Memoirs of the Faculty of Engineering, Okayama University, Vol. 46, pp. 21–33, January 2012

Optimal Computation of 3-D Similarity: Gauss-Newton vs. Gauss-Helmert

Kenichi KANATANI* and Hirotaka NIITSUMA
Department of Computer Science, Okayama University
Okayama 700-8530 Japan

(Received November 28, 2011)

Because 3-D data are acquired using 3-D sensing such as stereo vision and laser range finders, they have inhomogeneous and anisotropic noise. This paper studies optimal computation of the similarity (rotation, translation, and scale change) of such 3-D data. We first point out that the Gauss-Newton and the Gauss-Helmert methods, regarded as different techniques, have similar structures. We then combine them to define what we call the *modified Gauss-Helmert method* and do stereo vision simulation to show that it is superior to either of the two in convergence performance. Finally, we show an application to real GPS geodetic data and point out that the widely used homogeneous and isotropic noise model is insufficient and that GPS geodetic data are prone to numerical problems.

1. Introduction

The task of autonomous robots to reconstruct the 3-D structure of the scene using stereo vision and simultaneously compute its location in the map of the environment, called SLAM (Simultaneous Localization and Mapping), is one of the central themes of robotics studies today. One of the fundamental techniques for this is to compute the 3-D motion of the robot between two time instances. This information is obtained by tracking a particular 3-D object to compute its rotation, translation, and scale change. A similar task occurs in reconstructing the entire shape of a 3-D object by 3-D sensing, for which multiple sensors are necessary, because one sensor can reconstruct only the part that is visible to it. Hence, we need to map a partial shape obtained from one sensor to a partial shape obtained from another by computing an appropriate similarity between them. The same task arises for geodetic measurement of the earth surface from multiple satellite sensor data [1, 5, 7, 18].

Thus, 3-D similarity estimation is an important problem in many engineering applications. To this end, many researchers have focused on accurate rotation estimation since 1980s. This is because translation can be estimated from the displacement of the centroid of the object, and the scale change is easily perceived from its changing size, while rotation estimation is not so straightforward in the presence of noise. However, almost all rotation estimation algorithms proposed in the past [3, 4, 9, 10, 13, 24] have assumed homogeneous and isotropic noise. This is unrealistic for 3-D data acquired by 3-D sensing such as stereo vision and laser/ultrasonic range finders, because the accuracy is usually different between the depth direction and the direction orthogonal to it, resulting in an inhomogeneous and anisotropic noise distribution depending on the position, orientation, and type of the sensor.

It is [21] who first pointed out the inevitable inhomogeneity and anisotropy of the noise in 3-D data and presented a 3-D rotation estimation scheme that takes it into account. They used a technique called *renormalization*, which iteratively removes statistical bias of reweight least squares [14]. Recently, [19] presented a numerical scheme for computing an exact maximum likelihood (ML) solution. [20] extended it to 3-D similarity estimation. They used for the optimization computation the Levenberg-Marquardt (LM) method, the most widely used standard optimization technique in the field of computer vision [8]. The LM is basically the Gauss-Newton method, to which the gradient descent principle is combined to ensure convergence [23].

In geodetic science, on the other hand, the Gauss-Helmert method [2, 17, 18] is popular for similarity estimation; Helmert himself was a geodesist, and the similarity transformation is sometimes referred to as the "Helmert transformation". The Gauss-Helmert method is also used in some computer vision ap-

*E-mail kanatani@suri.cs.okayama-u.ac.jp

plications [6, 22]. However, the Gauss-Newton and Gauss-Helmert methods have been regarded as different principles, and no comparative studies of them are found, partly because they have been mainly used in different domains: the former in robotics and computer vision, the latter in geodetic science. In this paper, we reformulate the Gauss-Helmert method in a form that makes the comparison easier and point out that the two have a very similar mathematical structures. Then, we combine them to define what we call the "modified Gauss-Helmert method" and do stereo vision simulation to show that it is superior to either of the two in the convergence performance. Finally, we show an application to real GPS geodetic data and point out that the widely used homogeneous and isotropic noise model is insufficient and that the numerical problem sometimes regarded as a shortcoming of Gauss-Helmert method is due to the inherent nature of GPS geodetic data.

2. Maximum likelihood estimation

Suppose we are given 3-D position measurements $\boldsymbol{r}_\alpha$ and $\boldsymbol{r}'_\alpha$, $\alpha = 1, ..., N$, before and after a similarity motion. We model the measurement uncertainty by independent Gaussian noise of mean $\boldsymbol{0}$ and covariance matrices $\epsilon^2 V_0[\boldsymbol{r}_\alpha]$ and $\epsilon^2 V_0[\boldsymbol{r}'_\alpha]$, where ϵ, which we call the *noise level*, describes the magnitude and $V_0[\boldsymbol{r}_\alpha]$ and $V_0[\boldsymbol{r}'_\alpha]$, which we call the *normalized covariance matrices*, describe the directional characteristics of the noise. If the noise is isotropic and homogeneous, we can let $V_0[\boldsymbol{r}_\alpha] = V_0[\boldsymbol{r}'_\alpha] = \boldsymbol{I}$ (the unit matrix) for all α, but in general $V_0[\boldsymbol{r}_\alpha]$ and $V_0[\boldsymbol{r}'_\alpha]$ are not diagonal and also different from position to position.

Let $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$ be the true positions of $\boldsymbol{r}_\alpha$ and $\boldsymbol{r}'_\alpha$, respectively, that undergo a similarity of rotation $\boldsymbol{R}$, translation $\boldsymbol{t}$, and scale change s. Their optimal estimation in the sense of ML is to minimize the *Mahalanobis distance*, which we hereafter call "residual" for simplicity (the multiplier $1/2$ is merely for convenience),

$$J = \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha, V_0[\boldsymbol{r}_\alpha]^{-1}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha)) + \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha, V_0[\boldsymbol{r}'_\alpha]^{-1}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha)), \quad (1)$$

with respect to $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$ subject to

$$\bar{\boldsymbol{r}}'_\alpha = \boldsymbol{S}\bar{\boldsymbol{r}}_\alpha + \boldsymbol{t}, \quad (2)$$

for some rotation $\boldsymbol{R}$, translation $\boldsymbol{t}$, and scale change s. Here, we combine the rotation $\boldsymbol{R}$ and scale change s into a "scaled rotation" $\boldsymbol{S} = s\boldsymbol{R}$ and express it in terms of the *quaternion*[1] $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^\top$ as follows:

[1] Mathematically, $\boldsymbol{q}$ is called a "quaternion" when associated with its algebra, i.e., the rule of composition [11]. However, the quaternion algebra does not play any role in this paper.

$$\boldsymbol{S} = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_2q_1+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_3q_1-q_0q_2) & 2(q_3q_2+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}. \quad (3)$$

This matrix represents a rotation if $\boldsymbol{q}$ is normalized to unit norm [11]. If $\boldsymbol{q}$ is not restricted to a unit vector, the square norm $\|\boldsymbol{q}\|^2$ represents the scale change s. This quaternion representation of similarity is well known in computer vision and robotics, and it is also used in geodesic recently [2].

Introducing to (1) Lagrange multipliers $\boldsymbol{\lambda}_\alpha$ for the constraint (2), we let

$$\tilde{J} = \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha, V_0[\boldsymbol{r}_\alpha]^{-1}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha)) + \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha, V_0[\boldsymbol{r}'_\alpha]^{-1}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha)) - \sum_{\alpha=1}^{N}(\boldsymbol{\lambda}_\alpha, \bar{\boldsymbol{r}}'_\alpha - \boldsymbol{S}\bar{\boldsymbol{r}}_\alpha - \boldsymbol{t}). \quad (4)$$

The ML estimators of $\bar{\boldsymbol{r}}_\alpha$, $\bar{\boldsymbol{r}}'_\alpha$, $\boldsymbol{q}$ and $\boldsymbol{t}$ are obtained by letting the derivatives of (4) with respect to them be $\boldsymbol{0}$ and solving the resulting equations.

3. Gauss-Newton method

We first formulate the Gauss-Newton method, the most fundamental optimization technique for robotics and computer vision. Differentiating (4) with respect to $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$, we obtain

$$\nabla_{\bar{\boldsymbol{r}}_\alpha}\tilde{J} = -V_0[\boldsymbol{r}_\alpha]^{-1}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha) + \boldsymbol{S}^\top\boldsymbol{\lambda}_\alpha,$$
$$\nabla_{\bar{\boldsymbol{r}}'_\alpha}\tilde{J} = -V_0[\boldsymbol{r}'_\alpha]^{-1}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha) - \boldsymbol{\lambda}_\alpha. \quad (5)$$

Letting these be $\boldsymbol{0}$ and solving for $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$, we have

$$\bar{\boldsymbol{r}}_\alpha = \boldsymbol{r}_\alpha - V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{\lambda}_\alpha, \quad \bar{\boldsymbol{r}}'_\alpha = \boldsymbol{r}'_\alpha + V_0[\boldsymbol{r}'_\alpha]\boldsymbol{\lambda}_\alpha. \quad (6)$$

Substituting these into (2), we can obtain the Lagrange multipliers $\boldsymbol{\lambda}_\alpha$ in the form

$$\boldsymbol{\lambda}_\alpha = -\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha, \quad (7)$$

where we define

$$\boldsymbol{e}_\alpha = \boldsymbol{r}'_\alpha - \boldsymbol{S}\boldsymbol{r}_\alpha - \boldsymbol{t}, \quad \boldsymbol{W}_\alpha = (\boldsymbol{S}V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top + V_0[\boldsymbol{r}'_\alpha])^{-1}. \quad (8)$$

Substituting (7) into (6) and substituting the resulting $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$ into (1), we can express the residual J in the following form:

$$
\begin{aligned}
J &= \frac{1}{2}\sum_{\alpha=1}^{N}(V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha, V_0[\boldsymbol{r}_\alpha]^{-1}V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha) \\
&\quad + \frac{1}{2}\sum_{\alpha=1}^{N}(V_0[\boldsymbol{r}'_\alpha]\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha, V_0[\boldsymbol{r}'_\alpha]^{-1}V_0[\boldsymbol{r}'_\alpha]\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha) \\
&= \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{e}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{S}V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top + V_0[\boldsymbol{r}'_\alpha])\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha) \\
&= \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{e}_\alpha, \boldsymbol{W}_\alpha\boldsymbol{W}_\alpha^{-1}\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha) \\
&= \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{e}_\alpha, \boldsymbol{W}_\alpha\boldsymbol{e}_\alpha). \qquad (9)
\end{aligned}
$$

Differentiating (3) with respect to q_i, $i = 0, 1, 2, 3$, we obtain

$$
\frac{\partial \boldsymbol{S}}{\partial q_i} = 2\boldsymbol{Q}_i, \qquad (10)
$$

where we define

$$
\begin{aligned}
\boldsymbol{Q}_0 &= \begin{pmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix}, \quad \boldsymbol{Q}_1 = \begin{pmatrix} q_1 & q_2 & q_3 \\ q_2 & -q_1 & -q_0 \\ q_3 & q_0 & -q_1 \end{pmatrix}, \\
\boldsymbol{Q}_2 &= \begin{pmatrix} -q_2 & q_1 & q_0 \\ q_1 & q_2 & q_3 \\ -q_0 & q_3 & -q_2 \end{pmatrix}, \quad \boldsymbol{Q}_3 = \begin{pmatrix} -q_3 & -q_0 & q_1 \\ q_0 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \end{pmatrix}. \qquad (11)
\end{aligned}
$$

