LETTER

Calibration Method by Image Registration with Synthetic Image of 3D Model*

Toru TAMAKI† ***and*** **Masanobu YAMAMOTO**†, ***Regular Members***

SUMMARY We propose a method for camera calibration based on image registration. This method registers two images; one is a real image captured by a camera with a calibration object with known shape and texture, and the other is a synthetic image containing the object. The proposed method estimates the parameters of the rotation and translation of the object by using the depth information of the synthetic image. The Gauss-Newton method is used to minimize the residuals of intensities of the two images. The proposed method does not depend on initial values of the minimization, and is applicable to images with much noise. Experimental results using real images demonstrate the robustness against initial state and noise on the image.

key words: *camera calibration, image registration, Gauss-Newton method, Tsai's method, z buffer*

1. Introduction

Camera calibration is an important process for computer vision and has been well studied. In this paper, we propose a camera calibration method that is based on image registration. The proposed method uses information of a known geometric object with texture and doesn't need point correspondences.

Classical calibration methods require a number of point correspondences. The correspondences are established by manual operations or the marker detection. However, the small number of the correspondences produced by such operations may affect the accuracy of estimates.

The image registration technique [1]–[3] has been also used to estimate parameters. It does not require the correspondence because two images are registered by minimizing the difference in intensities. The problem of conventional image registration is to assume that objects to be registered are planar. Although registrations for recovering arbitrary depth of a scene have been proposed [1], [3], it can not estimate parameters such as rotation and translation.

The registration-based calibration method proposed in this article allows us to register images containing a three-dimensional object that is not planar, and to estimate rotation and translation accurately. The proposed method registers two images; one is a real image of a known geometric object captured by the camera to be calibrated, and the other is a synthetic CG image containing the object. The CG image is rendered using the shape and the texture of the real object. Then the Gauss-Newton method is used to minimize the residuals of intensities of the two images.

Experimental results shown in Sect. 4 demonstrate that the inaccuracy of initial values for the minimization does not affect estimates of the proposed method and the parameters are estimated with high accuracy even for images with much noise.

Manuscript received June 5, 2002.

Manuscript revised September 12, 2002.

†The authors are with the Faculty of Engineering, Niigata University, Niigata-shi, 950–2181 Japan.

*This research is supported in part by Research Foundation for the Electrotechnology of Chubu and Grant for the Promotion on Niigata University Research Projects.

2. Registration between Two Images

In this section, we describe the model of the transformation between two images using depth information. As shown in Fig. 1, the CG image I_1 is produced with a known object by given rotation R and translation $\boldsymbol{T}$ (f is known). The object in the real image is supposed to be slightly moved**, and the parameters Q and $\boldsymbol{S}$ are unknown.

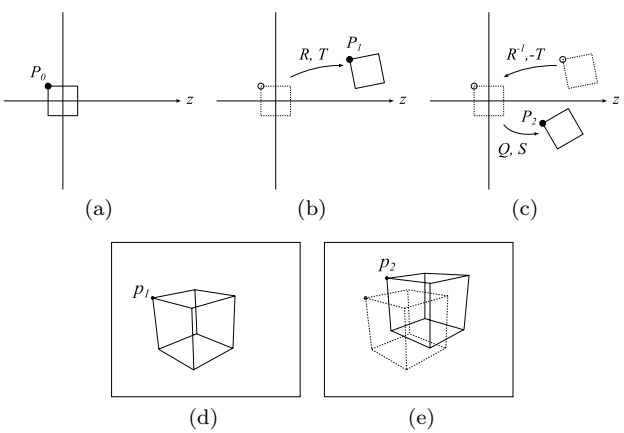


Fig. 1 Outline of the transformations. (a) The object and the camera coordinate system (the camera is looking toward positive direction of z). (b) Transfered object. (c) Relation of the object in CG and real images. (d) CG image I_1 of the object. (e) Real image I_2 with superimposed CG object. The object at the first image is drawn in dotted line.

