J Math Imaging Vis
DOI 10.1007/s10851-010-0206-6

Unified Computation of Strict Maximum Likelihood for Geometric Fitting

Kenichi Kanatani · **Yasuyuki Sugaya**

Received: 13 May 2009 / Accepted: xx Xxxx 0000

Abstract A new numerical scheme is presented for computing strict maximum likelihood (ML) of geometric fitting problems having an implicit constraint. Our approach is orthogonal projection of observations onto a parameterized surface defined by the constraint. Assuming a linearly separable nonlinear constraint, we show that a theoretically global solution can be obtained by iterative Sampson error minimization. Our approach is illustrated by ellipse fitting and fundamental matrix computation. Our method also encompasses optimal correction, computing, e.g., perpendiculars to an ellipse and triangulating stereo images. A detailed discussion is given to technical and practical issues about our approach.

Keywords Geometric fitting · Maximum likelihood · Ellipse fitting · Fundamental matrix · Stereo image triangulation

1 Introduction

This paper presents a unified numerical scheme for computing strict maximum likelihood (ML) for a problem called *geometric fitting* [16] having an *implicit* constraint. This type of problem very frequently appears in computer vision applications. By "strict", we mean Gaussian noise is assumed in the original data space, while many existing methods implicitly assume it in a transformed data space, minimizing what is known as the *Sampson error*. By "unified", we mean we need not derive a problem-specific cost function for particular applications. Assuming a linearly separable constraint, which is very common in computer vision applications, we show that a theoretically global solution can be obtained by iterative Sampson error minimization in a problem-independent manner.

Our approach is *orthogonal projection* of observations onto a parameterized surface defined by the constraint. The principle itself is well known, but we present a new scheme by exploiting the linear separability of the constraint. We illustrated our procedure by applying it to ellipse fitting and fundamental matrix computation. Our method also encompasses *optimal correction* [16], which, too, plays an important role in computer vision applications. We illustrate our approach by computing perpendiculars to an ellipse and triangulating stereo images.

We first state the problem and give mathematical fundamentals in Sect. 2 and 3. In Sect. 4 and 5, we contrast two types of ML. In Sect. 6, we review existing approaches and sketch our strategy. The computational details are described in Sect. 7 and 8. In Sect. 9, we apply our approach to ellipse fitting and fundamental matrix computation and show some numerical results. In Sect. 10, we reduce our method to optimal correction schemes, showing how to compute perpendiculars to an ellipse and triangulate stereo images. We summarize technical and practical issues of our approach in Sec. 11 and conclude in Sect. 12.

2 Geometric fitting

We consider the problem of fitting to noisy vector data $\mathbf{x}_\alpha$, $\alpha = 1, ..., N$, an *implicit* equation in the form

Kenichi Kanatani
Department of Computer Science, Okayama University, Okayama, 700-8530 Japan
Tel.: +81-86-251-8173
Fax.: +81-86-251-8173
E-mail: kanatani@suri.cs.okayama-u.ac.jp

Yasuyuki Sugaya
Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Aichi, 441-8580 Japan
Tel.: +81-532-44-6760
Fax.: +81-532-44-6757
E-mail: sugaya@iim.cs.tut.ac.jp

$$F(\mathbf{x};\theta) = 0, \tag{1}$$

parameterized by θ. In other words, we want to estimate the parameter θ in such a way that $F(\mathbf{x}_\alpha;\theta) \approx 0$ for all α. Many computer vision problems are formulated in this way [15,16]; one can infer the shapes and the positions of objects seen in images from the thus computed θ. In the statistics literature, this problem is sometimes called the *Gauss-Helmert model*, while it is called the *Gauss-Markoff model* if (1) can be explicitly solved for $\mathbf{x}$ in terms of θ [9,28].

The function $F(\mathbf{x};\theta)$ in (1) is generally nonlinear in the data vector $\mathbf{x}$. In a wide range of computer vision problems, however, $F(\mathbf{x};\theta)$ is frequently linear in the parameter θ or can be made linear by an appropriate reparameterization. In such a case, (1) can be rewritten as

$$(\xi(\mathbf{x}),\theta) = 0, \tag{2}$$

where and throughout this paper we denote by $(\mathbf{a},\mathbf{b})$ the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$. The ith component $\xi_i(\mathbf{x})$ of the vector $\xi(\mathbf{x})$ consists of (generally nonlinear) terms in $\mathbf{x}$ that are multiplied by θ_i. If terms that do not involve θ are added, they are regarded as multiplied by an unknown, which we identify with the final component θ_n of θ. Then, we should obtain a solution such that $\theta_n = 1$, but because (2) is homogeneous in θ, we can determine θ only up to scale. It follows that an arbitrary normalization can be imposed on θ, such as $\|\theta\| = 1$.

Example 1 (Ellipse Fitting) We want to fit to a point sequence (x_α, y_α), $\alpha = 1, ..., N$, an ellipse in the form

$$Ax^2 + 2Bxy + Cy^2 + 2(Dx + Ey) + F = 0, \tag{3}$$

(Fig. 1). If we define $\xi(x,y)$ and θ by

$$\xi(x,y) = (x^2\ 2xy\ y^2\ 2x\ 2y\ 1)^\top,\quad \theta = (A\ B\ C\ D\ E\ F)^\top, \tag{4}$$

(3) has the form of (2) [22].

Example 2 (Fundamental Matrix Computation) Consider two images of the same scene viewed from different positions. If point (x,y) in the first image corresponds to (x',y') in the second, the following *epipolar equation* is satisfied [15] (Fig. 2):

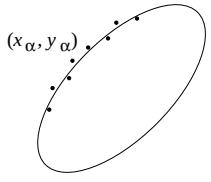
$$\left(\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \mathbf{F}\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}\right) = 0. \tag{5}$$

Here, $\mathbf{F}$ is a matrix of rank 2, called the *fundamental matrix*, which does not depend on the scene we are looking at; it depends only on the relative positions of the two cameras and their intrinsic parameters. By computing the fundamental matrix $\mathbf{F}$ from point correspondences, we can reconstruct the 3-D shape of the scene and the camera positions [19]. If we define

$$\begin{aligned} &\xi(x,y,x',y') = (xx'\ xy'\ x\ yx'\ yy'\ y\ x'\ y'\ 1)^\top, \\ &\theta = (F_{11}\ F_{12}\ F_{13}\ F_{21}\ F_{22}\ F_{23}\ F_{31}\ F_{32}\ F_{33})^\top, \end{aligned} \tag{6}$$

(5) has the form of (2) [20].

Fig. 1 Fitting an ellipse to a point sequence.



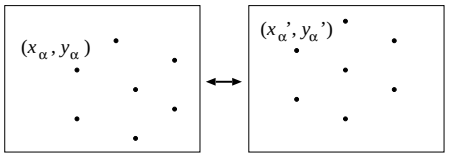


Fig. 2 Computing the fundamental matrix from corresponding points between two images.

3 Gaussian noise in the ξ-space

For statistical inference from noisy data, we need to specify:

- Noise model: What kind of property do we assume noise to have?
- Criterion of optimality: What kind of solution do we regard as optimal?

