Memoirs of the Faculty of Engineering, Okayama University, Vol. 46, pp. 1-9, January 2012

Optimal Computation of 3-D Similarity from Space Data with Inhomogeneous Noise Distributions

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

(Received November 21, 2011)

We optimally estimate the similarity (rotation, translation, and scale change) between two sets of 3-D data in the presence of inhomogeneous and anisotropic noise. Adopting the Lie algebra representation of the 3-D rotational change, we derive the Levenberg-Marquardt procedure for simultaneously optimizing the rotation, the translation, and the scale change. We test the performance of our method using simulated stereo data and real GPS geodetic sensing data. We conclude that the conventional method assuming homogeneous and isotropic noise is insufficient and that our simultaneous optimization scheme can produce an accurate solution.

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 (translation and rotation) of the robot between two time instances. A similar task occurs in reconstructing the entire shape of a 3-D object by 3-D sensing, for which we need multiple sensors, because one sensor can reconstruct only the part that is visible from it. Hence, we need to map the partial shape obtained from one sensor to the 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, 7, 8, 20].

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 accurately estimated from the displacement of the 3-D points, and the scale change is easily measured by comparing the size of the corresponding parts, while rotation estimation is not so straightforward in the presence of noise. However, almost all rotation estimation algorithms proposed in the past [2, 5, 9, 10, 13, 26] 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 Ohta and Kanatani [22] 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]. As a result, a solution statistically equivalent to maximum likelihood (ML) is obtained, but it does not necessarily coincide with the ML solution itself. Recently, Niitsuma and Kanatani [21] presented a numerical scheme for computing an exact ML solution. Following Ohta and Kanatani [22], they represented the 3-D rotation by a quaternion and computed the ML solution using the FNS (Fundamental Numerical Scheme) of Chojnacki et al. [4]. They demonstrated that the resulting solution nearly achieves the theoretical accuracy limit called the *KCR lower bound* [14, 15].

In this paper, we include the translation and scale change as well and optimize all the parameters simultaneously. Scale changes may not be considered if the measurement is done by the same sensor before and after the object motion, but in that case we can

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

simply drop the corresponding terms in the optimization. The unique aspect of our scheme is that we use the *Lie algebra representation* [11] of the 3-D rotational change, by which *we need not parameterize the rotation at all*; we can use the rotation matrix itself throughout.

This technique is well known in physics and used in many vision applications including robot navigation [3, 19], object tracking [6], 3-D shape reconstruction from images [16], panoramic image generation [24], fundamental matrix computation [25], but mostly for tracking rotational changes. In this paper, we use the Lie algebra representation for deriving the Levenberg-Marquardt procedure for simultaneously optimizing the rotation, the translation, and the scale change.

We test the performance of our scheme using simulated stereo data and real GPS geodetic sensing data and examine to what degree the accuracy improves by our simultaneous optimization.

2. Maximum Likelihood of Similarity

Suppose we are given 3-D position measurements $\boldsymbol{r}_\alpha$ and $\boldsymbol{r}'_\alpha$, $\alpha = 1, ..., N$, before and after a similiarity 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*, desicribes 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 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. Optimal estimation of similarity in the sense of maximum likelihood (ML) is to minimize the *Mahalanobis distance* [14] (the multiplier 1/e 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

$$\bar{\boldsymbol{r}}'_\alpha = s\boldsymbol{R}\bar{\boldsymbol{r}}_\alpha + \boldsymbol{t}, \qquad \alpha = 1, ..., N, \quad (2)$$

for some $\boldsymbol{R}$ (rotation), $\boldsymbol{t}$ (translation) and s (scale change). Throughout this paper, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$. Introducing Lagrange multipliers $\boldsymbol{\lambda}_\alpha$ for the constrain of Eq. (2), we let

$$\tilde{J} = \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha, V_0[\boldsymbol{r}_\alpha]^{-1}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha)) + \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha, V_0[\boldsymbol{r}'_\alpha]^{-1}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha)) - \sum_{\alpha=1}^{N}(\boldsymbol{\lambda}_\alpha, \bar{\boldsymbol{r}}'_\alpha - s\boldsymbol{R}\bar{\boldsymbol{r}}'_\alpha - \boldsymbol{t}). \quad (3)$$

The $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$ that minimize Eq. (1) subject to Eq. (2) are obtained by setting the derivatives of Eq. (3) with respect to $\bar{\boldsymbol{r}}_\alpha$ and $\bar{\boldsymbol{r}}'_\alpha$ to $\boldsymbol{0}$. Noting the identity $(\boldsymbol{\lambda}_\alpha, \boldsymbol{R}\bar{\boldsymbol{r}}'_\alpha) = (\boldsymbol{R}^\top\boldsymbol{\lambda}_\alpha, \bar{\boldsymbol{r}}'_\alpha)$, we obtain

$$\nabla_{\bar{\boldsymbol{r}}_\alpha}\tilde{J} = -V_0[\boldsymbol{r}_\alpha]^{-1}(\boldsymbol{r}_\alpha - \bar{\boldsymbol{r}}_\alpha) + s\boldsymbol{R}^\top\boldsymbol{\lambda}_\alpha, \quad \nabla_{\bar{\boldsymbol{r}}'_\alpha}\tilde{J} = -V_0[\boldsymbol{r}'_\alpha]^{-1}(\boldsymbol{r}'_\alpha - \bar{\boldsymbol{r}}'_\alpha) - \boldsymbol{\lambda}_\alpha. \quad (4)$$

Setting these to $\boldsymbol{0}$, we obtain

$$\bar{\boldsymbol{r}}_\alpha = \boldsymbol{r}_\alpha - sV_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{\lambda}_\alpha, \quad \bar{\boldsymbol{r}}'_\alpha = \boldsymbol{r}'_\alpha + V_0[\boldsymbol{r}'_\alpha]\boldsymbol{\lambda}_\alpha. \quad (5)$$