**We consider that the camera coordinate system coincides with the world coordinate system.

2.1 Transformations of the Object

At first, we have a CG image, I_1, of a known geometric object with texture. Let $\boldsymbol{P}_0$ be a point on the object of which coordinate system is the same with the world coordinate system (Fig. 1 (a)). $\boldsymbol{P}_0$ is moved to $\boldsymbol{P}_1$ as the object is transfered by the rotation matrix R and the translation vector $\boldsymbol{T}$ as

$$\boldsymbol{P}_1 = R\boldsymbol{P}_0 + \boldsymbol{T}. \tag{1}$$

Then the point $\boldsymbol{P}_1 = (X_1, Y_1, Z_1)^T$ is projected to $\boldsymbol{p}_1 = (x_1, y_1)^T$ on the image I_1 (Fig. 1 (d)) by

$$\boldsymbol{P}_1 = \left(\frac{x_1 Z_1}{f}, \frac{y_1 Z_1}{f}, Z_1\right)^T, \tag{2}$$

where f is the focal length. Note that Z_1 for each $\boldsymbol{p}_1$ is obtained by depth buffer (see Sect. 3.2).

Then we consider on I_2, the real image captured by the camera. We assume that I_2 is similar to but slightly different from I_1 (Fig. 1 (e)). The point $\boldsymbol{P}_2 = (X_2, Y_2, Z_2)^T$ is also transfered from $\boldsymbol{P}_0$ by the rotation Q and the translation $\boldsymbol{S}$ as

$$\boldsymbol{P}_2 = Q\boldsymbol{P}_0 + \boldsymbol{S}. \tag{3}$$

Then $\boldsymbol{P}_2$ is projected to $\boldsymbol{p}_2 = (x_2, y_2)^T$ I_2:

$$\boldsymbol{p}_2 = \left(f\frac{X_2}{Z_2}, f\frac{Y_2}{Z_2}\right)^T. \tag{4}$$

Therefore, $\boldsymbol{p}_1$ in I_1 corresponds to $\boldsymbol{p}_2$ in I_2 through Eqs. (2), (4) and the following equation:

$$\boldsymbol{P}_2 = Q\boldsymbol{P}_0 + \boldsymbol{S} = QR^{-1}(\boldsymbol{P}_1 - \boldsymbol{T}) + \boldsymbol{S}. \tag{5}$$

3. Estimating Parameters

The minimization of the proposed method estimates the parameters Q and $\boldsymbol{S}$ so that the two images I_1 and I_2 are close to each other. Ideally, the difference in intensities of corresponding points in I_1 and I_2 have to be zero, but practically residual errors exist. Image registration seeks to minimize the residuals r_i:

$$r_i = I_1(\boldsymbol{p}_{1i}) - I_2(\boldsymbol{p}_{2i}), \tag{6}$$

where $I_1(\boldsymbol{p}_1)$ is the intensity at the point $\boldsymbol{p}_1$ in the image I_1, and $I_2(\boldsymbol{p}_2)$ is the intensity at the point $\boldsymbol{p}_2$ in the image I_2.

Since the correspondence between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ is unknown, we make $\boldsymbol{p}_1$ fixed and $\boldsymbol{p}_2$ be a function of $\boldsymbol{p}_1, Q$ and $\boldsymbol{S}$ through Eqs. (2), (4), and (5). Therefore, the residual is also a function of the parameters, and the objective function to be totally minimized for the estimation is the sum of squares of the residuals over the image I_1 as

$$\min_{\boldsymbol{\theta}} \sum_i {r_i}^2, \tag{7}$$

where $\boldsymbol{\theta}$ is the parameters to be estimated, and the summation i is taken over the points $\boldsymbol{p}_{1i}$ that are in the object region in I_1 and are also visible in I_2 (see Sect. 3.1).

The objective function is minimized by the Gauss-Newton method (see Appendix).