The standard noise model is independent Gaussian noise of mean 0; each observation may have a different (non-isotropic) covariance matrix. For this, however, we have two alternatives: Gaussian noise in the original data $\mathbf{x}_\alpha$ or in the transformed data $\xi_\alpha = \xi(\mathbf{x}_\alpha)$. The covariance matrix $V[\mathbf{x}_\alpha]$ of $\mathbf{x}_\alpha$ and the covariance matrix $V[\xi_\alpha]$ of ξ_α are related by

$$V[\xi_\alpha] = \left(\frac{\partial \xi}{\partial \mathbf{x}}\right)_\alpha V[\mathbf{x}_\alpha]\left(\frac{\partial \xi}{\partial \mathbf{x}}\right)_\alpha^\top, \tag{7}$$

up to high (fourth[1] in Examples 1 and 2) order noise terms, where $(\partial\xi/\partial\mathbf{x})_\alpha$ denotes the Jacobian matrix of the mapping $\xi(\mathbf{x})$ evaluated at $\mathbf{x} = \mathbf{x}_\alpha$.

[1] Covariance matrices are expectation of second order noise statistics. For symmetrically distributing noise of mean 0, expectation of third order noise terms vanishes.

Example 3 (Ellipse Fitting) If each point (x_α, y_α) has independent noise of mean 0 and standard deviation σ in its x and y coordinates, the covariance matrix $V[\xi_\alpha]$ has the following form [22]:

$$V[\xi_\alpha] = 4\sigma^2 \begin{pmatrix} x_\alpha^2 & x_\alpha y_\alpha & 0 & x_\alpha & 0 & 0 \\ x_\alpha y_\alpha & x_\alpha^2 + y_\alpha^2 & x_\alpha y_\alpha & y_\alpha & x_\alpha & 0 \\ 0 & x_\alpha y_\alpha & y_\alpha^2 & 0 & y_\alpha & 0 \\ x_\alpha & y_\alpha & 0 & 1 & 0 & 0 \\ 0 & x_\alpha & y_\alpha & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{8}$$

Example 4 (Fundamental Matrix Computation) If each correspondence pair (x_α, y_α) and (x'_α, y'_α) has independent noise of mean 0 and standard deviation σ in its x and y coordinates, the covariance matrix $V[\xi_\alpha]$ is as follows [20]:

$$V[\xi_\alpha] = \sigma^2 \begin{pmatrix} x_\alpha^2 + x_\alpha'^2 & x'_\alpha y'_\alpha & x'_\alpha & x_\alpha y_\alpha & 0 & 0 & x_\alpha & 0 & 0 \\ x'_\alpha y'_\alpha & x_\alpha^2 + y_\alpha'^2 & y'_\alpha & 0 & x_\alpha y_\alpha & 0 & 0 & x_\alpha & 0 \\ x'_\alpha & y'_\alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_\alpha y_\alpha & 0 & 0 & y_\alpha^2 + x_\alpha'^2 & x'_\alpha y'_\alpha & x'_\alpha & y_\alpha & 0 & 0 \\ 0 & x_\alpha y_\alpha & 0 & x'_\alpha y'_\alpha & y_\alpha^2 + y_\alpha'^2 & y'_\alpha & 0 & y_\alpha & 0 \\ 0 & 0 & 0 & x'_\alpha & y'_\alpha & 1 & 0 & 0 & 0 \\ x_\alpha & 0 & 0 & y_\alpha & 0 & 0 & 1 & 0 & 0 \\ 0 & x_\alpha & 0 & 0 & y_\alpha & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{9}$$

4 ML in the ξ-space

The standard criterion for optimality is *maximum likelihood* (*ML*): the likelihood function obtained by substituting observed data into the probability density of the noise model is maximized, or equivalently its negative logarithm is minimized. It is known that the resulting solution achieves the theoretical accuracy bound called the *KCR lower bound* [6, 16, 17] up to higher order noise terms.

If Gaussian noise is assumed in the ξ-space, ML reduces to minimization of the *Mahalanobis distance*

$$J = \sum_{\alpha=1}^{N} (\xi_\alpha - \bar{\xi}_\alpha, V[\xi_\alpha]^{-1}(\xi_\alpha - \bar{\xi}_\alpha)), \tag{10}$$

between observed values $\{\xi_\alpha\}$ and their true values $\{\bar{\xi}_\alpha\}$ subject to

$$(\bar{\xi}_\alpha, \theta) = 0, \qquad \alpha = 1, ..., N, \tag{11}$$

with respect to $\bar{\xi}_\alpha$ and θ. Since the constraint is linear in $\bar{\xi}_\alpha$, it can be eliminated by introducing Lagrange multipliers, reducing (10) to[2] [18]

$$J = \sum_{\alpha=1}^{N} \frac{(\xi_\alpha, \theta)^2}{(\theta, V[\xi_\alpha]\theta)}, \tag{12}$$

which is called the *Sampson error*, originating from ellipse fitting by Sampson [29].

Various numerical schemes are available for minimizing (12) [18], including the *FNS* (*Fundamental Numerical Scheme*) of Chojnacki et al. [7], the *HEIV* (*Heteroscedastic Errors In Variables*) of Leedan and Meer [26], and the *projective Gauss-Newton iterations* of Kanatani and Sugaya [20, 22]. These apply when no special constraint (scale normalization aside) is imposed on θ. For computing the fundamental matrix, however, it has an additional constraint that it has rank 2. The FNS of Chojnacki et al. [7] can be extended to incorporate such constraints in the form of the *CFNS*[3] (*Constrained FNS*) of Chojnacki et al. [8] and the *EFNS* (*Extended FNS*) of Kanatani and Sugaya [21].

[2] If ξ_α has constant components as in (4) and (6), the covariance matrix $V[\xi_\alpha]$ becomes singular as seen in (8) and (9). In such a case, we replace $V[\xi_\alpha]^{-1}$ in (10) by the pseudoinverse, which means we focus only on those components of ξ_α that can vary. Still, (12) holds [16].

[3] It was pointed out that CFNS does not necessarily compute a correct solution [21].

In the past, minimizing the Sampson error (12) has been regarded by many as an *approximation* to ML; some called Sampson error minimization "approximate ML". We point out that Sampson error minimization is "exact ML" for *Gaussian noise in the ξ-space*: No approximation is introduced to go from (10) and (11) to (12). We will show that this observation plays a crucial role in computing ML in the **x**-space.

5 ML in the x-space

The preceding formulation suits numerical computation and accuracy analysis [18]. For ellipse fitting, however, it is natural to assume that each point (x_α, y_α) has independent Gaussian noise in its x and y coordinates. Then, the noise in ξ_α after the nonlinear transformation (4) is, strictly, no longer Gaussian. For fundamental matrix computation, too, it is natural to assume that each corresponding pair (x_α, y_α) and (x'_α, y'_α) has independent Gaussian noise in its x and y coordinates, but the noise in ξ_α after the nonlinear transformation (6) is, strictly, no longer Gaussian. Whether we assume Gaussian noise in the ξ-space or in the **x**-space may not make much difference as long as the noise is small, but some difference may arise when the noise is large. Studying this is the main motivation of this paper.

If Gaussian noise is assumed in the **x**-space, ML reduces to minimization of the Mahalanobis distances[4]

$$E = \sum_{\alpha=1}^{N} (\mathbf{x}_\alpha - \bar{\mathbf{x}}_\alpha, V[\mathbf{x}_\alpha]^{-1}(\mathbf{x}_\alpha - \bar{\mathbf{x}}_\alpha)), \tag{13}$$

between observations $\{\mathbf{x}_\alpha\}$ and their true values $\{\bar{\mathbf{x}}_\alpha\}$ subject to the nonlinear constraint

$$(\xi(\bar{\mathbf{x}}_\alpha), \theta) = 0, \qquad \alpha = 1, ..., N, \tag{14}$$

with respect to $\bar{\mathbf{x}}_\alpha$ and θ. To avoid confusion, we call the Mahalanobis distance (13) the *reprojection error* as opposed to the Sampson error (12), which equals the Mahalanobis distance (10) in the ξ-space. Traditionally, the term "reprojection" has been used in the context of 3-D reconstruction from images: an assumed 3-D shape is "reprojected" onto the image plane and compared with the actually observed image; the 3-D shape with the smallest discrepancy is sought. Here, we slightly abuse this term to mean the Mahalanobis distance between actual observations and their guesses *in the original* **x**-*space*.

