Machine Vision and Applications Manuscript-Nr.
(will be inserted by hand later)

Accuracy bounds and optimal computation of robot localization

Kenichi Kanatani[1], **Naoya Ohta**[2]

[1] Department of Information Technology, Okayama University, Okayama 700-8530, Japan
[2] Department of Computer Science, Gunma University, Kiryu, Gunma 376-8515, Japan

Received: 17 March 1998 / Accepted: 7 April 2001

Abstract. We present an optimal method for estimating the current location of a mobile robot by matching an image of the scene taken by the robot with the model of the known environment. We first derive a theoretical accuracy bound and then give a computational scheme that can attain 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.

Key words: Mobile robot – Localization – Statistical estimation – Theoretical accuracy bound – Bootstrap

1 Introduction

For a mobile robot to navigate autonomously, it must have a geometric model of the environment. This may be given as data or constructed by the robot itself using vision and sensor data. Here we consider the case in which a robot already has a three-dimensional (3-D) map of the environment and study the problem of identifying its current location relative to 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 (Suorsa and Sridhar 1994). However, the accuracy of the computed location quickly deteriorates as errors (due to slippages of the wheels, vibrations of the camera, etc.) accumulate in the course of the navigation. At some point, therefore, the current location needs to be estimated by some direct means.

A typical method for localization is to compute the current camera position by matching feature points 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 (Ayache and Faugeras 1988). 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 that are 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.

Computing the 3-D relationship between image and model features has been previously studied by many researchers as the problem called "PnP", in which the 3-D positions of the feature points are computed 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 (Sugihara 1988). It can also be applied to three-dimensional motion by replacing circles by spheres (Sutherland and Thompson 1994). 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 (Fischler and Bolles 1981). For a given match, however, it is obviously better to fuse all available information in an optimal manner. Such a method also exists (Betke and Gurvits 1997), but so far the main concern has been *methods* for estimation with little attention being given on *theoretical optimality* and *reliability of the solution*.

The aim of this paper is *not* to propose a new method that performs appreciably better than existing ones. Rather, we focus on theoretically guaranteeing optimality *a priori*. Introducing a model of noise, and viewing the problem as a *statistical estimation*, we first derive a *theoretical accuracy bound* independently of particular solution techniques. Then we present a computational scheme that can attain that bound; such a method alone can be called "optimal" in the sense that *no other method could possibly outperform it*.

We can view the bound as describing the "probability distribution" of the current location by using Gaussian approximation. This information can be obtained without any knowledge about the magnitude of image noise. We confirm the theoretical predictions of our theory by

[0] *Correspondence to*: K. Kanatani

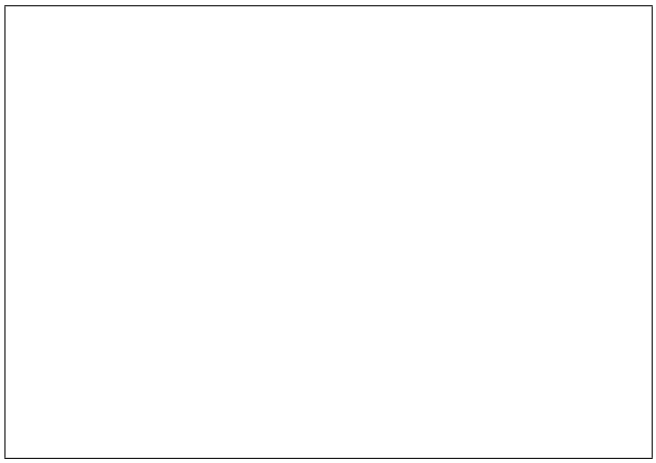


Fig. 1. Camera imaging geometry

using real images and applying the bootstrap procedure (Efron and Tibshirani 1993). We also discuss many practical issues for real implementation of our method.

2 Imaging geometry and noise model

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). Letting f be the 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 represent a point with image coordinates (x, y) by the following three-dimensional vector:

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

This vector indicates the line of sight starting from the camera coordinate origin and passing through the corresponding point in the scene (Fig. 1).

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

$$x_\alpha = \bar{x}_\alpha + \Delta x_\alpha, \qquad y_\alpha = \bar{y}_\alpha + \Delta y_\alpha. \tag{2}$$

We model the errors Δx_α and Δy_α as (generally correlated) Gaussian random variables with zero mean, independently for each α. 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 zero, the covariance matrix $V[\boldsymbol{x}_\alpha]$ is singular; its third row and third column consist of zeros.

The covariance matrix $V[\boldsymbol{x}_\alpha]$ measures the uncertainty of detecting the feature point $\boldsymbol{x}_\alpha$, but in practice it is very difficult to predict it precisely. Here we assume

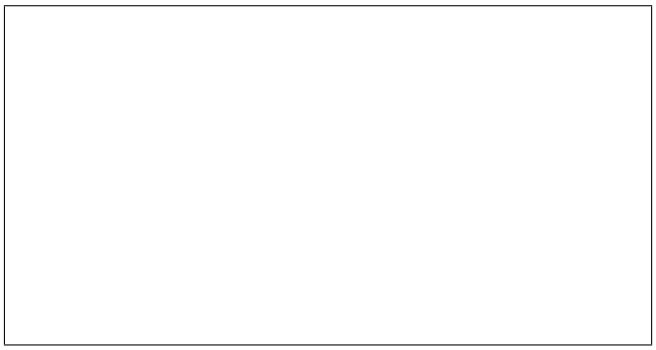


Fig. 2. The camera coordinate system and the world coordinate system

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 that the constant ϵ is unknown (see the discussions in Sect.10.4); we call ϵ the *noise level*, and $V_0[\boldsymbol{x}_\alpha]$ the *normalized covariance matrix* (Kanatani 1996).

If Δx_α and Δy_α are subject to an isotropic and identical Gaussian distribution of mean zero and standard deviation σ, 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.

3 Statistical robot localization

Suppose the camera coordinate system is in a position defined by translating the world coordinate system by $\boldsymbol{t}$ and rotating it by $\mathbf{R}$ with respect to the world O_w-$X_0Y_0Z_0$ coordinate system (Fig. 2). We call $\{\boldsymbol{t}, \mathbf{R}\}$ 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, are known with respect to the world coordinate system, optimally compute the motion parameters $\{\boldsymbol{t}, \mathbf{R}\}$ and their probability distribution.

The vector $\bar{\boldsymbol{x}}_\alpha$ representing the true position of the αth feature point is defined with respect to the camera coordinate system. If it is described with respect to the world coordinate system, it becomes $\mathbf{R}\bar{\boldsymbol{x}}_\alpha$ (Fig. 2). 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:

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

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

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

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

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

A naive but widely adopted approach to solving this type of estimation is the *least-squares method*, minimizing the square sum of the constraint:

$$\sum_{\alpha=1}^{N} \|(\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\mathbf{R}\bar{\boldsymbol{x}}_\alpha\|^2 \to \min. \tag{8}$$

We will show that this method is not optimal: the accuracy of the solution is inferior to the method that we going to describe.

4 Theoretical accuracy bound

Let $\{\hat{\boldsymbol{t}}, \hat{\mathbf{R}}\}$ be an estimator of the true motion parameters $\{\bar{\boldsymbol{t}}, \bar{\mathbf{R}}\}$ obtained by some means. Since translations form an Abelian group under addition, the deviation of a translation can be measured by the "difference"

$$\Delta\boldsymbol{t} = \hat{\boldsymbol{t}} - \bar{\boldsymbol{t}} \tag{9}$$

of the estimator $\hat{\boldsymbol{t}}$ from its true value $\bar{\boldsymbol{t}}$. Rotations, on the other hand, form a group denoted by SO(3) (*special orthogonal group*) under matrix multiplication. So, the deviation of a rotation can be measured by the "quotient" $\hat{\mathbf{R}}\bar{\mathbf{R}}^\top$, i.e., the rotation of $\hat{\mathbf{R}}$ relative to $\bar{\mathbf{R}}$. Let $\boldsymbol{l}$ (unit vector) and $\Delta\Omega$ be, respectively, the axis and angle of the relative rotation $\hat{\mathbf{R}}\bar{\mathbf{R}}^\top$, and define

$$\Delta\boldsymbol{\Omega} = \Delta\Omega\boldsymbol{l}. \tag{10}$$

We define the covariance matrices of the estimator $\{\hat{\boldsymbol{t}}, \hat{\mathbf{R}}\}$ as follows (Kanatani 1996; Kanatani and Morris 2001):

$$V[\hat{\boldsymbol{t}}] = E[\Delta\boldsymbol{t}\Delta\boldsymbol{t}^\top], \quad V[\hat{\boldsymbol{t}}, \hat{\mathbf{R}}] = E[\Delta\boldsymbol{t}\Delta\boldsymbol{\Omega}^\top],$$
$$V[\hat{\mathbf{R}}, \hat{\boldsymbol{t}}] = E[\Delta\boldsymbol{\Omega}\Delta\boldsymbol{t}^\top], \quad V[\hat{\mathbf{R}}] = E[\Delta\boldsymbol{\Omega}\Delta\boldsymbol{\Omega}^\top]. \tag{11}$$