3.1 Visible Test

We perform the following visible test (Fig. 2) to remove a point which is not visible in I_2. Let $\boldsymbol{P}_1$ be a visible point in I_1 on the object, then the normal vector $\boldsymbol{N}_1$ of the surface at $\boldsymbol{P}_1$ is given [4] by $\boldsymbol{N}_1 = \frac{\partial \boldsymbol{P}_1}{\partial x} \times \frac{\partial \boldsymbol{P}_1}{\partial y}$. As $\boldsymbol{P}_1$ is transformed to $\boldsymbol{P}_2$ by Eq. (5), $\boldsymbol{N}_1$ is also rotated to the normal $\boldsymbol{N}_2$ at $\boldsymbol{P}_2$ by QR^{-1} in Eq. (5).

Since the camera center is identical to the origin, the angle between the normal $\boldsymbol{N}_2$ and the viewing direction (from the origin O to the point) is given by $\cos^{-1}\left(\frac{|\boldsymbol{N}_2 \cdot \boldsymbol{P}_2|}{|\boldsymbol{N}_2||\boldsymbol{P}_2|}\right)$. If the angle is larger than 90° (Fig. 2 (b)), then $\boldsymbol{P}_2$ is not visible in I_2 and $\boldsymbol{P}_1$ (or $\boldsymbol{p}_1$) in I_1 is excluded from the calculation of Eq. (7).

3.2 Depth from the z Buffer

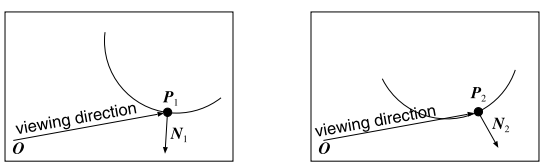
To obtain Z_1 in Eq. (2), the proposed method uses the depth information provided by a graphic library [5], [6]. However, a value in the z buffer is different from the actual depth because it is just used for relative depth.

The relation between a value stored in the z buffer, z_b, and the actual depth, Z, is given by the following equation [7];

$$Z = \frac{f_z n_z}{\frac{z_b}{s}(f_z - n_z) - f_z}, \tag{8}$$

where n_z and f_z are the distances from the camera center O to the front (near) and the rear (far) clipping planes between which the z buffer holds its depth, and s is the maximum value of the z buffer.

The precision of the z buffer is not uniform over the depth range but depends on the ratio of f_z to n_z. The larger the ratio is, the lesser the precision becomes at the rear of the z buffer. Usually the precision is better at the front of the z buffer.



(a) (b)

Fig. 2 Visible test. (a) $\boldsymbol{P}_1$ is visible in I_1. (b) $\boldsymbol{P}_2$ is not visible in I_2.

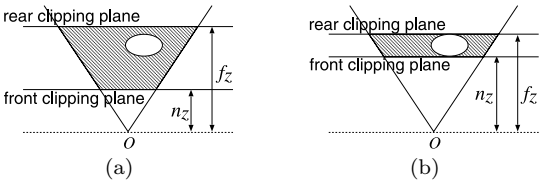


Fig. 3 Configuration of the z buffer. The ratio f_z to n_z is (a) large and (b) small.

Therefore, we create I_1 with the z buffer twice. At first time, we find the max/min distances between the camera and the object in the z buffer with a predetermined large ratio (Fig. 3 (a)). Then, f_z and n_z is placed just in front/behind of the object to make the ratio small as possible (Fig. 3 (b)).

4. Experimental Results

The proposed method was tested in two experiments. First, the estimation was analyzed using synthetic images whose parameters were known, and the results were compared with another calibration method. Second, the parameters of an actual camera were estimated using a real image taken by the camera.

4.1 Tests with Synthetic Images

The following test was performed with synthetic images. As shown in Fig. 4 (a), we made I_2 with a checkered cube as a calibration object ($30 \times 30 \times 30$); after the CG image of the cube was rendered by OpenGL [5] without shading and lighting, the contrast was changed and uniform noise was added. The true values of the parameters Q and $\boldsymbol{S}$ are shown in the top line of Table 1.

