Contents list available at ScienceDrect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

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

Kenichi Kanatani*, Hirotaka Niitsuma

Department of Computer Science, Okayama University, Okayama 700-8530, Japan

ARTICLE INFO

Article history:
Received 5 September 2011
Received in revised form 24 February 2012
Accepted 20 March 2012
Available online 29 March 2012

Keywords:
3-D Similarity estimation
Inhomogeneous anisotropic noise
Gauss-Newton method
Gauss-Helmert method
Stereo vision
Geodetic sensing

ABSTRACT

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 describe two well known methods for this: the Gauss-Newton and the Gauss-Helmert methods, which are often regarded as different techniques. We then point out that they have similar mathematical structures and combine them to define a hybrid, which we call the *modified Gauss-Helmert method*. Doing stereo vision simulation, we demonstrate that the proposed method is superior to either of the two methods 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. We also discuss some numerical issues about GPS data.

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 (Acar et al., 2006; Felus and Burch, 2009; Grafarend and Awange, 2003; Neitzel, 2010).

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. However, rotation estimation is not so straightforward in the presence of noise, and many rotation estimation algorithms that assume homogeneous and isotropic noise have been proposed in the past (Arun et al., 1987; Dorst, 2005; Horn, 1987; Horn et al., 1988; Kanatani, 1994; Umeyama, 1991). However, the assumption of homogeneous and isotropic noise is not realistic 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.

This observation motivated computer vision researchers to study optimal estimation under inhomogeneous and anisotropic noise in late 1990s (Kanatani, 1996; Matei and Meer, 1999). Ohta and Kanatani (1998) presented a 3-D

* Corresponding author. Tel.: +81 86 251 8173; fax +81 86 251 8173.
E-mail addresses: kanatani@suri.cs.okayama-u.ac.jp (K. kanatani), niitsuma@suri.cs.okayama-u.ac.jp (H. Niitsuma).

0167-9473/$–see front matter
doi:10.1016/j.csda.2012.03.014

rotation estimation scheme that takes the inhomogeneity and anisotropy of the noise in 3-D data it into account. They used a technique called *renormalization*, which iteratively removes statistical bias of reweighted least squares (Kanatani, 1996). Later, schemes for computing a maximum likelihood (ML) solution have been proposed for estimating the 3-D rotation (Niitsuma and Kanatani, 2011b), 3-D rigid motion (Matei and Meer, 1999). Niitsuma and Kanatani (2011b) extended it to 3-D similarity estimation, using the Levenberg-Marquardt (LM) method, the most widely used standard optimization technique in the field of computer vision (Triggs, et al., 1999; Fitzgibbon, 2003; Hartley and Zisserman, 2004). The LM is basically the Gauss-Newton method, to which the gradient descent principle is combined to ensure convergence (Press, et al., 1992).

In geodetic science, on the other hand, the Gauss-Helmert method is popular for similarity estimation (Mikhail and Ackermann, 1976; Neitzel, 2010); 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 applications (Förstner, 2000; Perwass et al., 2006; Gebken and Sommer, 2008). However, no comparative studies of the Gauss-Newton and Gauss-Helmert methods 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 general terms, the Gauss-Newton method first expresses the residual as a nonlinear function in the unknowns by eliminating the nonlinear constraint using Lagrange multipliers and then minimizes it by Newton iterations, introducing the Gauss-Newton approximation to the Hessian evaluation. The Gauss-Helmert method, on the other hand, first linearizes the nonlinear constraint around the current values of the unknowns and expresses the residual as a quadratic function in the increments of the variables. Then, the variables are updated by the increments that minimizes it, and this procedure is iterated.

In this paper, we start with the maximum likelihood framework of 3-D similarity estimation (Section 2) and formulate the Gauss-Newton method (Section 3). Then, we reformulate the Gauss-Helmert method in a form that makes the comparison easier (Section 4) and point out that the two have a very similar mathematical structures (Section 5). We then combine them to define a hybrid, which we call the "modified Gauss-Helmert method" (Section 6). Doing stereo vision simulation, we demonstrate that the proposed method is superior to either of the two methods in the convergence performance (Section 7). We also show an application to real GPS geodetic data and point out that the widely used homogeneous and isotropic noise model is insufficient with some discussions about numerical problems of GPS data (Section 8). Finally, we summarize our observations (Section 9).

2. Maximum likelihood estimation of 3-D similarity

Suppose we are given 3-D position measurements $\mathbf{r}_\alpha$ and $\mathbf{r}'_\alpha$, $\alpha = 1, ..., N$, before and after a similarity motion. We model the measurement uncertainty by independent Gaussian noise of mean $\mathbf{0}$ and covariance matrices $\sigma^2 V_0[\mathbf{r}_\alpha]$ and $\sigma^2 V_0[\mathbf{r}'_\alpha]$, where the scalar σ, which we call the *noise level*, describes the magnitude, and the matrices $V_0[\mathbf{r}_\alpha]$ and $V_0[\mathbf{r}'_\alpha]$, which we call the *normalized covariance matrices*, describe the directional characteristics of the noise. This separation is merely for computational convenience; there is no fixed rule as to how to define σ. This convention is introduced because first it is difficult to evaluate the absolute noise magnitude and second the optimization computation is not affected by multiplication of all the covariance matrices by a positive constant. Thus, we need not know the exact noise magnitude for optimization; relative values suffice.

We say that noise is *homogeneous* if its distribution is the same for all the data and *inhomogeneous* otherwise. We also say that noise is *isotropic* if the distribution is orientation independent, and *anisotropic* otherwise. If the noise is isotropic and homogeneous, we can let $V_0[\mathbf{r}_\alpha] = V_0[\mathbf{r}'_\alpha] = \mathbf{I}$ (the unit matrix) for all α. In this paper, we assume that $V_0[\mathbf{r}_\alpha]$ and $V_0[\mathbf{r}'_\alpha]$ are not necessarily diagonal and generally different from position to position. Thus, the noise is generally inhomogeneous and anisotropic, including homogeneous and isotropic noise as a special case.

Let $\bar{\mathbf{r}}_\alpha$ and $\bar{\mathbf{r}}'_\alpha$ be the true positions of $\mathbf{r}_\alpha$ and $\mathbf{r}'_\alpha$, respectively, that undergo a similarity of rotation $\mathbf{R}$, translation $\mathbf{t}$, 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),

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

where and throughout this paper $(\mathbf{a}, \mathbf{b})$ denotes the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$. The residual J is minimized with respect to $\bar{\mathbf{r}}_\alpha$ and $\bar{\mathbf{r}}'_\alpha$ subject to

$$\bar{\mathbf{r}}'_\alpha = \mathbf{S}\bar{\mathbf{r}}_\alpha + \mathbf{t}, \tag{2}$$

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

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

[1] Mathematically, $\mathbf{q}$ is called a "quaternion" when associated with its algebra, i.e., the rule of composition (Kanatani, 1990). However, the quaternion algebra does not play any role in this paper.

