Optimal Robot Self-Localization and Reliability Evaluation*

Kenichi Kanatani and Naoya Ohta

Department of Computer Science, Gunma University
Kiryu, Gunma 376-8515 Japan
{kanatani|ohta}@cs.gunma-u.ac.jp

Abstract. We discuss optimal estimation of the current location of a robot by matching an image of the scene taken by the robot with the model of the environment. We first present a theoretical accuracy bound and then give a method that attains that bound, which can be viewed as describing the probability distribution of the current location. Using real images, we demonstrate that our method is superior to the naive least-squares method. We also confirm the theoretical predictions of our theory by applying the bootstrap procedure.

1 Introduction

For a robot to navigate autonomously, it must have a geometric model of the environment; it may be given as data or constructed by the robot itself from vision and sensor data. Here, we consider the case in which a robot already has a three-dimensional map of the environment and study the problem of identifying its current location in the world model. In theory, the current location can be computed by tracing the history of motion from a known initial position, e.g., integrating the rotation of the wheels or incrementally correcting the position by estimating robot motion from images [8]. However, the accuracy of the computed location quickly deteriorates as errors (due to slippage of the wheels, vibration of the camera, etc.) are accumulated in the course of integration. At some point, therefore, we need to estimate the current location by some direct means.

A typical method for self-localization is computing the current position of the camera by matching feature points detected in the images with their corresponding positions in the world model. A direct method is stereo vision, by which the 3-D locations of the feature points can be computed relative to the cameras [1]. This fails, however, if the feature points are located very far away as compared with the baseline of the stereo system. In an outdoor environment, feature points easily discernible from a wide range of positions are usually those located very far away (e.g., towers and mountain tops). Hence, we need a method for computing the current position by matching a single image with the world model.

* This work was in part supported by the Ministry of Education, Science, Sports and Culture, Japan under a Grant in Aid for Scientific Research C(2) (No. 09680352).

The problem of matching image and model features is inseparable from the problem of computing the 3-D position; we first hypothesize a matching based on known clues (e.g., brightness, color, shape, etc.) and then validate the resulting 3-D position (e.g., by comparing it with that obtained by integrating the history of motion, examining the image if features that should be observed from that position actually exist, etc.) [10, 11]. Hence, computing the 3-D position for given matching between image and model features is crucial whether the matching is correct or not.

Computing the 3-D relationship between image and model features has been studied by many researchers in the past in the form known as "PnP", in which the goal is to compute the 3-D positions of the feature points relative to the camera, given a 3-D configuration of the feature points relative to each other. Here, we are interested in computing the absolute position of the camera, given absolute 3-D positions of feature points.

If the robot motion is constrained to be on a horizontal surface (e.g., the ground or a floor), a simple method based on elementary geometry of circles is well known for this purpose [7]. It can also be applied to three-dimensional motion by replacing circles by spheres [9]. But this technique uses only pairwise relative orientations of the lines of sight defined by the feature points; their absolute positions in the image are not used. Using minimal information has the advantage that it can be adapted to mismatch removal: we pick out multiple minimal sets of data and choose the solution supported by majority voting [4]. For assumed matching, however, it is obviously better to fuse all available information in an optimal manner. Such a method also exists [2], but so far the main concern has been *methods* for estimation; little attention has been given on *theoretical optimality* and *reliability of the solution*.

The aim of this paper is *not* to propose yet another new solution technique. Rather, we focus on *statistical* aspects. We first introduce a model of noise and view the problem as statistical estimation. Then, we present a *theoretical accuracy bound* that can be evaluated independently of particular solution techniques involved. Next, we give a computational scheme that attains that bound; such a method alone can be called "optimal".

Since the solution attains the accuracy bound, we can view it as quantitatively describing the "probability distribution" of the current location of the robot. We show that we can compute this distribution without any knowledge about the magnitude of image noise. This computation helps validate the hypothesized matching: if the evaluated distribution spreads out widely, the hypothesis is very questionable. We confirm the theoretical predictions of our theory by using real images and applying the bootstrap procedure [3].

2 Statistical Self-Localization

We regard the camera imaging geometry as perspective projection and define an XYZ camera coordinate system in such a way that its origin is at the center of projection and its optical axis is along the Z-axis (Fig. 1(a)). Letting f be the

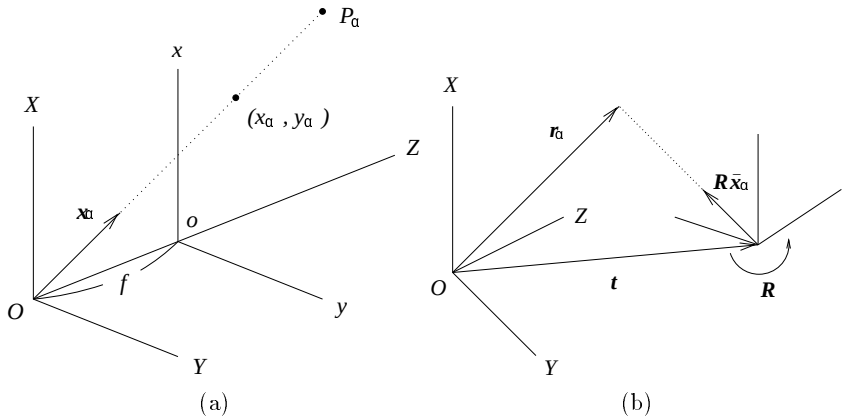


Fig. 1. (a) Camera imaging geometry. (b) The camera coordinate system and the world coordinate system.

focal length, we identify the plane $Z = f$ with the image plane, on which we define an xy image coordinate system in such a way that the origin is on the optical axis and the x- and y-axes are parallel to the X- and Y-axes, respectively.