Applying the general theory of statistical optimization (Kanatani 1996), we can obtain the following lower bound:

$$\begin{pmatrix} V[\hat{\boldsymbol{t}}] & V[\hat{\boldsymbol{t}}, \hat{\mathbf{R}}] \\ V[\hat{\mathbf{R}}, \hat{\boldsymbol{t}}] & V[\hat{\mathbf{R}}] \end{pmatrix} \succ \epsilon^2 \begin{pmatrix} \sum_{\alpha=1}^N \bar{\mathbf{A}}_\alpha^\top \bar{\mathbf{W}}_\alpha \bar{\mathbf{A}}_\alpha & \sum_{\alpha=1}^N \bar{\mathbf{A}}_\alpha^\top \bar{\mathbf{W}}_\alpha \bar{\mathbf{B}}_\alpha \\ \sum_{\alpha=1}^N \bar{\mathbf{B}}_\alpha^\top \bar{\mathbf{W}}_\alpha \bar{\mathbf{A}}_\alpha & \sum_{\alpha=1}^N \bar{\mathbf{B}}_\alpha^\top \bar{\mathbf{W}}_\alpha \bar{\mathbf{B}}_\alpha \end{pmatrix}^{-1}. \tag{12}$$

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

$$\bar{\mathbf{A}}_\alpha = -(\bar{\mathbf{R}}\bar{\boldsymbol{x}}_\alpha)\times\mathbf{I}, \tag{13}$$

$$\bar{\mathbf{B}}_\alpha = (\bar{\boldsymbol{t}}_\alpha - \boldsymbol{r}_\alpha, \bar{\mathbf{R}}_\alpha\bar{\boldsymbol{x}}_\alpha)\mathbf{I} - \bar{\mathbf{R}}\bar{\boldsymbol{x}}_\alpha(\bar{\boldsymbol{t}}-\boldsymbol{r}_\alpha)^\top, \tag{14}$$

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

Throughout this paper, the inner product of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted by $(\boldsymbol{u}, \boldsymbol{v})$. The product $\boldsymbol{v}\times\mathbf{U}$ of a vector $\boldsymbol{v}$ and a matrix $\mathbf{U}$ is the matrix whose columns are the vector products of $\boldsymbol{v}$ and the columns of $\mathbf{U}$. The product $\mathbf{U}\times\boldsymbol{v}$ of a matrix $\mathbf{U}$ and a vector $\boldsymbol{v}$ is the matrix whose rows are the vector products of the rows of $\mathbf{U}$ and vector $\boldsymbol{v}$. The operation $(\,\cdot\,)^-$ designates the (Moore-Penrose) generalized inverse (Kanatani 1996).

5 Optimal estimation

Applying the general theory of statistical optimization (Kanatani 1996), we can obtain a computational scheme for solving Problem 2 in such a way that the resulting solution attains the accuracy bound (12) in the first order (i.e., ignoring terms of $O(\epsilon^4)$): we minimize the sum of squared *Mahalanobis distances*

$$J = \sum_{\alpha=1}^{N} (\bar{\boldsymbol{x}}_\alpha - \boldsymbol{x}_\alpha, V_0[\boldsymbol{x}_\alpha]^-(\bar{\boldsymbol{x}}_\alpha - \boldsymbol{x}_\alpha)) \tag{16}$$

with respect to $\{\bar{\boldsymbol{x}}_\alpha\}$ subject to the constraint (7). The solution is given as follows (Kanatani 1996):

$$\bar{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - V_0[\boldsymbol{x}_\alpha]\mathbf{R}^\top\left((\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\mathbf{W}_\alpha\times(\boldsymbol{t}-\boldsymbol{r}_\alpha)\right)\mathbf{R}\boldsymbol{x}_\alpha, \tag{17}$$

$$\mathbf{W}_\alpha = \Big((\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\mathbf{R}V_0[\boldsymbol{x}_\alpha]\mathbf{R}^\top\times(\boldsymbol{t}-\boldsymbol{r}_\alpha)\Big)_2^-. \tag{18}$$

Here 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 (Kanatani 1996).

Substituting (17) into (16), we obtain the following expression to minimize with respect to the motion parameters $\{\boldsymbol{t}, \mathbf{R}\}$ alone:

$$J = \sum_{\alpha=1}^{N} ((\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\mathbf{R}\boldsymbol{x}_\alpha, \mathbf{W}_\alpha\left((\boldsymbol{t}-\boldsymbol{r}_\alpha)\times\mathbf{R}\boldsymbol{x}_\alpha\right)) \tag{19}$$

Let $\hat{J}$ be the *residual*, i.e., the minimum of J. It can be shown that $\hat{J}/\epsilon^2$ is subject to a χ^2 distribution with $2N-6$ degrees of freedom in the first order (Kanatani 1996). Hence, we obtain an unbiased estimator of the squared noise level ϵ^2 *a posteriori* in the following form:

$$\hat{\epsilon}^2 = \frac{\hat{J}}{2N-6}. \tag{20}$$

The solution $\{\hat{\boldsymbol{t}}, \hat{\mathbf{R}}\}$ of the minimization (19) is known to attain the accuracy bound (12) in the first order (Kanatani 1996). Hence, we can evaluate the covariance matrix of the solution by optimally estimating the true positions $\{\bar{\boldsymbol{x}}_\alpha\}$ (see Sect. 7) and substituting the solution $\{\hat{\boldsymbol{t}}, \hat{\mathbf{R}}\}$ and the estimator $\hat{\epsilon}^2$ given by (20) for their true values $\{\bar{\boldsymbol{t}}, \bar{\mathbf{R}}\}$ and ϵ^2 in (12). Approximating the error distribution to be Gaussian, we can obtain the probability distribution of the current location using the covariance matrix $V[\hat{\boldsymbol{t}}]$ in (12) in the following form:

$$p(\boldsymbol{r}) = \frac{1}{(2\pi|V[\hat{\boldsymbol{t}}]|)^{3/2}} \mathrm{e}^{-(\boldsymbol{r}-\hat{\boldsymbol{t}}, V[\hat{\boldsymbol{t}}]^{-1}(\boldsymbol{r}-\hat{\boldsymbol{t}}))/2}. \tag{21}$$

6 Numerical optimization scheme

Equation (19) can be minimized by whatever numerical scheme, and the optimality of the solution holds irrespective of the minimization scheme used. We adopt here, among many possible alternatives, the modified Newton method, which is known to converge very quickly with

quadratic speed. It also has the advantage that the theoretical accuracy bound (12) is automatically evaluated as a byproduct, as we now show.

If a rotation $\mathbf{R}$ is perturbed by a small rotation represented by the vector $\Delta\boldsymbol{\Omega}$ defined by (10), the perturbed rotation has the expression

$$\mathbf{R} + \Delta\boldsymbol{\Omega} \times \mathbf{R} + \frac{1}{2}\Delta\boldsymbol{\Omega}\Delta\boldsymbol{\Omega}^\top\mathbf{R} - \frac{1}{2}\|\Delta\boldsymbol{\Omega}\|^2\mathbf{R} + O(\Delta\boldsymbol{\Omega})^3, \tag{22}$$

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 $\boldsymbol{u}$, $\boldsymbol{v}$, ... Replacing $\boldsymbol{t}$ by $\boldsymbol{t} + \Delta\boldsymbol{t}$ and $\mathbf{R}$ by (22) in (19), and expanding it with respect to $\Delta\boldsymbol{t}$ and $\boldsymbol{\Omega}$, we obtain the following expression:

$$\begin{aligned} &J + (\nabla_{\boldsymbol{t}} J, \Delta\boldsymbol{t}) + (\nabla_{\mathbf{R}} J, \Delta\boldsymbol{\Omega}) + \frac{1}{2}(\Delta\boldsymbol{t}, \nabla^2_{\boldsymbol{t}\boldsymbol{t}} J\Delta\boldsymbol{t}) \\ &+(\Delta\boldsymbol{t}, \nabla^2_{\boldsymbol{t}\mathbf{R}} J\Delta\boldsymbol{\Omega}) + \frac{1}{2}(\Delta\boldsymbol{\Omega}, \nabla^2_{\mathbf{R}\mathbf{R}} J\Delta\boldsymbol{\Omega}) \\ &+O(\Delta\boldsymbol{t}, \Delta\boldsymbol{\Omega})^3. \end{aligned} \tag{23}$$