Letting

$$
\boldsymbol{V}_\alpha = \boldsymbol{S}V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top + V_0[\boldsymbol{r}'_\alpha], \qquad (12)
$$

and differentiating $\boldsymbol{V}_\alpha\boldsymbol{W}_\alpha = \boldsymbol{I}$ with respect to q_i on both sides, we obtain

$$
2(\boldsymbol{Q}_i V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top + \boldsymbol{S}V_0[\boldsymbol{r}_\alpha]\boldsymbol{Q}_i^\top)\boldsymbol{W}_\alpha + \boldsymbol{V}_\alpha\frac{\partial \boldsymbol{W}_\alpha}{\partial q_i} = \boldsymbol{O}, \qquad (13)
$$

from which $\partial\boldsymbol{W}_\alpha/\partial q_i$ is expressed as

$$
\frac{\partial \boldsymbol{W}_\alpha}{\partial q_i} = -4\boldsymbol{W}_\alpha\mathcal{S}[\boldsymbol{Q}_i V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top]\boldsymbol{W}_\alpha, \qquad (14)
$$

where $\mathcal{S}[\,\cdot\,]$ denotes symmetrization ($\mathcal{S}[\boldsymbol{A}] = (\boldsymbol{A} + \boldsymbol{A}^\top)/2$). Thus, the derivative of (9) with respect to q_i is

$$
\begin{aligned}
\frac{\partial J}{\partial q_i} &= -2\sum_{\alpha=1}^{N}(\boldsymbol{Q}_i\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha\boldsymbol{e}_\alpha) \\
&\quad -2\sum_{\alpha=1}^{N}(\boldsymbol{e}_\alpha, \boldsymbol{W}_\alpha\boldsymbol{Q}_i V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha). \qquad (15)
\end{aligned}
$$

The symmetrization operator $\mathcal{S}[\,\cdot\,]$ is not necessary in the second term on the right because only the symmetric part counts in a quadratic form. If we define the 3×4 matrix $\boldsymbol{U}_\alpha$

$$
\boldsymbol{U}_\alpha = 2\begin{pmatrix} \boldsymbol{Q}_0\boldsymbol{r}_\alpha & \boldsymbol{Q}_1\boldsymbol{r}_\alpha & \boldsymbol{Q}_2\boldsymbol{r}_\alpha & \boldsymbol{Q}_3\boldsymbol{r}_\alpha \end{pmatrix}, \qquad (16)
$$

we can write (15) in the form

$$
\begin{aligned}
\nabla_{\boldsymbol{q}} J &= -\sum_{\alpha=1}^{N}\boldsymbol{U}_\alpha^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha \\
&\quad -2\left(\textstyle\sum_{\alpha=1}^{N}(\boldsymbol{e}_\alpha, \boldsymbol{W}_\alpha\boldsymbol{Q}_i V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha)\right), \qquad (17)
\end{aligned}
$$

where the second term on the right means the 4-D vector with that term as the ith component, $i = 0, 1, 2, 3$. Differentiation of (9) with respect to $\boldsymbol{t}$ yields

$$
\nabla_{\boldsymbol{t}} J = -\sum_{\alpha=1}^{N}\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha. \qquad (18)
$$

Differentiating (15) with respect to q_j and doing the Gauss-Newton approximation, i.e., omitting the terms containing $\boldsymbol{e}_\alpha$, we obtain the second derivative

$$
\frac{\partial^2 J}{\partial q_i \partial q_j} = 4\sum_{\alpha=1}^{N}(\boldsymbol{Q}_i\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha\boldsymbol{Q}_j\boldsymbol{r}_\alpha). \qquad (19)
$$

From (18), the second derivative with respect to $\boldsymbol{t}$ is

$$
\nabla^2_{\boldsymbol{t}} J = \sum_{\alpha=1}^{N}\boldsymbol{W}_\alpha. \qquad (20)
$$

Differentiation (18) with respect to q_i and doing the Gauss-Newton approximation, we obtain the following mixed second derivative:

$$
\nabla_{\boldsymbol{t}}\frac{\partial J}{\partial q_i} = 2\sum_{\alpha=1}^{N}\boldsymbol{W}_\alpha\boldsymbol{Q}_i\boldsymbol{r}_\alpha. \qquad (21)
$$

Using the matrix $\boldsymbol{U}_\alpha$ in (16), we can thus express the Hessian of the residual J in the following form (Gauss-Newton approximation):

$$
\boldsymbol{H} = \begin{pmatrix} \sum_{\alpha=1}^{N}\boldsymbol{U}_\alpha^\top\boldsymbol{W}_\alpha\boldsymbol{U}_\alpha & \sum_{\alpha=1}^{N}\boldsymbol{U}_\alpha^\top\boldsymbol{W}_\alpha \\ \sum_{\alpha=1}^{N}\boldsymbol{W}_\alpha\boldsymbol{U}_\alpha & \sum_{\alpha=1}^{N}\boldsymbol{W}_\alpha \end{pmatrix}. \qquad (22)
$$

Thus, we obtain the following procedure for the Gauss-Newton method.

1. Provide an initial guess of $\boldsymbol{q}$ and $\boldsymbol{t}$, and let $J_0 = \infty$ (a sufficiently large number).
2. Compute the scaled rotation $\boldsymbol{S}$ in (3) for $\boldsymbol{q}$.
3. Compute the vectors $\boldsymbol{e}_\alpha$ and the matrices $\boldsymbol{W}_\alpha$ in (8), and evaluate the residual J in (9).
4. If $J \approx J_0$, return $\boldsymbol{q}$ and $\boldsymbol{t}$ and stop. Else, let $J_0 \leftarrow J$.
5. Compute the matrices $\boldsymbol{Q}_i$ in (11) and the matrices $\boldsymbol{U}_\alpha$ in (16).

6. Solve the following 7-D linear equation:

$$\begin{pmatrix} \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^\top \boldsymbol{W}_\alpha \boldsymbol{U}_\alpha & \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^\top \boldsymbol{W}_\alpha \\ \sum_{\alpha=1}^N \boldsymbol{W}_\alpha \boldsymbol{U}_\alpha & \sum_{\alpha=1}^N \boldsymbol{W}_\alpha \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{q} \\ \Delta \boldsymbol{t} \end{pmatrix} = \begin{pmatrix} \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^\top \boldsymbol{W}_\alpha \boldsymbol{e}_\alpha \\ \sum_{\alpha=1}^N \boldsymbol{W}_\alpha \boldsymbol{e}_\alpha \end{pmatrix} + 2 \begin{pmatrix} \sum_{\alpha=1}^N (\boldsymbol{e}_\alpha, \boldsymbol{W}_\alpha \boldsymbol{Q}_i V_0[\boldsymbol{r}_\alpha] \boldsymbol{S}^\top \boldsymbol{W}_\alpha \boldsymbol{e}_\alpha) \\ \boldsymbol{0} \end{pmatrix}. \quad (23)$$

7. Update $\boldsymbol{q}$ and $\boldsymbol{t}$ as follows, and return to Step 2:

$$\boldsymbol{q} \leftarrow \boldsymbol{q} + \Delta \boldsymbol{q}, \qquad \boldsymbol{t} \leftarrow \boldsymbol{t} + \Delta \boldsymbol{t}. \quad (24)$$

As described in textbooks of optimization, (23) and (24) mean approximating, via Taylor expansion, the residual J in (9) by a quadratic function around the current values of $\boldsymbol{q}$ and $\boldsymbol{t}$ and moving to the minimum of the approximated residual J until the iterations converge.

4. Gauss-Helmert method

We now formulate the Gauss-Helmert method popular in geodetic science [2, 6, 17, 18, 22]. Suppose we are given some approximations $\boldsymbol{r}_\alpha^{(0)}$ and $\boldsymbol{r}'^{(0)}_\alpha$ of the true positions $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$. Let $\boldsymbol{q}$ and $\boldsymbol{t}$ be the current estimates the true solution $\bar{\boldsymbol{q}}$ and $\bar{\boldsymbol{t}}$. We write

$$\bar{\boldsymbol{r}}_\alpha = \boldsymbol{r}_\alpha^{(0)} + \Delta \bar{\boldsymbol{r}}_\alpha, \qquad \bar{\boldsymbol{r}}'_\alpha = \boldsymbol{r}'^{(0)}_\alpha + \Delta \bar{\boldsymbol{r}}'_\alpha,$$
$$\bar{\boldsymbol{q}} = \boldsymbol{q} + \Delta \boldsymbol{q}, \qquad \bar{\boldsymbol{t}} = \boldsymbol{t} + \Delta \boldsymbol{t}, \quad (25)$$

and compute the correction terms $\Delta \bar{\boldsymbol{r}}_\alpha$, $\Delta \bar{\boldsymbol{r}}'_\alpha$, $\Delta \boldsymbol{q}$, and $\Delta \boldsymbol{t}$. Substituting (25) into (2), doing Taylor expansion, and omitting second and higher order terms in the correction terms, we obtain

$$\boldsymbol{r}'^{(0)}_\alpha + \Delta \bar{\boldsymbol{r}}'_\alpha = \boldsymbol{S}(\boldsymbol{r}_\alpha^{(0)} + \Delta \bar{\boldsymbol{r}}_\alpha) + \sum_{i=0}^3 \Delta q_i \frac{\partial \boldsymbol{S}}{\partial q_i} \boldsymbol{r}_\alpha^{(0)} + \boldsymbol{t} + \Delta \boldsymbol{t}, \quad (26)$$

If (25) and (26) are substituted into (4), we obtain

$$\begin{aligned} \tilde{J} = & \frac{1}{2} \sum_{\alpha=1}^N (\boldsymbol{r}_\alpha - \boldsymbol{r}_\alpha^{(0)} - \Delta \bar{\boldsymbol{r}}_\alpha, V_0[\boldsymbol{r}_\alpha]^{-1} (\boldsymbol{r}_\alpha - \boldsymbol{r}_\alpha^{(0)} - \Delta \bar{\boldsymbol{r}}_\alpha)) \\ & + \frac{1}{2} \sum_{\alpha=1}^N (\boldsymbol{r}'_\alpha - \boldsymbol{r}'^{(0)}_\alpha - \Delta \bar{\boldsymbol{r}}'_\alpha, V_0[\boldsymbol{r}'_\alpha]^{-1} (\boldsymbol{r}'_\alpha - \boldsymbol{r}'^{(0)}_\alpha - \Delta \bar{\boldsymbol{r}}'_\alpha)) \\ & - \sum_{\alpha=1}^N (\boldsymbol{\lambda}_\alpha, \Big(\boldsymbol{r}'^{(0)}_\alpha + \Delta \bar{\boldsymbol{r}}'_\alpha - \boldsymbol{S}(\boldsymbol{r}_\alpha^{(0)} + \Delta \bar{\boldsymbol{r}}_\alpha) \\ & - \sum_{i=0}^3 \Delta q_i \frac{\partial \boldsymbol{S}}{\partial q_i} \boldsymbol{r}_\alpha^{(0)} - \boldsymbol{t} - \Delta \boldsymbol{t} \Big)). \end{aligned} \quad (27)$$