Substituting these into Eq. (2), we have

$$\boldsymbol{r}'_\alpha + V_0[\boldsymbol{r}'_\alpha]\boldsymbol{\lambda}_\alpha = s\boldsymbol{R}(\boldsymbol{r}_\alpha - sV_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{\lambda}_\alpha) + \boldsymbol{t}, \quad (6)$$

from which $\boldsymbol{\lambda}_\alpha$ is obtained in the form

$$\boldsymbol{\lambda}_\alpha = -\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \quad (7)$$

where the matrix $\boldsymbol{W}_\alpha$ is defined by

$$\boldsymbol{W}_\alpha = (s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + V_0[\boldsymbol{r}'_\alpha])^{-1}. \quad (8)$$

Substituting Eqs. (5), after replacing $\boldsymbol{\lambda}_\alpha$ by Eq. (7), into Eq. (1), we can write Eq. (1) as follows:

$$\begin{aligned} J &= \frac{1}{2}\sum_{\alpha=1}^{N}(V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), V_0[\boldsymbol{r}_\alpha]^{-1}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &\quad + \frac{s^2}{2}\sum_{\alpha=1}^{N}(V_0[\boldsymbol{r}'_\alpha]\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), V_0[\boldsymbol{r}'_\alpha]^{-1}V_0[\boldsymbol{r}'_\alpha]\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &= \frac{s^2}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \boldsymbol{W}_\alpha\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]V_0[\boldsymbol{r}_\alpha]^{-1}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) + \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \boldsymbol{W}_\alpha V_0[\boldsymbol{r}'_\alpha]V_0[\boldsymbol{r}'_\alpha]^{-1}V_0[\boldsymbol{r}'_\alpha]\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &= \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \boldsymbol{W}_\alpha(s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + V_0[\boldsymbol{r}'_\alpha])\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &= \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \boldsymbol{W}_\alpha\boldsymbol{W}_\alpha^{-1}\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &= \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})). \quad (9) \end{aligned}$$

Our task is to minimize this function with respect to $\boldsymbol{R}$, $\boldsymbol{t}$, and s.

3. Gradient Computation

The rotation matrix $\boldsymbol{R}$ has nine elements, but the constraint $\boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}$ leaves only three degrees of freedom. Thus, the change of rotation is specified by three parameters. The important fact is that *we need not parameterize* $\boldsymbol{R}$ *itself*; we only need to specify its *changes*. From $\boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}$, we see that the variation of $\Delta\boldsymbol{R}$ of $\boldsymbol{R}$ satisfies $\Delta\boldsymbol{R}\boldsymbol{R}^\top + \boldsymbol{R}\Delta\boldsymbol{R}^\top = \boldsymbol{O}$ to a first approximation. So, $(\Delta\boldsymbol{R}\boldsymbol{R}^\top)^\top = -\Delta\boldsymbol{R}\boldsymbol{R}^\top$, which implies that $\Delta\boldsymbol{R}\boldsymbol{R}^\top$ is antisymmetric. Hence, there exist ω_1, ω_2, and ω_3 such that

$$\Delta\boldsymbol{R}\boldsymbol{R}^\top = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{10}$$

In mathematics, this form is called an *infinitesimal rotation*, and the set of all infinitesimal rotations is a linear space $so(3)$, called the *Lie algebra*[1] of the group of rotations $SO(3)$, spanned by ω_1, ω_2, and ω_3 [11].

Let us define the product $\boldsymbol{a} \times \boldsymbol{T}$ of a vector $\boldsymbol{a}$ and matrix $\boldsymbol{T}$ to be the matrix obtained by the vector products of $\boldsymbol{a}$ and the corresponding columns of $\boldsymbol{T}$ [14]. Then, the right-hand side of Eq. (10) is the product $\boldsymbol{\omega} \times \boldsymbol{I}$ of the vector $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^\top$ and the unit matrix $\boldsymbol{I}$. Note that the identities $(\boldsymbol{a} \times \boldsymbol{I})\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$ and $(\boldsymbol{a} \times \boldsymbol{I})\boldsymbol{T} = \boldsymbol{a} \times \boldsymbol{T}$ hold. Multiplying Eq. (10) by $\boldsymbol{R}$ from right on both sides, we obtain

$$\Delta\boldsymbol{R} = \boldsymbol{\omega} \times \boldsymbol{R}. \tag{11}$$

If we divide this by the time laps Δt and take the limit $\Delta t \to 0$, we obtain the instantaneous change of rotation $d\boldsymbol{R}/dt$, and the limit of $\boldsymbol{\omega}$ is identified with the *angular velocity*. This representation itself is well known for describing rotational changes, e.g., in tracking objects, but is rarely used in numerical optimization, for which many researchers are accustomed to use parameterization in terms of the Euler angles, axial rotations, or a quaternion. However, such parameterization complicates the derivative expressions. Using Eq. (11), we can write down the first variation (or derivative) δJ of Eq. (9) in a simple form, as we now show:

[1]Strictly speaking, this linear space is a Lie algebra if a product called the *commutator* is introduced, but in the following the commutator does not play any role.

$$\begin{aligned}\delta J &= s\sum_{\alpha=1}^N(-\delta\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &\quad+\frac{1}{2}\sum_{\alpha=1}^N(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \delta\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &= s\sum_{\alpha=1}^N(-\boldsymbol{\omega}\times\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\ &\quad+\frac{1}{2}\sum_{\alpha=1}^N(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \delta\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})).\end{aligned} \tag{12}$$

Using the scalar triplet product $|\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}| = (\boldsymbol{a}, \boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{b}, \boldsymbol{c} \times \boldsymbol{a}) = (\boldsymbol{c}, \boldsymbol{a} \times \boldsymbol{b})$, we can write the first term as

$$\begin{aligned}&-s\sum_{\alpha=1}^N|\boldsymbol{\omega}, \boldsymbol{R}, \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})| \\ &= -s\sum_{\alpha=1}^N(\boldsymbol{\omega}, \boldsymbol{R}\boldsymbol{r}_\alpha \times \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})).\end{aligned} \tag{13}$$

For the variation $\delta\boldsymbol{W}_\alpha$ in the second term in Eq. (12), consider the defining equation of $\boldsymbol{W}_\alpha$ in Eq. (8):

$$(s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + V_0[\boldsymbol{r}'_\alpha])\boldsymbol{W}_\alpha = \boldsymbol{I}. \tag{14}$$