The explicit expressions of $\nabla_{\boldsymbol{t}} J$, $\nabla_{\mathbf{R}} J$, $\nabla^2_{\boldsymbol{t}\boldsymbol{t}} J$, $\nabla^2_{\boldsymbol{t}\mathbf{R}} J$, and $\nabla^2_{\mathbf{R}\mathbf{R}} J$ are given in the appendix. Differentiating (23) 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:

$$\begin{pmatrix} \nabla^2_{\boldsymbol{t}\boldsymbol{t}} J & \nabla^2_{\boldsymbol{t}\mathbf{R}} J \\ (\nabla^2_{\boldsymbol{t}\mathbf{R}} J)^\top & \nabla^2_{\mathbf{R}\mathbf{R}} J \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{t} \\ \Delta\boldsymbol{\Omega} \end{pmatrix} = - \begin{pmatrix} \nabla_{\boldsymbol{t}} J \\ \nabla_{\mathbf{R}} J \end{pmatrix}. \tag{24}$$

Starting from an initial guess $\{\boldsymbol{t}, \mathbf{R}\}$, we solve (24) for the increments $\{\Delta\boldsymbol{t}, \Delta\boldsymbol{\Omega}\}$ and update the solution in the form

$$\boldsymbol{t} \leftarrow \boldsymbol{t} + \Delta\boldsymbol{t}, \qquad \mathbf{R} \leftarrow \mathcal{R}(\Delta\boldsymbol{\Omega})\mathbf{R}, \tag{25}$$

where $\mathcal{R}(\Delta\boldsymbol{\Omega})$ designates the rotation matrix by angle $\Delta\Omega$ $(= \|\Delta\boldsymbol{\Omega}\|)$ around axis $\boldsymbol{l}$ $(= \Delta\boldsymbol{\Omega}/\Delta\Omega)$ in the following form (the *Rodrigues formula*):

$$\mathcal{R}(\Delta\boldsymbol{\Omega}) = \cos\Delta\Omega\mathbf{I} + (1 - \cos\Delta\Omega)\boldsymbol{l}\boldsymbol{l}^\top + \sin\Delta\Omega\boldsymbol{l} \times \mathbf{I}. \tag{26}$$

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

After the iterations have converged, the matrix that appears on the left-hand side of (24) divided by ϵ^2 is called the *information matrix*, whose inverse is shown to coincide with the right-hand side of (12) if the true values $\{\bar{\boldsymbol{x}}_\alpha\}$ and $\{\bar{\boldsymbol{t}}, \bar{\mathbf{R}}\}$ are replaced by their estimates $\{\hat{\boldsymbol{x}}_\alpha\}$ and $\{\hat{\boldsymbol{t}}, \hat{\mathbf{R}}\}$ (Kanatani and Morris 2001). This means that our scheme allows us to not only compute an optimal solution but also evaluate its reliability at the same time.

We compute the initial guess $\{\boldsymbol{t}, \mathbf{R}\}$ by a *structure-from-motion* algorithm (Kanatani 1996; Hartley and Zisserman 2000). 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 correspondence between this hypothetical 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 (Kanatani 1994) using a technique called *renormalization* (Kanatani 1996).

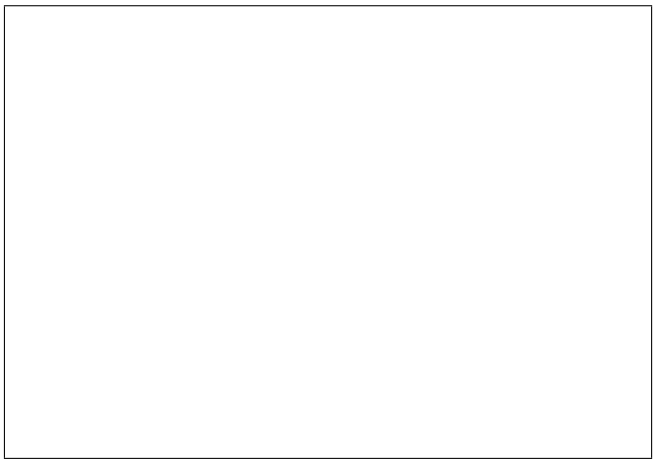


Fig. 3. An image of a toy house

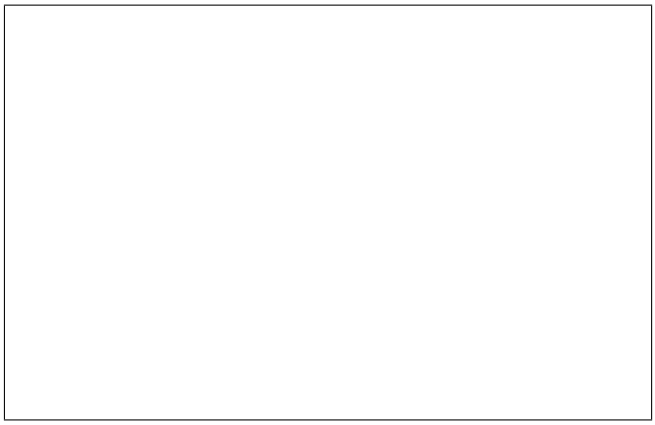
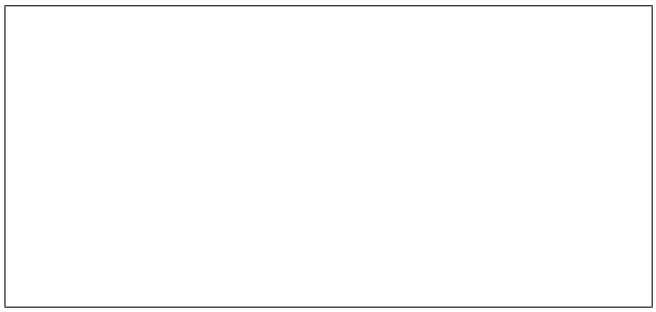


Fig. 4. Estimated current location

7 Example 1

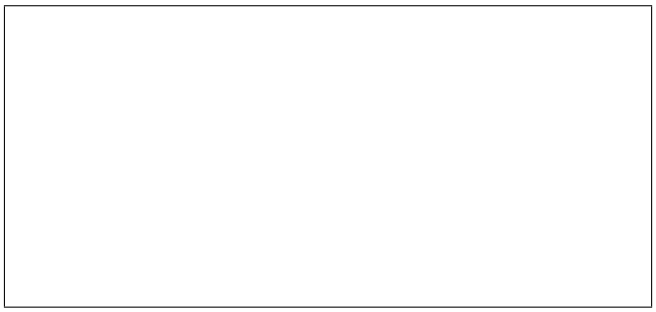
Figure 3 is a 640×480-pixel image of a toy house. The focal length was calibrated to be $f = 955$ pixels. We manually selected the feature points marked by white dots and assumed the noise model of (5). Their true 3-D positions relative to the world coordinate system, which was defined to coincide with the three orthogonal edges of the toy house, were mannually measured using a rule. The initial guess was computed by the optimal structure-from-motion algorithm (Kanatani 1994). The optimization converged after five iterations for thresholds $\varepsilon_{\boldsymbol{t}} = 0.01$cm (the height of the toy house is 8cm) and $\varepsilon_{\mathbf{R}} = 0.01°$. According to this, the initial camera position was displaced by about 16cm and rotated by about 4°. Figure 4 displays the toy house and the estimated camera coordinate axes viewed from above.

We evaluated the reliability of the computed solution $\{\hat{\boldsymbol{t}}, \hat{\mathbf{R}}\}$ in the following two ways: theoretical analysis and random noise simulation. The former is straightforward: since our method is known to attain the accuracy bound (12) in the first order, we can evaluate the reliability of the solution by approximating the true values by their estimates in (12). This computation can be done in the course of the minimization iterations as described earlier. The noise level estimated by (20) is $\hat{\epsilon} = 1.08 \times 10^{-3}$, which translates to $\sigma = 1.03$ pixels.



a **b**

Fig. 5. Bootstrap errors (our method): **a** translation; **b** rotation



a **b**

Fig. 6. Bootstrap errors (least squares): **a** translation; **b** rotation

An alternative method for reliability evaluation is *bootstrap* (Efron and Tibshirani 1993). This computation can be applied to any solution method *without knowing the ground truth*. Tentatively assuming that the computed solution is true, we correct the data so that they are exactly compatible with that solution. Then we add artificial noise to the modified data and observe how the solution fluctuates around the original value.