[4] The following argument holds if $V[\mathbf{x}_\alpha]$ is singular. All we need is to replace $V[\mathbf{x}_\alpha]^{-1}$ by its pseudoinverse and appropriately use projection operations [16].

6 Existing ML approaches

In the past, many researchers have studied minimization of the reprojection error (13) subject to a *general* constraint (1) instead of (14). Popular approaches can be roughly classified into *orthogonal projection* and *bundle adjustment*.

Orthogonal projection. The constraint $F(\mathbf{x}; \theta) = 0$ defines a hypersurface $\mathscr{S}$ in the **x**-space, and the problem of finding $\bar{\mathbf{x}}_\alpha$ that minimizes (13) can be interpreted as finding a point $\bar{\mathbf{x}}_\alpha \in \mathscr{S}$ closest to $\mathbf{x}_\alpha$ measured in (13). Hence, the solution is obtained by "orthogonally" projecting the observations $\mathbf{x}_\alpha$ onto $\mathscr{S}$, where orthogonality is defined with respect to $V[\mathbf{x}_\alpha]$: **a** and **b** are orthogonal if $(\mathbf{a}, V[\mathbf{x}_\alpha]^{-1}\mathbf{b}) = 0$ (Fig. 3). An initial guess $\{\mathbf{x}_\alpha^{(0)}, \theta^{(0)}\}$ ($\mathbf{x}_\alpha^{(0)}$ are the observations $\mathbf{x}_\alpha$ themselves) is assumed, and a sequence $\{\mathbf{x}_\alpha^{(0)}, \theta^{(0)}\}$, $\{\mathbf{x}_\alpha^{(1)}, \theta^{(1)}\}$, $\{\mathbf{x}_\alpha^{(2)}, \theta^{(2)}\}$, ... is generated by computing $\{\mathbf{x}_\alpha^{(k)}, \theta^{(k)}\}$ from $\{\mathbf{x}_\alpha^{(k-1)}, \theta^{(k-1)}\}$ [1,2,9,28].

Bundle adjustment. Introducing auxiliary variables, we solve the constraint $F(\bar{\mathbf{x}}_\alpha; \theta) = 0$ for each $\bar{\mathbf{x}}_\alpha$ in terms of θ and the auxiliary variables. The resulting expressions are substituted into (13) to define an explicit function E of θ and the auxiliary variables. Then, it is minimized by a standard tool such as the Levenberg-Marquardt method. Usually, a local minimum of E is obtained, but methods that can find a global minimum also exist, e.g., branch and bound [13]. Here, we are slightly abusing the term "bundle adjustment". It has traditionally been used in the context of 3-D reconstruction from images; we adjust the bundle of "rays" starting from camera projection center in such a way that the reprojection error is minimized [31].

The purpose of this paper is to show that if the constraint has the form of (14) we can exploit the close relationship between the ML in the ξ-space ((10), (11), and (12)) and the ML in the **x**-space ((13) and (14)), which seems to have been overlooked in the past.

Of course, we could adopt the traditional approach even for the constraint (14). Indeed, bundle adjustment has been very popular. For ellipse fitting, for example, auxiliary variables such as the center, the radii, the major and minor axes, and the angular parameters of individual points are introduced, and the resulting high dimensional space is searched

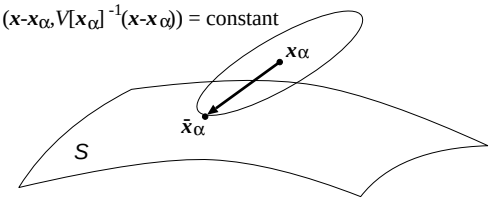


Fig. 3 Orthogonal projection of $\mathbf{x}_\alpha$ with respect to $V[\mathbf{x}_\alpha]$.

[4,10,11,30]. For fundamental matrix computation, auxiliary variables are introduced by tentatively reconstructing the 3-D positions of the observed points from an assumed fundamental matrix, which is a function of the camera parameters (the relative positions of the cameras and its intrinsic parameters). The resulting high dimensional space is searched so that the reprojection error is minimized [3].

However, the way we obtain an explicit form of E is problem-dependent, since the choice of auxiliary variables depends on particular properties of the problem. Also, the search space is usually very high dimensional, and the efficiency of computation depends largely on how we implement the search; the speed can be greatly accelerated by a clever preprocessing of sparse Hessians by considering the particularities of the problem.

In the following, we adopt the orthogonal projection approach to the constraint (14) and show that the optimization is formulated in a problem-independent way without using any particular properties of the problem. We show that the computation reduces to repeated minimization of the Sampson error (12). The resulting solution is a global optimum if the Sampson error can be globally minimized. Our approach appears to be the same in spirit as the extended HEIV of Matei and Meer [27], but we explicitly take advantage of the correspondence between ML in the $\mathbf{x}$-space and ML in the ξ-space.

Note that minimization of (10) is not affected by multiplication of $V[\xi_\alpha]$ by any positive constant. Similarly, $V[\mathbf{x}_\alpha]$ in (13) can be multiplied by any positive constant. Hence, the scales of $V[\xi_\alpha]$ and $V[\mathbf{x}_\alpha]$ are not important. However, they must be related by (7), which plays a key role in our subsequent analysis. In other words, we need to know $V[\xi_\alpha]$ and $V[\mathbf{x}_\alpha]$ up to a *common* scale.

7 Orthogonal projection computation

Our goal is to orthogonally project each observation $\mathbf{x}_\alpha$ onto the hypersurface $\mathscr{S}$ defined by (14) in the $\mathbf{x}$-space, where the "orthogonality" is defined with respect to the covariance matrix $V[\mathbf{x}_\alpha]$. The precise meaning is as follows. Let us call the direction specified by

$$\mathbf{n} = \mathbf{V}\Big(\frac{\partial \xi}{\partial \mathbf{x}}\Big)^\top \theta \tag{15}$$

the $\mathbf{V}$-*normal* of $\mathscr{S}$ at $\mathbf{x} \in \mathscr{S}$. It is easily seen that $(\mathbf{t}, \mathbf{V}\mathbf{n}) = 0$ for any tangent vector $\mathbf{t}$ at $\mathbf{x}$ (See Appendix). Our goal is to find for each $\mathbf{x}_\alpha$ a point $\bar{\mathbf{x}}_\alpha \in \mathscr{S}$ such that $\bar{\mathbf{x}} - \mathbf{x}_\alpha$ is in the direction of the $V[\mathbf{x}_\alpha]$-normal at $\bar{\mathbf{x}}_\alpha$. Such a point $\mathbf{x}_\alpha$, if it uniquely exists, minimizes $(\bar{\mathbf{x}} - \mathbf{x}_\alpha, V[\mathbf{x}_\alpha]^{-1}(\bar{\mathbf{x}} - \mathbf{x}_\alpha))$. However, we do not know the value of θ that defines $\mathscr{S}$; it should be determined so that (13) is minimized. We solve this problem by the following iterative procedure.