The first variation of the left-hand side is

$$\begin{aligned}&(s^2\delta\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\delta\boldsymbol{R}^\top)\boldsymbol{W}_\alpha \\ &\quad+(s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + V_0[\boldsymbol{r}'_\alpha])\delta\boldsymbol{W}_\alpha \\ &= (s^2\boldsymbol{\omega}\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha](\boldsymbol{\omega}\times\boldsymbol{R})^\top)\boldsymbol{W}_\alpha \\ &\quad+(s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + V_0[\boldsymbol{r}'_\alpha])\delta\boldsymbol{W}_\alpha \\ &= (s^2\boldsymbol{\omega}\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\times\boldsymbol{\omega})\boldsymbol{W}_\alpha \\ &\quad+(s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + V_0[\boldsymbol{r}'_\alpha])\delta\boldsymbol{W}_\alpha,\end{aligned} \tag{15}$$

where we define the product $\boldsymbol{T}\times\boldsymbol{a}$ of a matrix $\boldsymbol{T}$ and a vector $\boldsymbol{a}$ by $\boldsymbol{T}(\boldsymbol{a}\times\boldsymbol{I})^\top$. Since $\boldsymbol{I}$ on the right-hand side of Eq. (14) is a constant matrix, Eq. (15) should vanish, so we obtain

$$\begin{aligned}\delta\boldsymbol{W}_\alpha &= -s^2(s^2\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + V_0[\boldsymbol{r}'_\alpha])^{-1}(\boldsymbol{\omega} \\ &\quad\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top + \boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\times\boldsymbol{\omega})\boldsymbol{W}_\alpha \\ &= -s^2\boldsymbol{W}_\alpha(\boldsymbol{\omega}\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top \\ &\quad+\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\times\boldsymbol{\omega})\boldsymbol{W}_\alpha.\end{aligned} \tag{16}$$

Substituting this into Eq. (12), we can write the second term on the right-hand side of Eq. (12) as follows:

$$
\begin{aligned}
&-\frac{s^2}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \boldsymbol{W}_\alpha(\boldsymbol{\omega}\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top \\
&+\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\times\boldsymbol{\omega})\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - \boldsymbol{R}\boldsymbol{r}_\alpha)) \\
=&-\frac{s^2}{2}\sum_{\alpha=1}^{N}(\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \\
&\boldsymbol{\omega}\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\
&-\frac{s^2}{2}\sum_{\alpha=1}^{N}(\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top(\boldsymbol{\omega} \\
&\times\boldsymbol{I})^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\
=&-\frac{s^2}{2}\sum_{\alpha=1}^{N}|\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \boldsymbol{\omega}, \\
&\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})| \\
&+\frac{s^2}{2}\sum_{\alpha=1}^{N}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}, \boldsymbol{W}_\alpha\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top(\boldsymbol{\omega} \\
&\times\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}))) \\
=&-\frac{s^2}{2}\sum_{\alpha=1}^{N}|\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \boldsymbol{\omega}, \\
&\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})| \\
&+\frac{s^2}{2}\sum_{\alpha=1}^{N}(\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \boldsymbol{\omega} \\
&\times\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\
=&-\frac{s^2}{2}\sum_{\alpha=1}^{N}|\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \boldsymbol{\omega}, \\
&\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})| \\
&+\frac{s^2}{2}\sum_{\alpha=1}^{N}|\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \boldsymbol{\omega}, \\
&\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})| \\
=&\frac{s^2}{2}\sum_{\alpha=1}^{N}|\boldsymbol{\omega}, \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \\
&\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})| \\
&+\frac{s^2}{2}\sum_{\alpha=1}^{N}|\boldsymbol{\omega}, \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \\
&\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})| \\
=&\, s^2\sum_{\alpha=1}^{N}(\boldsymbol{\omega}, (\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\
&\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})). \qquad (17)
\end{aligned}
$$

Combining this with Eq. (13), we see that δJ in Eq. (12) has the form

$$
\begin{aligned}
\delta J =&\, s\sum_{\alpha=1}^{N}(\boldsymbol{\omega}, (\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}))\times\boldsymbol{R}\boldsymbol{r}_\alpha) \\
&+s^2\sum_{\alpha=1}^{N}(\boldsymbol{\omega}, (\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\
&\times\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\
=&\, s\sum_{\alpha=1}^{N}(\boldsymbol{\omega}, (\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})) \\
&\times(\boldsymbol{R}\boldsymbol{r}_\alpha + s\boldsymbol{R}V_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}))) \\
=&\,(\boldsymbol{\omega}, s\sum_{\alpha=1}^{N}(\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}))\times\boldsymbol{R}(\boldsymbol{r}_\alpha \\
&+sV_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}))). \qquad (18)
\end{aligned}
$$

This means that the first derivative of J with respect to the rotation is given by

$$
\begin{aligned}
\nabla_{\boldsymbol{\omega}}J =&\, s\sum_{\alpha=1}^{N}(\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}))\times\boldsymbol{R}(\boldsymbol{r}_\alpha \\
&+sV_0[\boldsymbol{r}_\alpha]\boldsymbol{R}^\top\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})). \qquad (19)
\end{aligned}
$$

Differentiating Eq. (9) with respect to $\boldsymbol{t}$ and s, we obtain

$$
\nabla_{\boldsymbol{t}}J = -\sum_{\alpha=1}^{N}\boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}), \qquad (20)
$$

$$
\frac{\partial J}{\partial s} = -\sum_{\alpha=1}^{N}(\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})). \qquad (21)
$$