The actual procedure goes as follows. We first optimally correct the observed positions $\{\boldsymbol{x}_\alpha\}$ into $\{\hat{\boldsymbol{x}}_\alpha\}$ in such a way that constraint (7) exactly holds. From (17), this correction is done as follows:

$$\hat{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - V_0[\boldsymbol{x}_\alpha]\hat{\mathbf{R}}^\top \Big((\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha) \times \hat{\mathbf{W}}_\alpha \times (\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha)\Big) \hat{\mathbf{R}}\boldsymbol{x}_\alpha, \tag{27}$$

$$\hat{\mathbf{W}}_\alpha = \Big((\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha) \times \hat{\mathbf{R}} V_0[\boldsymbol{x}_\alpha]\hat{\mathbf{R}}^\top \times (\hat{\boldsymbol{t}} - \boldsymbol{r}_\alpha)\Big)_2^-. \tag{28}$$

Estimating the noise level ϵ using (20), we generate random Gaussian noise that has the estimated noise level $\hat{\epsilon}$ and add it to the corrected positions independently. Then we compute the motion parameters $\{\boldsymbol{t}^*, \mathbf{R}^*\}$ and the angle $\Delta\Omega^*$ and axis $\boldsymbol{l}^*$ of the relative rotation $\hat{\mathbf{R}}^*\hat{\mathbf{R}}^\top$ each time.

Figure 5a,b 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

$$(\Delta\boldsymbol{t}^*, V[\hat{\boldsymbol{t}}]^{-1}\Delta\boldsymbol{t}^*) = 1, \quad (\Delta\boldsymbol{\Omega}^*, V[\hat{\mathbf{R}}]^{-1}\Delta\boldsymbol{\Omega}^*) = 1, \tag{29}$$

where $V[\hat{\boldsymbol{t}}]$ and $V[\hat{\mathbf{R}}]$ are computed by approximating $\bar{\mathbf{R}}$, $\{\bar{\boldsymbol{x}}_\alpha\}$, and ϵ^2 by $\hat{\mathbf{R}}$, $\{\hat{\boldsymbol{x}}_\alpha\}$, and $\hat{\epsilon}^2$, respectively, on the right-hand side of (12).

These ellipsoids indicate the lower bound on the standard deviation of the errors in each orientation (Kanatani 1996). The average diameter in the sense of root mean squares is 0.1cm for the translation and 0.1° for the rotation. The cubes in the figures are displayed as a reference; one edge of the cube is three times the average diameter of the respective ellipsoid. In Fig. 5a, the vertical edge is in the direction of the world X_0-axis taken to be perpendicular to the floor; in Fig. 5b, the vertical edge corresponds to the rotation components around the X_0-axis.

We compared our method with the least-squares method of (8). Figure 6a,b shows the result that corresponds to Fig. 5a,b (the ellipsoids and the cubes are the same as in Fig. 5a,b). Comparing Figs. 5a,b and 6a,b, we can confirm that our method improves the accuracy of the solution over the least-squares method (see the discussions in Sect. 10.6). We can also see that errors for our method distribute around the ellipsoids which correspond to the theoretical accuracy bound (12). This means that our method indeed attains the theoretical accuracy bound; no further improvement is possible.

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

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

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

$$S^*_{\boldsymbol{t}} = \sqrt{\frac{1}{B}\sum_{b=1}^B \|\Delta\boldsymbol{t}^*_b\|^2}, \quad S^*_{\mathbf{R}} = \sqrt{\frac{1}{B}\sum_{b=1}^B (\Delta\Omega^*_b)^2}, \tag{30}$$

where B is the number of bootstrap samples and the subscript b labels each sample. The corresponding standard deviations for the theoretical lower bound are

$$S_{\boldsymbol{t}} = \sqrt{\mathrm{tr}V[\hat{\boldsymbol{t}}]}, \qquad S_{\mathbf{R}} = \sqrt{\mathrm{tr}V[\hat{\mathbf{R}}]}, \tag{31}$$

respectively. Table 1 lists the values of $S^*_{\boldsymbol{t}}$ and $S^*_{\mathbf{R}}$ for our method and the least-squares method ($B = 1000$) together with their theoretical lower bounds $S_{\boldsymbol{t}}$ and $S_{\mathbf{R}}$. We can see 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 evaluating the theoretical accuracy bound given by (12) and using Gaussian approximation.

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

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

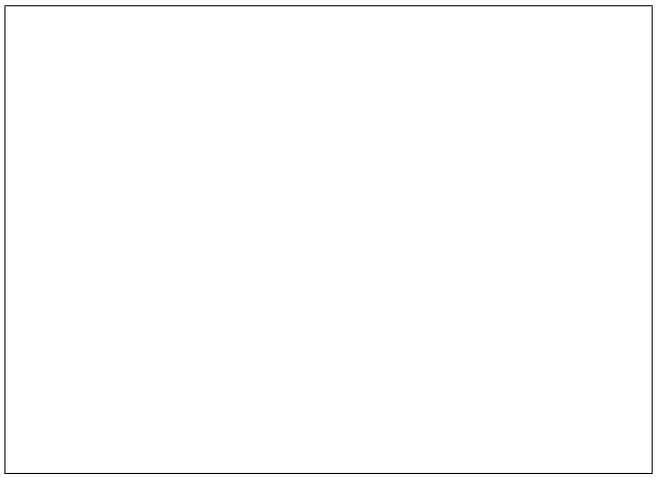


Fig. 7. An image of a real building

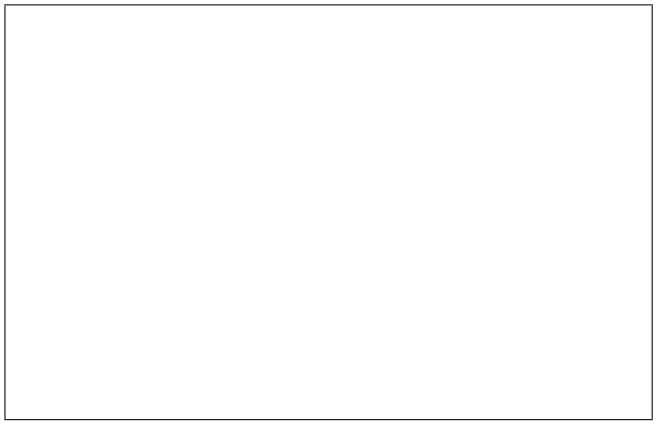


Fig. 8. Estimated current location and its reliability

8 Example 2

Figure 7 is a 640×480-pixel image of a real building for which a design plan is available. The focal length was calibrated to be $f = 849$ pixels. We manually selected the feature points marked by white dots and assumed the noise model of (5). The world coordinate system was fixed to the building, and the 3-D coordinates of the feature points were read from the design plan. The visible edge of this building is 13.4m long. The initial guess was computed by the optimal structure-from-motion algorithm (Kanatani 1994). The optimization converged after four iterations for thresholds $\varepsilon_{\boldsymbol{t}} = 0.1$cm and $\varepsilon_{\mathbf{R}} = 0.01^\circ$. According to this, the initial camera position was displaced by about 1.7m and rotated by about 4°.

Figure 8 displays the building and the estimated camera coordinate axes when viewed from above; the ellipse in the figure indicates the ellipsoid corresponding to those in Figs. 5**a** and 6**a** except that it is now enlarged by three times; its average diameter is about 3m. We also evaluated the reliability of the solution by both theoretical analysis and bootstrap. The noise level estimated by (20) is $\hat{\epsilon} = 4.89 \times 10^{-3}$, which translates to $\sigma = 1.05$ pixels. Table 2 shows 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.