This matrix represents a rotation if $\mathbf{q}$ is normalized to unit norm (Kanatani, 1990). If $\mathbf{q}$ is not restricted to a unit vector, the square norm $\|\mathbf{q}\|^2$ represents the scale change s. This quaternion representation of similarity has been used for computer vision and robotics, and recently it is also used in geodetics (Akylimaz, 2011).

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

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

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

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

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

$$\bar{\mathbf{r}}_\alpha = \mathbf{r}_\alpha - V_0[\mathbf{r}_\alpha]\mathbf{S}^\top\boldsymbol{\lambda}_\alpha, \qquad \bar{\mathbf{r}}'_\alpha = \mathbf{r}'_\alpha + V_0[\mathbf{r}'_\alpha]\boldsymbol{\lambda}_\alpha. \tag{6}$$

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

$$\boldsymbol{\lambda}_\alpha = -\mathbf{W}_\alpha\mathbf{e}_\alpha, \tag{7}$$

where we define

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

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

$$J = \frac{1}{2}\sum_{\alpha=1}^{N}(\mathbf{e}_\alpha, \mathbf{W}_\alpha\mathbf{e}_\alpha). \tag{9}$$

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

$$\frac{\partial\mathbf{S}}{\partial q_i} = 2\mathbf{Q}_i, \tag{10}$$

where we define

$$\mathbf{Q}_0 = \begin{pmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix}, \quad \mathbf{Q}_1 = \begin{pmatrix} q_1 & q_2 & q_3 \\ q_2 & -q_1 & -q_0 \\ q_3 & q_0 & -q_1 \end{pmatrix}, \quad \mathbf{Q}_2 = \begin{pmatrix} -q_2 & q_1 & q_0 \\ q_1 & q_2 & q_3 \\ -q_0 & q_3 & -q_2 \end{pmatrix}, \quad \mathbf{Q}_3 = \begin{pmatrix} -q_3 & -q_0 & q_1 \\ q_0 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \end{pmatrix}. \tag{11}$$

Letting

$$\mathbf{V}_\alpha = \mathbf{S}V_0[\mathbf{r}_\alpha]\mathbf{S}^\top + V_0[\mathbf{r}'_\alpha], \tag{12}$$

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

$$2(\mathbf{Q}_iV_0[\mathbf{r}_\alpha]\mathbf{S}^\top + \mathbf{S}V_0[\mathbf{r}_\alpha]\mathbf{Q}_i^\top)\mathbf{W}_\alpha + \mathbf{V}_\alpha\frac{\partial\mathbf{W}_\alpha}{\partial q_i} = \mathbf{O}, \tag{13}$$

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

$$\frac{\partial\mathbf{W}_\alpha}{\partial q_i} = -4\mathbf{W}_\alpha\mathcal{S}[\mathbf{Q}_iV_0[\mathbf{r}_\alpha]\mathbf{S}^\top]\mathbf{W}_\alpha, \tag{14}$$

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

$$\frac{\partial J}{\partial q_i} = -2\sum_{\alpha=1}^{N}(\mathbf{Q}_i\mathbf{r}_\alpha, \mathbf{W}_\alpha\mathbf{e}_\alpha) - 2\sum_{\alpha=1}^{N}(\mathbf{e}_\alpha, \mathbf{W}_\alpha\mathbf{Q}_iV_0[\mathbf{r}_\alpha]\mathbf{S}^\top\mathbf{W}_\alpha\mathbf{e}_\alpha). \tag{15}$$

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 $\mathbf{U}_\alpha$

$$\mathbf{U}_\alpha = 2\Big(\ \mathbf{Q}_0\mathbf{r}_\alpha\ \ \mathbf{Q}_1\mathbf{r}_\alpha\ \ \mathbf{Q}_2\mathbf{r}_\alpha\ \ \mathbf{Q}_3\mathbf{r}_\alpha\ \Big), \tag{16}$$

we can write (15) in the form

$$\nabla_{\mathbf{q}} J = -\sum_{\alpha=1}^{N} \mathbf{U}_\alpha^\top \mathbf{W}_\alpha \mathbf{e}_\alpha - 2\left(\sum_{\alpha=1}^{N} (\mathbf{e}_\alpha, \mathbf{W}_\alpha \mathbf{Q}_i V_0[\mathbf{r}_\alpha] \mathbf{S}^\top \mathbf{W}_\alpha \mathbf{e}_\alpha)\right)_{i=0,1,2,3}, \tag{17}$$

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 $\mathbf{t}$ yields

$$\nabla_{\mathbf{t}} J = -\sum_{\alpha=1}^{N} \mathbf{W}_\alpha \mathbf{e}_\alpha. \tag{18}$$

Differentiating (15) with respect to q_j and doing the Gauss-Newton approximation, we obtain the second derivative

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

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

$$\nabla_{\mathbf{t}}^2 J = \sum_{\alpha=1}^{N} \mathbf{W}_\alpha. \tag{20}$$

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

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

By "Gauss-Newton approximation", we mean omission of the terms of $O(\mathbf{e}_\alpha)$. Note that the residual J in (9) is quadratic in $\mathbf{e}_\alpha$ with a coefficient matrix of $O(1)$ and that $\mathbf{e}_\alpha$ would be $\mathbf{0}$ in the absence of noise. In general terms, the Gauss-Newton approximation means retaining only the leading term of the expansion of the Hessian in the quantities that would be zero in the absence of noise (Triggs, et al., 1999). If the residual had the form of the sum of squares $(1/2)\sum_{\alpha=1}^{N} \|\mathbf{e}_\alpha\|^2$, in particular, the Hessian with the Gauss-Newton approximation would be expressed only in terms of the Jacobian matrix of $\mathbf{e}_\alpha$ with respect to the parameters (Fitzgibbon, 2003).