We identify $\mathscr{S}$ with a "level set" of $(\xi(\mathbf{x}), \theta) = c$ for $c = 0$ and view the $\mathbf{x}$-space as filled with continuously varying level sets with varying c (*foliation* in mathematical terms). The observation $\mathbf{x}_\alpha$ is on the surface (or *leaf*) $\mathscr{S}_\alpha$: $(\xi(\mathbf{x}), \theta) = c_\alpha$, where $c_\alpha = (\xi_\alpha, \theta)$. We first project $\mathbf{x}_\alpha$ onto $\mathscr{S}$ in the direction of the $V[\mathbf{x}_\alpha]$-normal at $\mathbf{x}_\alpha \in \mathscr{S}_\alpha$, where the value of θ is not specified yet. The projection $\hat{\mathbf{x}}_\alpha = \mathbf{x}_\alpha - \lambda_\alpha \mathbf{n}_\alpha$ is given by computing the magnitude λ_α so as to satisfy

$$(\xi(\mathbf{x}_\alpha - \lambda_\alpha \mathbf{n}_\alpha), \theta) = 0. \tag{16}$$

If θ were given a specific value, we could determine λ_α by numerical line search. If each component of $\xi(\xi)$ is a polynomial of ξ, we might even be able to compute λ_α analytically. Here, however, we need to express λ_α as an explicit function of θ to be optimized later. Assuming that λ_α is small and using the Taylor expansion

$$\xi(\mathbf{x}_\alpha - \lambda_\alpha \mathbf{n}_\alpha) = \xi_\alpha - \lambda_\alpha \Big(\frac{\partial \xi}{\partial \mathbf{x}}\Big)_\alpha \mathbf{n}_\alpha + \cdots, \tag{17}$$

we express λ_α to a first approximation in the form

$$\begin{aligned}\lambda_\alpha &= \frac{(\xi_\alpha, \theta)}{((\partial \xi/\partial \mathbf{x})_\alpha \mathbf{n}_\alpha, \theta)} = \frac{(\xi_\alpha, \theta)}{((\partial \xi/\partial \mathbf{x})_\alpha V[\mathbf{x}_\alpha](\partial \xi/\partial \mathbf{x})_\alpha^\top \theta, \theta)} \\ &= \frac{(\xi_\alpha, \theta)}{(\theta, V[\xi_\alpha]\theta)},\end{aligned} \tag{18}$$

where the identity (7) is used. The reprojection error of the thus computed $\hat{\mathbf{x}}_\alpha$, $\alpha = 1, ..., N$, is

$$\begin{aligned}E &= \sum_{\alpha=1}^N (\mathbf{x}_\alpha - \hat{\mathbf{x}}_\alpha, V[\mathbf{x}_\alpha]^{-1}(\mathbf{x}_\alpha - \hat{\mathbf{x}}_\alpha)) \\ &= \sum_{\alpha=1}^N (\lambda_\alpha \mathbf{n}_\alpha, V[\mathbf{x}_\alpha]^{-1} \lambda_\alpha \mathbf{n}_\alpha) \\ &= \sum_{\alpha=1}^N \lambda_\alpha^2 (V[\mathbf{x}_\alpha]\Big(\frac{\partial \xi}{\partial \mathbf{x}}\Big)_\alpha^\top \theta, V[\mathbf{x}_\alpha]^{-1} V[\mathbf{x}_\alpha]\Big(\frac{\partial \xi}{\partial \mathbf{x}}\Big)_\alpha^\top \theta) \\ &= \sum_{\alpha=1}^N \lambda_\alpha^2 (\theta, \Big(\frac{\partial \xi}{\partial \mathbf{x}}\Big)_\alpha V[\mathbf{x}_\alpha]\Big(\frac{\partial \xi}{\partial \mathbf{x}}\Big)_\alpha^\top \theta) \\ &= \sum_{\alpha=1}^N \lambda_\alpha^2 (\theta, V[\xi_\alpha]\theta) \\ &= \sum_{\alpha=1}^N \frac{(\xi_\alpha, \theta)^2}{(\theta, V[\xi_\alpha]\theta)},\end{aligned} \tag{19}$$

which is nothing but the Sampson error (10). At this stage, we assign to θ the value $\hat{\theta}$ that minimizes (19) over the entire domain of θ, using some Sampson error minimization method such as FNS or HEIV. Then, the orthogonal projection is given by

$$\hat{\mathbf{x}}_\alpha = \mathbf{x}_\alpha - \frac{(\xi_\alpha, \hat{\theta}) V[\mathbf{x}_\alpha]}{(\hat{\theta}, V[\xi_\alpha]\hat{\theta})} \Big(\frac{\partial \xi}{\partial \mathbf{x}}\Big)_\alpha^\top \hat{\theta}. \tag{20}$$

8 Strict orthogonal projection

Now the nature of the Sampson error has become clear. The *core* of our algorithm is to iteratively modify it so that it *coincides* with the reprojection error. In the preceding computation, we projected $\mathbf{x}_\alpha$ along the $V[\mathbf{x}_\alpha]$-normal of the level set $\mathscr{S}_\alpha$ at $\mathbf{x}_\alpha$, while our goal is to project $\mathbf{x}_\alpha$ along the $V[\mathbf{x}_\alpha]$-normal of the hypersurface $\mathscr{S}$ itself at $\bar{\mathbf{x}}_\alpha$, which is unknown yet. So, we project $\mathbf{x}_\alpha$ along the $V[\mathbf{x}_\alpha]$-normal of the level set $\hat{\mathscr{S}}_\alpha$ at $\hat{\mathbf{x}}_\alpha$ that we have just computed (Fig. 4). To do this, we again view θ as a mere variable whose value is unspecified; the value assigned to it in the preceding computation is discarded. The $V[\mathbf{x}_\alpha]$-normal of $\hat{\mathscr{S}}_\alpha$ at $\hat{\mathbf{x}}_\alpha$ is

$$\hat{\mathbf{n}}_\alpha = V[\mathbf{x}_\alpha]\Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha^\top\theta, \tag{21}$$

where $(\partial\hat{\xi}/\partial\mathbf{x})_\alpha$ is the Jacobian matrix $(\partial\xi/\partial\mathbf{x})$ evaluated at $\hat{\mathbf{x}}_\alpha$. We project the *original* $\mathbf{x}_\alpha$, *not* the computed $\hat{\mathbf{x}}_\alpha$, onto $\mathscr{S}$ in the form $\hat{\mathbf{x}}_\alpha = \mathbf{x}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha$ (Fig. 4) and determine $\hat{\lambda}_\alpha$ so as to satisfy

$$(\xi(\mathbf{x}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha), \theta) = 0. \tag{22}$$