Differentiating this with respect to $\Delta \bar{\boldsymbol{r}}_\alpha$, $\Delta \bar{\boldsymbol{r}}'_\alpha$, Δq_i, and $\Delta \boldsymbol{t}$ and letting the results be $\boldsymbol{0}$, we have

$$-V_0[\boldsymbol{r}_\alpha]^{-1} (\boldsymbol{r}_\alpha - \boldsymbol{r}_\alpha^{(0)} - \Delta \bar{\boldsymbol{r}}_\alpha) + \boldsymbol{S}^\top \boldsymbol{\lambda}_\alpha = \boldsymbol{0},$$
$$-V_0[\boldsymbol{r}'_\alpha]^{-1} (\boldsymbol{r}'_\alpha - \boldsymbol{r}'^{(0)}_\alpha - \Delta \bar{\boldsymbol{r}}'_\alpha) - \boldsymbol{\lambda}_\alpha = \boldsymbol{0},$$
$$\sum_{\alpha=1}^N (\boldsymbol{\lambda}_\alpha, \frac{\partial \boldsymbol{S}}{\partial q_i} \boldsymbol{r}_\alpha^{(0)}) = 0, \qquad \sum_{\alpha=1}^N \boldsymbol{\lambda}_\alpha = \boldsymbol{0}. \quad (28)$$

From the first and second equations, we obtain

$$\boldsymbol{r}_\alpha^{(0)} + \Delta \bar{\boldsymbol{r}}_\alpha = \boldsymbol{r}_\alpha - V_0[\boldsymbol{r}_\alpha] \boldsymbol{S}^\top \boldsymbol{\lambda}_\alpha,$$
$$\boldsymbol{r}'^{(0)}_\alpha + \Delta \bar{\boldsymbol{r}}'_\alpha = \boldsymbol{r}'_\alpha + V_0[\boldsymbol{r}'_\alpha] \boldsymbol{\lambda}_\alpha. \quad (29)$$

Substitution of these into (26) results in

$$2 \sum_{i=0}^3 \Delta q_i \boldsymbol{Q}_i \boldsymbol{r}_\alpha^{(0)} + \Delta \boldsymbol{t} - (\boldsymbol{S} V_0[\boldsymbol{r}_\alpha] \boldsymbol{S}^\top + V_0[\boldsymbol{r}'_\alpha]) \boldsymbol{\lambda}_\alpha = \boldsymbol{e}_\alpha, \quad (30)$$

where $\boldsymbol{e}_\alpha$ is the first vector in (8). We have also used the matrices $\boldsymbol{Q}_i$ in (11) and the relation in (10). If we define the 3×4 matrices

$$\boldsymbol{U}_\alpha^{(0)} = 2 \begin{pmatrix} \boldsymbol{Q}_0 \boldsymbol{r}_\alpha^{(0)} & \boldsymbol{Q}_1 \boldsymbol{r}_\alpha^{(0)} & \boldsymbol{Q}_2 \boldsymbol{r}_\alpha^{(0)} & \boldsymbol{Q}_3 \boldsymbol{r}_\alpha^{(0)} \end{pmatrix}, \quad (31)$$

the third equation in (28) has the form

$$\sum_{\alpha=1}^N \boldsymbol{U}_\alpha^{(0)\top} \boldsymbol{\lambda}_\alpha = \boldsymbol{0}. \quad (32)$$

Using the matrices $\boldsymbol{U}_\alpha^{(0)}$ in (31) and the matrices $\boldsymbol{V}_\alpha$ in (12) we can write (30) as

$$\boldsymbol{U}_\alpha^{(0)} \Delta \boldsymbol{q} + \Delta \boldsymbol{t} - \boldsymbol{V}_\alpha \boldsymbol{\lambda}_\alpha = \boldsymbol{e}_\alpha. \quad (33)$$

We see that (30), (32), and (33) define linear equations in $\boldsymbol{\lambda}_1$, ..., $\boldsymbol{\lambda}_N$, $\Delta \boldsymbol{q}$, and $\Delta \boldsymbol{t}$. Solving these, we can determine $\bar{\boldsymbol{q}}$, $\bar{\boldsymbol{t}}$, $\bar{\boldsymbol{r}}_\alpha$, and $\bar{\boldsymbol{r}}'_\alpha$, which *exactly* minimize the residual J in (1) subject to the *linearized constraint* (26). However, (26) is an approximation, so we regard the computed solution $\boldsymbol{q}$ and $\boldsymbol{t}$ as new current values and upgrade $\boldsymbol{r}_\alpha^{(0)}$ by the left side of the first equation in (29) using the computed λ_α (the value of $\boldsymbol{r}_\alpha{}^{(0)}$ is not used in the computation). We repeat this until all variables converge. The procedure is summarized as follows:

1. Provide an initial guess of $\boldsymbol{q}$ and $\boldsymbol{t}$, and let $\boldsymbol{r}_\alpha^{(0)} = \boldsymbol{r}_\alpha$ and $J_0 = \infty$ (a sufficiently large number.
2. Compute the scaled rotation $\boldsymbol{S}$ in (3) for $\boldsymbol{q}$.
3. Compute the vectors $\boldsymbol{e}_\alpha$ and the matrices $\boldsymbol{W}_\alpha$ in (8), and evaluate the residual J in (9).
4. If $J \approx J_0$, return $\boldsymbol{q}$ and $\boldsymbol{t}$ and stop. Else, let $J_0 \leftarrow J$.
5. Compute the matrices $\boldsymbol{Q}_i$ in (11) and $\boldsymbol{U}_\alpha^{(0)}$ in (31).
6. Solve the following $(3N+7)$-D linear equation:

$$\begin{pmatrix} -\boldsymbol{V}_1 & & & \boldsymbol{U}_1^{(0)} & \boldsymbol{I} \\ & \ddots & & \vdots & \vdots \\ & & -\boldsymbol{V}_N & \boldsymbol{U}_N^{(0)} & \boldsymbol{I} \\ \boldsymbol{U}_1^{(0)\top} & \cdots & \boldsymbol{U}_N^{(0)\top} & & \\ \boldsymbol{I} & \cdots & \boldsymbol{I} & & \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_N \\ \Delta \boldsymbol{q} \\ \Delta \boldsymbol{t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_N \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}. \quad (34)$$

7. Update $\boldsymbol{r}_\alpha^{(0)}$, $\boldsymbol{q}$ and $\boldsymbol{t}$ as follows, and return to Step 2:

$$\boldsymbol{r}_\alpha^{(0)} \leftarrow \boldsymbol{r}_\alpha - V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{\lambda}_\alpha,$$
$$\boldsymbol{q} \leftarrow \boldsymbol{q} + \Delta\boldsymbol{q}, \qquad \boldsymbol{t} \leftarrow \boldsymbol{t} + \Delta\boldsymbol{t}. \tag{35}$$

5. Reduced Gauss-Helmert method

The above description gives an impression that the Gauss-Newton and Gauss-Helmert methods are very different disciplines. We now show that the Gauss-Helmert method can be expressed in a form very similar to the Gauss-Newton method. From (33), we can express λ_α in the form

$$\boldsymbol{\lambda}_\alpha = \boldsymbol{W}_\alpha\Big(\boldsymbol{U}_\alpha^{(0)}\Delta\boldsymbol{q} + \Delta\boldsymbol{t} - \boldsymbol{e}_\alpha\Big), \tag{36}$$

where $\boldsymbol{W}_\alpha$ is the matrix defined in (8). If (36) is substituted into the first and the second equalities in (28) to eliminate $\boldsymbol{\lambda}_\alpha$, we obtain linear equations only in $\Delta\boldsymbol{q}$ and $\Delta\boldsymbol{t}$. Hence, the procedure of the Gauss-Helmert method can also be written in the following form:

1. Provide an initial guess of $\boldsymbol{q}$ and $\boldsymbol{t}$, and let $\boldsymbol{r}_\alpha^{(0)} = \boldsymbol{r}_\alpha$ and $J_0 = \infty$ (a sufficiently large number).
2. Compute the scaled rotation $\boldsymbol{S}$ in (3) for $\boldsymbol{q}$.
3. Compute the vectors $\boldsymbol{e}_\alpha$ and the matrices $\boldsymbol{W}_\alpha$ in (8), and evaluate the residual J in (9).
4. If $J \approx J_0$, return $\boldsymbol{q}$ and $\boldsymbol{t}$ and stop. Else, let $J_0 \leftarrow J$.
5. Compute the matrices $\boldsymbol{Q}_i$ in (11) and the matrices $\boldsymbol{U}_\alpha^{(0)}$ in (31).
6. Solve the following 7-D linear equation:

$$\begin{pmatrix} \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^{(0)\top}\boldsymbol{W}_\alpha\boldsymbol{U}_\alpha^{(0)} & \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^{(0)\top}\boldsymbol{W}_\alpha \\ \sum_{\alpha=1}^N \boldsymbol{W}_\alpha\boldsymbol{U}_\alpha^{(0)} & \sum_{\alpha=1}^N \boldsymbol{W}_\alpha \end{pmatrix}\begin{pmatrix} \Delta\boldsymbol{q} \\ \Delta\boldsymbol{t} \end{pmatrix} = \begin{pmatrix} \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^{(0)\top}\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha \\ \sum_{\alpha=1}^N \boldsymbol{W}_\alpha\boldsymbol{e}_\alpha \end{pmatrix}. \tag{37}$$

7. Compute $\boldsymbol{\lambda}_\alpha$ by (36).
8. Update $\boldsymbol{r}_\alpha^{(0)}$, $\boldsymbol{q}$, and $\boldsymbol{t}$ as follows, and return to Step 2:

$$\boldsymbol{r}_\alpha^{(0)} \leftarrow \boldsymbol{r}_\alpha - V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{\lambda}_\alpha,$$
$$\boldsymbol{q} \leftarrow \boldsymbol{q} + \Delta\boldsymbol{q}, \qquad \boldsymbol{t} \leftarrow \boldsymbol{t} + \Delta\boldsymbol{t}. \tag{38}$$

This reduction of the algorithm reduces the memory usage in the computer; the effect is significant when the number N of the data is very large. Also, the similarity to the Gauss-Newton method becomes more apparent. Comparing (23) and (37), we see that the matrices $\boldsymbol{U}_\alpha$ of (16) are replaced by the matrices $\boldsymbol{U}_\alpha^{(0)}$ of (31). For the Gauss-Newton method, the right side of (23) is the gradient of the residual J with respect to $\boldsymbol{q}$ and $\boldsymbol{t}$, but the counterpart of the second term is missing on the right side of (37). However, when $\Delta\boldsymbol{q} = \Delta\boldsymbol{t} = \boldsymbol{0}$ at the time of the convergence of the Gauss-Helmert iterations, we see from (36) that $\boldsymbol{\lambda}_\alpha = -\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha$. Hence, the first equation in (38) implies

$$\boldsymbol{r}_\alpha^{(0)} = \boldsymbol{r}_\alpha + V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha, \tag{39}$$

which coincides with (6) if (7) is substituted. Substitution of (39) into (31) shows

$$\begin{aligned}\boldsymbol{U}_\alpha^{(0)} &= \boldsymbol{U}_\alpha \\ &+2\,\big(\,\boldsymbol{Q}_0V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha\ \boldsymbol{Q}_1V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha \\ &\quad \boldsymbol{Q}_2V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha\ \boldsymbol{Q}_3V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha\,\big).\end{aligned} \tag{40}$$

Hence, we have

$$\begin{aligned}\boldsymbol{U}_\alpha^{(0)\top}\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha &= \boldsymbol{U}_\alpha\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha \\ &+2\Big((\boldsymbol{e}_\alpha, \boldsymbol{W}_\alpha\boldsymbol{Q}_iV_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha)\Big).\end{aligned} \tag{41}$$

If this is substituted, the right side of (37) coincides with that of (23). Since this is $\boldsymbol{0}$ when the iterations have converged, we see that the Gauss-Helmert method returns the same solution as the Gauss-Newton method. Since the matrix on the left side of (23) is the Gauss-Newton approximation of the Hessian of the residual J, the matrix on the left side of (37) can also be viewed as an approximation of the Hessian of J. If we call it the *Gauss-Helmert approximation*, the Gauss-Newton and the Gauss-Helmert approximations differ by $O(\boldsymbol{e}_\alpha)$.