Using the matrix $\mathbf{U}_\alpha$ in (16), we can now express the Gauss-Newton approximated Hessian $\mathbf{H}$ of the residual J in the form

$$\mathbf{H} = \begin{pmatrix} \displaystyle\sum_{\alpha=1}^{N} \mathbf{U}_\alpha^\top \mathbf{W}_\alpha \mathbf{U}_\alpha & \displaystyle\sum_{\alpha=1}^{N} \mathbf{U}_\alpha^\top \mathbf{W}_\alpha \\ \displaystyle\sum_{\alpha=1}^{N} \mathbf{W}_\alpha \mathbf{U}_\alpha & \displaystyle\sum_{\alpha=1}^{N} \mathbf{W}_\alpha \end{pmatrix}. \tag{22}$$

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

(i) Provide an initial guess of $\mathbf{q}$ and $\mathbf{t}$, and let $J_0 = \infty$ (a sufficiently large number).
(ii) Compute the scaled rotation $\mathbf{S}$ in (3) for $\mathbf{q}$.
(iii) Compute the vectors $\mathbf{e}_\alpha$ and the matrices $\mathbf{W}_\alpha$ in (8), and evaluate the residual J in (9).
(iv) If $J \approx J_0$, return $\mathbf{q}$ and $\mathbf{t}$ and stop. Else, let $J_0 \leftarrow J$.
(v) Compute the matrices $\mathbf{Q}_i$ in (11) and the matrices $\mathbf{U}_\alpha$ in (16).
(vi) Solve the following 7-D linear equation:

$$\begin{pmatrix} \displaystyle\sum_{\alpha=1}^{N} \mathbf{U}_\alpha^\top \mathbf{W}_\alpha \mathbf{U}_\alpha & \displaystyle\sum_{\alpha=1}^{N} \mathbf{U}_\alpha^\top \mathbf{W}_\alpha \\ \displaystyle\sum_{\alpha=1}^{N} \mathbf{W}_\alpha \mathbf{U}_\alpha & \displaystyle\sum_{\alpha=1}^{N} \mathbf{W}_\alpha \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{t} \end{pmatrix} = \begin{pmatrix} \displaystyle\sum_{\alpha=1}^{N} \mathbf{U}_\alpha^\top \mathbf{W}_\alpha \mathbf{e}_\alpha \\ \displaystyle\sum_{\alpha=1}^{N} \mathbf{W}_\alpha \mathbf{e}_\alpha \end{pmatrix} + 2\begin{pmatrix} \Big(\displaystyle\sum_{\alpha=1}^{N} (\mathbf{e}_\alpha, \mathbf{W}_\alpha \mathbf{Q}_i V_0[\mathbf{r}_\alpha] \mathbf{S}^\top \mathbf{W}_\alpha \mathbf{e}_\alpha)\Big)_{i=0,3} \\ \mathbf{0} \end{pmatrix}. \tag{23}$$

(vii) Update $\mathbf{q}$ and $\mathbf{t}$ as follows, and return to Step (ii):

$$\mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}, \qquad \mathbf{t} \leftarrow \mathbf{t} + \Delta\mathbf{t}. \tag{24}$$

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 $\mathbf{q}$ and $\mathbf{t}$ 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, which is popular in geodetic science (Förstner, 2000; Mikhail and Ackermann, 1976; Neitzel, 2010; Perwass et al., 2006; Gebken and Sommer, 2008). Suppose we are given some approximations $\mathbf{r}_\alpha^{(0)}$ and $\mathbf{r}'_\alpha{}^{(0)}$ of the true positions $\bar{\mathbf{r}}_\alpha$ and $\bar{\mathbf{r}}'_\alpha$. Let $\mathbf{q}$ and $\mathbf{t}$ be the current estimates the true solution $\bar{\mathbf{q}}$ and $\bar{\mathbf{t}}$. We write

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

Substituting (25) into (2), doing Taylor expansion, and omitting second and higher order terms in the correction terms, we obtain

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

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

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

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

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

From the first and second equations, we obtain

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

Substitution of these into (26) results in

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

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

$$\mathbf{U}_\alpha^{(0)} = 2\Big(\mathbf{Q}_0\mathbf{r}_\alpha^{(0)} \ \mathbf{Q}_1\mathbf{r}_\alpha^{(0)} \ \mathbf{Q}_2\mathbf{r}_\alpha^{(0)} \ \mathbf{Q}_3\mathbf{r}_\alpha^{(0)}\Big), \tag{31}$$

the third equation in (28) has the form

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

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

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

We see that (30), (32), and (33) define linear equations in $\boldsymbol{\lambda}_1$, ..., $\boldsymbol{\lambda}_N$, $\Delta\mathbf{q}$, and $\Delta\mathbf{t}$. Solving these, we can determine $\bar{\mathbf{q}}$, $\bar{\mathbf{t}}$, $\bar{\mathbf{r}}_\alpha$, and $\bar{\mathbf{r}}'_\alpha$, 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 $\mathbf{q}$ and $\mathbf{t}$ as new current values and upgrade $\mathbf{r}_\alpha^{(0)}$ by the left side of the first equation in (29) using the computed $\boldsymbol{\lambda}_\alpha$ (the value $\mathbf{r}'_\alpha{}^{(0)}$ is not used in the computation). We repeat this until all variables converge. The procedure is summarized as follows:

(i) Provide an initial guess of $\mathbf{q}$ and $\mathbf{t}$, and let $\mathbf{r}_\alpha^{(0)} = \mathbf{r}_\alpha$ and $J_0 = \infty$ (a sufficiently large number).
(ii) Compute the scaled rotation $\mathbf{S}$ in (3) for $\mathbf{q}$.

(iii) Compute the vectors $\mathbf{e}_\alpha$ and the matrices $\mathbf{W}_\alpha$ in (8), and evaluate the residual J in (9).
(iv) If $J \approx J_0$, return $\mathbf{q}$ and $\mathbf{t}$ and stop. Else, let $J_0 \leftarrow J$.
(v) Compute the matrices $\mathbf{Q}_i$ in (11) and $\mathbf{U}_\alpha^{(0)}$ in (31).
(vi) Solve the following $(3N+7)$-D linear equation:

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

(vii) Update $\mathbf{r}_\alpha^{(0)}$, $\mathbf{q}$ and $\mathbf{t}$ as follows, and return to Step (ii):

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

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 $\boldsymbol{\lambda}_\alpha$ in the form

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

where $\mathbf{W}_\alpha$ 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 $\Delta\mathbf{q}$ and $\Delta\mathbf{t}$. Hence, the procedure of the Gauss-Helmert method can also be written in the following form:

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

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

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

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

This reduction of the algorithm reduces the memory usage in the computer. Since (34) is a $(3N+7)$-D linear equation, the coefficient matrix has $(3N+7)^2$ elements, and the unknown is a $(3N+7)$-D vector. In contrast, (37) is a 7-D linear equation; the coefficient matrix has 49 elements, and the unknown is a 7-D vector. The effect memory reduction becomes more significant as the number N of the data increases. Mathematically, this reduction of a high-dimensional linear equation to a smaller dimension is equivalent to replacing a part of the coefficient matrix with a submatrix known as the *Schur complement* (Triggs, et al., 1999).