We regard observed image coordinates (x_α, y_α) (in pixels) as perturbed from their true values $(\bar{x}_\alpha, \bar{y}_\alpha)$ by noise and write

$$x_\alpha = \bar{x}_\alpha + \Delta x_\alpha, \qquad y_\alpha = \bar{y}_\alpha + \Delta y_\alpha. \tag{1}$$

We regard Δx_α and Δy_α as (generally correlated) Gaussian random variables of mean 0, independent for each α.

Suppose the camera coordinate system is in a position defined by translating the world coordinate system by $\boldsymbol{t}$ and rotating it by $\boldsymbol{R}$ with respect to the world coordinate system (Fig. 1(b)). We call $\{\boldsymbol{t}, \boldsymbol{R}\}$ the *motion parameters*. Our goal is formally stated as follows:

Problem 1. Given image coordinates (x_α, y_α), $\alpha = 1, ..., N$, of feature points whose *3-D* positions $\boldsymbol{r}_\alpha$, $\alpha = 1, ..., N$, with respect to the world coordinate system are known, optimally compute the motion parameters $\{\boldsymbol{t}, \boldsymbol{R}\}$ and their probability distribution.

We represent a point with image coordinates (x, y) by the following three-dimensional vector:

$$\boldsymbol{x} = \begin{pmatrix} x/f \\ y/f \\ 1 \end{pmatrix}. \tag{2}$$

This vector indicates the line of sight starting from the camera coordinate origin and passing through the corresponding point in the scene (Fig. 1(a)). Let $\boldsymbol{x}_\alpha$ and $\bar{\boldsymbol{x}}_\alpha$ be the αth observed point and its true position, respectively. The error

$\Delta \boldsymbol{x}_\alpha = \boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha$ is a three-dimensional vector. We define its covariance matrix by

$$V[\boldsymbol{x}_\alpha] = E[\Delta \boldsymbol{x}_\alpha \Delta \boldsymbol{x}_\alpha^\top], \tag{3}$$

where $E[\,\cdot\,]$ denotes expectation and the superscript $\top$ denotes transpose. Since the Z component of $\Delta \boldsymbol{x}_\alpha$ is identically 0, the covariance matrix $V[\boldsymbol{x}_\alpha]$ is singular; its third row and third column consist of 0s.

The covariance matrix $V[\boldsymbol{x}_\alpha]$ measures the uncertainty of detecting the feature point $\boldsymbol{x}_\alpha$, but in practice it is usually very difficult to predict it precisely. However, it is often possible to predict the relative likelihood of noise. Here, we assume that the covariance matrix is known only *up to scale* and write

$$V[\boldsymbol{x}_\alpha] = \epsilon^2 V_0[\boldsymbol{x}_\alpha]. \tag{4}$$

We assume that $V_0[\boldsymbol{x}_\alpha]$ is known but the constant ϵ is unknown; we call ϵ the *noise level*, and $V_0[\boldsymbol{x}_\alpha]$ (generally different from point to point) the *normalized covariance matrix* [6].

For example, if Δx_α and Δy_α are subject to an isotropic and identical Gaussian distribution of mean 0 and variance σ^2, we have

$$\epsilon = \frac{\sigma}{f}, \qquad V_0[\boldsymbol{x}_\alpha] = \mathrm{diag}(1, 1, 0), \tag{5}$$

where $\mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$ denotes the diagonal matrix with diagonal elements λ_1, λ_2, and λ_3 in that order.

The vector $\bar{\boldsymbol{x}}_\alpha$ is defined with respect to the camera coordinate system. If it is described with respect to the world coordinate system, it becomes $\boldsymbol{R}\bar{\boldsymbol{x}}_\alpha$ (Fig. 1(b)). Hence, letting Z_α be the depth of the αth feature point in the scene from the camera coordinate origin, we obtain the following relationship:

$$\boldsymbol{r}_\alpha = \boldsymbol{t} + Z_\alpha \boldsymbol{R}\bar{\boldsymbol{x}}_\alpha. \tag{6}$$

Such a depth Z_α exists if and only if vector $\boldsymbol{r}_\alpha - \boldsymbol{t}$ is parallel to vector $\boldsymbol{R}\bar{\boldsymbol{x}}_\alpha$. Hence, Problem 1 reduces to the following statistical estimation:

Problem 2. Given $\{\boldsymbol{r}_\alpha\}$, estimate the motion parameters $\{\boldsymbol{t}, \boldsymbol{R}\}$ that satisfy

$$(\boldsymbol{t} - \boldsymbol{r}_\alpha) \times \boldsymbol{R}\bar{\boldsymbol{x}}_\alpha = \boldsymbol{0}, \qquad \alpha = 1, ..., N, \tag{7}$$

from the noisy data $\{\boldsymbol{x}_\alpha\}$. At the same time, compute the probability distribution of the estimated motion parameters $\{\boldsymbol{t}, \boldsymbol{R}\}$.