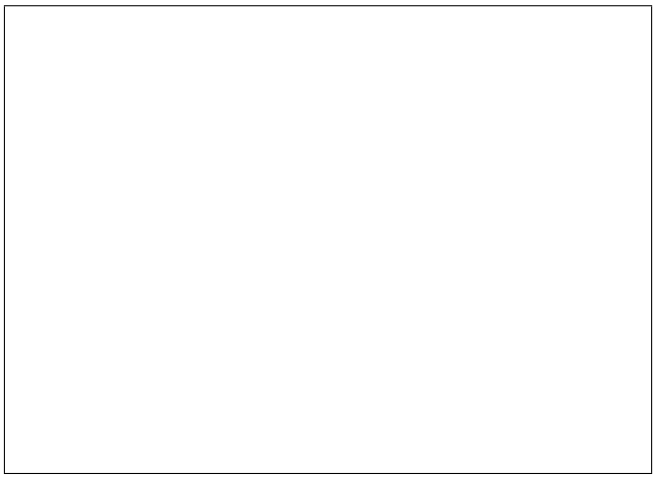


Fig. 9. An image of a city scene

9 Example 3

If the robot is constrained to be on a horizontal plane, the computation is considerably simplified. Figure 9 is a 640×480-pixel image of a city scene. The focal length was $f = 849$ pixels. We manually spotted nine features at the bottoms of the white vertical lines in the figure and computed the viewing location by matching the spotted positions to their corresponding locations in a city map (scaled by 1/10,000), out of which their coordinates were read. We defined the world coordinate system based on the latitude and longitude lines drawn in the map (the origin is at lat. $36^\circ 25' 22''$N and long. $139^\circ 21' 22''$E). The initial guess was computed by the method of circle geometry (Sugihara 1988; Sutherland and Thompson 1994). The optimization converged after five iterations for thresholds $\varepsilon_{\boldsymbol{t}} = 0.01$m and $\varepsilon_{\mathbf{R}} = 0.01^\circ$. According to this, the initial camera position was displaced by about 93m and rotated by 0.2°.

Figure 10 shows the angle of view and the estimated location superimposed on the city map; the locations of the feature points are marked by white dots. Figure 11 shows the estimated camera location superimposed on the enlarged image of the city map. One pixel corresponds to about 2.2m. The ellipse in the figure is a two-dimensional version of the ellipsoids in Figs. 5a and 6a; its longest and shortest radii are 36.7m and 6.8m, respectively. The white dot in the map indicates the place where the picture of Fig. 9 was actually taken (it is in the building shown in Fig.7; its horizontal size is 13.4m×20m). We see that the true position is within the ellipse. Its discrepancy from the estimate reads to be about 14m.

Figures 12a,b shows 100 bootstrap errors in the estimated location, which are plotted in the same way as Figs. 5a and 6a (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 (see the discussions in Sect. 10.6).

Fig. 10. Estimated angle of view

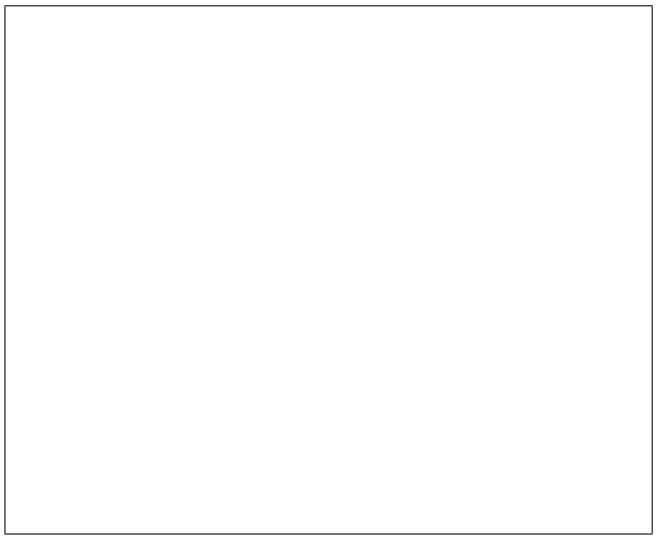
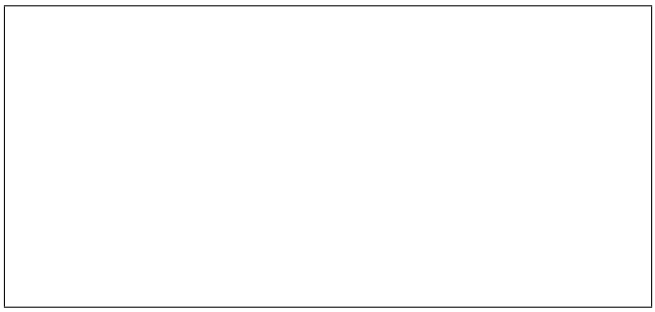


Fig. 11. Estimated current location



a **b**

Fig. 12. Bootstrap errors in the estimated location: **a** our method; **b** least squares

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

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

10 Discussions

10.1 Significance of theoretical approaches

Our approach appears to be excessively theoretical, but it has recently been recognized more and more in the computer vision community that for further advancement a theoretical approach such as ours is indispensable and that the gains obtained by mere combinations of existing methods and ad-hoc techniques are very much limited (Kanatani 1996; Hartley and Zisserman 2000).

In the past, the performance of a system was usually evaluated only *a posteriori* by experiments using real or synthetic data. But this does not tell us anything about situations that are not covered by the experiments. The optimality of a system can be guaranteed only by a theoretical analysis, and such an analysis also reveals the limitations of the system by telling us in what circumstances it performs well and in what circumstances not. Our approach is a typical example in that direction.

10.2 Theoretical background

The mathematical formulation used in this paper was not created specifically for solving the robot localization problem but rather given as a typical example of a general statistical optimization theory for geometric inference (Kanatani 1996). It can be proved that the solution of the constrained optimization of (16) can attain the theoretical accuracy bound in the first order, and simulations have confirmed that the solution indeed falls in the vicinity of the bound (Kanatani 1996, 2000; Ohta and Kanatani 1998; Kanatani and Ohta 1999). It has also been confirmed by simulations that the noise level estimation in the form of (20) indeed gives a very good estimate of the true noise level (Kanatani 1996; Kanazawa and Kanatani 1996a,b).

10.3 Feature matching procedure

In order to apply our localization scheme, we must first match observed image features to known model features. This can be done by first establishing the matching by human interaction and then tracing the selected features frame-by-frame in the course of the navigation (provided they are always visible from the robot). Otherwise, the robot hypothesizes a match based on known clues (e.g., brightness, color, shape) and then validates 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 (Yagi et al. 1995; Talluri and Aggarwal 1996).

Whatever strategy we adopt, the matching process requires many stages of image processing, which are in its own right a difficult research target beyond the scope of this paper. We have therefore assumed that the matching has already been established and focused on computational schemes for localization. It should be noted, however, that if we adopt the hypothesizing-validating approach, the ability to compute the 3-D camera position for a given match between the image and the model, and to evaluate the reliability of the solution, is crucial whether the match is correct or not. For example, we hypothesize the matching, optimally estimate the robot location, and evaluate the probability distribution of the estimated location by the procedure we have described. If the resulting distribution spreads out too widely, the hypothesis may be incorrect.

10.4 Validity of the noise model

We modeled the deviations of the detected feature points from their true positions as Gaussian noise. This is a reasonable approximation, because we are assuming that the deviations are within a few pixel magnitude. In fact, this assumption can be confirmed a posteriori using (20), as shown in our examples. If the estimated noise magnitude is unrealistically large, we must doubt the correctness of the matching, as mentioned above.

We have also assumed that the normalized covariance matrix is known. In fact, there are many methods for estimating it from the variations of the gray levels of the image (Förstner 1987; Singh 1990; Ohta 1991; Shi and Tomasi 1994; Morris and Kanade 1998; Kanazawa and Kanatani 2001). The result is generally different from point to point.

However, it has been observed (Kanazawa and Kanatani 2001) that it is sufficient to assume the isotropic and identical model of (5) *provided* the feature points are manually specified by humans or detected by a feature-detection operator (e.g., those described in Harris and Stephens 1988; Reisfeld et al. 1995; Smith and Brady 1997). This is because humans and feature detectors both implicitly evaluate the normalized covariance matrix and output those features with almost isotropic and identically distributed covariance. In contrast, the computed normalized covariance varies from point to point if the feature points are chosen randomly or independently of the image content (Kanazawa and Kanatani 2001).

The robot localization application falls in the former category as long as salient feature points are used for matching. Hence, it is sufficient to use the isotropic and homogeneous model of (5), as we did in our examples. For the sake of generality, however, we have described all equations using the notation $V_0[\boldsymbol{x}_\alpha]$.

10.5 Calibration errors and image distortions

We have assumed that the camera has already been calibrated and image distortions due to lens aberration has been corrected (e.g., by the procedure described in Weng et al. 1992). However, the calibration may not be precise, and the estimated focal length and the principal point (the point that corresponds to the lens optical axis) may not be correct. Also, the image distortion still exist. In our formulation, such deviations are regarded as "noise", and the noise level $\hat{\epsilon}$ estimated by (20) measures the total inaccuracies including such deviations.

Note that calibration errors and image distortions affect the accuracy of the feature points systematically with strong correlations among them. Hence, the a posteriori covariance matrix given by (12), which was derived on the assumption of independence of noise, may underestimate the true covariance.

