathbf{WQ}_i}\right), \tag{1}$$

so that $\mathbf{s}$ can be obtained so as to minimize the sum of the errors $\Delta\mathbf{u}_{\mathbf{CQ}_i}$:

$$\sum_{i=1}^{n} \Delta\mathbf{u}_{\mathbf{CQ}_i} = \sum_{i=1}^{n} \left(\mathbf{u}_{\mathbf{CQ}_i} - \tilde{\mathbf{u}}_{\mathbf{CQ}_i}\right). \tag{2}$$

If the right-hand side of Equation (1) can be partially differentiated with respect to each of the entries of $\mathbf{s}$, an error-minimizing method (such as the Gauss-Newton method) can be used to obtain vector $\mathbf{s}$ that minimizes the sum of the errors as follows.

[§] The contents of this section are a publicly known technology, but we organize it here since this technology is the basis for our methods.

First, let some $\mathbf{s}$ be given as an initial value. Find $\tilde{\mathbf{u}}_{\mathbf{CQ}_i}$ by using the given $\mathbf{s}$ and Equation (1) for each feature point, and calculate the error $\Delta\mathbf{u}_{\mathbf{CQ}_i}$ between the theoretical and the detected coordinates. Define $\mathbf{E_s}$ to be the $2n$-dimensional vector formed by listing all these entries vertically in one column. Next, for each feature point, calculate the 2-by-m Jacobian matrix $\mathbf{J}_{\mathbf{usQ}_i}$ (known as the "image Jacobian") by listing the partial derivatives of the right-hand side of Equation (1) with respect to all the entries of $\mathbf{s}$. Then, define the $2n$-by-m matrix $\mathbf{\Phi_s}$ by listing all of the image Jacobian in a vertical manner. Here, we have the relation $\mathbf{E_s} \approx \mathbf{\Phi_s} \cdot \Delta\mathbf{s}$, so the correction value $\Delta\mathbf{s}$ for $\mathbf{s}$ can be obtained by the following equation using $\mathbf{\Phi_s}^*$, the pseudo-inverse of $\mathbf{\Phi_s}$:

$$\Delta\mathbf{s} = \mathbf{\Phi_s}^* \cdot \mathbf{E_s} . \tag{3}$$

Thus, the value $\mathbf{s}$ is corrected by $\Delta\mathbf{s}$, and by repeating this procedure until the value converges, one can obtain vector $\mathbf{s}$ that minimizes the error. Here, Equation (3) is the formula for the Gauss-Newton method, but it is equally possible to use another error-minimizing method (e.g., Levenberg-Marquardt method).

When registration is performed within the framework of the above numerical computation, the following tasks will be required:

- Designing what the vector of unknown parameters $\mathbf{s}$ should be,
- Formulating the coordinate-transformation function $f_\mathbf{s}$ with respect to $\mathbf{s}$,
- Formulating the image Jacobian $\mathbf{J_{us}}$, which is the partial derivative of $f_\mathbf{s}$ by $\mathbf{s}$.

3.2. Obtaining the pose with 6DOF

Consider the typical situation in AR, where the intrinsic camera parameters have been calibrated but the camera pose is unknown. Here, it is sufficient to let $\mathbf{s}$ be the six-dimensional vector expressing the pose of camera C in world coordinate system W. A general solution method in this case is explained below.

■ Perspective projection and viewing transformation

If we denote the perspective projection from the camera to the image coordinate system by function P_C and the viewing transformation from the world to the camera coordinate system by function M_{CW}, Equation (1) becomes

$$f_\mathbf{s}\left(\mathbf{x}_{\mathbf{WQ}_i}\right) = P_C\left(M_{CW}\left(\mathbf{x}_{\mathbf{WQ}_i}\right)\right). \tag{4}$$

Hence, if we let $\mathbf{J_{uxc}}$ be the Jacobian matrix obtained by partially differentiating P_C by the coordinate-axis variable $\mathbf{x_C}$ in the camera coordinate system at feature point Q_i, and if we let $\mathbf{J_{xcs}}$ be the Jacobian matrix obtained by partially differentiating M_{CW} by the unknown parameter vector $\mathbf{s}$, then from Equation (4) we obtain the relation

$$\mathbf{J_{us}} = \mathbf{J_{uxc}} \cdot \mathbf{J_{xcs}} . \tag{5}$$

It is assumed that the intrinsic camera parameters are already known, as previously mentioned; therefore, P_C is fixed regardless of $\mathbf{s}$. Consequently, the only function that depends on the definition of $\mathbf{s}$ is M_{CW}. So the problem of formulating $f_\mathbf{s}$ and $\mathbf{J_{us}}$ reduces to the problem of formulating M_{CW} and $\mathbf{J_{xcs}}$ according to the definition of the unknown parameter vector $\mathbf{s}$.