3 Theoretical Accuracy Bound

Let $\{\hat{\boldsymbol{t}}, \hat{\boldsymbol{R}}\}$ be an estimator of the true motion parameters $\{\bar{\boldsymbol{t}}, \bar{\boldsymbol{R}}\}$. The deviation of translation can be measured by the "difference"

$$\Delta \boldsymbol{t} = \hat{\boldsymbol{t}} - \bar{\boldsymbol{t}} \tag{8}$$

of the estimator $\hat{\boldsymbol{t}}$ from its true value $\bar{\boldsymbol{t}}$. The deviation of rotation can be measured by the "quotient" $\hat{\boldsymbol{R}}\bar{\boldsymbol{R}}^\top$, i.e., the rotation of $\hat{\boldsymbol{R}}$ relative to $\bar{\boldsymbol{R}}$. Let $\boldsymbol{l}$ (unit vector) and $\Delta\Omega$ be, respectively, the axis and angle of the relative rotation $\hat{\boldsymbol{R}}\bar{\boldsymbol{R}}^\top$, and define

$$\Delta\boldsymbol{\Omega} = \Delta\Omega\boldsymbol{l}. \tag{9}$$

We define the covariance matrices of the estimator $\{\hat{\boldsymbol{t}},\ \hat{\boldsymbol{R}}\}$ as follows:

$$\begin{aligned} V[\hat{\boldsymbol{t}}] &= E[\Delta\boldsymbol{t}\Delta\boldsymbol{t}^\top], & V[\hat{\boldsymbol{t}},\hat{\boldsymbol{R}}] &= E[\Delta\boldsymbol{t}\Delta\boldsymbol{\Omega}^\top], \\ V[\hat{\boldsymbol{R}},\hat{\boldsymbol{t}}] &= E[\Delta\boldsymbol{\Omega}\Delta\boldsymbol{t}^\top], & V[\hat{\boldsymbol{R}}] &= E[\Delta\boldsymbol{\Omega}\Delta\boldsymbol{\Omega}^\top]. \end{aligned} \tag{10}$$

Applying the theory of Kanatani [6], we can obtain the following lower bound, which Kanatani called the *Cramer-Rao lower bound* in analogy with the corresponding bound in traditional statistical estimation:

$$\begin{pmatrix} V[\hat{\boldsymbol{t}}] & V[\hat{\boldsymbol{t}},\hat{\boldsymbol{R}}] \\ V[\hat{\boldsymbol{R}},\hat{\boldsymbol{t}}] & V[\hat{\boldsymbol{R}}] \end{pmatrix} \succ \epsilon^2 \begin{pmatrix} \sum_{\alpha=1}^N \bar{\boldsymbol{A}}_\alpha^\top \bar{\boldsymbol{W}}_\alpha \bar{\boldsymbol{A}}_\alpha & \sum_{\alpha=1}^N \bar{\boldsymbol{A}}_\alpha^\top \bar{\boldsymbol{W}}_\alpha \bar{\boldsymbol{B}}_\alpha \\ \sum_{\alpha=1}^N \bar{\boldsymbol{B}}_\alpha^\top \bar{\boldsymbol{W}}_\alpha \bar{\boldsymbol{A}}_\alpha & \sum_{\alpha=1}^N \bar{\boldsymbol{B}}_\alpha^\top \bar{\boldsymbol{W}}_\alpha \bar{\boldsymbol{B}}_\alpha \end{pmatrix}^{-1}. \tag{11}$$

Here, $\boldsymbol{U} \succ \boldsymbol{V}$ means that $\boldsymbol{U} - \boldsymbol{V}$ is a positive semi-definite symmetric matrix. The matrices $\bar{\boldsymbol{A}}_\alpha$, $\bar{\boldsymbol{B}}_\alpha$, and $\bar{\boldsymbol{W}}_\alpha$ are defined as follows ($\boldsymbol{I}$ is the unit matrix):

$$\bar{\boldsymbol{A}}_\alpha = -(\bar{\boldsymbol{R}}\bar{\boldsymbol{x}}_\alpha) \times \boldsymbol{I}, \qquad \bar{\boldsymbol{B}}_\alpha = (\bar{\boldsymbol{t}}_\alpha - \boldsymbol{r}_\alpha, \bar{\boldsymbol{R}}_\alpha \bar{\boldsymbol{x}}_\alpha)\boldsymbol{I} - \bar{\boldsymbol{R}}\bar{\boldsymbol{x}}_\alpha(\bar{\boldsymbol{t}} - \boldsymbol{r}_\alpha)^\top, \tag{12}$$

$$\bar{\boldsymbol{W}}_\alpha = \Big((\bar{\boldsymbol{t}} - \boldsymbol{r}_\alpha) \times \bar{\boldsymbol{R}} V_0[\boldsymbol{x}_\alpha] \bar{\boldsymbol{R}}^\top \times (\bar{\boldsymbol{t}} - \boldsymbol{r}_\alpha)\Big)^-. \tag{13}$$

Throughout this paper, the inner product of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted by $(\boldsymbol{u}, \boldsymbol{v})$. The product $\boldsymbol{v} \times \boldsymbol{U}$ of a vector $\boldsymbol{v}$ and a matrix $\boldsymbol{U}$ is the matrix whose columns are the vector products of $\boldsymbol{v}$ and the columns of $\boldsymbol{U}$. The product $\boldsymbol{U} \times \boldsymbol{v}$ of a matrix $\boldsymbol{U}$ and a vector $\boldsymbol{v}$ is the matrix whose rows are the vector products of the rows of $\boldsymbol{U}$ and vector $\boldsymbol{v}$. The operation $(\,\cdot\,)^-$ designates the (Moore-Penrose) generalized inverse.