4. Hessian Computation

If Eq. (12) is differentiated with respect to $\boldsymbol{R}$ along with the Gauss-Newton approximation, i.e., ignoring terms containing $\boldsymbol{r}'_\alpha - \boldsymbol{R}\boldsymbol{r}_\alpha$, the second variation of J is obtained as follows:

$$
\begin{aligned}
\delta^2 J =&\sum_{\alpha=1}^{N}(-s\delta\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha(-s\delta\boldsymbol{R}\boldsymbol{r}_\alpha)) \\
=&\, s^2\sum_{\alpha=1}^{N}(\boldsymbol{\omega}\times\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{\omega}\times\boldsymbol{R}\boldsymbol{r}_\alpha)) \\
=&\, s^2\sum_{\alpha=1}^{N}-(\boldsymbol{\omega}\times\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha((\boldsymbol{R}\boldsymbol{r}_\alpha)\times\boldsymbol{\omega})) \\
=&\, s^2\sum_{\alpha=1}^{N}-(\boldsymbol{\omega}\times\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha((\boldsymbol{R}\boldsymbol{r}_\alpha)\times\boldsymbol{I})\boldsymbol{\omega}) \\
=&\, s^2\sum_{\alpha=1}^{N}(\boldsymbol{\omega}\times\boldsymbol{R}\boldsymbol{r}_\alpha, (\boldsymbol{W}_\alpha\times(\boldsymbol{R}\boldsymbol{r}_\alpha))\boldsymbol{\omega}) \\
=&\, s^2\sum_{\alpha=1}^{N}|\boldsymbol{\omega}, \boldsymbol{R}\boldsymbol{r}_\alpha, (\boldsymbol{W}_\alpha\times(\boldsymbol{R}\boldsymbol{r}_\alpha))\boldsymbol{\omega}| \\
=&\,(\boldsymbol{\omega}, s^2\sum_{\alpha=1}^{N}((\boldsymbol{R}\boldsymbol{r}_\alpha)\times\boldsymbol{W}_\alpha\times(\boldsymbol{R}\boldsymbol{r}_\alpha))\boldsymbol{\omega}). \qquad (22)
\end{aligned}
$$

Thus, the second derivative of J for the rotation components has the form

$$\nabla_{\boldsymbol{\omega}}^2 J = s^2 \sum_{\alpha=1}^{N} (\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{W}_\alpha \times (\boldsymbol{R}\boldsymbol{r}_\alpha). \tag{23}$$

From Eqs. (20) and (21), the second derivatives of J with respect to $\boldsymbol{t}$ and s are given by

$$\nabla_{\boldsymbol{t}}^2 J = \sum_{\alpha=1}^{N} \boldsymbol{W}_\alpha, \quad \frac{\partial^2 J}{\partial s^2} = \sum_{\alpha=1}^{N} (\boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{W}_\alpha \boldsymbol{R}\boldsymbol{r}_\alpha). \tag{24}$$

Equation (20) implies that for t_i we have

$$\begin{aligned}\frac{\partial J}{\partial t_i} &= -\sum_{\alpha=1}^{N}\sum_{j=1}^{3} W_{\alpha ij}(\boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t})_i \\ &= -\sum_{\alpha=1}^{N} (\boldsymbol{w}_{\alpha i}, \boldsymbol{r}'_\alpha - s\boldsymbol{R}\boldsymbol{r}_\alpha - \boldsymbol{t}),\end{aligned} \tag{25}$$

where $W_{\alpha ij}$ is the (ij) element of the matrix $\boldsymbol{W}_\alpha$, and $\boldsymbol{w}_{\alpha i}$ is the ith column of $\boldsymbol{W}_\alpha$ (note that $\boldsymbol{W}_\alpha$ is a symmetric matrix). If the Gauss-Newton approximation is used, the variation of Eq. (25) for rotation is

$$\begin{aligned}\delta\frac{\partial J}{\partial t_i} &= s\sum_{\alpha=1}^{N} (\boldsymbol{w}_{\alpha i}, \boldsymbol{\omega} \times (\boldsymbol{R}\boldsymbol{r}_\alpha)) = s\sum_{\alpha=1}^{N} |\boldsymbol{w}_{\alpha i}, \boldsymbol{\omega}, \boldsymbol{R}\boldsymbol{r}_\alpha| \\ &= s\sum_{\alpha=1}^{N} |\boldsymbol{\omega}, \boldsymbol{R}\boldsymbol{r}_\alpha, \boldsymbol{w}_{\alpha i}| = s\sum_{\alpha=1}^{N} (\boldsymbol{\omega}, (\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{w}_{\alpha i}).\end{aligned} \tag{26}$$

This implies

$$\nabla_{\boldsymbol{\omega}} \frac{\partial J}{\partial t_i} = s\sum_{\alpha=1}^{N} (\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{w}_{\alpha i}, \tag{27}$$

which is rewritten as

$$\begin{aligned}\nabla_{\boldsymbol{\omega}\boldsymbol{t}} J &= \Big(\nabla_{\boldsymbol{\omega}}\frac{\partial J}{\partial t_1}, \nabla_{\boldsymbol{\omega}}\frac{\partial J}{\partial t_2}, \nabla_{\boldsymbol{\omega}}\frac{\partial J}{\partial t_3}\Big) \\ &= s\sum_{\alpha=1}^{N} \Big((\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{w}_{\alpha 1}, (\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{w}_{\alpha 2}, (\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{w}_{\alpha 3}\Big) \\ &= s\sum_{\alpha=1}^{N} (\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{W}_\alpha.\end{aligned} \tag{28}$$

Differentiating Eq. (19) with respect to s and using the Gauss-Newton approximation, we obtain

$$\begin{aligned}\frac{\partial \nabla_{\boldsymbol{\omega}} J}{\partial s} &= \nabla_{\boldsymbol{\omega}} \frac{\partial J}{\partial s} = -s\sum_{\alpha=1}^{N} (\boldsymbol{W}_\alpha \boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{R}\boldsymbol{r}_\alpha \\ &= s\sum_{\alpha=1}^{N} (\boldsymbol{R}\boldsymbol{r}_\alpha) \times \boldsymbol{W}_\alpha \boldsymbol{R}\boldsymbol{r}_\alpha.\end{aligned} \tag{29}$$

Finally, we differentiate Eq. (20) with respect to s to obtain

$$\frac{\partial \nabla_{\boldsymbol{t}} J}{\partial s} = \nabla_{\boldsymbol{t}} \frac{\partial J}{\partial s} = s\sum_{\alpha=1}^{N} \boldsymbol{W}_\alpha \boldsymbol{R}\boldsymbol{r}_\alpha. \tag{30}$$