At the same time, (37) makes the similarity to the Gauss-Newton method more apparent. Comparing (23) and (37), we see that the matrices $\mathbf{U}_\alpha$ in (16) are replaced by the matrices $\mathbf{U}_\alpha^{(0)}$ in (31). For the Gauss-Newton method, the right side of (23) is the gradient of the residual J with respect to $\mathbf{q}$ and $\mathbf{t}$, but the counterpart of the second term is missing on the right side of (37). However, when $\Delta\mathbf{q} = \Delta\mathbf{t} = \mathbf{0}$ at the time of the convergence of the Gauss-Helmert iterations, we see from (36) that $\boldsymbol{\lambda}_\alpha = -\mathbf{W}_\alpha\mathbf{e}_\alpha$. Hence, the first equation in (38) implies

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

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

$$\mathbf{U}_\alpha^{(0)} = \mathbf{U}_\alpha + 2\Big(\mathbf{Q}_0V_0[\mathbf{r}_\alpha]\mathbf{S}^\top\mathbf{W}_\alpha\mathbf{e}_\alpha\ \mathbf{Q}_1V_0[\mathbf{r}_\alpha]\mathbf{S}^\top\mathbf{W}_\alpha\mathbf{e}_\alpha\ \mathbf{Q}_2V_0[\mathbf{r}_\alpha]\mathbf{S}^\top\mathbf{W}_\alpha\mathbf{e}_\alpha\ \mathbf{Q}_3V_0[\mathbf{r}_\alpha]\mathbf{S}^\top\mathbf{W}_\alpha\mathbf{e}_\alpha\Big). \tag{40}$$

Hence, we have

$$\mathbf{U}_\alpha^{(0)\top}\mathbf{W}_\alpha\mathbf{e}_\alpha = \mathbf{U}_\alpha\mathbf{W}_\alpha\mathbf{e}_\alpha + 2\Big((\mathbf{e}_\alpha, \mathbf{W}_\alpha\mathbf{Q}_i V_0[\mathbf{r}_\alpha]\mathbf{S}^\top\mathbf{W}_\alpha\mathbf{e}_\alpha)\Big)_{i=0,1,2,3}. \tag{41}$$

If this is substituted, the right side of (37) coincides with that of (23). Thus, 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 $O(\mathbf{e}_\alpha)$.

6. Modified Gauss-Helmert method

It is easily seen that (6) of the Gauss-Newton method gives the maximum likelihood estimators of $\bar{\mathbf{r}}_\alpha$ and $\bar{\mathbf{r}}'_\alpha$, given the current estimates $\mathbf{q}$ and $\mathbf{t}$, while the first equation in (35) (or (38)) gives an iterative update using the Lagrange multipliers $\boldsymbol{\lambda}_\alpha$ in (36) expressed in the increments $\Delta\mathbf{q}$ and $\Delta\mathbf{t}$ of $\mathbf{q}$ and $\mathbf{t}$. Although the resulting $\mathbf{r}_\alpha^{(0)}$ converges 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:

(i) Provide an initial guess of $\mathbf{q}$ and $\mathbf{t}$, and let $J_0 = \infty$ (a sufficiently large number).

(ii) Compute the scaled rotation $\mathbf{S}$ in (3) for $\mathbf{q}$.

(iii) Compute the vectors $\mathbf{e}_\alpha$ and the matrices $\mathbf{W}_\alpha$ in (8), and compute $\mathbf{r}_\alpha^{(0)}$ as follows:

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

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

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

(vi) Solve the following 7-D linear equation:

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

(vii) Update $\mathbf{r}_\alpha^{(0)}$, $\mathbf{q}$, and $\mathbf{t}$ as follows, and return to Step (ii):

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

In other words, the Lagrange multipliers $\boldsymbol{\lambda}_\alpha$, which are the main characteristics of the Gauss-Helmert method, no longer appear in this modified algorithm. If the values of $\mathbf{r}_\alpha^{(0)}$ defined in (42) are used, the right side of (43) coincides with that of (23). In other words, while the right side of (37) of the (reduced) Gauss-Helmert method agrees with that of the Gauss-Newton method in the end, this modified algorithm uses the expression in the Gauss-Newton method from the beginning. In this sense, this algorithm is a hybrid between the Gauss-Helmert and the Gauss-Newton methods. The only difference is now whether the Gauss-Newton or the Gauss-Helmert approximation is used for the Hessian.

7. Simulated Stereo Vision

We now experimentally compare the convergence performance of the Gauss-Newton, the Gauss-Helmert, and the modified Gauss-Helmert methods by stereo vision simulation, for which inhomogeneous and anisotropic noise characteristics can be analytically modeled.

7.1. *Covariance evaluation*

A curved grid surface is rotated around the world origin O and translated after its scale is changed, as depicted on the left of 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 on the right of Fig. 1. 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 the images and computed their 3-D positions $\hat{\mathbf{r}}_\alpha$ and $\hat{\mathbf{r}}'_\alpha$ by the method described in Kanatani et al. (2008). For optimal similarity estimation, we need to evaluate the normalized covariances $V_0[\hat{\mathbf{r}}_\alpha]$ and $V_0[\hat{\mathbf{r}}'_\alpha]$ of the reconstructed 3-D positions $\hat{\mathbf{r}}_\alpha$ and $\hat{\mathbf{r}}'_\alpha$. Following Kanazawa and Kanatani (1995) and Niitsuma and Kanatani (2011a), 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 the x- and y-axis are parallel to the X- and Y-axis

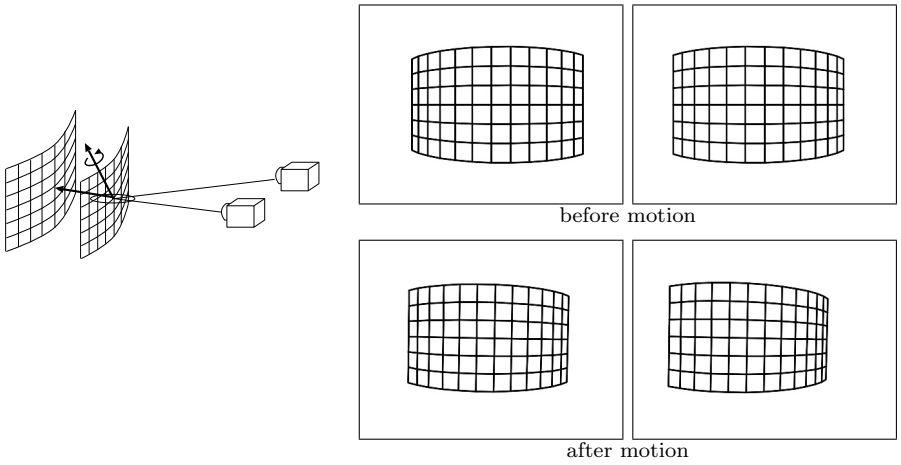

Fig. 1. A grid surface rotates, translates, and changes the scale. The 3-D position of the grid points are measured by stereo vision before and after the similarity motion. The ellipsoid illustrates the measurement uncertainty.