4 Optimal Estimation

Applying the general theory of Kanatani [6], we can obtain a computational scheme for solving Problem 2 in such a way that the resulting solution attains the accuracy bound (11) in the first order (i.e., ignoring terms of $O(\epsilon^4)$): we minimize the sum of squared *Mahalanobis distances*

$$J = \sum_{\alpha=1}^N (\bar{\boldsymbol{x}}_\alpha - \boldsymbol{x}_\alpha, V_0[\boldsymbol{x}_\alpha]^-(\bar{\boldsymbol{x}}_\alpha - \boldsymbol{x}_\alpha)) \tag{14}$$

with respect to $\{\bar{\boldsymbol{x}}_\alpha\}$ subject to the constraint (7). The solution is given by

$$\bar{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - V_0[\boldsymbol{x}_\alpha]\boldsymbol{R}^\top \Big((\boldsymbol{t} - \boldsymbol{r}_\alpha) \times \boldsymbol{W}_\alpha \times (\boldsymbol{t} - \boldsymbol{r}_\alpha)\Big)\boldsymbol{R}\boldsymbol{x}_\alpha, \tag{15}$$

$$\boldsymbol{W}_\alpha = \left((\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\boldsymbol{R}V_0[\boldsymbol{x}_\alpha]\boldsymbol{R}^\top\times(\boldsymbol{t}-\boldsymbol{r}_\alpha)\right)_2^-, \tag{16}$$

where the operation $(\cdot)_r^-$ designates the *rank-constrained* (Moore-Penrose) generalized inverse computed by transforming it into the canonical form, replacing its eigenvalues except the r largest ones by 0, and computing the (Moore-Penrose) generalized inverse (this operation is necessary for preventing numerical instability [6]).

Substituting eq. (15) into eq. (14), we obtain the following expression to be minimized with respect to the motion parameters $\{\boldsymbol{t},\ \boldsymbol{R}\}$ alone:

$$J = \sum_{\alpha=1}^{N}((\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\boldsymbol{R}\boldsymbol{x}_\alpha, \boldsymbol{W}_\alpha\left((\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\boldsymbol{R}\boldsymbol{x}_\alpha\right)). \tag{17}$$

The unknown noise level ϵ can be estimated *a posteriori*. Let $\hat{J}$ be the *residual*, i.e., the minimum of J. Since $\hat{J}/\epsilon^2$ is subject to a χ^2 distribution with $2N-6$ degrees of freedom in the first order [6], we obtain an unbiased estimator of the squared noise level ϵ^2 in the following form:

$$\hat{\epsilon}^2 = \frac{\hat{J}}{2N-6}. \tag{18}$$

Because the solution $\{\hat{\boldsymbol{t}},\ \hat{\boldsymbol{R}}\}$ of the minimization (17) attains the accuracy bound (11) in the first order, we can evaluate their covariance matrices by optimally estimating the true positions $\{\bar{\boldsymbol{x}}_\alpha\}$ (we discuss this in the next section) and substituting the solution $\{\hat{\boldsymbol{t}},\ \hat{\boldsymbol{R}}\}$ and the estimator (18) for their true values $\{\bar{\boldsymbol{t}},\ \bar{\boldsymbol{R}}\}$ and ϵ^2 in eqs. (11). Using the covariance matrix $V[\hat{\boldsymbol{t}}]$ in eqs. (11), we can estimate the probability distribution of the current location in the following form:

$$p(\boldsymbol{r}) = \frac{1}{(2\pi|V[\hat{\boldsymbol{t}}]|)^{3/2}}e^{-(\boldsymbol{r}-\hat{\boldsymbol{t}},V[\hat{\boldsymbol{t}}]^{-1}(\boldsymbol{r}-\hat{\boldsymbol{t}}))/2}. \tag{19}$$

We conduct the minimization (17) by modified Newton iterations. If rotation $\boldsymbol{R}$ is perturbed by a small rotation represented by the vector $\Delta\boldsymbol{\Omega}$ defined by eq. (9), the perturbed rotation has the expression

$$\boldsymbol{R}+\Delta\boldsymbol{\Omega}\times\boldsymbol{R}+\frac{1}{2}\Delta\boldsymbol{\Omega}\Delta\boldsymbol{\Omega}^\top\boldsymbol{R}-\frac{1}{2}\|\Delta\boldsymbol{\Omega}\|^2\boldsymbol{R}+O(\Delta\boldsymbol{\Omega})^3, \tag{20}$$

where $\|\boldsymbol{u}\|$ denotes the norm of a vector $\boldsymbol{u}$ and $O(\boldsymbol{u},\boldsymbol{v},...)^k$ designates terms of order k or higher in the elements of vectors $\boldsymbol{u}$, $\boldsymbol{v}$, ... Substituting eq. (20) and $\boldsymbol{t}+\Delta\boldsymbol{t}$ for $\boldsymbol{R}$ and $\boldsymbol{t}$, respectively, in eq. (17) and expanding it with respect to $\Delta\boldsymbol{t}$ and $\boldsymbol{\Omega}$, we obtain the following expression:

$$\begin{aligned} &J+(\nabla_{\boldsymbol{t}}J,\Delta\boldsymbol{t})+(\nabla_{\boldsymbol{R}}J,\Delta\boldsymbol{\Omega})+\frac{1}{2}(\Delta\boldsymbol{t},\nabla^2_{\boldsymbol{t}\boldsymbol{t}}J,\Delta\boldsymbol{t})\\ &\quad+(\Delta\boldsymbol{t},\nabla^2_{\boldsymbol{t}\boldsymbol{R}}J,\Delta\boldsymbol{\Omega})+\frac{1}{2}(\Delta\boldsymbol{\Omega},\nabla^2_{\boldsymbol{R}\boldsymbol{R}}J,\Delta\boldsymbol{\Omega})+O(\Delta\boldsymbol{t},\Delta\boldsymbol{\Omega})^3. \end{aligned} \tag{21}$$

Differentiating this with respect to $\Delta\boldsymbol{t}$ and $\Delta\boldsymbol{\Omega}$, letting the resulting expressions equal zero, and ignoring terms of $O(\Delta\boldsymbol{t}, \Delta\boldsymbol{\Omega})^3$, we obtain the following simultaneous linear equations:

$$\begin{pmatrix} \nabla^2_{\boldsymbol{tt}} J & \nabla^2_{\boldsymbol{t}\boldsymbol{R}} J \\ (\nabla^2_{\boldsymbol{t}\boldsymbol{R}} J)^\top & \nabla^2_{\boldsymbol{R}\boldsymbol{R}} J \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{t} \\ \Delta\boldsymbol{\Omega} \end{pmatrix} = - \begin{pmatrix} \nabla_{\boldsymbol{t}} J \\ \nabla_{\boldsymbol{R}} J \end{pmatrix}. \tag{22}$$

Starting from an initial guess $\{\boldsymbol{t}, \boldsymbol{R}\}$, we solve eq. (22) for the increments $\{\Delta\boldsymbol{t}, \Delta\boldsymbol{\Omega}\}$ and update the solution in the form $\boldsymbol{t} \leftarrow \boldsymbol{t} + \Delta\boldsymbol{t}$ and $\boldsymbol{R} \leftarrow \mathcal{R}(\Delta\boldsymbol{\Omega})\boldsymbol{R}$, where $\mathcal{R}(\Delta\boldsymbol{\Omega})$ designates the rotation matrix by angle $\|\Delta\boldsymbol{\Omega}\|$ around axis $\Delta\boldsymbol{\Omega}$:

$$\mathcal{R}(\Delta\boldsymbol{\Omega}) = \cos\Delta\Omega\boldsymbol{I} + (1 - \cos\Delta\Omega)\boldsymbol{l}\boldsymbol{l}^\top + \sin\Delta\Omega\boldsymbol{l} \times \boldsymbol{I}. \tag{23}$$

We iterate this until $\|\Delta\boldsymbol{t}\| < \epsilon_{\boldsymbol{t}}$ and $\|\Delta\boldsymbol{\Omega}\| < \epsilon_{\boldsymbol{R}}$ for specified thresholds $\epsilon_{\boldsymbol{t}}$ and $\epsilon_{\boldsymbol{R}}$.

We compute the initial guess $\{\boldsymbol{t}, \boldsymbol{R}\}$ by a structure-from-motion algorithm. First, we hypothetically place a reference camera coordinate system in a known position in the world model and compute the image coordinates of the feature points viewed from that position (we need not actually generate a graphics image). From the correspondences of image coordinates between this reference image and the actually observed image, we can reconstruct the 3-D motion of the camera and the 3-D positions of the feature points up to scale; since we know the absolute positions of the feature points, we can easily adjust the scale a posteriori. Here, we adopt the statistically optimal algorithm of Kanatani [5] using a technique called *renormalization*.

5 Example 1

Fig. 2(a) is a real image of a toy house. We manually input the feature points marked by white dots and used the noise model of eqs. (5). The computation converged after five iterations for thresholds $\epsilon_{\boldsymbol{t}} = 0.01$cm (the height of the house is 8cm) and $\epsilon_{\boldsymbol{R}} = 0.01°$. Fig. 2(b) displays the house and the estimated camera coordinate axes viewed from an angle.

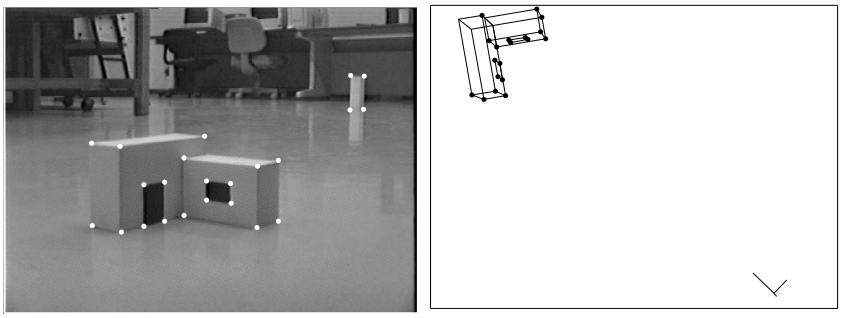
We evaluated the reliability of the computed solution $\{\hat{\boldsymbol{t}}, \hat{\boldsymbol{R}}\}$ in the following two ways:

- Theoretical analysis.
- Random noise simulation.