Now, P_C can be expressed as

$$u_x = -f_C \frac{x_C}{z_C} \ , \quad u_y = -f_C \frac{y_C}{z_C} \ , \tag{6}$$

where f_C is the known focal length, and $[u_x \ u_y]^T = \mathbf{u}$, $[x_C \ y_C \ z_C]^T = \mathbf{x_C}$, so $\mathbf{J_{uxc}}$ is as follows:

$$\mathbf{J_{uxc}} = \begin{bmatrix} \frac{\partial u_x}{\partial x_C} & \frac{\partial u_x}{\partial y_C} & \frac{\partial u_x}{\partial z_C} \\ \frac{\partial u_y}{\partial x_C} & \frac{\partial u_y}{\partial y_C} & \frac{\partial u_y}{\partial z_C} \end{bmatrix} = \begin{bmatrix} -\frac{f_C}{z_C} & 0 & \frac{f_C x_C}{z_C^2} \\ 0 & -\frac{f_C}{z_C} & \frac{f_C y_C}{z_C^2} \end{bmatrix} . \tag{7}$$

■ Formulating M_{CW} and $\mathbf{J_{xcs}}$

The pose of an object B in a coordinate system A can be expressed by a three-dimensional position vector $\mathbf{t_{AB}}$ (= $[x_{AB} \ y_{AB} \ z_{AB}]^T$) and a 3-by-3 orientation matrix $\mathbf{R_{AB}}$. The orientation has three degrees of freedom, and several methods are known to express this through three variables (such as by Euler angles). Now, suppose that we use one of these methods and express the orientation with a three-dimensional vector $\boldsymbol{\omega_{AB}} = [\xi_{AB} \ \psi_{AB} \ \zeta_{AB}]^T$ (in other words, $\mathbf{R_{AB}} = \mathbf{R}(\boldsymbol{\omega_{AB}})$). Then, the pose of object B can be expressed by six-dimensional vector $[x_{AB} \ y_{AB} \ z_{AB} \ \xi_{AB} \ \psi_{AB} \ \zeta_{AB}]^T$.

Using these values, we can define the transformation from the coordinates $\mathbf{x_B}$ on the coordinate system of object B to the coordinate system A as follows:

$$\mathbf{x_A} = \mathbf{R}\left(\boldsymbol{\omega}_{\mathbf{AB}}\right) \cdot \mathbf{x_B} + \mathbf{t_{AB}} . \tag{8}$$

When certain values are assigned to the unknown parameters $\mathbf{s} = [\mathbf{t_{WC}}^T\ \boldsymbol{\omega}_{\mathbf{WC}}{}^T]^T = [x_{WC}\ y_{WC}\ z_{WC}\ \xi_{WC}\ \psi_{WC}\ \zeta_{WC}]^T$, then from Equation (8) the viewing transformation M_{CW} is given by

$$\mathbf{x}_{\mathbf{CQ}_i} = M_{CW}\left(\mathbf{x}_{\mathbf{WQ}_i}\right) = \mathbf{R}\left(\boldsymbol{\omega}_{\mathbf{WC}}\right)^{-1} \cdot \left(\mathbf{x}_{\mathbf{WQ}_i} - \mathbf{t}_{\mathbf{WC}}\right). \quad (9)$$

To find $\mathbf{J}_{\mathbf{xcs}}$, it is necessary merely to expand the right-hand side of Equation (9) above and differentiate each component of it with respect to each variable of $\mathbf{s}$. Appendix A shows the details on how to derive $\mathbf{J}_{\mathbf{xcs}}$.

Under the conventional framework, in which the unknown variables are the pose (position and orientation) with six degrees of freedom, a necessary condition for finding a solution for Equation (3) is that at least three non-collinear feature points have been detected; more than four such points give a stable solution.

3.3. Application examples in AR

Examples of registration by this type of numerical computation include the following research projects thus far. Sundareswaran, et al. [11] used the framework of error-minimization by numerical computation in the registration of AR. The registration results of the previous frame are used as the initial value in order to estimate the current camera pose by iterative calculations. State, et al. [12] computed the camera pose by analytical calculation using three markers and used them as the initial value for the iterative calculations using all markers. Auer, et al. [5] obtained the camera pose by removing outliers using the RANSAC algorithm, and the pose was computed iteratively with respect only to dependable markers.

All these methods are, in a sense, similar in that each of them has six unknown parameters (position and orientation) and that they are deduced by using the image coordinates of feature points obtained by the user's camera. They are different from the methods proposed in this paper, which are not contrary to the above-mentioned methods but rather extensions of these methods.

4. Algorithms of the proposed methods

4.1. Line Constraint Method

Suppose that the position of the bird's-eye view camera $\mathbf{t_{WB}}$ in the world coordinate system, its orientation $\mathbf{R_{WB}}$, and its focal distance f_B are all known. One head marker H is mounted on the HMD to be observed from the bird's-eye view camera. When the head marker is detected from a bird's-eye view image, we can define a line that conceptually contains the head marker in the three-dimensional space.

If the head marker H is detected at the point with image coordinates $\mathbf{u_B} = [u_{Bx}\ u_{By}]^T$, points on this line in the coordinate system of the bird's-eye view camera trace $l_B(\tau) = [u_{Bx}\tau\ \ u_{By}\tau\ \ f_B\tau]^T$ as a function of parameter τ. Points on this line in the world coordinate system can then be expressed by the following equation, also as a function of parameter τ:

$$l_W(\tau) = \mathbf{R_{WB}} \cdot \begin{bmatrix} u_{Bx}\tau \\ u_{By}\tau \\ f_B\tau \end{bmatrix} + \mathbf{t_{WB}} = \begin{bmatrix} h_x\tau + x_{WB} \\ h_y\tau + y_{WB} \\ h_z\tau + z_{WB} \end{bmatrix}. \quad (10)$$

Here, $[h_x\ h_y\ h_z]^T$ is a constant term determined by $\mathbf{R_{WB}} \cdot [u_{Bx}\ u_{By}\ f_B]^T$.