If $\hat{\mathbf{x}}_\alpha$ is a good approximation to the true solution $\bar{\mathbf{x}}_\alpha$ as compared with the original data $\mathbf{x}_\alpha$, we have $\|\hat{\bar{\mathbf{x}}}_\alpha - \hat{\mathbf{x}}_\alpha\| \ll \|\hat{\bar{\mathbf{x}}}_\alpha - \mathbf{x}_\alpha\|$ (Fig. 4). Hence, if we write

$$\begin{aligned}\xi(\mathbf{x}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha) &= \xi(\hat{\mathbf{x}}_\alpha + \mathbf{x}_\alpha - \hat{\mathbf{x}}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha)\\ &= \xi(\hat{\mathbf{x}}_\alpha + \tilde{\mathbf{x}}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha),\end{aligned} \tag{23}$$

where we define

$$\tilde{\mathbf{x}}_\alpha \equiv \mathbf{x}_\alpha - \hat{\mathbf{x}}_\alpha, \tag{24}$$

then $\tilde{\mathbf{x}}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha$ is a small quantity of a higher order. Using the Taylor expansion

$$\xi(\mathbf{x}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha) = \hat{\xi}_\alpha + \Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha(\tilde{\mathbf{x}}_\alpha - \hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha) + \cdots, \tag{25}$$

where $\hat{\xi}_\alpha \equiv \xi(\hat{\mathbf{x}}_\alpha)$, we obtain from (22) to a first approximation the expression

$$\hat{\lambda}_\alpha = \frac{(\hat{\xi}_\alpha, \theta) + (\theta, (\partial\hat{\xi}/\partial\mathbf{x})_\alpha\tilde{\mathbf{x}}_\alpha)}{(\theta, V[\hat{\xi}_\alpha]\theta)} = \frac{(\hat{\xi}^*_\alpha, \theta)}{(\theta, V[\hat{\xi}_\alpha]\theta)}. \tag{26}$$

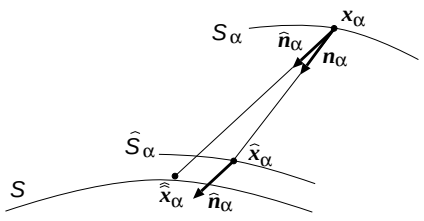


Fig. 4 Successive projection along the $V[\mathbf{x}_\alpha]$-normal of the level set.

Here, we define

$$\begin{aligned}\hat{\xi}^*_\alpha &\equiv \hat{\xi}_\alpha + \Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha\tilde{\mathbf{x}}_\alpha,\\ V[\hat{\xi}_\alpha] &\equiv \Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha V[\mathbf{x}_\alpha]\Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha^\top.\end{aligned} \tag{27}$$

The reprojection error of the thus computed $\hat{\bar{\mathbf{x}}}_\alpha$, α = 1, ..., N, is

$$\begin{aligned}E &= \sum_{\alpha=1}^N (\mathbf{x}_\alpha - \hat{\bar{\mathbf{x}}}_\alpha, V[\mathbf{x}_\alpha]^{-1}(\mathbf{x}_\alpha - \hat{\bar{\mathbf{x}}}_\alpha))\\ &= \sum_{\alpha=1}^N (\hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha, V[\mathbf{x}_\alpha]^{-1}\hat{\lambda}_\alpha\hat{\mathbf{n}}_\alpha)\\ &= \sum_{\alpha=1}^N \hat{\lambda}_\alpha^2(V[\mathbf{x}_\alpha]\Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha^\top\theta, V[\mathbf{x}_\alpha]^{-1}V[\mathbf{x}_\alpha]\Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha^\top\theta)\\ &= \sum_{\alpha=1}^N \hat{\lambda}_\alpha^2(\theta, \Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha V[\mathbf{x}_\alpha]\Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha^\top\theta)\\ &= \sum_{\alpha=1}^N \hat{\lambda}_\alpha^2(\theta, V[\hat{\xi}_\alpha]\theta) = \sum_{\alpha=1}^N \frac{(\hat{\xi}^*_\alpha, \theta)^2}{(\theta, V[\hat{\xi}_\alpha]\theta)}.\end{aligned} \tag{28}$$

Again, this has the same form as (12), so we assign to θ, whose value has been unspecified, the value $\hat{\hat{\theta}}$ that minimizes (28) over the entire domain of θ, using some Sampson error minimization method such as FNS or HEIV. Then, the true value $\bar{\mathbf{x}}$ is estimated to be

$$\hat{\bar{\mathbf{x}}}_\alpha = \mathbf{x}_\alpha - \frac{(\hat{\xi}^*_\alpha, \hat{\hat{\theta}})V[\mathbf{x}_\alpha]}{(\hat{\hat{\theta}}, V[\hat{\xi}_\alpha]\hat{\hat{\theta}})}\Big(\frac{\partial\hat{\xi}}{\partial\mathbf{x}}\Big)_\alpha^\top\hat{\hat{\theta}}. \tag{29}$$

Now, we again discard the value $\hat{\hat{\theta}}$, viewing θ as a mere variable. Identifying $\hat{\bar{\mathbf{x}}}_\alpha$ with $\hat{\mathbf{x}}_\alpha$, we repeat the same process until it converges. In the end, $\hat{\mathbf{x}}_\alpha$ is on the surface $\mathscr{S}$: $(\xi(\mathbf{x}), \theta)$ = 0 and is a projection of $\mathbf{x}_\alpha$ along the $V[\mathbf{x}_\alpha]$-normal of $\mathscr{S}$ itself at $\hat{\mathbf{x}}_\alpha$, i.e., the desired exact orthogonal projection. From (28), we see that strict ML in the $\mathbf{x}$-space coincides with *Sampson error minimization in the modified* $\hat{\xi}^*$*-space*. As we can see from (27), however, the mapping from $\mathbf{x}$ to $\hat{\xi}^*$ is defined only *dynamically* in the course of iterations.

9 Examples of strict ML

We now apply the above procedure to typical examples, where we assume that the x and y coordinates of each point have independent and identical Gaussian noise of mean 0

and variance σ^2. Then, the Mahalanobis distance (13) in the **x**-space coincides with the Euclidean distance. The noise variance σ^2 need not be known, since minimization of (13) is not affected by multiplication of $V[\mathbf{x}_\alpha]$ by any positive constant. So, we regard σ to be 1 in the computation.

Example 5 (Ellipse Fitting) The procedure for fitting an ellipse to a point sequence (x_α, y_α), $\alpha = 1, ..., N$, is obtained as follows[5]. We remove the scale indeterminacy of the ellipse parameter θ in (4) by normalizing it to $\|\theta\| = 1$:

1. Let $E_0 = \infty$ (a sufficiently large number), $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = 0$, $\alpha = 1, ..., N$.
2. Let $V[\hat{\xi}_\alpha]$ be the matrix obtained by substituting $\hat{x}_\alpha$ and $\hat{y}_\alpha$ for x_α and y_α, respectively, in (8).
3. Compute the following ξ^*_α, $\alpha = 1, ..., N$:

$$\xi^*_\alpha = \begin{pmatrix} \hat{x}^2_\alpha + 2\hat{x}_\alpha\tilde{x}_\alpha \\ 2(\hat{x}_\alpha\hat{y}_\alpha + \hat{y}_\alpha\tilde{x}_\alpha + \hat{x}_\alpha\tilde{y}_\alpha) \\ \hat{y}^2_\alpha + 2\hat{y}_\alpha\tilde{y}_\alpha \\ 2(\hat{x}_\alpha + \tilde{x}_\alpha) \\ 2(\hat{y}_\alpha + \tilde{y}_\alpha) \\ 1 \end{pmatrix}. \tag{30}$$

4. Compute the 6-D unit vector $\theta = (\theta_i)$ that minimizes the following function (e.g., by FNS or HEIV):

$$E(\theta) = \sum_{\alpha=1}^{N} \frac{(\theta, \xi^*_\alpha)^2}{(\theta, V[\hat{\xi}_\alpha]\theta)}. \tag{31}$$