Thus, the Hessian of J with respect to $\boldsymbol{R}$, $\boldsymbol{t}$, and s is given in the following form:

$$\boldsymbol{H} = \begin{pmatrix} \nabla_{\boldsymbol{\omega}}^2 J & \nabla_{\boldsymbol{\omega}\boldsymbol{t}} J & \nabla_{\boldsymbol{\omega}} \partial J/\partial s \\ (\nabla_{\boldsymbol{\omega}\boldsymbol{t}} J)^\top & \nabla_{\boldsymbol{t}}^2 J & \nabla_{\boldsymbol{t}} \partial J/\partial s \\ (\nabla_{\boldsymbol{\omega}} \partial J/\partial s)^\top & (\nabla_{\boldsymbol{t}} \partial J/\partial s)^\top & \partial^2 J/\partial s^2 \end{pmatrix}. \tag{31}$$

5. Levenberg-Marquardt Iterations

The Levenberg-Marquardt procedure for minimizing Eq. (9) is written as follows [23]:

1. Provide the initial guesses of $\boldsymbol{R}$, $\boldsymbol{t}$, and s, and compute the corresponding value of J in Eq. (9). Let $C = 0.0001$.

2. Compute the gradients $\nabla_{\boldsymbol{\omega}} J$, $\nabla_{\boldsymbol{t}} J$, and $\partial J/\partial s$ in Eqs. (19), (20), and (21) and the Hessian $\nabla_{\boldsymbol{\omega}}^2 J$ in Eq. (31).

3. Compute $\boldsymbol{\omega}$, $\Delta\boldsymbol{t}$, and Δs by solving the simultaneous linear equations

$$\Big(\boldsymbol{H} + C\mathcal{D}[\boldsymbol{H}]\Big)\begin{pmatrix}\boldsymbol{\omega} \\ \Delta\boldsymbol{t} \\ \Delta s\end{pmatrix} = -\begin{pmatrix}\nabla_{\boldsymbol{\omega}} J \\ \nabla_{\boldsymbol{t}} J \\ \partial J/\partial s\end{pmatrix}, \tag{32}$$

   where $\mathcal{D}[\,\cdot\,]$ denotes the diagonal matrix obtained by selecting the diagonal elements (or we may simply let $\mathcal{D}[\,\cdot\,] = \boldsymbol{I}$ [23]).

4. Compute the following rotation matrix $\tilde{\boldsymbol{R}}$, the translation vector $\tilde{\boldsymbol{t}}$, and the scale change $\tilde{s}$:

$$\tilde{\boldsymbol{R}} = \mathcal{R}[\boldsymbol{\omega}]\boldsymbol{R}, \quad \tilde{\boldsymbol{t}} = \boldsymbol{t} + \Delta\boldsymbol{t}, \quad \tilde{s} = s + \Delta s. \tag{33}$$

   Here, $\mathcal{R}[\boldsymbol{\omega}]$ is the rotation around an axis $\mathcal{N}[\boldsymbol{\omega}]$ ($\mathcal{N}[\,\cdot\,]$ denotes normalization to unit norm) by angle $\|\boldsymbol{\omega}\|$, given by the Rodorigues formula[2]

$$\mathcal{R}[\boldsymbol{\omega}] = \boldsymbol{I}\cos\Omega + \boldsymbol{l} \times \boldsymbol{I}\sin\Omega + \boldsymbol{l}\boldsymbol{l}^\top(1 - \cos\Omega), \tag{34}$$

   where $\boldsymbol{l} = \mathcal{N}[\boldsymbol{\omega}]$ and $\Omega = \|\boldsymbol{\omega}\|$.

5. Let $\tilde{J}$ be the value value of Eq. (9) for $\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{t}}$, $\tilde{s}$.

6. If $\tilde{J} \approx J$ or $\tilde{J} < J$, go to the next step. Else, let $C \leftarrow 10C$ and go back to Step 3.

7. Let $\boldsymbol{R} \leftarrow \tilde{\boldsymbol{R}}$, $\boldsymbol{t} \leftarrow \tilde{\boldsymbol{t}}$, and $s \leftarrow \tilde{s}$. If $J \approx \tilde{J}$, return $\boldsymbol{R}$, $\boldsymbol{t}$, and s and stop. Else, let $J \leftarrow \tilde{J}$ and $C \leftarrow C/10$, and go back to Step 2.

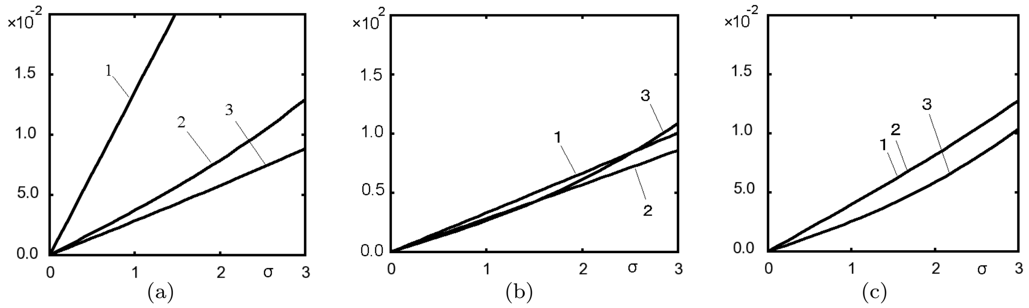

Figure 3: The RMS error vs. the standard deviation σ of the noise added to the stereo images: (a) rotation, (b) translation, (c) scale change. The numbers 1, 2, and 3 corresponds to Methods 1, 2, 3, respectively.

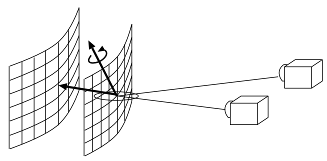

Figure 1: A grid surface rotates, translates, and changes the scale. The 3-D position of the grid points are measured by stereo vision. The ellipsoid illustrates the measurement uncertainty.