Now, the relationship between the position $\mathbf{x_{WH}}$ of head marker H and the position $\mathbf{t_{WC}}$ of the user's view camera in the world coordinate system is described by the following equation:

$$\mathbf{t_{WC}} = \mathbf{x_{WH}} - \mathbf{R}\left(\boldsymbol{\omega}_{\mathbf{WC}}\right) \cdot \mathbf{x_{CH}}. \quad (11)$$

Here, $\mathbf{x_{CH}} = [x_{CH}\ y_{CH}\ z_{CH}]^T$ is the position of head marker H in the user's view camera coordinate system C, i.e., the position where the head marker was set up on the HMD. This value is known through calibration. By substituting $l_W(\tau)$ in Equation (10) for $\mathbf{x_{WH}}$ in Equation (11), the camera position $\mathbf{t_{WC}}$ can be expressed as a four-parameter (τ, $\boldsymbol{\omega}_{\mathbf{WC}}$) function:

$$\mathbf{t_{WC}} = l_W(\tau) - \mathbf{R}\left(\boldsymbol{\omega}_{\mathbf{WC}}\right) \cdot \mathbf{x_{CH}}. \quad (12)$$

By this relation, the viewing transformation M_{CW} described in Section 3 can be written as a four-parameter (τ, $\boldsymbol{\omega}_{\mathbf{WC}}$) function:

$$\mathbf{x}_{\mathbf{CQ}_i} = M_{CW}\left(\mathbf{x}_{\mathbf{WQ}_i}\right) = \mathbf{R}\left(\boldsymbol{\omega}_{\mathbf{WC}}\right)^{-1} \cdot \left(\mathbf{x}_{\mathbf{WQ}_i} - l_W(\tau)\right) + \mathbf{x_{CH}}. \quad (13)$$

Hence, taking $\mathbf{s}=[\tau\ \xi_{WC}\ \psi_{WC}\ \zeta_{WC}]^T$ as the unknown parameters, registration can be performed based on a user's view image. Appendix B shows the computation of $\mathbf{J}_{\mathbf{xcs}}$ under this structure.

4.2. Global Error Minimization

In this method, the number of head markers H_j (j is an index) is flexible, but the position $\mathbf{x}_{\mathbf{CH}_j}$ (= [x_{CH_j} y_{CH_j} $z_{\mathbf{CH}_j}$]^T) of each marker in the user's view camera coordinate system C must be known beforehand by calibration. Now, suppose that n_1 feature points have been detected from a user's view image and that n_2 head markers have been detected from a bird's-eye view image. Let n be the total number of detected feature points, i.e. $n = n_1+n_2$.

■ Basic algorithm

In this method, a six-dimensional vector $\mathbf{s} = [x_{WC}\, y_{WC}\, z_{WC}\, \xi_{WC}\, \psi_{WC}\, \zeta_{WC}]^T$ defining the pose of the user's view camera in the world coordinate system is considered the unknown parameter, just as in the conventional method. The principle of this algorithm is as follows: first, the bird's-eye view camera is used to obtain the registration error $\Delta\mathbf{u}_{\mathbf{BH}_j}$ between the image coordinates $\mathbf{u}_{\mathbf{BH}_j}$ detected for head marker H_j and its theoretical value $\tilde{\mathbf{u}}_{\mathbf{BH}_j}$. Then, $\mathbf{s}$ is calculated to minimize the error-evaluator:

$$\sum_{i=1}^{n_1} \Delta\mathbf{u}_{\mathbf{CQ}_i} + \sum_{j=1}^{n_2} \Delta\mathbf{u}_{\mathbf{BH}_j}, \tag{14}$$

which represents the sum of the errors on head marker H_j plus the sum of errors on feature points Q_i obtained from the user's view.

In order to incorporate the bird's-eye view camera into the framework of numerical computation, first, the coordinate-transformation function $g_{\mathbf{s}}$ from the user's view camera to the bird's-eye view image coordinate system should be formulated with respect to $\mathbf{s}$. Then, the image Jacobian $\mathbf{J}_{\mathbf{us}}$ that expresses the infinitesimal change in the image coordinates corresponding to the infinitesimal change in $\mathbf{s}$ should be formulated.

$\mathbf{s}$ can be induced in the same way as when only the user's view camera is used. Specifically, for a given $\mathbf{s}$, find vector $\mathbf{E}_{\mathbf{s}}$ listing the error values as well as matrix $\mathbf{\Phi}_{\mathbf{s}}$ listing $\mathbf{J}_{\mathbf{us}}$, and use Equation (3) to calculate the correction values for $\mathbf{s}$. The difference between this method and the conventional one is that $\mathbf{E}_{\mathbf{s}}$ and $\mathbf{\Phi}_{\mathbf{s}}$ are formed using all feature points, including feature points Q_i detected from the user's view and head markers H_j detected from the bird's-eye view.

In this method, when no head marker is detected on the bird's-eye view image, the system functions under the conventional method described in Section 3. When no feature point is detected on the user's view image, the system functions only with the bird's-eye view camera.

■ Image Jacobian for bird's-eye view

To incorporate the bird's-eye view camera in the framework described above, we formulate both the coordinate-transformation function $g_{\mathbf{s}}$ and the image Jacobian $\mathbf{J}_{\mathbf{us}}$ corresponding to it.