5. Update $\tilde{x}_\alpha$, $\tilde{y}_\alpha$, $\hat{x}_\alpha$, and $\hat{y}_\alpha$ in the form

$$\begin{pmatrix} \tilde{x}_\alpha \\ \tilde{y}_\alpha \end{pmatrix} \leftarrow \frac{2(\theta, \xi^*_\alpha)}{(\theta, V[\hat{\xi}_\alpha]\theta)} \begin{pmatrix} \theta_1 & \theta_2 & \theta_4 \\ \theta_2 & \theta_3 & \theta_5 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ 1 \end{pmatrix}, \tag{32}$$

$$\hat{x}_\alpha \leftarrow x_\alpha - \tilde{x}_\alpha, \qquad \hat{y}_\alpha \leftarrow y_\alpha - \tilde{y}_\alpha. \tag{33}$$

6. Compute the reprojection error

$$E = \sum_{\alpha=1}^{N} (\tilde{x}^2_\alpha + \tilde{y}^2_\alpha). \tag{34}$$

7. If $E \approx E_0$, return θ and stop. Else, let $E_0 \leftarrow E$ and go back to Step 2.

This scheme was obtained by Kanatani and Sugaya [23] by using differentiation and Lagrange multipliers based on a specific analysis for ellipse fitting. Here, it is derived as a special case of our general theory.

As pointed out by Kanatani and Sugaya [23], however, the resulting accuracy is practically the same as Sampson error minimization; the difference is only in the last few significant digits. Figure 5(a) shows 20 points on an ellipse, and Fig. 5(b) shows the RMS error of the fitted ellipse over 1000 trials with independent Gaussian noise of mean 0 and standard deviation σ (pixels) added to the x and y coordinates of each point. The plots of ML in the ξ-space and ML in the **x**-space completely overlap. For the Sampson error minimization, we used the FNS of Chojnacki et al. [7]. As a comparison, the RMS error of the least squares (or algebraic distance minimization [15]) is also plotted. The dotted line shows the theoretical accuracy limit (*KCR lower bound* [6, 16, 17]).

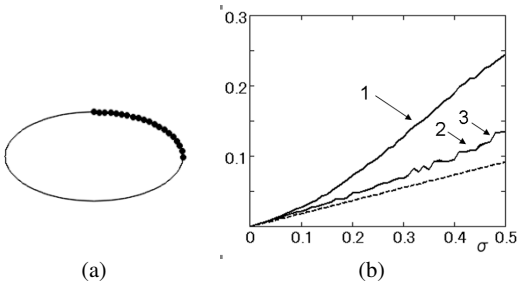


Fig. 5 (a) 20 points on an ellipse. (b) RMS error of the fitted ellipse over 1000 trials vs. the noise level σ. *1.* Least squares. *2.* ML in the **x**-space. *3.* ML in the ξ-space. The dotted line shows the KCR lower bound.

Example 6 (Fundamental Matrix Computation) The vector θ in (6) that encodes the fundamental matrix **F** of rank 2 is computed from corresponding points (x_α, y_α) and (x'_α, y'_α), $\alpha = 1, ..., N$, as follows[5]. We remove the scale indeterminacy of θ by normalizing it to $\|\theta\| = 1$:

1. Let $E_0 = \infty$ (a sufficiently large number), $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, $\hat{x}'_\alpha = x'_\alpha$, $\hat{y}'_\alpha = y'_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = \tilde{x}'_\alpha = \tilde{y}'_\alpha = 0$, $\alpha = 1, ..., N$.
2. Let $V[\hat{\xi}_\alpha]$ be the matrix obtained by substituting $\hat{x}_\alpha$, $\hat{y}_\alpha$, $\hat{x}'_\alpha$, and $\hat{y}'_\alpha$ for x_α, y_α, x'_α, and y'_α, respectively, in (9).
3. Compute the following ξ^*_α, $\alpha = 1, ..., N$:

$$\xi^*_\alpha = \begin{pmatrix} \hat{x}_\alpha\hat{x}'_\alpha + \hat{x}'_\alpha\tilde{x}_\alpha + \hat{x}_\alpha\tilde{x}'_\alpha \\ \hat{x}_\alpha\hat{y}'_\alpha + \hat{y}'_\alpha\tilde{x}_\alpha + \hat{x}_\alpha\tilde{y}'_\alpha \\ \hat{x}_\alpha + \tilde{x}_\alpha \\ \hat{y}_\alpha\hat{x}'_\alpha + \hat{x}'_\alpha\tilde{y}_\alpha + \hat{y}_\alpha\tilde{x}'_\alpha \\ \hat{y}_\alpha\hat{y}'_\alpha + \hat{y}'_\alpha\tilde{y}_\alpha + \hat{y}_\alpha\tilde{y}'_\alpha \\ \hat{y}_\alpha + \tilde{y}_\alpha \\ \hat{x}'_\alpha + \tilde{x}'_\alpha \\ \hat{y}'_\alpha + \tilde{y}'_\alpha \\ 1 \end{pmatrix}. \tag{35}$$

4. Compute the 9-D unit vector $\theta = (\theta_i)$ that minimizes the following function subject to the constraint that the re-

[5] The source code is available at:
`http://www.iim.cs.tut.ac.jp/~sugaya/public-e.html`

sulting fundamental matrix $\mathbf{F}$ has rank 2 (e.g., by EFNS [21]):

$$E(\theta) = \sum_{\alpha=1}^{N} \frac{(\theta, \xi_\alpha^*)^2}{(\theta, V[\hat{\xi}_\alpha]\theta)}. \tag{36}$$

5. Update $\tilde{x}_\alpha$, $\tilde{y}_\alpha$, $\tilde{x}'_\alpha$, $\tilde{y}'_\alpha$, $\hat{x}_\alpha$, $\hat{y}_\alpha$, $\hat{x}'_\alpha$, and $\hat{y}'_\alpha$ in the form

$$\begin{pmatrix} \tilde{x}_\alpha \\ \tilde{y}_\alpha \end{pmatrix} \leftarrow \frac{(\theta, \xi_\alpha^*)}{(\theta, V[\hat{\xi}_\alpha]\theta)} \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{pmatrix} \begin{pmatrix} \hat{x}'_\alpha \\ \hat{y}'_\alpha \\ 1 \end{pmatrix},$$

$$\begin{pmatrix} \tilde{x}'_\alpha \\ \tilde{y}'_\alpha \end{pmatrix} \leftarrow \frac{(\theta, \xi_\alpha^*)}{(\theta, V[\hat{\xi}_\alpha]\theta)} \begin{pmatrix} \theta_1 & \theta_4 & \theta_7 \\ \theta_2 & \theta_5 & \theta_8 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ 1 \end{pmatrix}, \tag{37}$$

$$\hat{x}_\alpha \leftarrow x_\alpha - \tilde{x}_\alpha, \quad \hat{y}_\alpha \leftarrow y_\alpha - \tilde{y}_\alpha,$$
$$\hat{x}'_\alpha \leftarrow x'_\alpha - \tilde{x}'_\alpha, \quad \hat{y}'_\alpha \leftarrow y'_\alpha - \tilde{y}'_\alpha. \tag{38}$$

6. Compute the reprojection error

$$E = \sum_{\alpha=1}^{N} (\tilde{x}_\alpha^2 + \tilde{y}_\alpha^2 + \tilde{x}_\alpha'^2 + \tilde{y}_\alpha'^2). \tag{39}$$

7. If $E \approx E_0$, return θ and stop. Else, let $E_0 \leftarrow E$ and go back to Step 2.

This scheme was obtained by Kanatani and Sugaya [24] by using differentiation and Lagrange multipliers based on a specific analysis for fundamental matrix computation. Here, it is derived as a special case of our general theory.