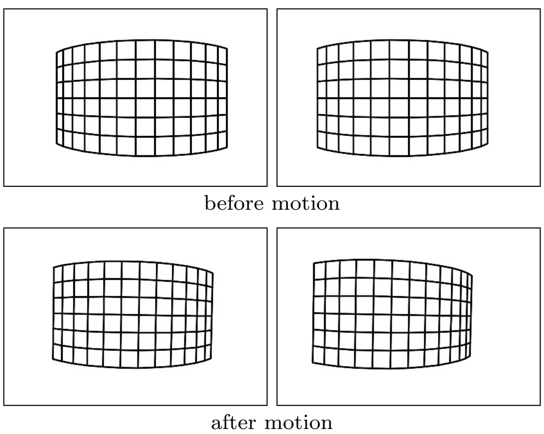

Figure 2: Simulated stereo image pairs before and after the similarity motion.

6. Simulated Stereo Vision

We rotate a curved grid surface around the world origin O, translate it, and change its scale, as depicted in Fig. 1. We measure the 3-D positions of the grid points by stereo vision before and after the similarity motion. The simulated stereo images are shown in Fig. 2. We set the image size 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 [17]. Then, the normalized covariances $V_0[\hat{\boldsymbol{r}}_\alpha]$ and $V_0[\hat{\boldsymbol{r}}'_\alpha]$ of the measurement uncertainty were evaluated as described in [18, 21] (see Appendix A), and the similarity was estimated by the Levenberg-Marquardt method, where we judged the $\approx$ in Steps 6 and 7 if the difference is less than 10^{-10} in absolute value.

Then, we measured the accuracy of the resulting 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 $\delta s = \hat{s} - \bar{s}$. We repeated this 1000 times with σ fixed, each time using different image noise, and computed the RMS errors

$$E_{\boldsymbol{R}} = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}(\delta\Omega^{(a)})^2}, \quad E_{\boldsymbol{t}} = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}\|\delta\boldsymbol{t}^{(a)}\|^2},$$
$$E_s = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}(\delta s^{(a)})^2}, \tag{35}$$

where the superscript (a) denotes the value of the ath trial. We compared the following three methods.

Method 1 Let $\boldsymbol{r}_c$ and $\boldsymbol{r}'_c$ be the centroids of $\{\boldsymbol{r}_\alpha\}$ and $\{\boldsymbol{r}'_\alpha\}$, respectively, and let $\tilde{\boldsymbol{r}}_\alpha = \boldsymbol{r}_\alpha - \boldsymbol{r}_c$ and $\tilde{\boldsymbol{r}}'_\alpha = \boldsymbol{r}'_\alpha - \boldsymbol{r}'_c$ be the deviations from the centroids. We estimate the scale change by $s = \sqrt{\sum_{\alpha=1}^N \|\tilde{\boldsymbol{r}}'_\alpha\|^2 / \sum_{\alpha=1}^N \|\tilde{\boldsymbol{r}}_\alpha\|^2}$ and optimally computed $\boldsymbol{R}$ from $\{\tilde{\boldsymbol{r}}_\alpha\}$ and $\{\tilde{\boldsymbol{r}}'_\alpha/s\}$, assuming homogeneous and isotropic noise (see Appendix B). Finally, we let $\boldsymbol{t} = \boldsymbol{r}'_c - s\boldsymbol{R}\boldsymbol{r}_c$.

Method 2 In Method 1, we let the normalized covariance matrices $\{\tilde{\boldsymbol{r}}_\alpha\}$ and $\{\tilde{\boldsymbol{r}}'_\alpha/s\}$ to be $V_0[\boldsymbol{r}_\alpha]$ and $V_0[\boldsymbol{r}'_\alpha]/s^2$, respectively. Using these, we optimally estimate the rotation $\boldsymbol{R}$ by the method of Niitsuma and Kanatani [21].

Method 3 The rotation $\boldsymbol{R}$, the translation $\boldsymbol{t}$, and the scale change s are simultaneously optimized by the Levenberg-Marquardt method, for which the initial guess is computed by Method 1.

[2] This is written as $\exp(\boldsymbol{\omega}\times\boldsymbol{I})$ in Lie group theory and called the *exponential map* of $\boldsymbol{\omega}$.

Table 1: 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

Table 2: The translation $\boldsymbol{t} = (t_1, t_2, t_3)^\top$ (in meter), 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 reprojection error J estimated from the data in Table 1 by the three methods.

	Method 1	Method 2	Method 3
t_1	−199.86035620	−237.32542737	−273.58000610
t_2	42.52530293	85.27928886	99.29808570
t_3	143.65787065	158.06078612	141.67312764
s	1.00000370	1.00000370	1.00000837
l_1	−0.04950650	−0.03494625	−0.01117288
l_2	0.93285277	0.85967794	0.82289933
l_3	−0.35684003	−0.50963968	−0.56807733
Ω	0.00224281	0.00267166	0.00288150
J	9.2429×10^{-6}	8.7283×10^{-6}	6.4095×10^{-6}

Figure 3 shows the plots of the RMS errors $E_{\boldsymbol{R}}$, $E_{\boldsymbol{t}}$, and E_s for the noise level σ. We can see that the conventional method based on the homogeneous and isotropic noise assumption has very low accuracy. We see that Method 2 exhibits a considerable improvement and that Method 3 further improves the accuracy. The scale change estimation is also slightly improved by our simultaneous optimization, while the translation accuracy is more or less the same. Thus, we can conclude that it is the rotation that is most affected by the noise assumption and the optimization technique. Although the optimal scheme in [21] produces a highly accurate rotation estimate, our simultaneous optimization results in an even better solution.

7. Real Data Example

Turkey is a country with frequent earthquakes, and researchers monitor the land deformation using GPS data. Table 1 shows the X, Y, and Z coordinates (in meter) 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 a statistical method. 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 2 lists the translation $\boldsymbol{t}$, the scale change s, the rotation (axis $\boldsymbol{l}$ and angle Ω), and the reprojection error J computed by the three described in the preceding section. We observe slight differences among the three methods. A precise estimation for such a small deformation requires an accurate optimization technique such as our simultaneous ML scheme. In the past, however, various types of weighted least squares scheme have been widely used in geodetics [1, 7, 8, 20].