Then we produced another image I_1 (Fig. 4 (b)) of the cube. The parameters R and $\boldsymbol{T}$ were given manually using a GUI tool so that the appearance of the cube in I_2 is similar to that in I_1 (so, R and $\boldsymbol{T}$ were not identical to Q and $\boldsymbol{S}$). After that, uniform noise was added to R and $\boldsymbol{T}$. Note that the z buffer was also produced when I_1 was rendered.

Figure 4 (c) shows the difference between I_2 and I_1 (except the background) at the initial state and we can see the large difference (the larger is the brighter) because we set $Q = R$ and $\boldsymbol{S} = \boldsymbol{T}$ as the initial value. Figures 4 (d)–(f) show the difference between I_2 and the image in which every point is warped from I_1 by currently estimated Q and $\boldsymbol{S}$. After the optimization converges (Fig. 4 (f)), we can see that the cube is accurately fitted to that in I_2.

To analyze statistically the results of the proposed method, the test described above was performed 10 times. The noise added to R and $\boldsymbol{T}$ were uniform (± 0.1 [rad] for α, β and γ, ± 5 for s_x and s_y, ± 50 for s_z).

The first row of Table 1 shows the results of the proposed method. The estimates have some bias, however, the standard deviation is small. The second row of Table 1 shows the results by Tsai's method [8] using manually selected 10 feature points to which uniform noise (± 1.5) was added to the x and y coordinates[†]. From the viewpoint of the standard deviation of 10 tests, the proposed method is more robust to noise. Note that the computational time for each test was

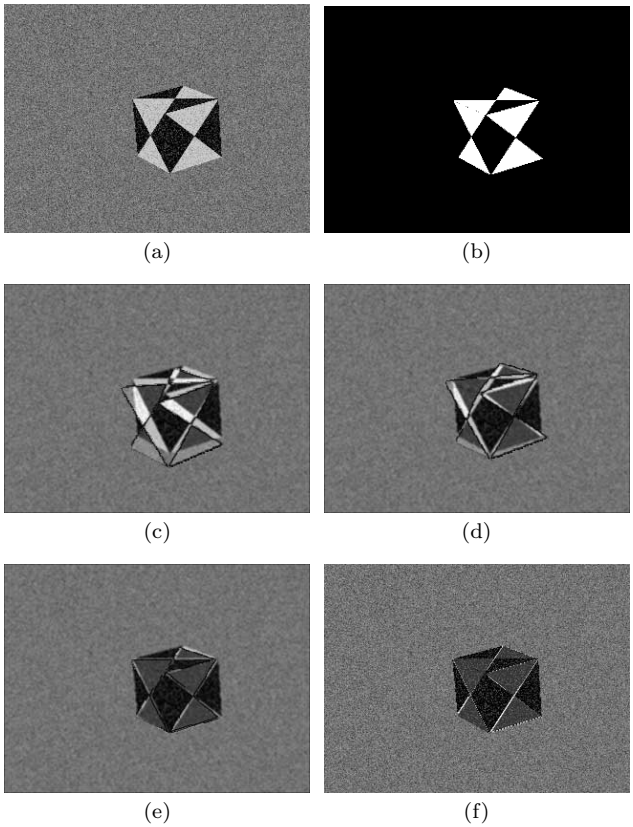


Fig. 4 Experimental result with the synthesis image. (a) Cube image I_2 (added noise and changed contrast). (b) CG cube model image I_1. Difference between transformed I_1 and I_2 at (c) 0 (d) 1 (e) 3 (f) 18 iterations.

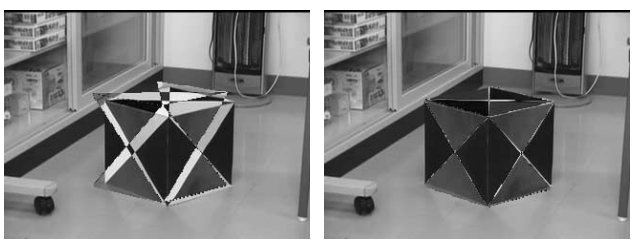


Fig. 5 Experimental result with the real image. The difference between the real image I_2 and the synthetic image (a) before optimization, and (b) after convergence.