Nevertheless, our formalism is expected to serve an approximate measure of the reliability of the solution in practice. After all, a theory is an idealization of the reality, and discrepancies of the theory from the reality must be tolerated to some degree in practical applications.

10.6 Why is the least squares not optimal?

Comparing (8) and (19) reveals that the difference between the least-squares method and our optimal method is in the weight matrix $\mathbf{W}_\alpha$ given by (15). We can see from (15) that the magnitude of $\mathbf{W}_\alpha$ is inversely proportional to $\|\boldsymbol{t} - \boldsymbol{r}_\alpha\|^2$.

Our optimal method minimizes the *image feature discrepancies* uniformly over all the feature points in the form of (16). In order to do so, we must weight the *constraint discrepancies* by $\mathbf{W}_\alpha$ in the form of (19), giving more weight on near features and less weight on remote features.

The least-squares method, on the other hand, minimizes constraint discrepancies uniformly over all the feature points in the form of (7). This means that the least squares gives, in effect, more weight on remote features, which are less sensitive to the camera position, and less weight on near features, which are more sensitive to the camera position. This is an intuitive explanation as to why the least-squares method is not optimal.

This observation implies that the difference between the least-squares method and our optimal method will become most apparent when the observed features have large depth variations, while not much difference will be found when the depth variations are small. In fact, the remote square pillar in Fig. 3 was placed to test this observation, and Figs. 5 and 6 indeed confirm our prediction.

In Examples 2 and 3, on the other hand, all the feature points have more or less similar depth magnitudes. This explains that the gain of our optimal method over the least squares is not so marked as in Example 1.

10.7 Other applications

This localization scheme can also be applied if all the feature points are coplanar in the scene. This situation occurs, for example, in *virtual studio applications*, where the position of a moving camera is computed frame-by-frame real time by observing a reference pattern placed in the scene (Matsunaga and Kanatani 2000; Seo et al. 2000). In such a situation, we can simplify the computation and calibrate the focal length as well by exploiting the fact that the relationship between the reference pattern and the observed image is a transformation called *homography* (Hartley and Zisserman 2000). Using the principle described here, we can optimally compute the solution and evaluate its reliability in quantitative terms. The theoretical predictions were confirmed by simulations (Matsunaga and Kanatani 2000).

The two-dimensional version of our scheme used in Example 3 can also be useful outside the domain of robot localization. For instance, it can be used off-line in forensic applications for estimating the location from which a given photograph was taken. The estimation may not be exact, as seen from Fig. 11, but we can also obtain its covariance, which indicates where to search. For example, the ellipse in Fig. 11 tells us that we need not search all directions concentrically; rather, we only need to search particular directions. Such information would greatly save costs and efforts.

Application of the mathematical framework presented here is not limited to position estimation. It can be applied to any problem of geometric estimation including optimally fitting lines and conic sections (Kanazawa and Kanatani 1996a,b), motion estimation (Kanatani 2000), homography estimation (Kanatani and Ohta 2000), and rotation estimation (Ohta and Kanatani 1998).

11 Concluding remarks

We have described 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 derived a theoretical accuracy bound independently of solution techniques and then presented a method that attains it; our method is truly "optimal" in that sense. We can view the bound as describing the probability distribution of the estimated location by using Gaussian approximation; this information does not require any knowledge about the noise magnitude. Using real images, we demonstrated that our method is superior to the naive least-squares method. We also confirmed the theoretical predictions of our theory by applying the bootstrap procedure. Finally, we discussed various theoretical and practical issues for using our method.

Acknowledgement. The authors thank Hiroyuki Tsuzuki of Oki Data Corporation and Yoshiyuki Kanazawa of Sharp, Ltd. for their assistance in our real image experiments and Chikara Matsunaga of For-A, Co. Ltd. for helpful discussions. This work was in part supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C(2) (No. 13680432).

Appendix: Gradient and Hessian computation

The gradient terms $\nabla_{\boldsymbol{t}} J$ and $\nabla_{\mathbf{R}} J$ that appear in (22) are computed as follows. First, compute the vectors $\boldsymbol{p}_\alpha$ and $\boldsymbol{q}_\alpha$, $\alpha = 1, ..., N$, as follows:

$$\boldsymbol{p}_\alpha = \mathbf{E}(\boldsymbol{r}_\alpha)\boldsymbol{x}_\alpha, \qquad \boldsymbol{q}_\alpha = \mathbf{W}_\alpha \mathbf{E}(\boldsymbol{r}_\alpha)\boldsymbol{x}_\alpha. \tag{32}$$

Here $\mathbf{E}(\,\cdot\,)$ is a matrix function defined by

$$\mathbf{E}(\boldsymbol{r}) = (\boldsymbol{t} - \boldsymbol{r}) \times \mathbf{R}. \tag{33}$$

Next, compute the matrices $\mathbf{P}_\alpha$ and $\mathbf{Q}_\alpha$, $\alpha = 1, ..., N$, as follows:

$$\begin{aligned}\mathbf{P}_\alpha = \mathbf{W}_\alpha \big(&\mathbf{A}(\boldsymbol{x}_\alpha) - \mathbf{A}(V_0[\boldsymbol{x}_\alpha]\mathbf{E}(\boldsymbol{r}_\alpha)^\top \boldsymbol{q}_\alpha) \\ &- \mathbf{E}(\boldsymbol{r}_\alpha) V_0[\boldsymbol{x}_\alpha] \mathbf{C}(\boldsymbol{q}_\alpha)\big),\end{aligned} \tag{34}$$

$$\begin{aligned}\mathbf{Q}_\alpha = \mathbf{W}_\alpha \big(&\mathbf{B}(\boldsymbol{r}_\alpha, \boldsymbol{x}_\alpha) - \mathbf{B}(\boldsymbol{r}_\alpha, V_0[\boldsymbol{x}_\alpha]\mathbf{E}(\boldsymbol{r}_\alpha)^\top \boldsymbol{q}_\alpha) \\ &- \mathbf{E}(\boldsymbol{r}_\alpha) V_0[\boldsymbol{x}_\alpha] \mathbf{D}(\boldsymbol{r}_\alpha, \boldsymbol{q}_\alpha)\big).\end{aligned} \tag{35}$$

Here $\mathbf{A}(\,\cdot\,)$, $\mathbf{B}(\,\cdot\,,\,\cdot\,)$, $\mathbf{C}(\,\cdot\,)$, and $\mathbf{D}(\,\cdot\,,\,\cdot\,)$ are matrix functions defined as follows:

$$\begin{aligned}&\mathbf{A}(\boldsymbol{x}) = -(\mathbf{R}\boldsymbol{x}) \times \mathbf{I}, \\ &\mathbf{B}(\boldsymbol{r}, \boldsymbol{x}) = (\boldsymbol{t} - \boldsymbol{r}, \mathbf{R}\boldsymbol{x})\mathbf{I} - \mathbf{R}\boldsymbol{x}(\boldsymbol{t} - \boldsymbol{r})^\top, \\ &\mathbf{C}(\boldsymbol{x}) = \mathbf{R}^\top(\boldsymbol{x} \times \mathbf{I}), \\ &\mathbf{D}(\boldsymbol{r}, \boldsymbol{x}) = \mathbf{R}^\top((\boldsymbol{t} - \boldsymbol{r})\boldsymbol{x}^\top - \boldsymbol{x}(\boldsymbol{t} - \boldsymbol{r})^\top).\end{aligned} \tag{36}$$

Then $\nabla_{\boldsymbol{t}} J$ and $\nabla_{\mathbf{RR}} J$ are given by

$$\begin{aligned}\nabla_{\boldsymbol{t}} J &= \sum_{\alpha=1}^{N} \left(\mathbf{P}_\alpha^\top \boldsymbol{p}_\alpha + \mathbf{A}(\boldsymbol{x}_\alpha)^\top \boldsymbol{q}_\alpha\right), \\ \nabla_{\mathbf{R}} J &= \sum_{\alpha=1}^{N} \left(\mathbf{Q}_\alpha^\top \boldsymbol{p}_\alpha + \mathbf{B}(\boldsymbol{r}_\alpha, \boldsymbol{x}_\alpha)^\top \boldsymbol{q}_\alpha\right),\end{aligned} \tag{37}$$

where $S[\,\cdot\,]$ designate the symmetrization operation: $S[\mathbf{A}] = (\mathbf{A} + \mathbf{A}^\top)/2$.