The former is straightforward: since our method attains the accuracy bound (11) in the first order, we can evaluate the reliability of the solution by approximating the true values by their estimates in eq. (11).

A well known method for the latter is *bootstrap* [3], which can be applied to any solution method. Here, we adopt the following procedure. We first optimally correct the observed positions $\{\boldsymbol{x}_\alpha\}$ into $\{\hat{\boldsymbol{x}}_\alpha\}$ so that constraint (7) exactly holds. From eq. (15), this optimal correction is done as follows:

$$\hat{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - V_0[\boldsymbol{x}_\alpha]\hat{\boldsymbol{R}}^\top \Big((\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha) \times \hat{\boldsymbol{W}}_\alpha \times (\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha) \Big) \hat{\boldsymbol{R}}\boldsymbol{x}_\alpha, \tag{24}$$



(a) (b)

Fig. 2. (a) A real image of a toy house. (b) Estimated current location.

$$\hat{\boldsymbol{W}}_\alpha = \Big((\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha) \times \hat{\boldsymbol{R}} V_0[\boldsymbol{x}_\alpha] \hat{\boldsymbol{R}}^\top \times (\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha) \Big)_2^- . \tag{25}$$

Estimating the noise variance by eq. (18), we generate random Gaussian noise that has the estimated variance and add it to the corrected positions independently. Then, we compute the motion parameters $\{\boldsymbol{t}^*, \boldsymbol{R}^*\}$ and the angle $\Delta\Omega^*$ and axis $\boldsymbol{l}^*$ of the relative rotation $\hat{\boldsymbol{R}}^* \hat{\boldsymbol{R}}^\top$.

Fig. 3 shows three-dimensional plots of the error vectors $\Delta\boldsymbol{t}^* = \boldsymbol{t}^* - \hat{\boldsymbol{t}}$ and $\Delta\boldsymbol{\Omega}^* = \Delta\Omega^*\boldsymbol{l}^*$ for 100 trials. The ellipsoids in the figures are respectively defined by

$$(\Delta\boldsymbol{t}^*, V[\hat{\boldsymbol{t}}]^{-1}\Delta\boldsymbol{t}^*) = 1, \qquad (\Delta\boldsymbol{\Omega}^*, V[\hat{\boldsymbol{R}}]^{-1}\Delta\boldsymbol{\Omega}^*) = 1, \tag{26}$$

where $V[\hat{\boldsymbol{t}}]$ and $V[\hat{\boldsymbol{R}}]$ are computed by approximating $\bar{\boldsymbol{R}}$, $\{\bar{\boldsymbol{x}}_\alpha\}$, and ϵ^2 by $\hat{\boldsymbol{R}}$, $\{\hat{\boldsymbol{x}}_\alpha\}$, and $\hat{\epsilon}^2$ on the right-hand side of eq. (11). These ellipsoids indicate the standard deviation of the errors in each orientation [6]. The cubes in the figures are displayed as a reference.

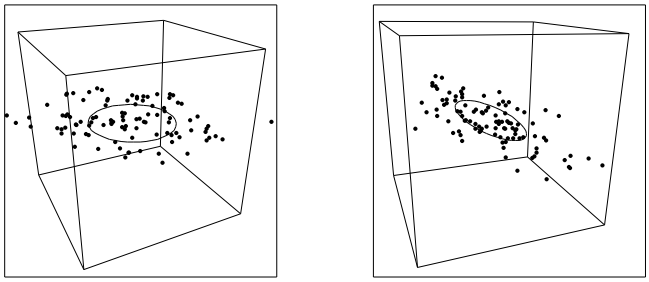
We compared our method with the naive least-squares method; we simply replaced the matrix $\boldsymbol{W}_\alpha$ by the unit matrix $\boldsymbol{I}$. Fig. 4 shows the result that corresponds to Fig. 3 (the ellipsoids and the cubes are the same as in Fig. 3).

Comparing Figs. 3 and 4, we can confirm that our method improves the accuracy of the solution as compared with the least-squares method. We can also see that errors for our method distribute around the ellipsoids, indicating that our method already attains the theoretical accuracy bound; no further improvement is possible.

The above visual observation can be given quantitative measures. We define the *bootstrap standard deviations* by

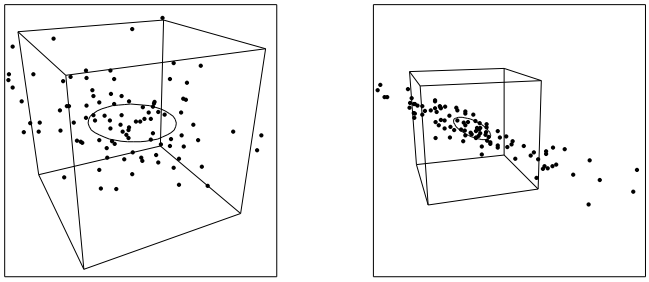
$$S^*_{\boldsymbol{t}} = \sqrt{\frac{1}{B}\sum_{b=1}^{B} \|\Delta\boldsymbol{t}^*_b\|^2}, \qquad S^*_{\boldsymbol{R}} = \sqrt{\frac{1}{B}\sum_{b=1}^{B} (\Delta\Omega^*_b)^2}, \tag{27}$$

where B is the number of bootstrap samples and the subscript b labels each sample. The corresponding standard deviations for the (estimated) theoretical



(a) (b)

Fig. 3. Bootstrap errors (our method): (a) translation; (b) rotation.