Denote the modeling transformation from the user's view camera to the world coordinate system by function M_{WC}, the viewing transformation from the world to the bird's-eye view camera coordinate system by function M_{BW}, and the perspective projection from the bird's-eye view camera to the bird's-eye view image coordinate system by function P_B. We then have the equation

$$g_s(\mathbf{x}_{\mathbf{CH}_j}) = P_B(M_{BW}(M_{WC}(\mathbf{x}_{\mathbf{CH}_j}))). \tag{15}$$

P_B can be obtained using the known intrinsic camera parameters in a manner similar to the way in which Equation (6) was used. Further, M_{WC} and M_{BW} can be defined as follows, using Equation (8):

$$\mathbf{x}_{\mathbf{BH}_j} = M_{BW}(\mathbf{x}_{\mathbf{WH}_j}) = \mathbf{R}_{\mathbf{WB}}^{-1} \cdot (\mathbf{x}_{\mathbf{WH}_j} - \mathbf{t}_{\mathbf{WB}}) \tag{16}$$

$$\mathbf{x}_{\mathbf{WH}_j} = M_{WC}(\mathbf{x}_{\mathbf{CH}_j}) = \mathbf{R}(\boldsymbol{\omega}_{\mathbf{WC}}) \cdot \mathbf{x}_{\mathbf{CH}_j} + \mathbf{t}_{\mathbf{WC}}. \tag{17}$$

Image Jacobian $\mathbf{J}_{\mathbf{us}}$ can then be expressed as follows from Equation (15):

$$\mathbf{J}_{\mathbf{us}} = \mathbf{J}_{\mathbf{ux}b} \cdot \mathbf{J}_{\mathbf{x}b\mathbf{x}w} \cdot \mathbf{J}_{\mathbf{x}w\mathbf{s}}. \tag{18}$$

$\mathbf{J}_{\mathbf{ux}b}$ can also be obtained in a manner similar to using Equation (7). Appendix C shows the calculations of $\mathbf{J}_{\mathbf{x}b\mathbf{x}w}$ and $\mathbf{J}_{\mathbf{x}w\mathbf{s}}$.

5. Experiments

This section shows the experiments in which the proposed methods were compared to the conventional methods in order to prove the effectiveness and advantages of the proposed methods.

5.1. Settings for the experiment

The experiment was conducted using Canon's *MR*

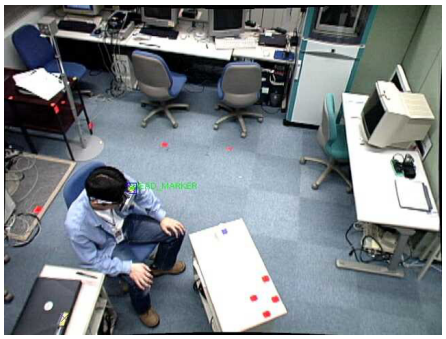


Figure 2. Bird's-eye view image

Figure 3. Markers on the scene

Figure 4. HMD with a head marker

Platform Basic Kit [13]; this includes a stereo video see-through HMD VH-2002 as well as *MR Platform SDK* (Software Development Kit). The proposed methods were implemented by extending the C++ classes in the SDK.

On the HMD, two NTSC cameras with horizontal view angle of about 51 degrees were installed internally, one at the right eye and the other at the left eye of the user. For the bird's-eye view camera, we installed an ELMO TN42H, equipped with a FUJINON TF4DA-8 lens with horizontal view angle of about 61.5 degrees, on the ceiling. The image of each camera is captured with a resolution of 640 by 240. The intrinsic parameters including radial lens distortion of each camera and the extrinsic parameters of the bird's-eye view camera are previously calibrated. Figure 2 shows an example of an image taken by the bird's-eye view camera.

The scene includes four red markers, which we set up as feature points for registration. To evaluate the error of this registration, one purple marker was also set up but was not used for registration. Figure 3 shows these markers set up in the scene. On the HMD, one yellow spherical marker was set up as the head marker, as shown in Figure 4. The world coordinates of the red and purple markers were measured by using a total station. The position of the head marker in the user's view camera coordinate system was measured manually.

For detecting markers, we used the marker-detecting function of the *MR platform SDK*. This function enables us to obtain the barycenter of the designated colored region as the image coordinates for a marker candidate. Further, the markers were identified based on the image coordinates of each marker estimated from the camera pose of the previous frame.

The parameters estimated in the previous frame were also used as the initial values in the iteration process at each frame. In the initial frame, some appropriate values had been chosen in advance to indicate an "initial pose". The CG image was rendered based on these values, and the operator first tried to overlay the CG image with the image actually taken by moving the HMD, and thereafter began the procedure.

All the processes were carried out using one PC (Xeon 2.0GHz Dual) equipped with a dual display graphics card and three video capture cards; a throughput of 30 fps was achieved during the experiment. Although our implementation could use the two user's view cameras and the bird's-eye camera simultaneously for registration, we show the result on only the left-eye camera and the bird's eye camera for registration to simplify the experiments.

5.2. Results

We executed various registration algorithms under the same conditions and compared the error. As a standard for evaluating the error, we measured the registration error between the estimated value on the image coordinates of the purple marker calculated by each method and the actual value detected for the purple marker.

The comparison was carried out in the following manner. First, the HMD was put on the head, and the head was moved in such a way that the number of red markers detected varied from four to two (four to zero only on the last experiment) while a log was taken and the previously-mentioned error was recorded at each frame.