As pointed out by Kanatani and Sugaya [24], however, the resulting accuracy is practically the same as Sampson error minimization; the difference is only in the last few significant digits. Figure 6(a) shows a curved grid viewed from two positions. Figure 6(b) plots the RMS error of the computed fundamental matrix over 1000 trials with independent Gaussian noise of mean 0 and standard deviation σ (pixels) added to the x and y coordinates of each grid point. Again, the plots of ML in the ξ-space and ML in the $\mathbf{x}$-space completely overlap. For the Sampson error minimization with the rank constraint, we used the EFNS of Kanatani and Sugaya [21]. As a comparison, the RMS error of the least squares (known as algebraic distance minimization [15] or Hartley's 8-point algorithm [12]) is also plotted. The dotted line shows the KCR lower bound.

We also conducted conventional bundle adjustment. First, we computed the fundamental matrix by least squares, from which we computed the focal lengths of the two frames, the relative camera displacement and rotation, and reconstructed the 3-D positions of the grid points. Then, we adjusted them in their joint space of dimension 280 (2 for focal lengths, 2 for the translation, 3 for the rotation, and 273 for the 91 grid points) by the Levenberg-Marquardt method so that the reprojection error is minimized.

Figure 6(c) shows an example of convergence for σ = 0.1. The marks $*$ indicate the reprojection error (in pixel per point per frame) in each iteration. For comparison, we started the iteration from the true values of the 280 parameters; the marks $\square$ indicate the resulting reprojection error. The marks $\bullet$ indicate the reprojection error of our orthogonal projection. All converged to the same solution. The numerical values of the reprojection errors are listed in Fig. 6(d). Our orthogonal projection converged to 9 significant digits after three iterations, while bundle adjustment converged to the same significant digits after four iterations when starting from the least squares estimates and after seven iterations starting from the true values.

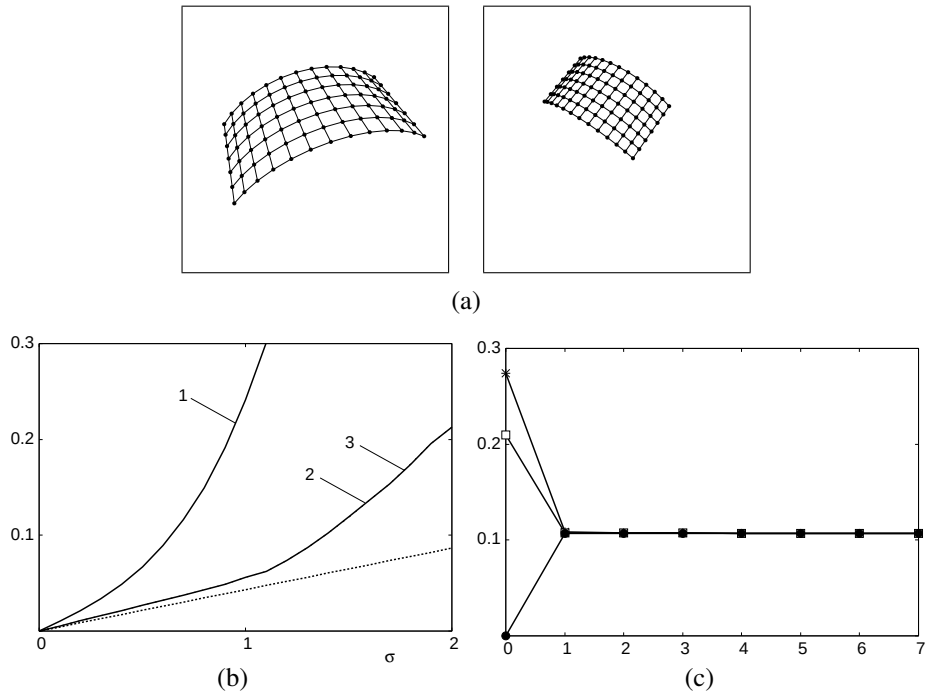
For both examples, we should note the following:

- During the iterations of orthogonal projection, the reprojection error E generally *increases*. By definition, E is the sum of square Mahalanobis (Euclidean in the above examples) distances between the observation $\mathbf{x}_\alpha$ and our guess $\hat{\mathbf{x}}_\alpha$. We start by regarding $\mathbf{x}_\alpha$ as our guess $\hat{\mathbf{x}}_\alpha$, so initially E is 0. Since the guess does not satisfy the constraint, we modify $\hat{\mathbf{x}}_\alpha$ so as to satisfy it. Maximum likelihood means that we do this by increasing E *by a minimal amount*. The increase of E may not always be monotonic; it can oscillate near the convergence. In each step, on the other hand, we *minimize E with respect to θ*. Optimization with constraints is known to involve both minimization and maximization [5].
- In both examples, the Sampson error minimization in Step 4 is iterative, so we stop it if the update of θ is smaller than a specified threshold. However, the Sampson error $E(\theta)$ in the form of (31) and (36) does not depend on the sign of θ, so it may happen that new value of θ has an opposite direction to the previous one. Hence, we must align their orientations of before comparing them.
- In Step 7 in both examples, we stop the iteration when the change of E is small enough. We could stop when the current value of θ is close to the previous one, but we must be careful about the *interference* of the nested iteration loops. If we stop the outer loop by demanding a stricter threshold on θ than can be afforded by the inner Sampson error minimization, the outer iteration may continue forever. No such interferences can occur if we stop the inner loop by θ and the outer loop by E.

10 Application to optimal correction

As a byproduct, our strict ML leads to a new numerical scheme for *optimal correction* [16]: we optimally correct a given $\mathbf{x}$ so as to satisfy the constraint $(\xi(\mathbf{x}), \theta) = 0$, where the parameter θ is given and fixed. For this, we minimize

$$E = (\mathbf{x} - \bar{\mathbf{x}}, V[\mathbf{x}]^{-1}(\mathbf{x} - \bar{\mathbf{x}})), \tag{40}$$



	●	∗	□
0	0.0000000000000	0.2740543086661	0.2097120015146
1	0.1071688468318	0.1083766529404	0.1075557226703
2	0.1071686014356	0.1076009069457	0.1073337819788
3	0.1071686015030	0.1076005713017	0.1072415103709
4	0.1071686013682	0.1071718714030	0.1071691179312
5	0.1071686015030	0.1071686014673	0.1071691179312
6	0.1071686013682	0.1071686014580	0.1071686014609
7	0.1071686016378	0.1071686014580	0.1071686014580

(d)[0.2cm]

Fig. 6 (a) Curved grids viewed from two angles. (b) RMS error of the fitted fundamental matrix over 1000 trials vs. the noise level σ. *1.* Least squares. *2.* ML in the **x**-space. *3.* ML in the ξ-space. The dotted line shows the KCR lower bound. (c) An example of reprojection error convergence for $\sigma = 0.1$ (pixel) vs. the number of iterations. ●: our orthogonal projection. ∗: bundle adjustment starting from least-squares estimates. □: bundle adjustment starting from true values. (d) Numerical values of the reprojection errors in (c).

subject to

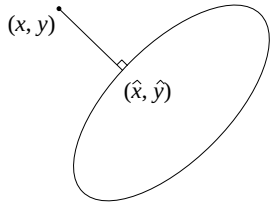
$$(\xi(\bar{\mathbf{x}}), \theta) = 0, \tag{41}$$

for a given θ. All we need is to remove the computation of θ in the procedure described in Sect. 7.