8. Concluding Remarks

Unlike 2-D image data, 3-D data are acquired by 3-D sensing such as stereo vision and laser range finding. Hence, the measurement uncertainty in 3-D is inhomogeneous and anisotropic, depending on the type, position, and orientation of the sensor. In this paper, we have presented a numerical scheme for maximum likelihood (ML) estimation of the similarity (rotation, translation, scale change) between two sets of 3-D measurement data. Using the Lie algebra representation of the rotational change, we derived the Levenberg-Marquardt procedure for simultaneously optimizing the rotation, the translation, and the scale change.

We tested the performance of our method using simulated stereo data and real GPS geodetic sensing data. We conclude that it is the rotation that is most affected by the noise assumption and the optimization technique and that the conventional method assuming homogeneous and isotropic noise produces very inaccurate results. We demonstrated that our simultaneous optimization produces an even better rotation estimate than using an optimal rotation estimation scheme.

Acknowledgments. The authors thank Orhan Akyilmaz of Istanbul Institute of Technology, Turkey for providing the GPS data and doing helpful discussions. They also thank 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] M. Acar, M. T. Özlüdemir, O. Akyilmaz, R. N. Celik and T. Ayan, Deformation analysis with total least Squares, Nat. Hazards Earth Syst. Sci, **6**-4 (2006-6), 663–669.

[2] K. S. Arun, T. S. Huang and S. D. Blostein, Least squares fitting of two 3-D point sets, *IEEE Trans. Patt. Anal. Mach. Intell.*, **9**-5 (1987-5), 698–700.

[3] N. Ayache and O. D. Faugeras, Building, registrating, and fusing noisy visual data, *Int. J. Comput. Vision*, **7**-6 (1988-12), 45–65.

[4] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**-11 (2000), 1294–1303.

[5] L. Dorst, First order error propagation of the Procrustes method for 3D attitude estimation, *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**-2 (2005-2), 221–229.

[6] T. Drummond and R. Cipolla, Application of Lie algebra to visual servoing, *Int. J. Comput. Vision*, **37**-1 (2000-6), 21–41.

[7] Y. A. Felus and R. C. Burch, On symmetrical three-dimensional datum conversion, *GPS Solutions* **13**-1 (2009-1), 65–74.

[8] E. W. Grafarend and J. L. Awange, Nonlinear analysis of the three-dimensional datum transformation [conformal group $C_7(3)$], *J. Geodesy*, **77**-1/2 (2003-5), 66–76.

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

[10] B. K. P. Horn, H. M. Hildren and S. Negahdaripour, Closed-form solution of absolute orientation, using orthonormal matrices, *Int. J. Opt. Soc. Am.* **A-5**-7 (1988-7), 1127–1135.

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

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

[13] K. Kanatani, Analysis of 3-D rotation fitting, *IEEE Trans. Patt. Anal. Mach. Intell.*, **16**-5 (1994-5), 543–449.

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

[15] K. Kanatani, Statistical optimization for geometric fitting: Theoretical accuracy analysis and high order error analysis, *Int. J. Comput. Vision*, **80**-2 (2008-11), 167–188.

[16] K. Kanatani and Y. Sugaya, Implementation and evaluation of bundle adjustment for 3-D Reconstruction, Proc. 17th Symp Sensing via Imaging Information, June 2010, Yokohama, Japan.

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

[18] Y. Kanazawa and K. Kanatani, Reliability of 3-D reconstruction by stereo vision, *IEICE Trans. Inf. & Syst.*, **E78-D**-10 (1995-10), 1301–1306.

[19] L. Matthies and S. A. Shafer, Error modeling in stereo navigation, *IEEE J. Robot. Automation*, **3**-3 (1987-6), 239–248.

[20] F. Neitzel, Generalization of total least-squares on example of unweighted and weighted 2D similarity transformations, *J. Geodesy*, **84**-12 (2010-12), 751–762.

[21] H. Niitsuma and K. Kanatani, Optimal computation of 3-D rotation under inhomogeneous anisotropic noise, Proc. 12th IAPR Conf. Machine Vision Applications, June 2011, Nara, Japan.

[22] N. Ohta and K. Kanatani, Optimal estimation of three-dimensional rotation and reliability evaluation, *IEICE Trans. Inf. & Syst.*, *E81-D*-11 (1998-11), 1247–1252.

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

[24] M. Sakamoto, Y. Sugaya, and K. Kanatani, Homography optimization for consistent circular panorama generation, *Proc. 2006 IEEE Pacific Rim-Symp. Image Video Technology (PSIVT'06)*, December 2006, Hsinchu, Taiwan, pp. 1195–1205.

[25] Y. Sugaya and K. Kanatani, High accuracy computation of rank-constrained fundamental matrix, *Proc. 18th British Machine Vision Conf.*, September 2007, Coventry, U.K., Vol. 1, pp. 282–291.

[26] S. Umeyama, Least-squares estimation of transformation parameters between two point sets, *IEEE Trans. Patt. Anal. Mach. Intell.*, **13**-4 (1991-4), 379–380.

Appendix

A. Covariance matrix evaluation

The covariance matrix of a 3-D position reconstructed by stereo vision can be evaluated as follows. We consider a reference camera placed at the world 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 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 O by $\boldsymbol{R}$ and translated by $\boldsymbol{t}$ from the reference position. 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:

$$\boldsymbol{x} \simeq \boldsymbol{P}\boldsymbol{X}, \qquad \boldsymbol{x} \equiv \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{X} \equiv \begin{pmatrix} \boldsymbol{r} \\ 1 \end{pmatrix}. \tag{36}$$

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

$$\boldsymbol{P} = \begin{pmatrix} f/f_0 & 0 & 0 \\ 0 & f/f_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}^\top & -\boldsymbol{R}^\top \boldsymbol{t} \end{pmatrix}, \tag{37}$$

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 $\boldsymbol{r}$ 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 [14], 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 [17]. From the corrected positions $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$, the corresponding 3-D position $\hat{\boldsymbol{r}}$ is uniquely determined.