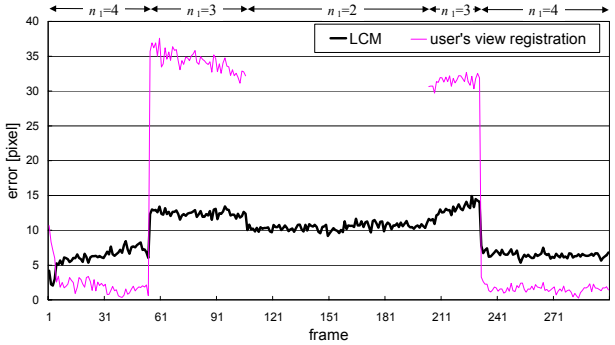
The first comparison is between the *LCM* and registration method described in Section 3 using only a user's view camera (hereafter referred to as user's view

registration). Figure 5(a) shows the errors of the respective methods. There was no significant error in the user's view registration while the user's viewpoint detected four markers; however, the *LCM* gave more stable results when only three markers were detected. Moreover, the user's view registration did not give a solution whereas the *LCM* provided a solution when only two markers were detected.

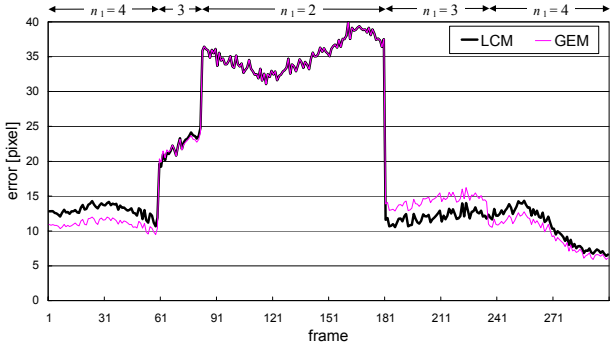
Next, we compared the two methods proposed here: the *LCM* and the *GEM*. Since only one head marker was used in this experiment, the input data in both methods were identical. Figure 5(b) compares the error in these methods. When two markers are detected from the user's view, these methods find a same solution. The *LCM* gave more accurate results when three markers were detected; on the other hand, the *GEM* got better results in case four markers were detected. This is because the *LCM* has a stronger inclination than the *GEM* to make the solution more stable and to force the user's view camera at correct position even if it increases the error on the detected markers at user's view image.

Finally, we compared the *GEM* with the registration using only the bird's-eye view camera (hereafter referred to as bird's-eye view registration). Only in this experiment we did use four head markers placed at the four vertices of a square with side length of approximately 10 cm. In *GEM*, registration was performed using all the feature points detected. In the bird's-eye view registration, the calculations were similar to those used in the *GEM*, except that no information from the user's view camera was used at all. We also carried out a comparison with a method that corrects the error from the bird's-eye view registration by using the information from the user's view, in which the bird's-eye view registration was considered a 6DOF sensor, and its error was corrected using a method similar to the one described in [12]. Figure 5(c) shows a comparison of these errors.

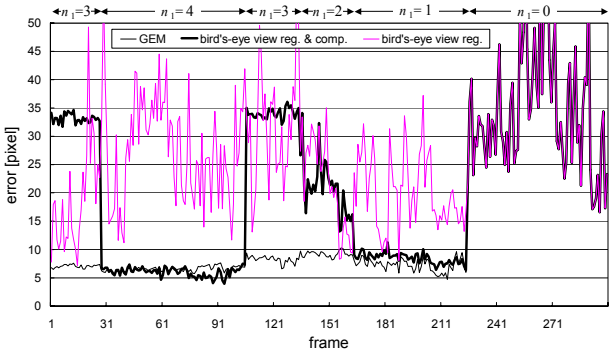
The accuracy of the bird's-eye view registration is clearly poor, which is due to the insufficient resolution of the bird's-eye view camera, calibration error, and the low level of accuracy in marker detection. When three markers were detected at the user's view, the method in which the bird's-eye view registration error was corrected by the user's view yielded unstable results, because the information from the bird's-eye view is not incorporated in



(a) LCM vs. user's view registration



(b) LCM vs. GEM



(c) GEM vs. bird's-eye view registration

Figure 5. Experimental Results

the result. Moreover, the methods based on the bird's-eye view registration cannot give a solution when some head markers are occluded.

In actual operation, it may be effective to dynamically switch the registration method according to the number of detected feature points.

6. Discussion and conclusion

In this paper, we proposed new vision-based registration methods using both a bird's-eye view camera and a user's view camera. By using information obtained from the

bird's-eye view camera, the number of feature points that needed to be observed by the user's view camera is reduced, thus improving the stability of registration.

Because of restrictions imposed by using a bird's-eye view camera, the proposed methods are not suitable for large-area applications such as outdoors. However, these methods could assist the development of vision-based registration in applications where a space about the size of a room needs to be viewed (and perhaps moving around in the area), for which an expensive 6DOF sensor was traditionally required. We used simple color markers in our experiments, but our proposed methods do not depend on the marker type or detection method; they work equally well with natural feature points or with identifiable markers [2][3].

We chose to use only one bird's-eye view camera in this study, but registration would be more stable if more than one bird's-eye view camera were available. It is easy to incorporate the multiple bird's-eye views in the *GEM* by listing information from all of the cameras to form $\mathbf{E_s}$ and $\mathbf{\Phi_s}$. In the *LCM*, the position $\mathbf{x_{WH}}$ of head marker H can be uniquely obtained from the multiple bird's-eye views, so registration can be performed with the unknown parameter **s** consisting of only the orientation components.

Further, these methods can be carried out simply by adding one inexpensive video capture card and a normal camera to a vision-based registration system using only a video see-though HMD, so it is very cost-efficient. In addition, in some AR applications, the bird's-eye view camera can be used not only for registration but also for generating an augmented bird's-eye view image to be shown to an audience.