[†]Note that the Tsai's method estimates f and s_z simultaneously after other parameters are estimated. Therefore, the estimates f affects the accuracy of s_z, but other parameters are intact.

Table 1 Results of 10 tests with synthetic images ($f = 911.490494$). Top : result of the proposed method. Bottom : result of Tsai's method.

true value	$\gamma = 35$	$\beta = 54$	$\alpha = 25$	$s_x = 6$	$s_y = 26$	$s_z = 411$
mean	35.139	54.871	25.118	6.0039	26.015	411.25
std.	4.847e-3	4.745e-3	4.950e-3	3.697e-4	4.922e-4	3.102e-2
mean	35.757	55.072	25.547	6.2903	26.123	431.79
std.	1.829	0.5342	1.559	0.1910	0.1976	113.6

Table 2 Results with different focal lengths. The actual focal length is $f' = 911.49$, and actual s_z is $s'_z = 410$.

f	γ	β	α	s_x	s_y	s_z	std.s_z	$s'_z f'/f$
717.09	35.526	54.994	25.455	5.9632	25.980	321.47	4.393e-3	322.56
802.94	35.397	54.914	25.348	5.9739	25.987	360.54	2.280e-3	361.17
911.49	35.139	54.871	25.118	6.0039	26.015	411.25	3.102e-2	410.00
1141.7	35.071	54.887	25.053	6.0293	25.980	515.40	3.037e-3	513.56
1371.6	34.777	54.770	24.836	6.0589	25.968	621.12	1.269e-3	616.97

Table 3 Experimental result with the real image.

γ	β	α	s_x	s_y	s_z
22.684	44.248	17.567	2.3529	27.594	485.08

less than two minutes on a PC (866 MHz CPU) implemented with GNU C++ and CLAPACK.

The proposed method assumes that the focal length f is known, but f is not often estimated accurately. Table 2 shows how the error of f affects the estimates. For this, several different f were used to produce I_1 (we used the same I_2 as Fig. 4 (a) with the actual focal length $f' = 911.49$). Initial values with noise were the same with the experiments above.

Although f was changed in the range of 0.8–1.5 times of f', the estimates of Q, s_x and s_y were still correct and robust (std. were smaller than 1e-2). s_z was affected by the change of f and close to $s'_z f'/f$ (in the most right column of Table 2), where s'_z is the actual s_z. The reason is that the focal length and the translation along the optical axis are related by $s'_z/f' = s_z/f$.

4.2 Calibration with Real Image

We conducted experiments with the proposed method using real images. A checkered cube whose size is $30 \times 30 \times 30$ cm was used as the calibration object. The procedure is the same with the experiments with the synthetic images, and $f = 1582$ is given by other calibration method.

The estimated parameters are shown in Table 3. After the optimization converged (Fig. 5 (b)) in two minutes, the cube model is accurately fitted to the actual object in the real image, while the difference before the optimization is large as shown in Fig. 5 (a).

Actually, the appearance of the object in real world depends not only on the reflectance but also on the lighting condition and the lens system of a real camera [9]. The proposed method does not take these effects into account, and assumes that the surface of the object has no specular and uniformly lighted. In spite of the simple assumptions, the proposed method worked well.

5. Conclusions

We have proposed a camera calibration method, based on image registration, for estimating the parameters of the transformation of a known geometric object with texture using depth information. The advantage of the proposed method is its robustness against initial state and noise on the image as shown in the experiments. Besides, the proposed method can applicable to any kind of a known geometric object with texture; not only a cube but also a texture-mapped 3D object scanned by a laser range finder or created by a CG modeler.

The proposed method does not estimate the focal length. Another method is required for estimation of the focal length, however, it is not necessary so accurate because the error in the focal length does not affect the estimates. The initial values do not need to be accurate, which facilitates using a GUI tool for the initial fitting.