6. Modified Gauss-Helmert method

It is easily seen that (6) of the Gauss-Newton method gives the maximum likelihood estimators of $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$, given the current estimates $\boldsymbol{q}$ and $\boldsymbol{t}$, while the first equation in (35) (or (38)) gives an iterative update using the Lagrange multipliers λ_α of (36) expressed in the increments $\Delta\boldsymbol{q}$ and $\Delta\boldsymbol{t}$ of $\boldsymbol{q}$ and $\boldsymbol{t}$. Although the resulting $\boldsymbol{r}_\alpha^{(0)}$ converge to the first equation of (6), as we showed in the preceding section, we can expect higher accuracy if we directly use the first equation of (6) rather than the iterative update as in (35) or (38). The modified procedure is summarized as follows:

1. Provide an initial guess of $\boldsymbol{q}$ and $\boldsymbol{t}$, and let $J_0 = \infty$ (a sufficiently large number).
2. Compute the scaled rotation $\boldsymbol{S}$ in (3) for $\boldsymbol{q}$.
3. Compute the vectors $\boldsymbol{e}_\alpha$ and the matrices $\boldsymbol{W}_\alpha$ in (8), and compute $\boldsymbol{r}_\alpha^{(0)}$ as follows:

$$\boldsymbol{r}_\alpha^{(0)} = \boldsymbol{r}_\alpha + V_0[\boldsymbol{r}_\alpha]\boldsymbol{S}^\top\boldsymbol{W}_\alpha\boldsymbol{e}_\alpha. \tag{42}$$

4. Evaluate the residual J in (9). If $J \approx J_0$, return $\boldsymbol{q}$ and $\boldsymbol{t}$ and stop. Else, let $J_0 \leftarrow J$.

5. Compute the matrices $\boldsymbol{Q}_i$ in (11) and the matrices $\boldsymbol{U}_\alpha^{(0)}$ in (31).

6. Solve the following 7-D linear equation:

$$\begin{pmatrix} \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^{(0)\top} \boldsymbol{W}_\alpha \boldsymbol{U}_\alpha^{(0)} & \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^{(0)\top} \boldsymbol{W}_\alpha \\ \sum_{\alpha=1}^N \boldsymbol{W}_\alpha \boldsymbol{U}_\alpha^{(0)} & \sum_{\alpha=1}^N \boldsymbol{W}_\alpha \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{q} \\ \Delta \boldsymbol{t} \end{pmatrix} = \begin{pmatrix} \sum_{\alpha=1}^N \boldsymbol{U}_\alpha^{(0)\top} \boldsymbol{W}_\alpha \boldsymbol{e}_\alpha \\ \sum_{\alpha=1}^N \boldsymbol{W}_\alpha \boldsymbol{e}_\alpha \end{pmatrix}. \tag{43}$$

7. Update $\boldsymbol{r}_\alpha^{(0)}$, $\boldsymbol{q}$, and $\boldsymbol{t}$ as follows, and return to Step 2:

$$\boldsymbol{q} \leftarrow \boldsymbol{q} + \Delta \boldsymbol{q}, \qquad \boldsymbol{t} \leftarrow \boldsymbol{t} + \Delta \boldsymbol{t} \tag{44}$$

As shown in the preceding section, if the values of $\boldsymbol{r}_\alpha^{(0)}$ defined in (42) are used, the right side of (43) coincide with that of (23). In other words, while the right side of (37) of the (reduced) Gauss-Helmert method converges to that of the Gauss-Newton method, the above modified Gauss-Helmert method uses, from the beginning, the same value as the Gauss-Newton method on the right side of (43). Consequently, the only difference between the Gauss-Newton and the Gauss-Helmert methods is whether the Gauss-Newton approximation or the Gauss-Helmert approximation is used on the left side.

7. Simulated Stereo Vision

7.1 Covariance evaluation

A curved grid surface is rotated around the world origin O and translated after its scale is changed, as depicted in Fig. 1. The 3-D positions of the grid points are reconstructed by stereo vision before and after the similarity motion. The simulated stereo images are shown in Fig. 2. The image size is set to 500×800 pixels and the focal length to 600 pixels. The two cameras are positioned so that the disparity angle, or the parallax, of the world origin O is 10°. We added independent Gaussian noise of mean 0 and standard deviation σ pixels to the x and y coordinates of each of the grid points in these images and computed their 3-D positions $\hat{\boldsymbol{r}}_\alpha$ and $\hat{\boldsymbol{r}}'_\alpha$ by the method described in [15]. For optimal similarity estimation, we need to evaluate the normalized covariances $V_0[\hat{\boldsymbol{r}}_\alpha]$ and $V_0[\hat{\boldsymbol{r}}'_\alpha]$ of the reconstructed 3-D positions $\hat{\boldsymbol{r}}_\alpha$ and $\hat{\boldsymbol{r}}'_\alpha$. Following [16] and [19], we evaluated these as follows.

We fix an XYZ world coordinate system and regard the reference camera position as placed at the coordinate origin O with the optical axis aligned to the Z-axis. The image xy coordinate system is defined in such a way that its origin o is at the principal point (the intersection with the optical axis) and

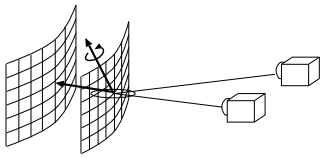

Figure 1: 3-D measurement of a grid point by stereo vision and its uncertainty ellipsoid.

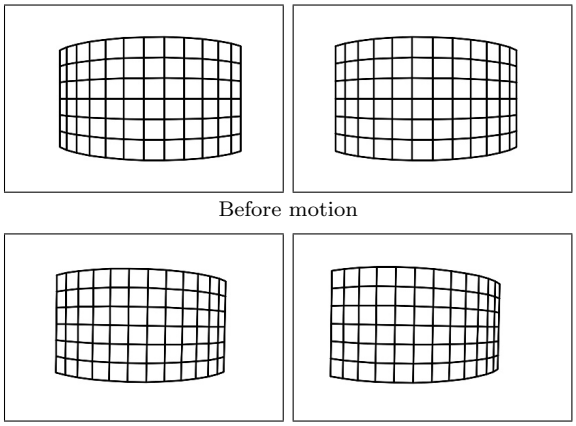

Figure 2: Simulated stereo images of the grid before and after the motion.

the x- and y-axis are parallel to the X- and Y-axis of the world coordinate system, respectively. Then, the camera is rotated around the world coordinate origin O by $\boldsymbol{R}$ (rotation matrix) and translated by $\boldsymbol{t}$ from the reference position. We call $\{\boldsymbol{R}, \boldsymbol{t}\}$ the *motion parameters* of the camera. The camera imaging geometry is modeled by perspective projection with focal length f, projecting a 3-D point onto a 2-D point (x, y) by the following relationship [8]:

$$\boldsymbol{x} \simeq \boldsymbol{P}\boldsymbol{X}, \qquad \boldsymbol{x} \equiv \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{X} \equiv \begin{pmatrix} \boldsymbol{r} \\ 1 \end{pmatrix}. \tag{45}$$

The symbol $\simeq$ means equality up to a nonzero constant multiplier, and f_0 is a scale constant of approximately the image size for stabilizing finite length computation. The 3×4 projection matrix $\boldsymbol{P}$ is given by

$$\boldsymbol{P} = \begin{pmatrix} f/f_0 & 0 & 0 \\ 0 & f/f_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}^\top & -\boldsymbol{R}^\top \boldsymbol{t} \end{pmatrix}, \tag{46}$$

where the aspect ratio is assumed to be 1 with no image skews, or so corrected by prior calibration.

We consider two cameras with motion parameters $\{\boldsymbol{R}, \boldsymbol{t}\}$ and $\{\boldsymbol{R}', \boldsymbol{t}'\}$ with focal lengths f and f', respectively. Let $\boldsymbol{P}$ and $\boldsymbol{P}'$ be the projection matrices of the respective cameras, and $\boldsymbol{x}$ and $\boldsymbol{x}'$ the images of a point in 3-D observed by the respective cameras. Image processing for correspondence detection entails uncertainty to some extent, and we model it by independent isotropic Gaussian noise of mean $\boldsymbol{0}$ and standard deviation σ (pixels). Due to noise, the

detected points $\boldsymbol{x}$ and $\boldsymbol{x}'$ do not exactly satisfy the epipolar constraint (Appendix A), so we correct $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively, to $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ that exactly satisfy the epipolar constraint in an optimal manner (Appendix B). From the corrected positions $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$, the corresponding 3-D position $\hat{\boldsymbol{r}}$ is uniquely determined (Appendix A). Note that although the noise in $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}'_\alpha$ is assumed to be independent, the noise in the corrected positions $\hat{\boldsymbol{x}}_\alpha$ and $\hat{\boldsymbol{x}}'_\alpha$ is no longer independent [14]. The normalized covariance matrices $V_0[\hat{\boldsymbol{x}}]$ and $V_0[\hat{\boldsymbol{x}}']$ and the normalized correlation matrices $V_0[\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}']$ and $V_0[\hat{\boldsymbol{x}}', \hat{\boldsymbol{x}}]$ are given as follows [14, 16]:

$$
\begin{aligned}
V_0[\hat{\boldsymbol{x}}] &= \frac{1}{f_0^2}\Big(\boldsymbol{P_k} - \frac{(\boldsymbol{P_k}\boldsymbol{F}\hat{\boldsymbol{x}}')(\boldsymbol{P_k}\boldsymbol{F}\hat{\boldsymbol{x}}')^\top}{\|\boldsymbol{P_k}\boldsymbol{F}\hat{\boldsymbol{x}}'\|^2 + \|\boldsymbol{P_k}\boldsymbol{F}^\top\hat{\boldsymbol{x}}\|^2}\Big),\\
V_0[\hat{\boldsymbol{x}}'] &= \frac{1}{f_0^2}\Big(\boldsymbol{P_k} - \frac{(\boldsymbol{P_k}\boldsymbol{F}^\top\hat{\boldsymbol{x}})(\boldsymbol{P_k}\boldsymbol{F}^\top\hat{\boldsymbol{x}})^\top}{\|\boldsymbol{P_k}\boldsymbol{F}\hat{\boldsymbol{x}}'\|^2 + \|\boldsymbol{P_k}\boldsymbol{F}^\top\hat{\boldsymbol{x}}\|^2}\Big),\\
V_0[\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}'] &= \frac{1}{f_0^2}\Big(-\frac{(\boldsymbol{P_k}\boldsymbol{F}\hat{\boldsymbol{x}}')(\boldsymbol{P_k}\boldsymbol{F}^\top\hat{\boldsymbol{x}})^\top}{\|\boldsymbol{P_k}\boldsymbol{F}\hat{\boldsymbol{x}}'\|^2 + \|\boldsymbol{P_k}\boldsymbol{F}^\top\hat{\boldsymbol{x}}\|^2}\Big)\\
&= V_0[\hat{\boldsymbol{x}}', \hat{\boldsymbol{x}}]^\top. \qquad (47)
\end{aligned}
$$