of the world coordinate system, respectively. Then, the camera is rotated around the world coordinate origin O by $\mathbf{R}$ (rotation matrix) and translated by $\mathbf{t}$ from the reference position. We call $\{\mathbf{R}, \mathbf{t}\}$ 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 (Hartley and Zisserman, 2004):

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

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 (Hartley, 1997). The 3×4 projection matrix $\mathbf{P}$ is given by

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

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 $\{\mathbf{R}, \mathbf{t}\}$ and $\{\mathbf{R}', \mathbf{t}'\}$ with focal lengths f and f', respectively. Let $\mathbf{P}$ and $\mathbf{P}'$ be the projection matrices of the respective cameras, and $\mathbf{x}$ and $\mathbf{x}'$ 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 $\mathbf{0}$ and standard deviation σ (pixels). Due to noise, the detected points $\mathbf{x}$ and $\mathbf{x}'$ do not exactly satisfy the epipolar constraint, so we correct $\mathbf{x}$ and $\mathbf{x}'$, respectively, to $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ that exactly satisfy the epipolar constraint in an optimal manner (Appendix A). From the corrected positions $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$, the corresponding 3-D position $\hat{\mathbf{r}}$ is uniquely determined. Note that although the noise in $\mathbf{x}_\alpha$ and $\mathbf{x}'_\alpha$ is assumed to be independent, the noise in the corrected positions $\hat{\mathbf{x}}_\alpha$ and $\hat{\mathbf{x}}'_\alpha$ is no longer independent (Kanatani, 1996). The normalized covariance matrices $V_0[\hat{\mathbf{x}}]$ and $V_0[\hat{\mathbf{x}}']$ and the normalized correlation matrices $V_0[\hat{\mathbf{x}}, \hat{\mathbf{x}}']$ and $V_0[\hat{\mathbf{x}}', \hat{\mathbf{x}}]$ are given as follows (Kanatani, 1996; Kanazawa and Kanatani, 1995):

$$V_0[\hat{\mathbf{x}}] = \frac{1}{f_0^2}\Big(\mathbf{P_k} - \frac{(\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}')(\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}')^\top}{\|\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}'\|^2 + \|\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}}\|^2}\Big), \qquad V_0[\hat{\mathbf{x}}'] = \frac{1}{f_0^2}\Big(\mathbf{P_k} - \frac{(\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})(\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})^\top}{\|\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}'\|^2 + \|\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}}\|^2}\Big),$$

$$V_0[\hat{\mathbf{x}}, \hat{\mathbf{x}}'] = \frac{1}{f_0^2}\Big(-\frac{(\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}')(\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})^\top}{\|\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}'\|^2 + \|\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}}\|^2}\Big) = V_0[\hat{\mathbf{x}}', \hat{\mathbf{x}}]^\top. \tag{47}$$

Here, $\mathbf{F}$ is the fundamental matrix between the two images (Hartley and Zisserman, 2004), and we define $\mathbf{P_k} \equiv \mathrm{diag}(1, 1, 0)$. Since the vector $\hat{\mathbf{X}}$ reconstructed from $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ satisfies the projection relationship in (45), vectors $\hat{\mathbf{x}}$ and $\mathbf{P}\hat{\mathbf{X}}$ are parallel, and so are $\hat{\mathbf{x}}'$ and $\mathbf{P}'\hat{\mathbf{X}}$. Thus, we have

$$\hat{\mathbf{x}} \times \mathbf{P}\hat{\mathbf{X}} = \mathbf{0}, \qquad \hat{\mathbf{x}}' \times \mathbf{P}'\hat{\mathbf{X}} = \mathbf{0} \tag{48}$$

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

$$\Delta\hat{\mathbf{x}} \times \mathbf{P}\hat{\mathbf{X}} + \hat{\mathbf{x}} \times \mathbf{P}\Delta\hat{\mathbf{X}} = \mathbf{0}, \qquad \Delta\hat{\mathbf{x}}' \times \mathbf{P}'\hat{\mathbf{X}}' + \hat{\mathbf{x}}' \times \mathbf{P}'\Delta\hat{\mathbf{X}} = \mathbf{0}. \tag{49}$$

These are combined into one equation in the form

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

where $\Delta\hat{\mathbf{r}}$ is the 3-D vector of the first three components of $\Delta\hat{\mathbf{X}}$ and $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{P}}'$ are the left 3×3 submatrices of the 3×4 projection matrices $\mathbf{P}$ and $\mathbf{P}'$, respectively. Here, we define the product $\mathbf{a} \times \mathbf{A}$ of a 3-D vector $\mathbf{a}$ and a 3×3 matrix $\mathbf{A}$ to be the 3×3 matrix whose columns are the vector products of $\mathbf{a}$ and the respective columns of $\mathbf{A}$ (Kanatani, 1996). Multiplying both sides by the transpose of the left side from left, we obtain

$$\Big((\hat{\mathbf{x}} \times \tilde{\mathbf{P}})^\top (\hat{\mathbf{x}} \times \tilde{\mathbf{P}}) + (\hat{\mathbf{x}}' \times \tilde{\mathbf{P}}')^\top (\hat{\mathbf{x}}' \times \tilde{\mathbf{P}}')\Big)\Delta\hat{\mathbf{r}} = \Big((\hat{\mathbf{x}} \times \tilde{\mathbf{P}})^\top ((\mathbf{P}\hat{\mathbf{X}}) \times \mathbf{I}) \;\; (\hat{\mathbf{x}}' \times \tilde{\mathbf{P}}')^\top ((\mathbf{P}'\hat{\mathbf{X}}) \times \mathbf{I})\Big) \begin{pmatrix} \Delta\hat{\mathbf{x}} \\ \Delta\hat{\mathbf{x}}' \end{pmatrix}, \tag{51}$$

which can be rewritten this in form

$$\mathbf{A}\Delta\hat{\mathbf{r}} = \mathbf{B} \begin{pmatrix} \Delta\hat{\mathbf{x}} \\ \Delta\hat{\mathbf{x}}' \end{pmatrix}, \tag{52}$$