In Newton iterations, the Hessian controls the speed of convergence, not the accuracy of the solution. So, we apply the *Gauss-Newton approximation* in computing the Hessian: we regard the matrix $\mathbf{W}_\alpha$ in (19) as constant. The resulting iterations are called the *Gauss-Newton iterations* and are known to be almost as effective as Newton iterations in practice. The matrices $\nabla^2_{\boldsymbol{tt}} J$, $\nabla^2_{\boldsymbol{t}\mathbf{R}} J$, and $\nabla^2_{\mathbf{RR}} J$ in (22) are given as follows:

$$\nabla^2_{\boldsymbol{tt}} J = 2\sum_{\alpha=1}^{N} (\mathbf{R}\boldsymbol{x}_\alpha) \times \mathbf{W}_\alpha \times (\mathbf{R}\boldsymbol{x}_\alpha), \tag{38}$$

$$\begin{aligned}\nabla^2_{\boldsymbol{t}\mathbf{R}} J = &\,2\sum_{\alpha=1}^{N} (\boldsymbol{t} - \boldsymbol{r}_\alpha, \mathbf{R}\boldsymbol{x}_\alpha)(\mathbf{R}\boldsymbol{x}_\alpha) \times \mathbf{W}_\alpha \\ &-2\sum_{\alpha=1}^{N} \left((\mathbf{R}\boldsymbol{x}_\alpha) \times \mathbf{W}_\alpha \mathbf{R}\boldsymbol{x}_\alpha\right)(\boldsymbol{t} - \boldsymbol{r}_\alpha)^\top \\ &+2\sum_{\alpha=1}^{N} \mathbf{R}\boldsymbol{x}_\alpha((\boldsymbol{t} - \boldsymbol{r}_\alpha) \times \mathbf{R}\boldsymbol{x}_\alpha)^\top \mathbf{W}_\alpha \\ &-2\sum_{\alpha=1}^{N} (\mathbf{R}\boldsymbol{x}_\alpha, \mathbf{W}_\alpha((\boldsymbol{t} - \boldsymbol{r}_\alpha) \times \mathbf{R}\boldsymbol{x}_\alpha))\mathbf{I},\end{aligned} \tag{39}$$

$$\nabla^2_{\mathbf{RR}} J = 2\sum_{\alpha=1}^{N} (\boldsymbol{t} - \boldsymbol{r}_\alpha, \mathbf{R}\boldsymbol{x}_\alpha)^2 \mathbf{W}_\alpha$$
$$-4\sum_{\alpha=1}^{N} (\boldsymbol{t} - \boldsymbol{r}_\alpha, \mathbf{R}\boldsymbol{x}_\alpha) S[\mathbf{W}_\alpha \mathbf{R}\boldsymbol{x}_\alpha (\boldsymbol{t} - \boldsymbol{r}_\alpha)^\top]$$
$$+2\sum_{\alpha=1}^{N} (\mathbf{R}\boldsymbol{x}_\alpha, \mathbf{W}_\alpha \mathbf{R}\boldsymbol{x}_\alpha)(\boldsymbol{t} - \boldsymbol{r}_\alpha)(\boldsymbol{t} - \boldsymbol{r}_\alpha)^\top$$
$$+2\sum_{\alpha=1}^{N} S[(\mathbf{R}\boldsymbol{x}_\alpha)(\mathbf{R}\boldsymbol{x}_\alpha)^\top ((\boldsymbol{t} - \boldsymbol{r}_\alpha) \times \mathbf{W}_\alpha \times (\boldsymbol{t} - \boldsymbol{r}_\alpha))]$$
$$-2\sum_{\alpha=1}^{N} ((\boldsymbol{t} - \boldsymbol{r}_\alpha) \times \mathbf{R}\boldsymbol{x}_\alpha, \mathbf{W}_\alpha ((\boldsymbol{t} - \boldsymbol{r}_\alpha) \times \mathbf{R}\boldsymbol{x}_\alpha))\mathbf{I}. \quad (40)$$

References

Ayache N, Faugeras OD (1988) Building, registrating, and fusing noisy visual maps. Int J Robot. Res. 7: 45–65

Betke M, Gurvits L (1997) Mobile robot localization using landmarks. IEEE Trans Robot Autom 13: 251–263

Efron B, Tibshirani RJ (1993) An Introduction to Bootstrap. Chapman-Hall, New York

Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm ACM 24: 381–395

Förstner W (1987) Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. Comput Vis Graph Image Process 40:273–310

Harris C, Stephens M (1988) A combined corner and edge detector. In: Proc 4th Alvey Vision Conf, Manchester, U.K, August, pp 147–151

Hartley R, Zisserman A (2000) Multiple View Geometry, Cambridge University Press, Cambridge

Kanatani K (1994) Renormalization for motion analysis: Statistically optimal algorithm. IEICE Trans Inform Syst E77-D: 1233–1239

Kanatani K (1996) Statistical Optimization for Geometric Computation: Theory and Practice, Elsevier, Amsterdam

Kanatani K (2000) Optimal fundamental matrix computation: Algorithm and reliability analysis. In: Proc 6th Symp Sensing Image Inform, Yokohama, Japan, June, pp 291–298

Kanatani K, Morris DD (2001) Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. IEEE Trans Inform Theor 47: 2017–2028

Kanatani K, Ohta N (2000) Accuracy bounds and optimal computation of homography for image mosaicing applications. In: Proc 7th Int Conf Comput Vis, Kerkya, Greece, September, pp 73–78

Kanazawa Y, Kanatani K (1996a) Optimal line fitting and reliability evaluation. IEICE Trans Inform Syst E79-D: 1317–1322

Kanazawa Y, Kanatani K (1996b) Optimal conic fitting and reliability evaluation. IEICE Trans Inform Syst E79-D: 1323–1328

Kanazawa Y, Kanatani K (2001) Do we really have to consider covariance matrices for image features? In: Proc 8th Int Conf Comput Vis, Vancouver, Canada, July, Vol. 2, pp. 301–306

Matsunaga C, Kanatani K (2000) Calibration of a moving camera using a planar pattern: Optimal computation, reliability evaluation and stabilization by model selection. In: Proc 6th Euro Conf Comput Vis, Dublin, Ireland, June–July, Vol.2, pp 595–609

Morris DD, Kanade T (1998) A unified factorization algorithm for points, line segments and planes with uncertainty models. In: Proc 6th Int Conf Comput Vis, Bombay, India, January, pp 696–702

Ohta N (1991) Image movement detection with reliability indices. IEICE Trans E74: 3379–3388

Ohta N, Kanatani K (1998) Optimal estimation of three-dimensional rotation and reliability evaluation. In: Proc 5th Euro Conf Comput Vis, Freiburg, Germany, June, Vol 1, pp 175–187

Reisfeld D, Wolfson H, Yeshurun Y (1995) Context-free attentional operators: The generalized symmetry transform. Int J Comput Vis 14:119–130

Seo Y, Ahn M-H, Hong K-S (2000) A multiple view approach for auto-calibration of a rotating and zooming camera. IEICE Trans Inform Syst E83-D: 1375–1385

Singh A (1990) An estimation-theoretic framework for image-flow computation. In: Proc 3rd Int Conf Comput Vis, Osaka, Japan, December, pp 168–177

Shi J, Tomasi C (1994) Good features to track. In: Proc IEEE Conf Comput Vis Patt Recog Seattle, Washi., June, pp 593–600

Smith SM, Brady JM (1997) SUSAN – A new approach to low level image processing. Int J Comput Vis 23: 45–78

Sugihara K (1988) Some location problems for robot navigation using a single camera. Comput Vis Graph Image Process 42:112–129

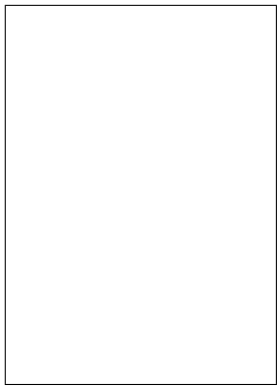
Suorsa RE, Sridhar B (1994) A parallel implementation of a multisensor feature-based range-estimation method. IEEE Trans Robot. Autom 10: 755–768

Sutherland KT, Thompson WB (1994) Localizing in unconstrained environment: Dealing with the errors. IEEE Trans Robot Autom 10: 740–754

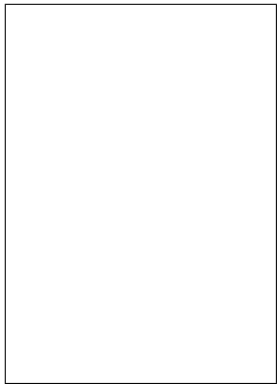
Talluri R, Aggarwal JK (1996) Mobile robot self-location using model-image feature correspondence. IEEE Trans Robot Autom 12: 63–77

Weng J, Cohen P, Herniou M (1992) Camera calibration with distortion models and accuracy evaluation. IEEE Trans Patt Anal Mach Intell 14: 965–980

Yagi M, Nishimitsu Y, Yachida M (1995) Map-based navigation for a mobile robot with omnidirectional image sensor COPIS. IEEE Trans Robot Autom 11: 634–648



Kenichi Kanatani received his Ph.D. in applied mathematics from the University of Tokyo in 1979. After serving as Professor of computer science at Gunma University, Japan, he has been Professor of information technology at Okayama University, Japan, since April 2001. He is the author of Group-Theoretical Methods in Image Understanding (Springer, 1990), Geometric Computation for Machine Vision (Oxford University Press, 1993) and Statistical Optimization for Geometric Computation: Theory and Practice (Elsevier Science, 1996).