The concept of using an additional bird's-eye view camera has a potential to improve not only the registration method based on the error-minimization framework as shown in this paper, but also the most of known vision-based camera tracking methods. Especially, the idea of reducing the degree-of-freedom to be solved, as presented in the *LCM*, can be applied to camera tracking generally.

The proposed methods assume precise calibration and reliable marker detection. Future works include analyzing stability against these errors and developing robust algorithm.

References

[1] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem", *IJCV*, vol.13, no.3, pp.331-356, 1994.

[2] G. A. Thomas, J. Jin, T. Niblett, and C. Urquhart, "A versatile camera position measurement system for virtual reality TV production", *Proc. International Broadcasting Convention '97*, pp.284-289, 1997.

[3] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top AR environment", *Proc. ISAR 2000*, pp.111-119, 2000.

[4] J. Park, B. Jiang, and U. Neumann, "Vision-based pose computation: robust and accurate augmented reality tracking", *Proc. IWAR '99*, pp.3-12, 1999.

[5] T. Auer and A. Pinz, "Building a hybrid tracking system - integration of optical and magnetic tracking", *ibid*, pp.13-22, 1999.

[6] R. Azuma and G. Bishop, "Improving static and dynamic registration in an optical see-through HMD", *Proc. SIGGRAPH '94*, pp.197-204, 1994.

[7] Y. Yokokohji, Y. Sugawara, and T. Yoshikawa, "Accurate image overlay on see-through head-mounted displays using vision and accelerometers", *Proc. IEEE VR 2000*, pp.247-254, 2000.

[8] K. Satoh, M. Anabuki, H. Yamamoto, and H. Tamura, "A hybrid registration method for outdoor augmented reality", *Proc. ISAR 2001*, pp.67-76, 2001.

[9] Vicon motion system, http://www.vicon.com/main/applications/virtual_reality.html

[10] D. G. Lowe, "Fitting parameterized three-dimensional models to images", *IEEE Trans. PAMI*, vol.13, no.5, pp.441-450, 1991.

[11] V. Sundareswaran and R. Behringer, "Visual servoing-based augmented reality", *Proc. IWAR '98*, pp.193-200, 1998.

[12] A. State, G. Hirota, D. T. Chen, B. Garrett, and M. Livingston, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking", *Proc. SIGGRAPH '96*, pp.429-438, 1996.

[13] S. Uchiyama, K. Takemoto, K. Satoh, H. Yamamoto, and H. Tamura, "MR Platform: a basic body on which mixed reality applications are built", *Proc. ISMAR '02*, pp.246-253, 2002.

[14] W. A. Hoff, "Fusion of data from head-mounted and fixed sensors", *Proc. IWAR '98*, pp.167-182, 1998.

Appendix A: Calculations of $\mathbf{J_{xcs}}$ to estimate 6DOF

To derive $\mathbf{J_{xcs}}$ as $\mathbf{J_{xcs}} = [\mathbf{J_{xct}}\ \mathbf{J_{xc\omega}}]$, consider the translation and rotation component of **s** separately and calculate the Jacobian matrix $\mathbf{J_{xct}}$ and $\mathbf{J_{xc\omega}}$ with respect to each. Let

$$\mathbf{R}(\boldsymbol{\omega}_{\mathbf{WC}}) = \begin{bmatrix} r^{11}(\boldsymbol{\omega}_{\mathbf{WC}}) & r^{12}(\boldsymbol{\omega}_{\mathbf{WC}}) & r^{13}(\boldsymbol{\omega}_{\mathbf{WC}}) \\ r^{21}(\boldsymbol{\omega}_{\mathbf{WC}}) & r^{22}(\boldsymbol{\omega}_{\mathbf{WC}}) & r^{23}(\boldsymbol{\omega}_{\mathbf{WC}}) \\ r^{31}(\boldsymbol{\omega}_{\mathbf{WC}}) & r^{32}(\boldsymbol{\omega}_{\mathbf{WC}}) & r^{33}(\boldsymbol{\omega}_{\mathbf{WC}}) \end{bmatrix} = \begin{bmatrix} r^{11}_{\mathbf{WC}} & r^{12}_{\mathbf{WC}} & r^{13}_{\mathbf{WC}} \\ r^{21}_{\mathbf{WC}} & r^{221}_{\mathbf{WC}} & r^{23}_{\mathbf{WC}} \\ r^{31}_{\mathbf{WC}} & r^{32}_{\mathbf{WC}} & r^{33}_{\mathbf{WC}} \end{bmatrix}$$

and expand Equation (9) to get

$$\mathbf{x}_{\mathbf{CQ}_i} = \begin{bmatrix} (x_{WQ_i} - x_{WC})r^{11}_{WC} + (y_{WQ_i} - y_{WC})r^{21}_{WC} + (z_{WQ_i} - z_{WC})r^{31}_{WC} \\ (x_{WQ_i} - x_{WC})r^{12}_{WC} + (y_{WQ_i} - y_{WC})r^{22}_{WC} + (z_{WQ_i} - z_{WC})r^{32}_{WC} \\ (x_{WQ_i} - x_{WC})r^{13}_{WC} + (y_{WQ_i} - y_{WC})r^{23}_{WC} + (z_{WQ_i} - z_{WC})r^{33}_{WC} \end{bmatrix}. \qquad \text{(A-1)}$$

By Equation (A-1), $\mathbf{J_{xct}}$ can be calculated by

$$\mathbf{J}_{\mathbf{xct}} = \begin{bmatrix} -r_{WC}^{11} & -r_{WC}^{21} & -r_{WC}^{31} \\ -r_{WC}^{12} & -r_{WC}^{22} & -r_{WC}^{32} \\ -r_{WC}^{13} & -r_{WC}^{23} & -r_{WC}^{33} \end{bmatrix}. \quad \text{(A-2)}$$