(a) (b)

Fig. 4. Bootstrap errors (least squares): (a) translation; (b) rotation.

lower bound are

$$S_{\boldsymbol{t}} = \sqrt{\mathrm{tr}V[\hat{\boldsymbol{t}}]}, \qquad S_{\boldsymbol{R}} = \sqrt{\mathrm{tr}V[\hat{\boldsymbol{R}}]}, \tag{28}$$

respectively. Table 1 lists the values of $S^*_{\boldsymbol{t}}$ and $S^*_{\boldsymbol{R}}$ for our method and the least-squares method ($B = 1000$) together with their theoretical lower bounds $S_{\boldsymbol{t}}$ and $S_{\boldsymbol{R}}$. We can see from this that our method is indeed superior to the least-squares method and that the accuracy of our solution is very close to the theoretical lower bound.

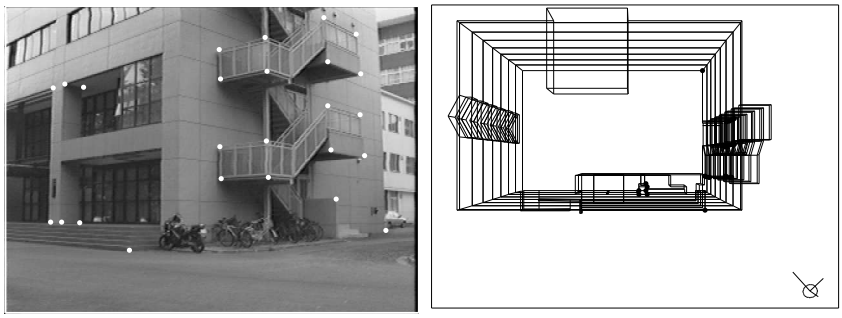
This observation confirms that we can evaluate the probability distribution of the estimated location by (approximately) evaluating the theoretical accuracy bound given by eq. (11).

6 Example 2

Fig. 5(a) is a real image of a real building for which a design plan is available. We manually input the feature points marked by white dots and used the noise model of eq. (5). The computation converged after four iterations for thresholds $\epsilon_{\boldsymbol{t}} = 0.1$cm and $\epsilon_{\boldsymbol{R}} = 0.01°$.

Table 1. Bootstrap standard deviations and the theoretical lower bounds.

	Translation	Rotation
Our method	0.16cm	0.16°
Least squares	0.25cm	0.45°
Lower bounds	0.16cm	0.15°



(a) (b)

Fig. 5. (a) A real image of a real building. (b) Estimated current location and its reliability.

Fig. 5(b) displays the building and the estimated camera coordinate axes viewed from above; the ellipse in the figure indicates the ellipsoid corresponding to those in Figs. 3(a) and 4(a) enlarged by three times. We also evaluated the reliability of the solution by both theoretical analysis and bootstrap. Table 2 is the result corresponding to Table 1. We can again confirm that our method is superior to the least-squares method and that our method almost attains the theoretical bound, which can be viewed as describing the probability distribution of the estimated location.

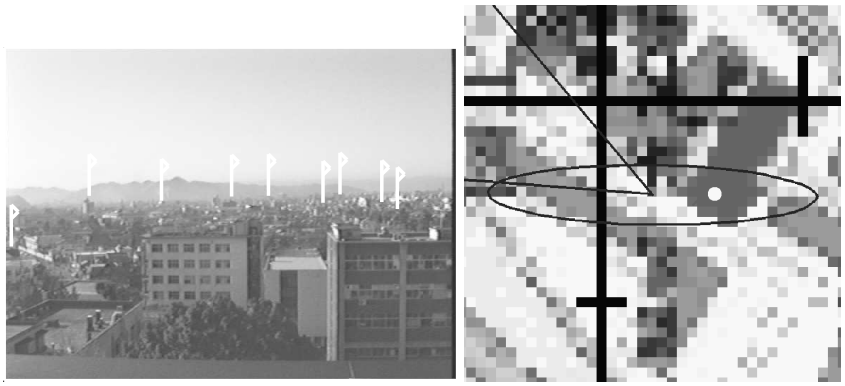
7 Example 3

If the robot is constrained to be on a horizontal plane, the computation is considerably simplified. Fig. 6(a) is a real image of a city scene. We manually spotted nine features at the bottoms of the white vertical bars in the figure and computed the viewer location by matching the positions of the bars to their corresponding locations in the city map. The initial guess was computed by the method of circle geometry [7, 9]; the computation converged after five iterations for thresholds $\epsilon_{\boldsymbol{t}} = 0.01$m and $\epsilon_{\boldsymbol{R}} = 0.01°$.

Fig. 6(b) shows the estimated current location superimposed on the city map. The ellipse in the figure is the two-dimensional version of the ellipsoids in Figs. 3(a) and 4(a). The white dot indicates the place where we actually took

Table 2. Bootstrap standard deviations and the theoretical lower bounds.

	Translation	Rotation
Our method	44.1cm	1.31°
Least squares	47.3cm	1.39°
Lower bounds	43.7cm	1.29°



(a) (b)

Fig. 6. (a) A real image of a city scene. (b) Estimated current location.

the picture of Fig. 6(a), and it is within the ellipse. Fig. 7 shows the angle of view from the estimated location superimposed on the city map; the locations of the feature points are marked by white dots.