References

[1] R. Szeliski, "Image mosaicing for tele-reality applications," Technical Report CRL 94/2, Digital Equipment Corporation, Cambridge Research Lab., 1994.
[2] R. Szeliski, "Video mosaics for virtual environment," IEEE Comput. Graph. Appl., vol.16, no.3, pp.22–30, 1996.
[3] H.S. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," IEEE Trans. Pattern Anal. Mach. Intell., vol.18, no.8, pp.814–830, 1996.
[4] E. Kreyszig, Advanced Engineering Mathematics, 8th ed., Wiley, 1999.
[5] OpenGL.org, "OpenGL developer FAQ and troubleshooting guide," 2000. http://www.opengl.org/developers/faqs/technical/depthbuffer.htm
[6] Microsoft, "DirectX C/C++ SDK documentation," http://msdn.microsoft.com/library/en-us/dx8_c/hh/dx8_c/graphics_using_8shf.asp
[7] A. Watt, 3D Computer Graphics, Addison-Wesley, 1993.
[8] R.Y. Tsai, "An efficient and accurate camera calibration

technique for 3D machine vision," Proc. CVPR'86, pp.364–374, 1986.
[9] E.J. Giorgianni and T.E. Madden, Digital Color Management: Encoding solutions, Addison-Wesley, 1998.
[10] G.A.F. Seber and C.J. Wild, Nonlinear Regression, Wiley, New York, 1989.
[11] D.G. Lowe, "Fitting parameterized three-dimensional models to images," IEEE Trans. Pattern Anal. Mach. Intell., vol.13, no.5, pp.441–450, 1991.
[12] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Numerical recipes in C, Cambridge University Press, 1992.

Appendix: Estimating Parameters

Here we explain how to apply the minimization Eq. (7) to the Gauss-Newton method.

Let $R, Q, \boldsymbol{T}$ and $\boldsymbol{S}$ appeared in Sect. 2 as:

$$R = Q_z(a)Q_y(b)Q_x(c) = \begin{pmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos c & -\sin c \\ 0 & \sin c & \cos c \end{pmatrix}, \quad (\mathrm{A}\cdot 1)$$

$$\boldsymbol{T} = (t_x, t_y, t_z)^T, \quad (\mathrm{A}\cdot 2)$$

$$Q = Q_z(\alpha)Q_y(\beta)Q_x(\gamma), \quad (\mathrm{A}\cdot 3)$$

$$\boldsymbol{S} = (s_x, s_y, s_z)^T. \quad (\mathrm{A}\cdot 4)$$

Estimating the parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_6)^T \equiv (\alpha, \beta, \gamma, s_x, s_y, s_z)^T$, Eq. (7) is minimized by the Gauss-Newton method [10]. The parameters are updated, with an initial value, by the following rule:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\, \delta\boldsymbol{\theta}. \quad (\mathrm{A}\cdot 5)$$

The decent direction [10] $\delta\boldsymbol{\theta} = (\delta\theta_1, \ldots, \delta\theta_6)^T$ is

$$\delta\boldsymbol{\theta} = -(J^T J)^{-1} J^T \boldsymbol{r}, \quad (\mathrm{A}\cdot 6)$$

$$J = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\theta}}, \quad (\mathrm{A}\cdot 7)$$

where $\boldsymbol{r} = (r_1, r_2, \ldots)^T$. This is the same as the least square formulation, that is, the system of linear equations [3] written as

$$\sum_i \sum_l \frac{\partial r_i}{\partial \theta_k}\frac{\partial r_i}{\partial \theta_l}\delta\theta_l = -\sum_i r_i \frac{\partial r_i}{\partial \theta_k}, \quad (\mathrm{A}\cdot 8)$$

for $k = 1, \ldots, 6$. The partial derivatives are derived by the chain rule of vector differentiation [10]:

$$\frac{\partial r}{\partial \theta_k} = -\frac{\partial \boldsymbol{p}_2}{\partial \theta_k}\frac{\partial I_2}{\partial \boldsymbol{p}_2} = -\left(\frac{\partial x_2}{\partial \theta_k}, \frac{\partial y_2}{\partial \theta_k}\right)^T \nabla I_2(\boldsymbol{p}_2). \quad (\mathrm{A}\cdot 9)$$