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, 18]:

$$\begin{aligned} V_0[\hat{\boldsymbol{x}}] &= \frac{1}{f_0^2}\Big(\boldsymbol{P_k} - \frac{(\boldsymbol{P_k F}\hat{\boldsymbol{x}}')(\boldsymbol{P_k F}\hat{\boldsymbol{x}}')^\top}{\|\boldsymbol{P_k F}\hat{\boldsymbol{x}}'\|^2 + \|\boldsymbol{P_k F}^\top \hat{\boldsymbol{x}}\|^2}\Big), \\ V_0[\hat{\boldsymbol{x}}'] &= \frac{1}{f_0^2}\Big(\boldsymbol{P_k} - \frac{(\boldsymbol{P_k F}^\top\hat{\boldsymbol{x}})(\boldsymbol{P_k F}^\top\hat{\boldsymbol{x}})^\top}{\|\boldsymbol{P_k F}\hat{\boldsymbol{x}}'\|^2 + \|\boldsymbol{P_k F}^\top \hat{\boldsymbol{x}}\|^2}\Big), \\ V_0[\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}'] &= \frac{1}{f_0^2}\Big(- \frac{(\boldsymbol{P_k F}\hat{\boldsymbol{x}}')(\boldsymbol{P_k F}^\top\hat{\boldsymbol{x}})^\top}{\|\boldsymbol{P_k F}\hat{\boldsymbol{x}}'\|^2 + \|\boldsymbol{P_k F}^\top \hat{\boldsymbol{x}}\|^2}\Big) \\ &= V_0[\hat{\boldsymbol{x}}', \hat{\boldsymbol{x}}]^\top. \end{aligned} \tag{38}$$

Here, 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 Eq. (36), 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

$$\hat{\boldsymbol{x}} \times \boldsymbol{P}\hat{\boldsymbol{X}} = \boldsymbol{0}, \qquad \hat{\boldsymbol{x}}' \times \boldsymbol{P}'\hat{\boldsymbol{X}} = \boldsymbol{0} \tag{39}$$

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

$$\begin{aligned} \Delta\hat{\boldsymbol{x}} \times \boldsymbol{P}\hat{\boldsymbol{X}} + \hat{\boldsymbol{x}} \times \boldsymbol{P}\Delta\hat{\boldsymbol{X}} &= \boldsymbol{0}, \\ \Delta\hat{\boldsymbol{x}}' \times \boldsymbol{P}'\hat{\boldsymbol{X}}' + \hat{\boldsymbol{x}}' \times \boldsymbol{P}'\Delta\hat{\boldsymbol{X}} &= \boldsymbol{0}. \end{aligned} \tag{40}$$

From these, we obtain the following relation (the details are given in [21]):

$$\boldsymbol{A}\Delta\hat{\boldsymbol{r}} = \boldsymbol{B}\begin{pmatrix} \Delta\hat{\boldsymbol{x}} \\ \Delta\hat{\boldsymbol{x}}' \end{pmatrix}, \tag{41}$$

$$\boldsymbol{A} \equiv \|\hat{\boldsymbol{x}}\|^2 \tilde{\boldsymbol{P}}^\top \boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}]}\tilde{\boldsymbol{P}} + \|\hat{\boldsymbol{x}}'\|^2 \tilde{\boldsymbol{P}}'^\top \boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}']}\tilde{\boldsymbol{P}}',$$

$$\begin{aligned} \boldsymbol{B} \equiv \Big(\tilde{\boldsymbol{P}}^\top \Big((\hat{\boldsymbol{x}}, \boldsymbol{P}\hat{\boldsymbol{X}})\boldsymbol{I} - (\boldsymbol{P}\hat{\boldsymbol{X}})\hat{\boldsymbol{x}}^\top \\ \tilde{\boldsymbol{P}}'^\top \Big((\hat{\boldsymbol{x}}', \boldsymbol{P}'\hat{\boldsymbol{X}})\boldsymbol{I} - (\boldsymbol{P}'\hat{\boldsymbol{X}})\hat{\boldsymbol{x}}'^\top\Big)\Big). \end{aligned} \tag{42}$$

$$\boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}]} \equiv \boldsymbol{I} - \mathcal{N}[\hat{\boldsymbol{x}}]\mathcal{N}[\hat{\boldsymbol{x}}]^\top, \quad \boldsymbol{P}_{\mathcal{N}[\hat{\boldsymbol{x}}']} \equiv \boldsymbol{I} - \mathcal{N}[\hat{\boldsymbol{x}}']\mathcal{N}[\hat{\boldsymbol{x}}']^\top. \tag{43}$$

Hence, we obtain

$$\Delta\hat{\boldsymbol{r}}\Delta\hat{\boldsymbol{r}}^\top = \boldsymbol{A}^{-1}\boldsymbol{B}\begin{pmatrix} \Delta\hat{\boldsymbol{x}}\Delta\hat{\boldsymbol{x}}^\top & \Delta\hat{\boldsymbol{x}}\Delta\hat{\boldsymbol{x}}^\top \\ \Delta\hat{\boldsymbol{x}}'\Delta\hat{\boldsymbol{x}}^\top & \Delta\hat{\boldsymbol{x}}'\Delta\hat{\boldsymbol{x}}'^\top \end{pmatrix}\boldsymbol{B}^\top(\boldsymbol{A}^{-1})^\top. \tag{44}$$

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:

$$V_0[\hat{\boldsymbol{r}}] = \boldsymbol{A}^{-1}\boldsymbol{B}\begin{pmatrix} V_0[\hat{\boldsymbol{x}}] & V_0[\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}'] \\ V_0[\hat{\boldsymbol{x}}', \hat{\boldsymbol{x}}] & V_0[\hat{\boldsymbol{x}}'] \end{pmatrix}\boldsymbol{B}^\top(\boldsymbol{A}^{-1})^\top. \tag{45}$$

B. Homogeneous isotropic noise solution

Various methods are known for optimally estimating the 3-D rotation for homogeneous and isotropic noise [2, 9, 10, 13, 26], 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:

$$\boldsymbol{N} = \sum_{\alpha=1}^{N} \boldsymbol{r}'_\alpha \boldsymbol{r}_\alpha^\top. \tag{46}$$

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

$$\boldsymbol{N} = \boldsymbol{U}\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)\boldsymbol{V}^\top, \tag{47}$$

   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:

$$\boldsymbol{R} = \boldsymbol{U}\mathrm{diag}(1, 1, \det(\boldsymbol{U}\boldsymbol{V}^\top))\boldsymbol{V}^\top. \tag{48}$$