Here, $\boldsymbol{F}$ is the fundamental matrix between the two images (see Appendix A), and we define $\boldsymbol{P_k} \equiv \mathrm{diag}(1, 1, 0)$. Since the vector $\hat{\boldsymbol{X}}$ reconstructed from $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ satisfies the projection relationship in (45), vectors $\hat{\boldsymbol{x}}$ and $\boldsymbol{P}\hat{\boldsymbol{X}}$ are parallel, and so are $\hat{\boldsymbol{x}}'$ and $\boldsymbol{P}'\hat{\boldsymbol{X}}$. Thus, we have

$$
\hat{\boldsymbol{x}} \times \boldsymbol{P}\hat{\boldsymbol{X}} = \boldsymbol{0}, \qquad \hat{\boldsymbol{x}}' \times \boldsymbol{P}'\hat{\boldsymbol{X}} = \boldsymbol{0} \qquad (48)
$$

It follows that if the noise in $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ is $\Delta\hat{\boldsymbol{x}}$ and $\Delta\hat{\boldsymbol{x}}'$, respectively, the noise $\Delta\hat{\boldsymbol{X}}$ in $\hat{\boldsymbol{X}}$ satisfies to a first approximation

$$
\begin{aligned}
\Delta\hat{\boldsymbol{x}} \times \boldsymbol{P}\hat{\boldsymbol{X}} + \hat{\boldsymbol{x}} \times \boldsymbol{P}\Delta\hat{\boldsymbol{X}} = \boldsymbol{0},\\
\Delta\hat{\boldsymbol{x}}' \times \boldsymbol{P}'\hat{\boldsymbol{X}}' + \hat{\boldsymbol{x}}' \times \boldsymbol{P}'\Delta\hat{\boldsymbol{X}} = \boldsymbol{0}. \qquad (49)
\end{aligned}
$$

These are combined into one equation in the form

$$
\begin{pmatrix} \hat{\boldsymbol{x}} \times \tilde{\boldsymbol{P}} \\ \hat{\boldsymbol{x}}' \times \tilde{\boldsymbol{P}}' \end{pmatrix} \Delta\hat{\boldsymbol{r}} = \begin{pmatrix} (\boldsymbol{P}\hat{\boldsymbol{X}}) \times \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & (\boldsymbol{P}'\hat{\boldsymbol{X}}) \times \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \Delta\hat{\boldsymbol{x}} \\ \Delta\hat{\boldsymbol{x}}' \end{pmatrix}, \qquad (50)
$$

where $\Delta\hat{\boldsymbol{r}}$ is the 3-D vector of the first three components of $\Delta\hat{\boldsymbol{X}}$ and $\tilde{\boldsymbol{P}}$ and $\tilde{\boldsymbol{P}}'$ are the left 3×3 submatrices of the 3×4 projection matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$, respectively. Here, we define the product $\boldsymbol{a} \times \boldsymbol{A}$ of a 3-D vector $\boldsymbol{a}$ and a 3×3 matrix $\boldsymbol{A}$ to be the 3×3 matrix whose columns are the vector products of $\boldsymbol{a}$ and the respective columns of $\boldsymbol{A}$ [14]. Multiplying both sides by the transpose of the left side from left, we obtain

$$
\begin{aligned}
&\Big((\hat{\boldsymbol{x}} \times \tilde{\boldsymbol{P}})^\top(\hat{\boldsymbol{x}} \times \tilde{\boldsymbol{P}}) + (\hat{\boldsymbol{x}}' \times \tilde{\boldsymbol{P}}')^\top(\hat{\boldsymbol{x}}' \times \tilde{\boldsymbol{P}}')\Big)\Delta\hat{\boldsymbol{r}}\\
&= \Big((\hat{\boldsymbol{x}} \times \tilde{\boldsymbol{P}})^\top((\boldsymbol{P}\hat{\boldsymbol{X}}) \times \boldsymbol{I}) \;\; (\hat{\boldsymbol{x}}' \times \tilde{\boldsymbol{P}}')^\top((\boldsymbol{P}'\hat{\boldsymbol{X}}) \times \boldsymbol{I})\Big) \begin{pmatrix} \Delta\hat{\boldsymbol{x}} \\ \Delta\hat{\boldsymbol{x}}' \end{pmatrix}. \qquad (51)
\end{aligned}
$$

The following identities hold [14]:

$$
\begin{aligned}
&(\hat{\boldsymbol{x}} \times \tilde{\boldsymbol{P}})^\top(\hat{\boldsymbol{x}} \times \tilde{\boldsymbol{P}}) = \tilde{\boldsymbol{P}}^\top(\hat{\boldsymbol{x}} \times \boldsymbol{I})^\top(\hat{\boldsymbol{x}} \times \boldsymbol{I})\tilde{\boldsymbol{P}}\\
&= \|\hat{\boldsymbol{x}}\|^2\tilde{\boldsymbol{P}}^\top\boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}]}\tilde{\boldsymbol{P}},\\
&(\hat{\boldsymbol{x}}' \times \tilde{\boldsymbol{P}}')^\top(\hat{\boldsymbol{x}}' \times \tilde{\boldsymbol{P}}') = \tilde{\boldsymbol{P}}'^\top(\hat{\boldsymbol{x}}' \times \boldsymbol{I})^\top(\hat{\boldsymbol{x}}' \times \boldsymbol{I})\tilde{\boldsymbol{P}}'\\
&= \|\hat{\boldsymbol{x}}'\|^2\tilde{\boldsymbol{P}}'^\top\boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}']}\tilde{\boldsymbol{P}}'. \qquad (52)
\end{aligned}
$$

Here, we define

$$
\begin{aligned}
\boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}]} &\equiv \boldsymbol{I} - \mathcal{N}[\hat{\boldsymbol{x}}]\mathcal{N}[\hat{\boldsymbol{x}}]^\top,\\
\boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}']} &\equiv \boldsymbol{I} - \mathcal{N}[\hat{\boldsymbol{x}}']\mathcal{N}[\hat{\boldsymbol{x}}']^\top, \qquad (53)
\end{aligned}
$$

where $\mathcal{N}[\,\cdot\,]$ denotes normalization to unit norm ($\mathcal{N}[\boldsymbol{a}] = \boldsymbol{a}/\|\boldsymbol{a}\|$). Similarly, we have

$$
\begin{aligned}
&(\hat{\boldsymbol{x}} \times \tilde{\boldsymbol{P}})^\top((\boldsymbol{P}\hat{\boldsymbol{X}}) \times \boldsymbol{I}) = \tilde{\boldsymbol{P}}^\top\Big((\hat{\boldsymbol{x}}, \boldsymbol{P}\hat{\boldsymbol{X}})\boldsymbol{I} - (\boldsymbol{P}\hat{\boldsymbol{X}})\hat{\boldsymbol{x}}^\top\Big),\\
&(\hat{\boldsymbol{x}}' \times \tilde{\boldsymbol{P}}')^\top((\boldsymbol{P}'\hat{\boldsymbol{X}}) \times \boldsymbol{I}) = \tilde{\boldsymbol{P}}'^\top\Big((\hat{\boldsymbol{x}}', \boldsymbol{P}'\hat{\boldsymbol{X}})\boldsymbol{I} - (\boldsymbol{P}'\hat{\boldsymbol{X}})\hat{\boldsymbol{x}}'^\top\Big). \qquad (54)
\end{aligned}
$$

Using these, we can rewrite (51) in the following form:

$$
\boldsymbol{A}\Delta\hat{\boldsymbol{r}} = \boldsymbol{B}\begin{pmatrix} \Delta\hat{\boldsymbol{x}} \\ \Delta\hat{\boldsymbol{x}}' \end{pmatrix},
$$

$$
\begin{aligned}
\boldsymbol{A} &\equiv \|\hat{\boldsymbol{x}}\|^2\tilde{\boldsymbol{P}}^\top\boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}]}\tilde{\boldsymbol{P}} + \|\hat{\boldsymbol{x}}'\|^2\tilde{\boldsymbol{P}}'^\top\boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}']}\tilde{\boldsymbol{P}}',\\
\boldsymbol{B} &\equiv \Big(\tilde{\boldsymbol{P}}^\top\Big((\hat{\boldsymbol{x}}, \boldsymbol{P}\hat{\boldsymbol{X}})\boldsymbol{I} - (\boldsymbol{P}\hat{\boldsymbol{X}})\hat{\boldsymbol{x}}^\top\Big)\\
&\quad \tilde{\boldsymbol{P}}'^\top\Big((\hat{\boldsymbol{x}}', \boldsymbol{P}'\hat{\boldsymbol{X}})\boldsymbol{I} - (\boldsymbol{P}'\hat{\boldsymbol{X}})\hat{\boldsymbol{x}}'^\top\Big)\Big). \qquad (55)
\end{aligned}
$$

Hence, we obtain

$$
\Delta\hat{\boldsymbol{r}}\Delta\hat{\boldsymbol{r}}^\top = \boldsymbol{A}^{-1}\boldsymbol{B}\begin{pmatrix} \Delta\hat{\boldsymbol{x}}\Delta\hat{\boldsymbol{x}}^\top & \Delta\hat{\boldsymbol{x}}\Delta\hat{\boldsymbol{x}}'^\top \\ \Delta\hat{\boldsymbol{x}}'\Delta\hat{\boldsymbol{x}}^\top & \Delta\hat{\boldsymbol{x}}'\Delta\hat{\boldsymbol{x}}'^\top \end{pmatrix}\boldsymbol{B}^\top(\boldsymbol{A}^{-1})^\top. \qquad (56)
$$

Taking expectation on both sides, we obtain the normalized covariance matrix $V_0[\hat{\boldsymbol{r}}]$ of the reconstructed position $\hat{\boldsymbol{r}}$ in the following form:

$$
V_0[\hat{\boldsymbol{r}}] = \boldsymbol{A}^{-1}\boldsymbol{B}\begin{pmatrix} V_0[\hat{\boldsymbol{x}}] & V_0[\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}'] \\ V_0[\hat{\boldsymbol{x}}', \hat{\boldsymbol{x}}] & V_0[\hat{\boldsymbol{x}}'] \end{pmatrix}\boldsymbol{B}^\top(\boldsymbol{A}^{-1})^\top. \qquad (57)
$$

Evaluating the normalized covariance matrix $V_0[\hat{\boldsymbol{r}}_\alpha]$ in (57), we find that the uncertainty distribution has an ellipsoidal shape elongated in the depth direction, as illustrated in Fig 1. The ratio of radii is, on average over all the points, $1.00 : 1.685 : 5.090$ in the vertical, horizontal, and depth directions, respectively, meaning that the error in the depth direction is approximately five times as large as in the vertical direction. We actually measured this ratio by adding noise to the images many times and found that it is about $1.00 : 1.686 : 5.095$, a very close value to the prediction by (57).