$$\mathbf{A} \equiv \|\hat{\mathbf{x}}\|^2 \tilde{\mathbf{P}}^\top \mathbf{P}_{\mathcal{N}[\hat{\mathbf{x}}]} \tilde{\mathbf{P}} + \|\hat{\mathbf{x}}'\|^2 \tilde{\mathbf{P}}'^\top \mathbf{P}_{\mathcal{N}[\hat{\mathbf{x}}']} \tilde{\mathbf{P}}', \qquad \mathbf{B} \equiv \Big(\tilde{\mathbf{P}}^\top \Big((\hat{\mathbf{x}}, \mathbf{P}\hat{\mathbf{X}})\mathbf{I} - (\mathbf{P}\hat{\mathbf{X}})\hat{\mathbf{x}}^\top\Big) \;\; \tilde{\mathbf{P}}'^\top \Big((\hat{\mathbf{x}}', \mathbf{P}'\hat{\mathbf{X}})\mathbf{I} - (\mathbf{P}'\hat{\mathbf{X}})\hat{\mathbf{x}}'^\top\Big)\Big), \tag{53}$$

where we define

$$\mathbf{P}_{\mathcal{N}[\hat{\mathbf{x}}]} \equiv \mathbf{I} - \mathcal{N}[\hat{\mathbf{x}}]\mathcal{N}[\hat{\mathbf{x}}]^\top, \qquad \mathbf{P}_{\mathcal{N}[\hat{\mathbf{x}}']} \equiv \mathbf{I} - \mathcal{N}[\hat{\mathbf{x}}']\mathcal{N}[\hat{\mathbf{x}}']^\top, \tag{54}$$

and $\mathcal{N}[\,\cdot\,]$ denotes normalization to unit norm ($\mathcal{N}[\mathbf{a}] = \mathbf{a}/\|\mathbf{a}\|$). From (52), we obtain

$$\Delta\hat{\mathbf{r}}\Delta\hat{\mathbf{r}}^\top = \mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} \Delta\hat{\mathbf{x}}\Delta\hat{\mathbf{x}}^\top & \Delta\hat{\mathbf{x}}\Delta\hat{\mathbf{x}}'^\top \\ \Delta\hat{\mathbf{x}}'\Delta\hat{\mathbf{x}}^\top & \Delta\hat{\mathbf{x}}'\Delta\hat{\mathbf{x}}'^\top \end{pmatrix} \mathbf{B}^\top (\mathbf{A}^{-1})^\top. \tag{55}$$

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

$$V_0[\hat{\mathbf{r}}] = \mathbf{A}^{-1}\mathbf{B} \begin{pmatrix} V_0[\hat{\mathbf{x}}] & V_0[\hat{\mathbf{x}}, \hat{\mathbf{x}}'] \\ V_0[\hat{\mathbf{x}}', \hat{\mathbf{x}}] & V_0[\hat{\mathbf{x}}'] \end{pmatrix} \mathbf{B}^\top (\mathbf{A}^{-1})^\top. \tag{56}$$

Note that here we have replaced only $\Delta\hat{\mathbf{x}}\Delta\hat{\mathbf{x}}^\top$, $\Delta\hat{\mathbf{x}}\Delta\hat{\mathbf{x}}'^\top$, $\Delta\hat{\mathbf{x}}'\Delta\hat{\mathbf{x}}^\top$, and $\Delta\hat{\mathbf{x}}'\Delta\hat{\mathbf{x}}'^\top$ by their expectations. Strictly speaking, $\hat{\mathbf{x}}_\alpha$ and $\hat{\mathbf{x}}'_\alpha$ are also random variables, but we use the values computed from the data for them. Replacing only those values that cannot be observed by their expectations is a practical compromise. Evaluating the normalized covariance matrix $V_0[\hat{\mathbf{r}}_\alpha]$ in (56), we find that the uncertainty distribution has an ellipsoidal shape elongated in the depth direction, as illustrated on the left of 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 (56).

This stereo vision simulation is merely for evaluating the performance of the Gauss-Newton and the (original and modified) Gauss-Helmert methods; we are not intending to obtain a practical stereo vision system, for which many realistic issues including robust image matching are to be resolved.

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 ($\mathbf{R} = \mathbf{I}$, $\mathbf{t} = \mathbf{0}$, $s = 1$) and imposed no threshold for convergence: we stopped if J stops decreasing. In each step, unchanged digits are underlined. Note that this does not mean that these digits are correct. Here, we are interested in how quickly the residual decreases and how quickly the number of converged digits increases. We are doing double precision computation with a 64 bit length register, so 16 decimal digits should be unchanged after convergence.

From Table 1, we see that the initial decrease in J of the Gauss-Helmert method is conspicuous; it drops sharply and almost abruptly in the first few iterations. In contrast, the residual J of the Gauss-Newton method reduces continuously, and the number of converged digits increases 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 $\bar{\mathbf{r}}_\alpha$ for the "current" estimate of the similarity. Since the initial guess (the identity) is far from the truth, the values $\mathbf{r}_\alpha^{(0)}$ computed by (42) are very poor approximations, while the Gauss-Helmert method initializes $\mathbf{r}_\alpha^{(0)}$ by the data $\mathbf{r}_\alpha$ themselves, so they are better estimates of $\bar{\mathbf{r}}_\alpha$. *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.

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.

	Gauss-Newton	Gauss-Helmert	Modified G-H
$\sigma = 1.0$			
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
$\sigma = 2.0$			
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	
$\sigma = 3.0$			
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		

Figure 2**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. 2**b** shows the result initialized using the widely used homogeneous and isotropic noise model assuming $V_0[\mathbf{r}_\alpha] = V_0[\mathbf{r}_\alpha] = \mathbf{I}$ for all α. In this case, we compute the centroids $\mathbf{r}_c$ and $\mathbf{r}'_c$ of the data $\{\mathbf{r}_\alpha\}$ and $\{\mathbf{r}'_\alpha\}$, respectively, and the deviations $\tilde{\mathbf{r}}_\alpha = \mathbf{r}_\alpha - \mathbf{r}_c$ and $\tilde{\mathbf{r}}'_\alpha = \mathbf{r}'_\alpha - \mathbf{r}'_c$ from their respective centroids. The scale change s is estimated by

$$s = \sqrt{\frac{\sum_{\alpha=1}^{N} \|\tilde{\mathbf{r}}'_\alpha\|^2}{\sum_{\alpha=1}^{N} \|\tilde{\mathbf{r}}_\alpha\|^2}}, \tag{57}$$

and the rotation $\mathbf{R}$ is computed from $\{\tilde{\mathbf{r}}_\alpha\}$ and $\{\tilde{\mathbf{r}}'_\alpha\}$ by the method of singular value decomposition (SVD) (Appendix B). The translation $\mathbf{t}$ is determined from $\mathbf{t} = \mathbf{r}'_c - s\mathbf{R}\mathbf{r}_c$. As we see from Fig. 2**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 $\hat{\mathbf{R}}$, the translation $\hat{\mathbf{t}}$, and the scale change $\hat{s}$. Note that the accuracy is independent of the convergence performance observed above. As the analysis in Section 5 shows, the Gauss-Newton, the Gauss-Helmert, and the modified Gauss-Helmert methods all should produce the same solution when