On the other hand, it is not so easy to solve $\mathbf{J_{xc\omega}}$ directly, so first find the Jacobian matrix $\mathbf{J_{xcr}}$ consisting of the partial derivative of $\mathbf{x_C}$ with respect to $\mathbf{r} = [r_{WC}^{11} \quad r_{WC}^{12} \quad \cdots \quad r_{WC}^{33}]^T$ and the Jacobian matrix $\mathbf{J_{r\omega}}$ consisting of the partial derivative of $\mathbf{r}$ with respect to $\boldsymbol{\omega}_{\mathbf{WC}}$. Then, use $\mathbf{J_{xc\omega}} = \mathbf{J_{xcr}} \cdot \mathbf{J_{r\omega}}$ to obtain the solution. The former is expressed as follows, by Equation (A-1):

$$\mathbf{J}_{\mathbf{xcr}} = \begin{bmatrix} x' & 0 & 0 & y' & 0 & 0 & z' & 0 & 0 \\ 0 & x' & 0 & 0 & y' & 0 & 0 & z' & 0 \\ 0 & 0 & x' & 0 & 0 & y' & 0 & 0 & z' \end{bmatrix} \quad \text{(A-3)}$$

where $x' = x_{WQ_i} - x_{WC}$, $y' = y_{WQ_i} - y_{WC}$, $z' = z_{WQ_i} - z_{WC}$. The latter depends on how the orientation is expressed. For instance, if the orientation is expressed in the rotation axis and the angle of rotation, calculations of $\mathbf{J_{r\omega}}$ are described in Appendix D.

Appendix B: Calculations of $\mathbf{J_{xcs}}$ in LCM

It is clear that $\mathbf{J_{xcs}} = [\mathbf{J_{xct}} \cdot \mathbf{J_{t\tau}} \quad \mathbf{J_{xcr}} \cdot \mathbf{J_{r\omega}}]$, so $\mathbf{J_{xcs}}$ can be calculated by finding each of $\mathbf{J_{xct}}$, $\mathbf{J_{t\tau}}$, $\mathbf{J_{xcr}}$, and $\mathbf{J_{r\omega}}$. From Equations (10) and (12), $\mathbf{J_{t\tau}}$ is deduced as

$$\mathbf{J}_{\mathbf{t\tau}} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix}. \quad \text{(A-4)}$$

From equation (13) we have:

$$\mathbf{J}_{\mathbf{xcr}} = \begin{bmatrix} x'' & 0 & 0 & y'' & 0 & 0 & z'' & 0 & 0 \\ 0 & x'' & 0 & 0 & y'' & 0 & 0 & z'' & 0 \\ 0 & 0 & x'' & 0 & 0 & y'' & 0 & 0 & z'' \end{bmatrix} \quad \text{(A-5)}$$

where $x'' = x_{WQ_i} - (h_x\tau + x_{WB})$, $y'' = y_{WQ_i} - (h_y\tau + y_{WB})$, $z'' = z_{WQ_i} + (h_z\tau + z_{WB})$. $\mathbf{J_{xct}}$ and $\mathbf{J_{r\omega}}$ can be obtained in the same way as the 6DOF pose estimation.

Appendix C: Calculations of $\mathbf{J_{xbxw}}$ and $\mathbf{J_{xws}}$

Expanding Equation (16) like Equation (A-1), we get

$$\mathbf{x}_{\mathbf{BH}_j} = \begin{bmatrix} (x_{WH_j} - x_{WB})r_{WB}^{11} + (y_{WH_j} - y_{WB})r_{WB}^{21} + (z_{WH_j} - z_{WB})r_{WB}^{31} \\ (x_{WH_j} - x_{WB})r_{WB}^{12} + (y_{WH_j} - y_{WB})r_{WB}^{22} + (z_{WH_j} - z_{WB})r_{WB}^{32} \\ (x_{WH_j} - x_{WB})r_{WB}^{13} + (y_{WH_j} - y_{WB})r_{WB}^{23} + (z_{WH_j} - z_{WB})r_{WB}^{33} \end{bmatrix} \quad \text{(A-6)}$$

leading to

$$\mathbf{J}_{\mathbf{xbxw}} = \begin{bmatrix} r_{WB}^{11} & r_{WB}^{21} & r_{WB}^{31} \\ r_{WB}^{12} & r_{WB}^{22} & r_{WB}^{32} \\ r_{WB}^{13} & r_{WB}^{23} & r_{WB}^{33} \end{bmatrix}. \quad \text{(A-7)}$$

On the other hand, since $\mathbf{J_{xws}} = [\mathbf{J_{xwt}} \quad \mathbf{J_{xwr}} \cdot \mathbf{J_{r\omega}}]$, it is necessary to find $\mathbf{J_{xwt}}$ and $\mathbf{J_{xwr}}$ to obtain $\mathbf{J_{xws}}$. From Equation (17), we get

$$\mathbf{x}_{\mathbf{WH}_j} = \begin{bmatrix} x_{CH_j} r_{WC}^{11} + y_{CH_j} r_{WC}^{12} + z_{CH_j} r_{WC}^{13} + x_{WC} \\ x_{CH_j} r_{WC}^{21} + y_{CH_j} r_{WC}^{22} + z_{CH_j} r_{WC}^{23} + y_{WC} \\ x_{CH_j} r_{WC}^{31} + y_{CH_j} r_{WC}^{32} + z_{CH_j} r_{WC}^{33} + z_{WC} \end{bmatrix}, \quad \text{(A-8)}$$

so we have

$$\mathbf{J}_{\mathbf{xwt}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(A-9)}$$

$$\mathbf{J}_{\mathbf{xwr}} = \begin{bmatrix} x_{CH_j} & y_{CH_j} & z_{CH_j} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{CH_j} & y_{CH_j} & z_{CH_j} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{CH_j} & y_{CH_j} & z_{CH_j} \end{bmatrix}. \quad \text{(A-10)}$$