Figs. 8(a) and (b) show 100 bootstrap errors in the estimated location plotted in the same way as Figs. 3(a) and 4(b) (we omit errors in rotation; they are very small). Table 3 corresponds to Tables 1 and 2 (this time $B = 10000$); our method is still superior to the least-squares method, although the difference is not so marked as in the three-dimensional case. At any rate, our method almost attains the theoretical bound, which can be viewed as describing the probability distribution of the estimated location.

8 Concluding Remarks

We have discussed optimal estimation of the current location of a robot by matching an image of the scene taken by the robot with the model of the environment. We have first presented a theoretical accuracy bound defined independently of solution techniques and then given a method that attains it; our method is truly "optimal" in that sense. Since the solution attains the accuracy bound, we can view it as describing the probability distribution of the estimated location; the computation does not require any knowledge about the noise magnitude. Using real images, we have demonstrated that our method is superior to

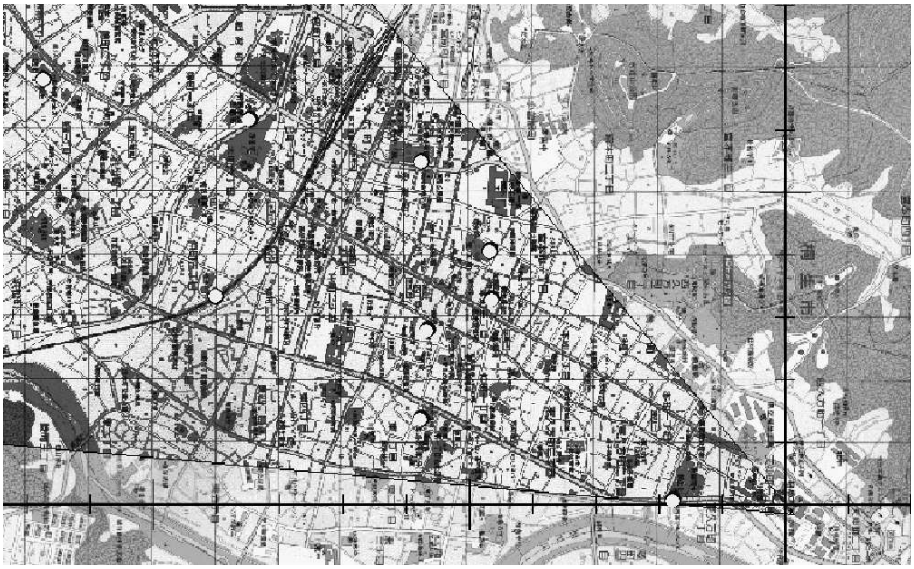
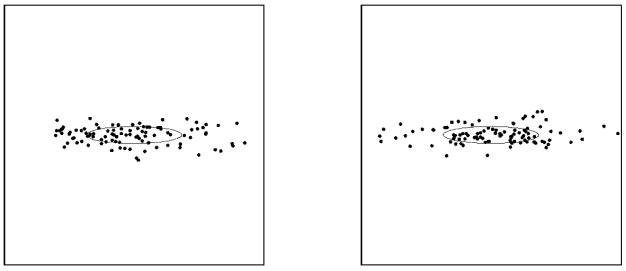


Fig. 7. Estimated angle of view.



(a) (b)

Fig. 8. Bootstrap errors in the estimated location: (a) our method; (b) least squares.

the naive least-squares method. We have also confirmed the theoretical predictions of our theory by applying the bootstrap procedure.

References

1. N. Ayache and O. D. Faugeras, "Building, registrating, and fusing noisy visual maps," *Int. J. Robotics Research*, **7**-6 (1988), 45–65.
2. M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Trans. Robotics Automation*, **13**-2 (1997), 251–263.
3. B. Efron and R. J. Tibshirani, *An Introduction to Bootstrap*, Chapman-Hall, New York, 1993.

Table 3. Bootstrap standard deviations and the theoretical lower bounds.

	Translation	Rotation
Our method	37.1m	0.78°
Least squares	37.9m	0.79°
Lower bounds	37.3m	0.78°

4. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, **24**-6 (1981), 381–395.
5. K. Kanatani, "Renormalization for motion analysis: Statistically optimal algorithm," *IEICE Trans. Inf. & Syst.*, **E77-D**-11 (1994), 1233–1239.
6. K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam 1996.
7. K. Sugihara, "Some location problems for robot navigation using a single camera," *Comput. Vis. Gr. Image Process.*, **42** (1988), 112–129.
8. R. E. Suorsa and B. Sridhar, "A parallel implementation of a multisensor feature-based range-estimation method," *IEEE Trans. Robotics Automation*, **10**-6 (1994), 755–768.
9. K. T. Sutherland and W. B. Thompson, "Localizing in unconstrained environment: Dealing with the errors," *IEEE Trans. Robotics Automation*, **10**-6 (1994), 740–754.
10. R. Talluri and J. K. Aggarwal, "Mobile robot self-location using model-image feature correspondence," *IEEE Trans. Robotics Automation*, **12**-1 (1996), 63–77.
11. Y. Yagi, Y. Nishimitsu and M. Yachida, "Map-based navigation for a mobile robot with ominidirectional image sensor COPIS," *IEEE Trans. Robotics Automation*, **11**-5 (1995), 634–648.

This article was processed using the LaTeX macro package with LLNCS style