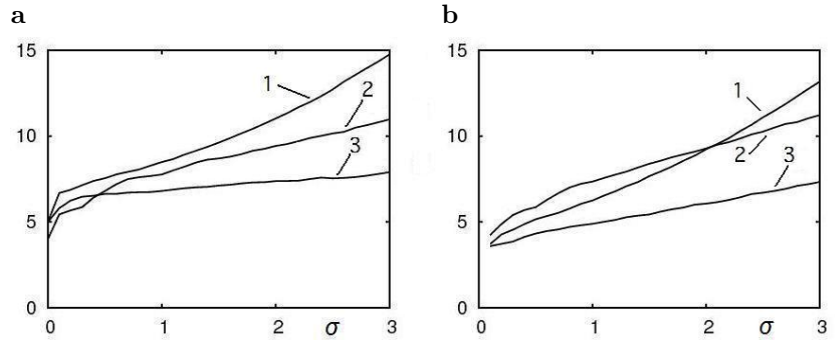

Fig. 2. The average number of iterations for various noise level σ (pixels) over 1000 independent trials: (1) Gauss-Newton, (2) Gauss-Helmert, and (3) modified Gauss-Helmert. (a) Initialized by the identity. (b) Initialized using the homogeneous and isotropic noise model.

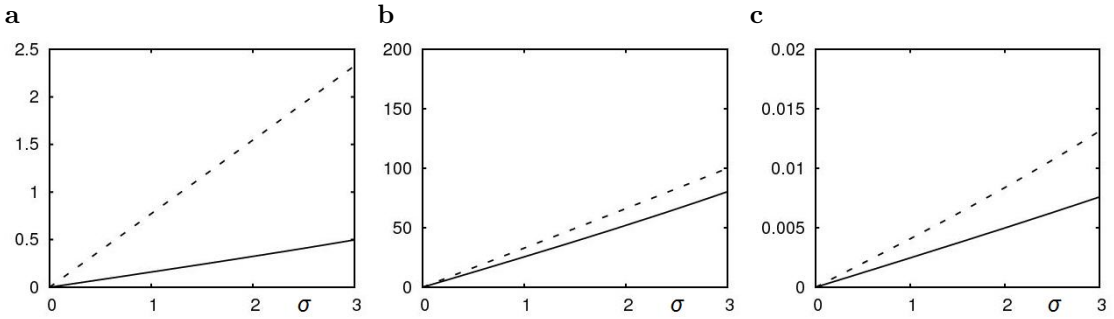

Fig. 3. The RMS error vs. the standard deviation σ (pixels) of the noise added to the stereo images: (a) rotation, (b) translation, and (c) scale change. The dotted lines are for the homogeneous and isotropic noise model.

they have converged, irrespective of the speed of convergence. Here, we compare their accuracy with the method using homogeneous and isotropic noise. Let $\bar{\mathbf{R}}$, $\bar{\mathbf{t}}$, and $\bar{s}$ be their true values, respectively. We computed the rotation angle $\delta\Omega$ (degree) of the relative rotation $\hat{\mathbf{R}}\bar{\mathbf{R}}^{-1}$, the translation error $\delta\mathbf{t} = \hat{\mathbf{t}} - \bar{\mathbf{t}}$ 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

$$E_{\mathbf{R}} = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}(\delta\Omega^{(a)})^2}, \qquad E_{\mathbf{t}} = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}\|\delta\mathbf{t}^{(a)}\|^2}, \qquad E_s = \sqrt{\frac{1}{1000}\sum_{a=1}^{1000}(\delta s^{(a)})^2}, \tag{58}$$

where the superscript (a) denotes the value of the ath trial. Figure 3 plots these for various σ. It is clearly demonstrated that accurate estimation cannot be done unless the inhomogeneity and anisotropy of the 3-D sensing data are taken into consideration.

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 (Acar et al., 2006). 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 normalized covariance matrices (up to a common noise level) are in the order listed in the table

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

For the 1998 data,

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

The noise level σ is around the order of 10^{-4} (meters).

Table 2
The 3-D data (in meters) of five points near Istanbul in October 1997 and March 1998 (Acar et al., 2006)

X	Y	Z
October 1997		
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		
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 3
a. Changes in the residual J ($\times 10^{-6}$) for the data in Table 2. Unchanged digits are underlined. b. The translation $\mathbf{t} = (t_1, t_2, t_3)^\top$ (meters), the scale change s, the rotation angle Ω (degrees), and the residual J ($\times 10^{-6}$) estimated from the data in Table 2.

	Gauss-Newton	Gauss-Helmert	Modified G-H
a			
0	13.90466081612066	13.90466081612066	13.90466081612066
1	6.409224221023396	6.409224221568889	6.409224220271752
2	6.409224212420786	6.409224212419703	6.409224212421016
3	6.409224212420786	6.409224212419703	6.409224212421016

	Conventional	Optimal
b		
t_1	−199.8603562003	−274.6708352865
t_2	42.52530292571	100.2332078221
t_3	143.6578706433	140.7879487881
s	1.000003703184	1.000008522356
l_1	−0.0495064987939	−0.0085468407697
l_2	0.9328527741966	0.8213706338012
l_3	−0.3568400317382	−0.5703308105318
Ω	0.002242810319013	0.002887644251870
J	$9.242857990972 \times 10^{-6}$	$6.40922421242 \times 10^{-6}$

We regarded $(4233000, 2308000, 4161000)$ as a tentative world coordinate origin and applied the three methods. Table 3(a) shows the changes in J initialized by the identity (unchanged digits are underlined). Table 3(b) lists the computed translation $\mathbf{t}$ (meters), the scale change s, the rotation axis $\mathbf{l}$ (unit vector), and the rotation angle Ω (degrees) with respect to the original world coordinate system, which are compared with the solution using the conventional homogeneous and isotropic noise model. The two solutions agree at most in the first digit. This illustrates that accurate estimation cannot be done without using the optimal computation shown here.

Note that all the numbers in Table 2 have 11 digits. Retaining such long digit sequences in the computation would cause shortage of bits for intermediate values with sufficient accuracy even in double precision. In fact, if we use the numbers in Table 2 as are, we observe that the residual converges only up to seven digits; the remaining digits keep fluctuating after that. Subtracting $(4233000, 2308000, 4161000)$ makes all the values in Table 2 seven-digit numbers. As a result, as we see from Table 3(a), the residual converges to 16 digits in three iterations for all the methods. The necessity of appropriate normalization of image data for stabilizing finite length computation is well recognized for computer vision applications (Hartley, 1997), but this is also true for GPS data, which usually consist of long digit sequences.