Table 1: Three examples of the decrease in the residual J for the standard deviation $\sigma = 1.0$, 2.0, 3.0 (pixels) of the error added to the stereo images. Unchanged digits are underlined.

$\sigma = 1.0$

	Gauss-Newton	Gauss-Helmert	modified G-H
0	23368.98044646554	23368.98044646554	23368.98044646554
1	5923.560464358145	151.2986897231218	1285.065292480236
2	260.2294019664706	138.7852882171576	157.4589569299990
3	138.6397722443412	138.4925647492029	138.5000828752851
4	138.4925871308799	138.4925039364953	138.4925004345684
5	138.4924995387721	138.4924994558186	138.4924994516441
6	138.4924994515073	138.4924994515843	138.4924994514191
7	138.4924994514190	138.4924994514190	138.4924994514189
8	138.4924994514186	138.4924994514186	138.4924994514183

$\sigma = 2.0$

	Gauss-Newton	Gauss-Helmert	modified G-H
0	23705.92405252490	23705.92405252490	23705.92405252490
1	6631.055953257285	594.2288594040884	1561.554831323493
2	736.3892569773028	558.1047339694743	558.6320435827311
3	553.9802729910044	553.1013519598023	553.0948890717049
4	553.1072304698275	553.0944258828818	553.0931283140471
5	553.0934173237943	553.0931334967796	553.0931253383568
6	553.9031314363093	553.0931261287393	553.0931253320340
7	553.9031254597525	553.0931253408035	553.0931253320183
8	553.9031253346927	553.0931253326902	
9	553.9031253320766	553.0931253320288	
10	553.9031253320209	553.0931253320218	
11	553.9031253320202	553.0931253320192	

$\sigma = 3.0$

	Gauss-Newton	Gauss-Helmert	modified G-H
0	24182.94641626991	24182.94641626991	24182.94641626991
1	7749.683523117275	1385.515352477767	2074.419348255112
2	1602.792794683083	1264.397843471071	1237.583977963944
3	1243.659456346911	1237.444020868650	1237.288888227983
4	1237.725240249172	1237.323793869136	1237.288729761082
5	1237.327534736445	1237.289379736924	1237.288728527720
6	1237.292254473387	1237.288833051470	1237.288728517618
7	1237.289050047267	1237.288731797017	1237.288728517540
8	1237.288757863381	1237.288728948894	1237.288728517537
9	1237.288731196586	1237.288728533946	1237.288728517535
10	1237.288728762133	1237.288728519425	
11	1237.288728539869	1237.288728517618	
12	1237.288728519574	1237.288728517545	
13	1237.288728517725	1237.288728517538	
14	1237.288728517554		
15	1237.288728517537		

7.2 Results

Table 1 shows how the residual J of each method decreases with iterations. The standard deviation of the error added to the stereo images is $\sigma = 1.0$, 2.0, 3.0 (pixels), respectively. We started from the identity ($\boldsymbol{R} = \boldsymbol{I}$, $\boldsymbol{t} = \boldsymbol{0}$, $s = 1$) and imposed no threshold for convergence: we stopped if J stops decreasing. In each step, unchanged digits are underlined. What is marked is that the initial decrease in J of the Gauss-Helmert method is conspicuous; it drops sharply and almost abruptly. In contrast, Gauss-Newton method reduces J continuously and steadily. On the other hand, the initial decrease in J of the modified Gauss-Helmert method is smaller than the Gauss-Helmert method. This is because the modified Gauss-Helmert method computes the ML estimator of $\bar{\boldsymbol{r}}_\alpha$ for the "current" estimate of the similarity. Since the initial guess (the identity) is far from the truth, the values $\boldsymbol{r}_\alpha^{(0)}$ computed by (42) are very poor approximations, while the Gauss-Helmert method initializes $\boldsymbol{r}_\alpha^{(0)}$ by the data $\boldsymbol{r}_\alpha$ themselves, so they are better estimates of $\bar{\boldsymbol{r}}_\alpha$. *Nevertheless*, the modified Gauss-Helmert method exhibits the best convergence performance of all, and the effect is more marked as the noise in the data increases.

Figure 3**a** shows for various σ the average number of iterations over independent 1000 trials, each time with different noise. We can see that the Gauss-Helmert method converges faster than the Gauss-Newton method and in most cases the modified Gauss-Helmert method converges even faster. For comparison, Fig. 3**b** shows the result initialized using the widely used homogeneous and isotropic noise model which assumes $V_0[\boldsymbol{r}_\alpha] = V_0[\boldsymbol{r}_\alpha] = \boldsymbol{I}$ for all α. In this case, we compute the centroids $\boldsymbol{r}_c$ and $\boldsymbol{r}'_c$ of the data $\{\boldsymbol{r}_\alpha\}$ and $\{\boldsymbol{r}'_\alpha\}$, respectively, and the deviations $\tilde{\boldsymbol{r}}_\alpha = \boldsymbol{r}_\alpha - \boldsymbol{r}_c$ and $\tilde{\boldsymbol{r}}'_\alpha = \boldsymbol{r}'_\alpha - \boldsymbol{r}'_c$ from their respective centroids. As is well known, the scale change s can be estimated by

$$s = \sqrt{\frac{\sum_{\alpha=1}^N \|\tilde{\boldsymbol{r}}'_\alpha\|^2}{\sum_{\alpha=1}^N \|\tilde{\boldsymbol{r}}_\alpha\|^2}}, \tag{58}$$

and the rotation $\boldsymbol{R}$ is computed from $\{\tilde{\boldsymbol{r}}_\alpha\}$ and $\{\tilde{\boldsymbol{r}}'_\alpha\}$ by the method of singular value decomposition (SVD) (Appendix C). The translation $\boldsymbol{t}$ is determined from $\boldsymbol{t} = \boldsymbol{r}'_c - s\boldsymbol{R}\boldsymbol{r}_c$. As we see from Fig. 3**b**, the Gauss-Newton method converges faster than the Gauss-Helmert method for such an accurate initialization when the noise in the data is small. However, the Gauss-Helmert method becomes more efficient as the noise increases. *Yet*, the modified Gauss-Helmert method is always the most efficient for all the noise level.

We next evaluated the accuracy of the computed rotation $\hat{\boldsymbol{R}}$, the translation $\hat{\boldsymbol{t}}$, and the scale change $\hat{s}$. Let $\bar{\boldsymbol{R}}$, $\bar{\boldsymbol{t}}$, and $\bar{s}$ be their true values, respectively. We computed the rotation angle $\delta\Omega$ (in degree) of the relative rotation $\hat{\boldsymbol{R}}\bar{\boldsymbol{R}}^{-1}$, the translation error $\delta\boldsymbol{t} = \hat{\boldsymbol{t}} - \bar{\boldsymbol{t}}$ and the scale change error $\delta s = \hat{s} - \bar{s}$. Repeating this 1000 times with σ fixed, each time using different image noise, we evaluated the RMS errors

$$E_{\boldsymbol{R}} = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}(\delta\Omega^{(a)})^2},$$
$$E_{\boldsymbol{t}} = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}\|\delta\boldsymbol{t}^{(a)}\|^2},$$
$$E_s = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}(\delta s^{(a)})^2}, \tag{59}$$

where the superscript (a) denotes the value of the ath trial. Figure 4 plots these for various σ. Since, as far as accuracy is concerned, there are no discernible differences between the Gauss-Newton, the Gauss-Helmert, and the modified Gauss-Helmert methods, we compared their accuracy with the use of the homogeneous and isotropic noise model. It is clearly demonstrated that accurate estimation cannot be done unless the inhomogeneity and anisotropy of the 3-D sensing data.

8. Real Data Example

Turkey is a country with frequent earthquakes, and researchers monitor the land deformation using GPS data. Table 2 shows the X, Y, and Z coordinates (in meters) of five positions selected from a landslide area near Istanbul in October 1997 and March 1998 [1]. The absolute positions are corrected in reference to control points in stable areas. The covariance matrices of these values are estimated using statistical regression analysis. For the 1997 data, their covariance matrices (in the order listed in the table) are

$$\begin{pmatrix} 34 & 10 & 17 \\ 10 & 12 & 7 \\ 17 & 7 & 33 \end{pmatrix}, \quad \begin{pmatrix} 234 & 83 & 136 \\ 83 & 97 & 58 \\ 136 & 58 & 245 \end{pmatrix}, \quad \begin{pmatrix} 24 & 8 & 12 \\ 8 & 10 & 6 \\ 12 & 6 & 25 \end{pmatrix},$$
$$\begin{pmatrix} 63 & 25 & 36 \\ 25 & 28 & 16 \\ 36 & 16 & 53 \end{pmatrix}, \quad \begin{pmatrix} 22 & 8 & 12 \\ 8 & 9 & 5 \\ 12 & 5 & 23 \end{pmatrix},$$

multiplied by 10^{-8}. For the 1998 data,

$$\begin{pmatrix} 51 & 18 & 23 \\ 18 & 18 & 13 \\ 23 & 13 & 30 \end{pmatrix}, \quad \begin{pmatrix} 323 & 140 & 159 \\ 140 & 148 & 100 \\ 159 & 100 & 218 \end{pmatrix}, \quad \begin{pmatrix} 41 & 14 & 19 \\ 14 & 16 & 11 \\ 19 & 11 & 28 \end{pmatrix},$$
$$\begin{pmatrix} 141 & 47 & 70 \\ 47 & 49 & 38 \\ 70 & 38 & 96 \end{pmatrix}, \quad \begin{pmatrix} 59 & 20 & 29 \\ 20 & 24 & 16 \\ 29 & 16 & 43 \end{pmatrix}$$

multiplied by 10^{-8}.

Table 3**a** shows the changes in J of the three methods initialized by the identity (unchanged digits are

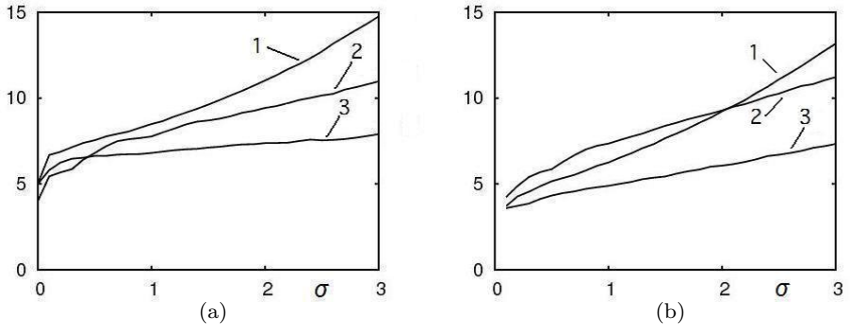

Figure 3: The average number of iterations for various noise level σ over 1000 independent trials: 1. Gauss-Newton, 2. Gauss-Helmert, 3. modified Gauss-Helmert. (a) Initialized by the identity. (b) Initialized using the homogeneous and isotropic noise model.