Appendix D: Calculations of $\mathbf{J_{r\omega}}$

There are several ways to express the rotation matrix $\mathbf{R}$ using mutually independent variables with three degrees of freedom. Here, we assume the orientation expression using the rotation axis and the angle of rotation.

Let the axis of rotation be (ξ', ψ', ζ') (unit vectors) and the angle of rotation be θ, and suppose these four variables $\boldsymbol{\omega}' = [\xi' \; \psi' \; \zeta' \; \theta]^T$ express the orientation. Then, the following relation exists between the orientation $\boldsymbol{\omega}'$ and $\mathbf{R}$:

$$\mathbf{R} = \begin{bmatrix} \xi'^2(1-\cos\theta) + \cos\theta & \xi'\psi'(1-\cos\theta) - \zeta'\sin\theta & \zeta'\xi'(1-\cos\theta) + \psi'\sin\theta \\ \xi'\psi'(1-\cos\theta) + \zeta'\sin\theta & \psi'^2(1-\cos\theta) + \cos\theta & \psi'\zeta'(1-\cos\theta) - \xi'\sin\theta \\ \zeta'\xi'(1-\cos\theta) - \psi'\sin\theta & \psi'\zeta'(1-\cos\theta) + \xi'\sin\theta & \zeta'^2(1-\cos\theta) + \cos\theta \end{bmatrix} \quad \text{(A-11)}$$

Since $\boldsymbol{\omega}'$ has four variables, we represent the angle of rotation by the length of the rotation axis so that the number of variables can be reduced to three. The following holds between the orientation expressed with three variables $\boldsymbol{\omega} = [\xi \; \psi \; \zeta]^T$ and the previously expressed orientation $\boldsymbol{\omega}'$:

$$\boldsymbol{\omega}' = \begin{bmatrix} \dfrac{\xi}{\sqrt{\xi^2+\psi^2+\zeta^2}} & \dfrac{\psi}{\sqrt{\xi^2+\psi^2+\zeta^2}} & \dfrac{\zeta}{\sqrt{\xi^2+\psi^2+\zeta^2}} & \sqrt{\xi^2+\psi^2+\zeta^2} \end{bmatrix}^T. \quad \text{(A-12)}$$

These lead to the rotation matrix $\mathbf{R}$ defined by the 3-variable orientation $\boldsymbol{\omega}$. Since $\mathbf{J_{r\omega}} = \mathbf{J_{r\omega'}} \cdot \mathbf{J_{\omega'\omega}}$, it is necessary to find $\mathbf{J_{r\omega'}}$ and $\mathbf{J_{\omega'\omega}}$ to obtain $\mathbf{J_{r\omega}}$. By Equation (A-11) and (A-12), we get the followings:

$$\mathbf{J}_{\mathbf{r\omega'}} = \begin{bmatrix} 2\xi'(1-\cos\theta) & 0 & 0 & (\xi'^2-1)\sin\theta \\ \psi'(1-\cos\theta) & \xi'(1-\cos\theta) & -\sin\theta & \xi'\psi'\sin\theta - \zeta'\cos\theta \\ \zeta'(1-\cos\theta) & \sin\theta & \xi'(1-\cos\theta) & \zeta'\xi'\sin\theta + \psi'\cos\theta \\ \psi'(1-\cos\theta) & \xi'(1-\cos\theta) & \sin\theta & \xi'\psi'\sin\theta + \zeta'\cos\theta \\ 0 & 2\psi'(1-\cos\theta) & 0 & (\psi'^2-1)\sin\theta \\ -\sin\theta & \zeta'(1-\cos\theta) & \psi'(1-\cos\theta) & \psi'\zeta'\sin\theta - \xi'\cos\theta \\ \zeta'(1-\cos\theta) & -\sin\theta & \xi'(1-\cos\theta) & \zeta'\xi'\sin\theta - \psi'\cos\theta \\ \sin\theta & \zeta'(1-\cos\theta) & \psi'(1-\cos\theta) & \psi'\zeta'\sin\theta + \xi'\cos\theta \\ 0 & 0 & 2\zeta'(1-\cos\theta) & (\zeta'^2-1)\sin\theta \end{bmatrix}, \quad \text{(A-13)}$$

$$\mathbf{J}_{\boldsymbol{\omega'\omega}} = \frac{1}{\theta}\begin{bmatrix} 1-\dfrac{\xi^2}{\theta^2} & -\dfrac{\xi\psi}{\theta^2} & -\dfrac{\xi\zeta}{\theta^2} \\ -\dfrac{\psi\xi}{\theta^2} & 1-\dfrac{\psi^2}{\theta^2} & -\dfrac{\psi\zeta}{\theta^2} \\ -\dfrac{\zeta\xi}{\theta^2} & -\dfrac{\zeta\psi}{\theta^2} & 1-\dfrac{\zeta^2}{\theta^2} \\ \xi & \psi & \zeta \end{bmatrix}. \quad \text{(A-14)}$$