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, which is widely regarded as the standard optimization technique by computer vision and robotics engineers, and

the Gauss-Helmert method, which is popular among geodetic scientists. Doing stereo vision simulation with inhomogeneous and anisotropic noise in 3-D data, we observed that

(i) The Gauss-Helmert method has a very similar structure to the Gauss-Newton method. This becomes evident if the unknowns are reduced to the similarity variables alone by eliminating 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".

(ii) 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. However, both converge to the same solution up to 16 digits in double precision. 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 them to define a hybrid, which we call the "modified Gauss-Helmert method". We observed that:

(iii) The initial drop of the residual is not so sharp as the Gauss-Helmert method, but it converges to the same solution (up to 16 digits in double precision), and the convergence is as smooth as the Gauss-Newton method.

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

We also applied our method to real GPS geodetic data and found that the widely used homogeneous and isotropic noise model is insufficient for accurate estimation. We also found that numerical inaccuracy can result if long digit sequences of GPS data are used without appropriate normalization.

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 JSPS Grant-in-Aid for Challenging Exploratory Research (24650086).

Appendix A. 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. 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. 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 (Kanatani et al., 2008):

(i) 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 $\mathbf{F} = (F_{ij})$ as the 9-D vector $\mathbf{f} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top$.

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

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

$$\boldsymbol{\xi}^* = \begin{pmatrix} \hat{x}\hat{x}' + \hat{x}'\tilde{x} + \hat{x}\tilde{x}' \\ \hat{x}\hat{y}' + \hat{y}'\tilde{x} + \hat{x}\tilde{y}' \\ f_0(\hat{x} + \tilde{x}) \\ \hat{y}\hat{x}' + \hat{x}'\tilde{y} + \hat{y}\tilde{x}' \\ \hat{y}\hat{y}' + \hat{y}'\tilde{y} + \hat{y}\tilde{y}' \\ f_0(\hat{y} + \tilde{y}) \\ f_0(\hat{x}' + \tilde{x}') \\ f_0(\hat{y}' + \tilde{y}') \\ f_0^2 \end{pmatrix}. \tag{A.1}$$

Here, f_0 is a fixed reference length of approximately the image size.

(iii) Update $\tilde{x}$, $\tilde{y}$, $\tilde{x}'$, $\tilde{y}'$, $\hat{x}$, $\hat{y}$, $\hat{x}'$, and $\hat{y}'$ as follows:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \leftarrow \frac{(\mathbf{f}, \boldsymbol{\xi}^*)}{(\mathbf{f}, V_0[\hat{\boldsymbol{\xi}}]\mathbf{f})} \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \end{pmatrix} \begin{pmatrix} \hat{x}' \\ \hat{y}' \\ 1 \end{pmatrix}, \qquad \begin{pmatrix} \tilde{x}' \\ \tilde{y}' \end{pmatrix} \leftarrow \frac{(\mathbf{f}, \boldsymbol{\xi}^*)}{(\mathbf{f}, V_0[\hat{\boldsymbol{\xi}}]\mathbf{f})} \begin{pmatrix} F_{11} & F_{21} & F_{31} \\ F_{12} & F_{22} & F_{32} \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix}, \tag{A.2}$$

$$\hat{x} \leftarrow x - \tilde{x}, \qquad \hat{y} \leftarrow y - \tilde{y}, \qquad \hat{x}' \leftarrow x' - \tilde{x}', \qquad \hat{y}' \leftarrow y' - \tilde{y}'. \tag{A.3}$$

(iv) Compute the reprojection error E by

$$E = \tilde{x}^2 + \tilde{y}^2 + \tilde{x}'^2 + \tilde{y}'^2. \tag{A.4}$$

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 (ii).

Appendix B. Optimal rotation estimation for homogeneous isotropic noise

Various methods are known for optimally computing the 3-D rotation for homogeneous and isotropic noise (Arun et al., 1987; Horn, 1987; Horn et al., 1988; Kanatani, 1994; Umeyama, 1991), but all are mathematically equivalent. The simplest is the following method in terms of the singular value decomposition (SVD) (Kanatani, 1993):

(i) Compute the following correlation matrix $\mathbf{N}$ between the 3-D positions $\mathbf{r}_\alpha$ and $\mathbf{r}'_\alpha$ before and after the rotations:

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

(ii) Compute the SVD of $\mathbf{N}$ in the form

$$\mathbf{N} = \mathbf{U}\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)\mathbf{V}^\top, \tag{B.2}$$

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

(iii) Return the following rotation matrix:

$$\mathbf{R} = \mathbf{U}\mathrm{diag}(1, 1, \det(\mathbf{U}\mathbf{V}^\top))\mathbf{V}^\top. \tag{B.3}$$

References

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

Akyilmaz, O., 2011. Solution of the heteroscedastic datum transformation problems. In: Abstract of 1st Int. Workshop the Quality of Geodetic Observation and Monitoring Systems.
http://www.gih.uni-hannover.de/qugoms2011/ProgramAbstracts1.htm

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

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

Felus, Y.A., Burch, R.C., 2009. On symmetrical three-dimensional datum conversion. GPS Solut. 13, 65–74.

Fitzgibbon, A., 2003. Robust registration of 2D and 3D point sets. Image Vis. Comput. 1145–1153.

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

Gebken, C., Sommer, G., 2008. Stochastically optimal epipole estimation in omnidirectional images with geometric algebra. In: Sommer, G., Klette, R. (Eds.), RobVis 2008, pp. 85–97.

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

Hartley, R.I., 1997. In defense of the eight-point algorithm. IEEE Trans. Patt. Anal. Mach. Intell. 19, 580–593.

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

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

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

Kanatani, K., 1990. Group-Theoretical Methods in Image Understanding. Springer, Berlin, Germany.

Kanatani, K., 1993. Geometric Computation for Machine Vision. Oxford University Press, Oxford, U.K.

Kanatani, K., 1994. Analysis of 3-D rotation fitting. IEEE Trans. Patt. Anal. Mach. Intell. 16, 543–549.

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

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

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

Matei, B., Meer, P., 1999. Optimal rigid motion estimation and performance evaluation with bootstrap. In: Proc. IEEE Conf. Computer Vision Pattern Recognition, pp. 339–345.

Mikhail, E.M., Ackermann, F., 1976. Observations and Least Squares. University Press of America, Lanham, MD., U.S.A.

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

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

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

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

Perwass, C., Gebken, C., Sommer, G., 2006. Geometry and kinematics with uncertain data. In: Proc. 9th Euro. Conf. Comput. Vision, 1, pp. 225–237.

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

Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.W., 1999. Bundle adjustment—a modern synthesis. In: Triggs, B., Zissermann, A., Hartley, R. (Eds.), Vision Algorithms: Theory and Practice, Springer, Berlin, pp. 298–372.

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