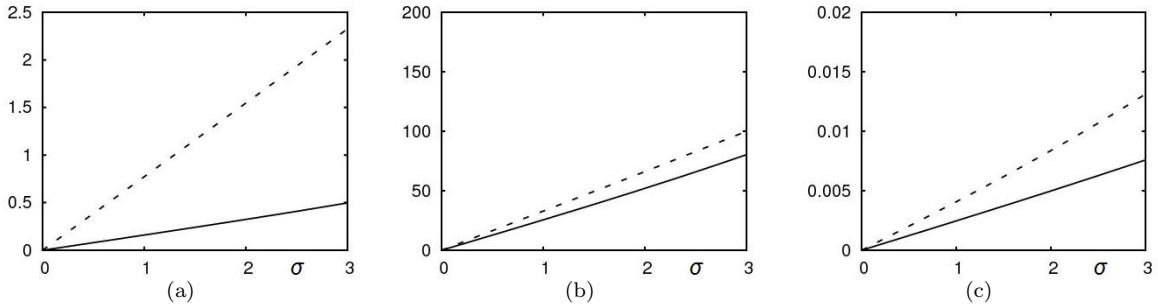

Figure 4: The RMS error vs. the standard deviation σ of the noise added to the stereo images: (a) Rotation, (b) Translation, (c) Scale change. The dotted lines are for the homogeneous and isotropic noise model.

underlined). We see that the values fluctuate in the digits below the eighth or ninth place. This phenomenon has been known in geodesic science and has sometimes been regarded as a shortcoming of the Gauss-Helmert method. However, this is common to the three methods. We see that although the data in Table 2 have many digits, the four or five digits are common to all the five observation points and that the differences between 1997 and 1998 occur only in the last two or three digits. As is well known in numerical analysis, such data are very prone to the magnification of the finite length computation error. We conclude that for all the methods the value of J in Table 3**a** has practically converged at the second iteration and that the subsequently fluctuates in the digits below the eighth or ninth place indicate the accuracy limit due to finite length computation (double precision in our case). Accordingly, the similarity solutions also converge up to six or seven digits. Table 3**b** lists the meaningful digits with utmost attainable accuracy, to which the solution using the conventional homogeneous and isotropic noise model agree at most in the first digit. This illustrates that accurate estimation cannot be done without using the optimal computation shown here.

Table 2: The 3-D data of five points near Istanbul in October 1997 and March 1998 [1].

October 1997

X	Y	Z
4233187.8344	2308228.6785	4161469.1229
4233190.6059	2308518.3249	4161336.2582
4233429.1004	2307875.2240	4161292.4034
4233259.8205	2307712.3025	4161553.4880
4233770.4580	2308340.5240	4160740.3286

March 1998

X	Y	Z
4233187.8612	2308228.7042	4161469.1383
4233190.6124	2308518.3166	4161336.2682
4233429.1008	2307875.2239	4161292.4029
4233259.8309	2307712.2990	4161553.5007
4233770.4534	2308340.5219	4160740.3181

9. Conclusions

Because 3-D data are acquired using 3-D sensing such as stereo vision and laser range finders, they have inhomogeneous and anisotropic noise. In this paper, we studied optimal computation of the similarity (rotation, translation, and scale change) of such 3-D data. We compared the two well known methods that are suitable for this purpose: the Gauss-Newton method widely regarded as the standard optimization technique by computer vision and robotics engineers, and the the Gauss-Helmert method popular among geodetic scientists. Our observations are summarized as follows:

1. The Gauss-Helmert method has a very similar form to the Gauss-Newton method. This

Table 3: (a) Changes in the residual J ($\times 10^{-6}$) for the data in Table 2. Unchanged digits are underlined. (b) The translation $\boldsymbol{t} = (t_1, t_2, t_3)^\top$ (in meters), the scale change s, the rotation axis $\boldsymbol{l} = (l_1, l_2, l_3)^\top$ (unit vector), the rotation angle Ω (in degree), and the residual J ($\times 10^{-6}$) estimated from the data in Table 2. The optimal solution lists meaningful digits with utmost accuracy.

	Gauss-Newton	Gauss-Helmert	modified G-H
0	13.90466081612066	13.90466081612066	13.90466081612066
1	6.891471483617726	6.891561230647212	6.891490551983246
2	6.409224270624541	6.409224054092592	6.409224341043299
3	6.409224500361636	6.409224582261247	6.409224199848933
4	6.409224272986374	6.409224257902020	6.409224110716818
5	6.409224117901836	6.409224682772476	6.409224152869662
6	6.409224095494097	6.409224242530478	6.409224325787738
7	6.409224093119093	6.409223929884663	6.409224417144383
8	6.409224304185821	6.409223999141919	6.409224110716818
9	6.409223917076926	6.409224189433356	6.409224080748898
10	6.409224111129707	6.409224068655141	6.409224155956410

(a)

	conventional	optimal
t_1	-199.8604	-274.6708
t_2	42.52530	100.2332
t_3	143.6579	140.7879
s	1.000004	1.000009
l_1	-0.04950650	-0.008546834
l_2	0.9328528	0.8213706
l_3	-0.3568400	-0.5703308
Ω	0.002242810	0.002887644
J	9.242858×10^{-6}	6.409224×10^{-6}

(b)

becomes evident if the unknowns are reduced to the similarity variables alone by eliminating out the Lagrange multipliers. We can view the Gauss-Helmert iterations a variant of the Gauss-Newton iterations with a special approximation of the Hessian of the residual, which may be called the "Gauss-Helmert approximation".

2. In the course of the iterations, the Gauss-Helmert method sharply drops the residual at first, but the subsequent convergence is not necessarily fast. In contrast, the Gauss-Newton iterations continuously and steadily reduce the residual. Overall, the Gauss-Newton method is more efficient when the initialization is accurate and the data noise is low, but the Gauss-Helmert method becomes more efficient when the initialization is poor or the noise level is high.

Then, we combined the merits of the two methods to define the "modified Gauss-Helmert method". We observed that:

3. The initial drop of the residual is not so sharp as the Gauss-Helmert method, but the convergence is smooth as the Gauss-Newton method.

4. Irrespective of the accuracy of the initialization or the noise level of the data, the convergence is always faster than the Gauss-Newton or the Gauss-Helmert method.

We also described how to evaluate the covariance matrices of the inhomogeneous and anisotropic noise associated with stereo vision. Finally, we showed an application to real GPS geodetic data and found that:

5. The widely used homogeneous and isotropic noise model is insufficient for accurate estimation.

6. The limited accuracy, for which the Gauss-Helmert method has sometimes been to blame, is due to the inherent nature of GPS geodetic data.

It is expected that our proposed modified Gauss-Helmert will become a standard tool in both computer vision and robotics engineering and geodetic science.

Acknowledgments. The authors thank Orhan Akyilmaz of Istanbul Institute of Technology, Turkey for providing the GPS data and doing helpful discussions. They also thank Takuto Honda of Okayama University and Hiroki Hara of SANYO Electric Co. Ltd. for helping our numerical experiments. This work was supported in part by the Ministry of Education, Culture, Sports, Science, and Technology, Japan, under a Grant in Aid for Scientific Research (C 21500172).

References

[1] Acar, A., Özlüdemir, M.T., Akyilmaz, O., Celik, R.N., Ayan, T., Deformation analysis with total least squares, *Nat. Hazards Earth Syst. Sci.*, **6**, 663–669, 2006.

[2] Akyilmaz, O., Solution of the heteroscedastic datum transformation problems, *Abstract of 1st Int. Workshop the Quality of Geodetic Observation and Monitoring Systems*, April, 2011, Garching/Munich, Germany. `http://www.gih.uni-hannover.de/qugoms2011/ProgramAbstracts1.htm`

[3] Arun, K.S., Huang, T.S., Blostein, S.D., Least squares fitting of two 3-D point sets, *IEEE Trans. Patt. Anal. Mach. Intell.*, **9**, 698–700, 1987.

[4] Dorst, L., First order error propagation of the Procrustes method for 3D attitude estimation, *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**, 221–229, 2005.

[5] Felus, Y.A., Burch, R.C., On symmetrical three-dimensional datum conversion, *GPS Solutions*, **13**, 65–74, 2009.

[6] Förstner, W., On weighting and choosing constraints for optimally reconstructing the geometry of image triplets, *Proc. 6th Euro. Conf. Comput. Vision*, **2**, 669–701, 2000.

[7] Grafarend, E.W., Awange, J.L., Nonlinear analysis of the three-dimensional datum transformation [conformal group $C_7(3)$], *J. Geodesy*, **77**, 66–76, 2003.

[8] Hartley, R., Zisserman, A., *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, Cambridge, U.K., 2004.

[9] Horn, B.K.P., Closed-form solution of absolute orientation, using quaternions, *Int. J. Opt. Soc. Am.*, **A-4**, 629–642, 1987.

[10] Horn, B.K.P., Hildren, H.M., Negahdaripour, S., Closed-form solution of absolute orientation, using orthonormal matrices, *Int. J. Opt. Soc. Am.*, **A-5**, 1127–1135, 1988.

[11] Kanatani, K., *Group-Theoretical Methods in Image Understanding*, Springer, Berlin, Germany, 1990.

[12] Kanatani, K., *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.

[13] Kanatani, K., Analysis of 3-D rotation fitting, *IEEE Trans. Patt. Anal. Mach. Intell.*, **16**, 543–449, 1994.

[14] Kanatani, K., *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, the Netherlands, 1996; reprinted, 2005, Dover, New York, U.S.A.

[15] Kanatani, K., Sugaya, Y., Niitsuma, H., Triangulation from two views revisited: Hartley-Sturm vs. optimal correction, *Proc. 19th British Machine Vision Conf.*, 173–182, 2008.

[16] Kanazawa, Y., Kanatani, K., Reliability of 3-D reconstruction by stereo vision, *IEICE Trans. Inf. Syst.*, **E78-D**, 1301–1306, 1995.

[17] Mikhail, E.M., Ackermann, F., (1976). *Observations and Least Squares*, University Press of America, Lanham, MD., U.S.A., 1976.

[18] Neitzel, F., Generalization of total least-squares on example of unweighted and weighted 2D similarity transformations, *J. Geodesy*, **84**, 751–762, 2010.

[19] Niitsuma, H., Kanatani, K., Optimal computation of 3-D rotation under inhomogeneous anisotropic noise, *Proc. 12th IAPR Conf. Machine Vision Applications*, 112–115, 2011.

[20] Niitsuma, H., Kanatani, K., Optimal computation of 3-D similarity from space data with inhomogeneous noise distributions, *Pro. 16th Symp. Sensing via Imaging Information*, IS4/03-1–IS4/03-8, 2011.

[21] Ohta, N., Kanatani, K., Optimal estimation of three-dimensional rotation and reliability evaluation, IEICE Trans. Inf. Syst., E81-D, 1247–1252, 1998.

[22] Perwass, C., Gebken, C., Sommer, G., Geometry and kinematics with uncertain data, Proc. 9th Euro. Conf. Comput. Vision, **1**, 225–237, 2006.

[23] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, U.K., 1992.

[24] Umeyama, S., Least-squares estimation of transformation parameters between two point sets, *IEEE Trans. Patt. Anal. Mach. Intell.*, **13**, 379–380, 1991.