Naoya Ohta received his ME degree in information science from the Tokyo Institute of Technology in 1985 and his Ph.D. in applied mathematics from the University of Tokyo in 1998. He engaged in research and development of image processing systems at the Pattern Recognition Research Laboratories of NEC, Japan. He was a research affiliate of the Media Laboratory in MIT from 1991 to 1992. He is currently Associate Professor of computer science at Gunma University, Japan. His research interests include image processing and computer vision.

This article was processed by the author using the LaTeX style file *cljour2* from Springer-Verlag.

Figures

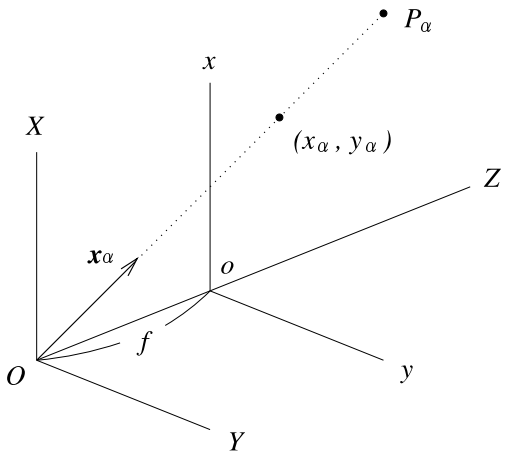


Fig. 1. Camera imaging geometry

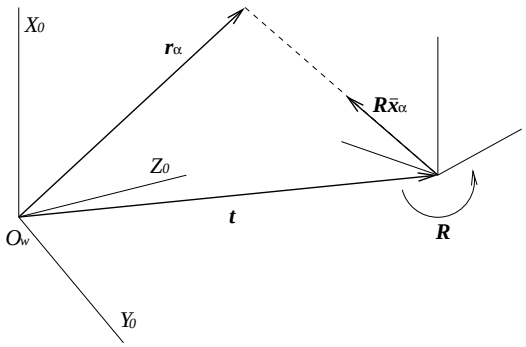


Fig. 2. The camera coordinate system and the world coordinate system

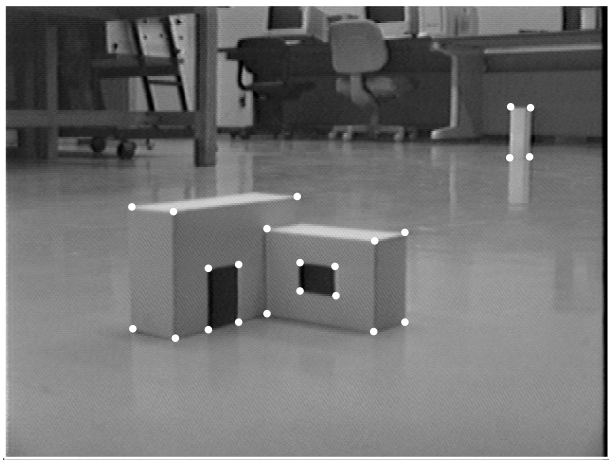


Fig. 3. An image of a toy house

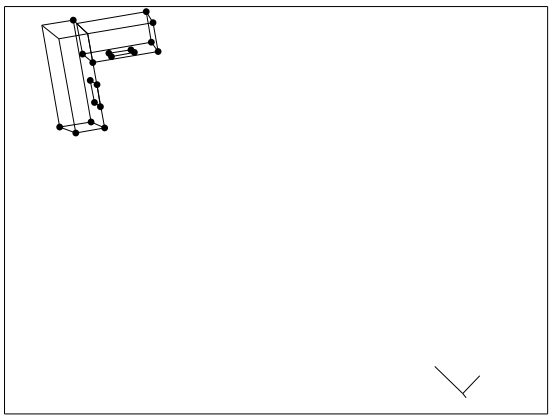
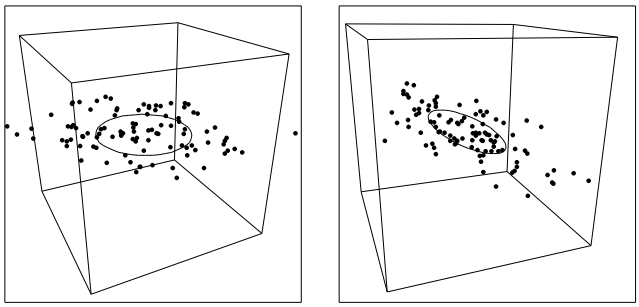
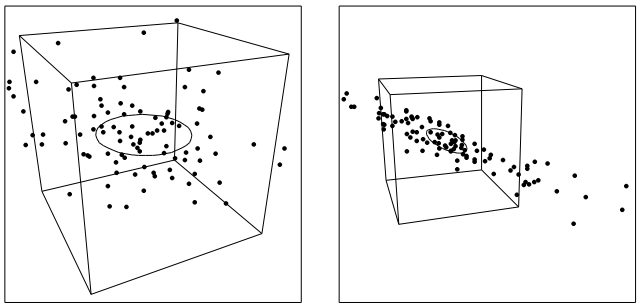


Fig. 4. Estimated current location



a **b**

Fig. 5ab. Bootstrap errors (our method): **a** translation; **b** rotation



a **b**

Fig. 6ab. Bootstrap errors (least squares): **a** translation; **b** rotation

Fig. 7. An image of a real building

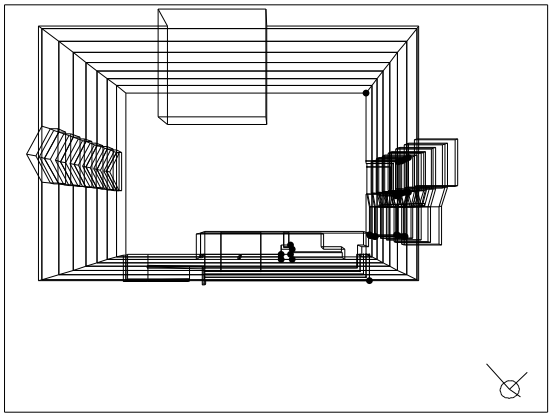


Fig. 8. Estimated current location and its reliability

Fig. 9. An image of a city scene

Fig. 10. Estimated angle of view

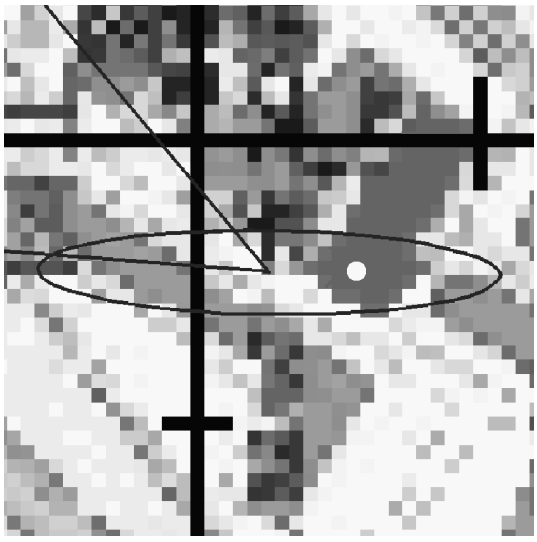
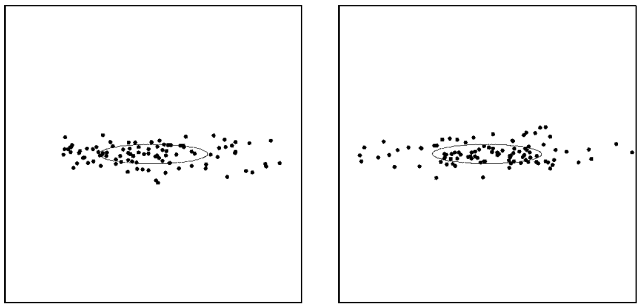


Fig. 11. Estimated current location



a **b**

Fig. 12ab. Bootstrap errors in the estimated location: **a** our method; **b** least squares