The partial derivatives of x and y [11] are as

$$\frac{\partial x_2}{\partial \theta_k} = \frac{f}{Z_2}\left(\frac{\partial X_2}{\partial \theta_k} - \frac{X_2}{Z_2}\frac{\partial Z_2}{\partial \theta_k}\right), \quad (\mathrm{A}\cdot 10)$$

$$\frac{\partial y_2}{\partial \theta_k} = \frac{f}{Z_2}\left(\frac{\partial Y_2}{\partial \theta_k} - \frac{Y_2}{Z_2}\frac{\partial Z_2}{\partial \theta_k}\right). \quad (\mathrm{A}\cdot 11)$$

The derivatives in the above equations are the elements of the following Jacobian:

$$\frac{\partial \boldsymbol{P}_2}{\partial \theta_k} = \left(\frac{\partial X_2}{\partial \theta_k}, \frac{\partial Y_2}{\partial \theta_k}, \frac{\partial Z_2}{\partial \theta_k}\right)^T, \quad (\mathrm{A}\cdot 12)$$

and the derivatives with respect to θ_k are as

$$\frac{\partial \boldsymbol{P}_2}{\partial \alpha} = \frac{\partial Q_z(\alpha)}{\partial \alpha}Q_y(\beta)Q_x(\gamma)R^{-1}(\boldsymbol{P}_1 - \boldsymbol{T}), \quad (\mathrm{A}\cdot 13)$$

$$\frac{\partial \boldsymbol{P}_2}{\partial \beta} = Q_z(\alpha)\frac{\partial Q_y(\beta)}{\partial \beta}Q_x(\gamma)R^{-1}(\boldsymbol{P}_1 - \boldsymbol{T}), \quad (\mathrm{A}\cdot 14)$$

$$\frac{\partial \boldsymbol{P}_2}{\partial \gamma} = Q_z(\alpha)Q_y(\beta)\frac{\partial Q_x(\gamma)}{\partial \gamma}R^{-1}(\boldsymbol{P}_1 - \boldsymbol{T}), \quad (\mathrm{A}\cdot 15)$$

$$\frac{\partial \boldsymbol{P}_2}{\partial \boldsymbol{S}} = \left(\frac{\partial \boldsymbol{P}_2}{\partial s_x}, \frac{\partial \boldsymbol{P}_2}{\partial s_y}, \frac{\partial \boldsymbol{P}_2}{\partial s_z}\right)^T = I, \quad (\mathrm{A}\cdot 16)$$

$$\frac{\partial Q_z(\alpha)}{\partial \alpha} = \begin{pmatrix} -\sin\alpha & -\cos\alpha & 0 \\ \cos\alpha & -\sin\alpha & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (\mathrm{A}\cdot 17)$$

$$\frac{\partial Q_y(\beta)}{\partial \beta} = \begin{pmatrix} -\sin\beta & 0 & \cos\beta \\ 0 & 0 & 0 \\ -\cos\beta & 0 & -\sin\beta \end{pmatrix}, \quad (\mathrm{A}\cdot 18)$$

$$\frac{\partial Q_x(\gamma)}{\partial \gamma} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin\gamma & -\cos\gamma \\ 0 & \cos\gamma & -\sin\gamma \end{pmatrix}. \quad (\mathrm{A}\cdot 19)$$

At the beginning of the iteration, we set $Q = R$, $\boldsymbol{S} = \boldsymbol{T}$ as the initial state because we assumes that the difference between the two images is small.

Once the direction is decided by solving the system of equations in Eq. (A·8), the step length α is optimized by line minimization [12]. Update by Eq. (A·5) is repeated until it converges. At each iteration, the parameters estimated in the previous iteration are used for the current Jacobian.