Appendix

A. 3-D reconstruction by stereo Vision

If a point (x, y) in the first image of a stereo pair corresponds to (x', y') in the second, they satisfy the following *epipolar constraint* [8]:

$$\left(\begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \boldsymbol{F} \begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix}\right) = 0, \tag{60}$$

Here, the matrix $\boldsymbol{F} = (F_{ij})$, called the *fundamental matrix*, is defined by

$$\begin{aligned}
F_{11} &= \begin{vmatrix} P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P'_{21} & P'_{22} & P'_{23} & P'_{24} \\ P'_{31} & P'_{32} & P'_{33} & P'_{34} \end{vmatrix}, \quad F_{12} = \begin{vmatrix} P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P'_{31} & P'_{32} & P'_{33} & P'_{34} \\ P'_{11} & P'_{12} & P'_{13} & P'_{14} \end{vmatrix}, \\
F_{13} &= \begin{vmatrix} P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P'_{11} & P'_{12} & P'_{13} & P'_{14} \\ P'_{21} & P'_{22} & P'_{23} & P'_{24} \end{vmatrix}, \quad F_{21} = \begin{vmatrix} P_{31} & P_{32} & P_{33} & P_{34} \\ P_{11} & P_{12} & P_{13} & P_{14} \\ P'_{21} & P'_{22} & P'_{23} & P'_{24} \\ P'_{31} & P'_{32} & P'_{33} & P'_{34} \end{vmatrix}, \\
F_{22} &= \begin{vmatrix} P_{31} & P_{32} & P_{33} & P_{34} \\ P_{11} & P_{12} & P_{13} & P_{14} \\ P'_{31} & P'_{32} & P'_{33} & P'_{34} \\ P'_{11} & P'_{12} & P'_{13} & P'_{14} \end{vmatrix}, \quad F_{23} = \begin{vmatrix} P_{31} & P_{32} & P_{33} & P_{34} \\ P_{11} & P_{12} & P_{13} & P_{14} \\ P'_{11} & P'_{12} & P'_{13} & P'_{14} \\ P'_{21} & P'_{22} & P'_{23} & P'_{24} \end{vmatrix}, \\
F_{31} &= \begin{vmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P'_{21} & P'_{22} & P'_{23} & P'_{24} \\ P'_{31} & P'_{32} & P'_{33} & P'_{34} \end{vmatrix}, \quad F_{32} = \begin{vmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P'_{31} & P'_{32} & P'_{33} & P'_{34} \\ P'_{11} & P'_{12} & P'_{13} & P'_{14} \end{vmatrix}, \\
F_{33} &= \begin{vmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P'_{11} & P'_{12} & P'_{13} & P'_{14} \\ P'_{21} & P'_{22} & P'_{23} & P'_{24} \end{vmatrix},
\end{aligned} \tag{61}$$

where P_{ij} and P'_{ij} are the (ij) elements of the projection matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$ of the first and the second camera, respectively, as defined in (46). If we let $\boldsymbol{r} = (X, Y, Z)^\top$ be the 3-D point we are looking at, we obtain from the perspective projection relationship in (45)

$$\begin{aligned}
x &= f_0 \frac{P_{11}X + P_{12}X + P_{13}X + P_{14}f_0}{P_{31}X + P_{32}X + P_{33}X + P_{34}f_0}, \\
y &= f_0 \frac{P_{21}X + P_{22}X + P_{23}X + P_{24}f_0}{P_{31}X + P_{32}X + P_{33}X + P_{34}f_0}, \\
x' &= f_0 \frac{P'_{11}X + P'_{12}X + P'_{13}X + P'_{14}f_0}{P'_{31}X + P'_{32}X + P'_{33}X + P'_{34}f_0}, \\
y' &= f_0 \frac{P'_{21}X + P'_{22}X + P'_{23}X + P'_{24}f_0}{P'_{31}X + P'_{32}X + P'_{33}X + P'_{34}f_0}.
\end{aligned} \tag{62}$$

Clearing the fraction, we obtain the following linear equations:

$$\begin{aligned}
&\begin{pmatrix} xP_{31} - f_0P_{11} & xP_{32} - f_0P_{12} & xP_{33} - f_0P_{13} \\ yP_{31} - f_0P_{21} & yP_{32} - f_0P_{22} & yP_{33} - f_0P_{23} \\ x'P'_{31} - f_0P'_{11} & x'P'_{32} - f_0P'_{12} & x'P'_{33} - f_0P'_{13} \\ y'P'_{31} - f_0P'_{21} & y'P'_{32} - f_0P'_{22} & y'P'_{33} - f_0P'_{23} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \\
&= - \begin{pmatrix} xP_{34} - f_0P_{14} \\ yP_{34} - f_0P_{24} \\ x'P'_{34} - f_0P'_{14} \\ y'P'_{34} - f_0P'_{24} \end{pmatrix}.
\end{aligned} \tag{63}$$

These are four equations for three unknowns X, Y, and Z, but because the epipolar constraint in (60)

is satisfied, the solution is unique. In fact, (60) is derived as the necessary and sufficient condition for (63) to have a unique solution.

B. Optimal Triangulation

Let (x, y) and (x', y') be a pair of corresponding points between stereo images. Since correspondence detection by an image processing operations inevitably entails uncertainty to some degree, they do not necessarily satisfy the epipolar constraint in (60). Geometrically, this corresponds to the fact that the lines of sight starting from the lens center of the two cameras and passing through (x, y) and (x', y') in the image plane do not necessarily meet in the scene. For optimal 3-D reconstruction, we need to correct (x, y) and (x', y') optimally to $(\hat{x}, \hat{y})$ and $(\hat{x}', \hat{y}')$ so that their lines of sight intersect, i.e., (60) is satisfied. By "optimally", we mean that the correction is done in such a way that the *reprojection error* $(\hat{x}-x)^2+(\hat{y}-y)^2+(\hat{x}'-x')^2+(\hat{y}'-y')^2$ is minimized. This correction procedure goes as follows [15]:

1. Let $E_0 = \infty$ (a sufficiently large number), $\hat{x} = x$, $\hat{y} = y$, $\hat{x}' = x'$, $\hat{y}' = y'$, and $\tilde{x} = \tilde{y} = \tilde{x}' = \tilde{y}' = 0$, and express the fundamental matrix $\boldsymbol{F} = (F_{ij})$ as the 9-D vector $\boldsymbol{f} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top$.

2. Compute the following 9×9 matrix $V_0[\hat{\boldsymbol{\xi}}]$ and the 9-D vector $\boldsymbol{\xi}^*$:

$$V_0[\hat{\boldsymbol{\xi}}] = \begin{pmatrix} \hat{x}^2+\hat{x}'^2 & \hat{x}'\hat{y}' & f_0\hat{x}' & \hat{x}\hat{y} & 0 \\ \hat{x}'\hat{y}' & \hat{x}^2+\hat{y}'^2 & f_0\hat{y}' & 0 & \hat{x}\hat{y} \\ f_0\hat{x}' & f_0\hat{y}' & f_0^2 & 0 & 0 \\ \hat{x}\hat{y} & 0 & 0 & \hat{y}^2+\hat{x}'^2 & \hat{x}'\hat{y}' \\ 0 & \hat{x}\hat{y} & 0 & \hat{x}'\hat{y}' & \hat{y}^2+\hat{y}'^2 \\ 0 & 0 & 0 & f_0\hat{x}' & f_0\hat{y}' \\ f_0\hat{x} & 0 & 0 & f_0\hat{y} & 0 \\ 0 & f_0\hat{x} & 0 & 0 & f_0\hat{y} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & f_0\hat{x} & 0 & 0 \\ 0 & 0 & f_0\hat{x} & 0 \\ 0 & 0 & 0 & 0 \\ f_0\hat{x}' & f_0\hat{y} & 0 & 0 \\ f_0\hat{y}' & 0 & f_0\hat{y} & 0 \\ f_0^2 & 0 & 0 & 0 \\ 0 & f_0^2 & 0 & 0 \\ 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\boldsymbol{\xi}^* = \begin{pmatrix} \hat{x}\hat{x}'+\hat{x}'\tilde{x}+\hat{x}\tilde{x}' \\ \hat{x}\hat{y}'+\hat{y}'\tilde{x}+\hat{x}\tilde{y}' \\ \hat{x}+\tilde{x} \\ \hat{y}\hat{x}'+\hat{x}'\tilde{y}+\hat{y}\tilde{x}' \\ \hat{y}\hat{y}'+\hat{y}'\tilde{y}+\hat{y}\tilde{y}' \\ \hat{y}+\tilde{y} \\ \hat{x}'+\tilde{x}' \\ \hat{y}'+\tilde{y}' \\ 1 \end{pmatrix}. \tag{64}$$

3. Update $\tilde{x}$, $\tilde{y}$, $\tilde{x}'$, $\tilde{y}'$, $\hat{x}$, $\hat{y}$, $\hat{x}'$, and $\hat{y}'$ as follows:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \leftarrow \frac{(\boldsymbol{f}, \boldsymbol{\xi}^*)}{(\boldsymbol{f}, V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{f})} \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \end{pmatrix} \begin{pmatrix} \hat{x}' \\ \hat{y}' \\ 1 \end{pmatrix},$$

$$\begin{pmatrix} \tilde{x}' \\ \tilde{y}' \end{pmatrix} \leftarrow \frac{(\boldsymbol{f}, \boldsymbol{\xi}^*)}{(\boldsymbol{f}, V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{f})} \begin{pmatrix} F_{11} & F_{21} & F_{31} \\ F_{12} & F_{22} & F_{32} \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix},$$

$$\hat{x} \leftarrow x-\tilde{x}, \quad \hat{y} \leftarrow y-\tilde{y}, \quad \hat{x}' \leftarrow x'-\tilde{x}', \quad \hat{y}' \leftarrow y'-\tilde{y}'. \tag{65}$$

4. Compute the reprojection error E by

$$E = \tilde{x}^2+\tilde{y}^2+\tilde{x}'^2+\tilde{y}'^2. \tag{66}$$

If $E \approx E_0$, return $(\hat{x}, \hat{y})$ and $(\hat{x}', \hat{y}')$ and stop. Else, let $E_0 \leftarrow E$ and go back to Step 2.

C. Homogeneous isotropic noise model

Various methods are known for optimally computing the 3-D rotation for homogeneous and isotropic noise [3, 9, 10, 13, 24], but all are mathematically equivalent. The simplest is the following method in terms of the singular value decomposition (SVD) [12]:

1. Compute the following correlation matrix $\boldsymbol{N}$ between the 3-D positions $\boldsymbol{r}_\alpha$ and $\boldsymbol{r}'_\alpha$ before and after the rotations:

$$\boldsymbol{N} = \sum_{\alpha=1}^{N} \boldsymbol{r}'_\alpha \boldsymbol{r}_\alpha^\top. \tag{67}$$

2. Compute the SVD of $\boldsymbol{N}$ in the form

$$\boldsymbol{N} = \boldsymbol{U}\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)\boldsymbol{V}^\top, \tag{68}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ (≥ 0) are the singular values.

3. Return the following rotation matrix:

$$\boldsymbol{R} = \boldsymbol{U}\mathrm{diag}(1, 1, \det(\boldsymbol{U}\boldsymbol{V}^\top))\boldsymbol{V}^\top. \tag{69}$$