Example 7 (Perpendicular to an Ellipse) Given a point (x, y) and an ellipse in the form of (3), we want to compute the foot $(\hat{x}, \hat{y})$ of the perpendicular from (x, y) (Fig. 7). It is computed as follows (θ is defined by (4)):

1. Let $E_0 = \infty$ (a sufficiently large number), $\hat{x} = x$, $\hat{y} = y$, and $\tilde{x} = \tilde{y} = 0$.
2. Let $V[\hat{\xi}]$ be the matrix obtained by substituting $\hat{x}$ and $\hat{y}$ for x_α and y_α, respectively, in (8).

Fig. 7 Drawing a perpendicular to an ellipse.



3. Compute the following ξ^*:

$$\xi^* = \begin{pmatrix} \hat{x}^2 + 2\hat{x}\tilde{x} \\ 2(\hat{x}\hat{y} + \hat{y}\tilde{x} + \hat{x}\tilde{y}) \\ \hat{y}^2 + 2\hat{y}\tilde{y} \\ 2(\hat{x} + \tilde{x}) \\ 2(\hat{y} + \tilde{y}) \\ 1 \end{pmatrix}. \tag{42}$$

4. Update $\tilde{x}$, $\tilde{y}$, $\hat{x}$, and $\hat{y}$ in the form

$$\begin{pmatrix}\tilde{x}\\ \tilde{y}\end{pmatrix} \leftarrow \frac{2(\theta,\xi^*)}{(\theta,V[\hat{\xi}]\theta)} \begin{pmatrix}\theta_1 & \theta_2 & \theta_4\\ \theta_2 & \theta_3 & \theta_5\end{pmatrix} \begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix}, \tag{43}$$

$$\hat{x} \leftarrow x-\tilde{x}, \quad \hat{y} \leftarrow y-\tilde{y}. \tag{44}$$

5. Compute the reprojection error

$$E = \tilde{x}^2+\tilde{y}^2. \tag{45}$$

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

The perpendicular to an ellipse can be obtained by solving simultaneous algebraic equations. It seems, however, the above simple procedure has not been known. The computation converges after at most two or three iterations (usually with one iteration), but even the first solution has sufficient accuracy for practical use.

Example 8 (Triangulation) When the fundamental matrix **F** is known and a noisy correspondence pair (x,y) and (x',y') is given, we can optimally reconstruct its 3-D position by minimally correcting (x,y) and (x',y') so as to satisfy the epipolar equation in (5) (Fig. 8(a)), since the lines of sight determined by points (x,y) and (x',y') can intersect if and only if the epipolar equation in (5) holds [15]. Today, there are still many who use the non-optimal method of regarding the "midpoint" of the shortest line segment connecting the two lines of sight as the intersection (Fig. 8(b)).

The optimally corrected positions $(\hat{x},\hat{y})$ and $(\hat{x}',\hat{y}')$ that minimize the sum of square distances from (x,y) and (x',y') are computed as follows (θ is defined by (6)):

1. Let $E_0 = \infty$ (a sufficiently large number), $\hat{x} = x$, $\hat{y} = y$, $\hat{x}' = x'$, and $\hat{y}' = y'$, $\tilde{x} = \tilde{y} = \tilde{x}' = \tilde{y}' = 0$.
2. Let $V[\hat{\xi}]$ be the matrix obtained by substituting $\hat{x}$, $\hat{y}$, $\hat{x}'$, and $\hat{y}'$ for x_α, y_α, x'_α, and y'_α, respectively, in (9).

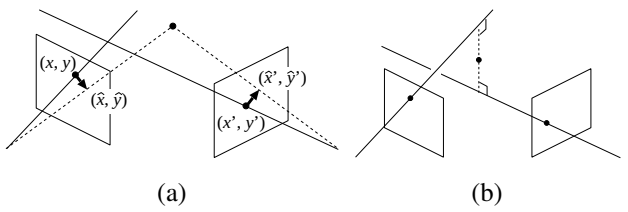


Fig. 8 Computing the 3-D position from noisy correspondence pair. (a) Optimal triangulation. (b) The mid-point method.

3. Compute the following ξ^*:

$$\xi^* = \begin{pmatrix}\hat{x}\hat{x}'+\hat{x}'\tilde{x}+\hat{x}\tilde{x}'\\ \hat{x}\hat{y}'+\hat{y}'\tilde{x}+\hat{x}\tilde{y}'\\ \hat{x}+\tilde{x}\\ \hat{y}\hat{x}'+\hat{x}'\tilde{y}+\hat{y}\tilde{x}'\\ \hat{y}\hat{y}'+\hat{y}'\tilde{y}+\hat{y}\tilde{y}'\\ \hat{y}+\tilde{y}\\ \hat{x}'+\tilde{x}'\\ \hat{y}'+\tilde{y}'\\ 1\end{pmatrix}. \tag{46}$$

4. Update $\tilde{x}$, $\tilde{y}$, $\tilde{x}'$, $\tilde{y}'$, $\hat{x}$, $\hat{y}$, $\hat{x}'$, and $\hat{y}'$ in the form

$$\begin{pmatrix}\tilde{x}\\ \tilde{y}\end{pmatrix} \leftarrow \frac{(\theta,\xi^*)}{(\theta,V[\hat{\xi}]\theta)} \begin{pmatrix}\theta_1 & \theta_2 & \theta_3\\ \theta_4 & \theta_5 & \theta_6\end{pmatrix} \begin{pmatrix}\hat{x}'\\ \hat{y}'\\ 1\end{pmatrix},$$

$$\begin{pmatrix}\tilde{x}'\\ \tilde{y}'\end{pmatrix} \leftarrow \frac{(\theta,\xi^*)}{(\theta,V[\hat{\xi}]\theta)} \begin{pmatrix}\theta_1 & \theta_4 & \theta_7\\ \theta_2 & \theta_5 & \theta_8\end{pmatrix} \begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix}, \tag{47}$$

$$\begin{aligned}\hat{x} &\leftarrow x-\tilde{x}, & \hat{y} &\leftarrow y-\tilde{y},\\ \hat{x}' &\leftarrow x'-\tilde{x}', & \hat{y}' &\leftarrow y'-\tilde{y}'.\end{aligned} \tag{48}$$

5. Compute the reprojection error

$$E = \tilde{x}^2+\tilde{y}^2+\tilde{x}'^2+\tilde{y}'^2. \tag{49}$$

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

This procedure is nothing but the stereo triangulation of Kanatani et al. [25], who derived this algorithm by directly minimizing the reprojection error, using differentiation and Lagrange multipliers. Here, it is derived as a special case of our general theory. A popular method for optimal triangulation is due to Hartley and Sturm [14], who determined the epipolar lines of the corresponding points by algebraically solving a 6-degree polynomial. Kanatani et al. [25] experimentally confirmed that their solution is identical to that of Hartley and Sturm [14] yet the computation is significantly faster (Fig. 9).

11 Observations

We summarize the main characteristics of our approach:

– Accuracy: We compute the strict ML solution, so the accuracy is the same as all existing ML-based methods. Our approach can be regarded as a special implementation of the "Gold Standard" of Hartley and Zisserman [15]. Our Example 6 demonstrates that our computation converged to the same solution as bundle adjustment. We have also confirmed for ellipse fitting that our solution is

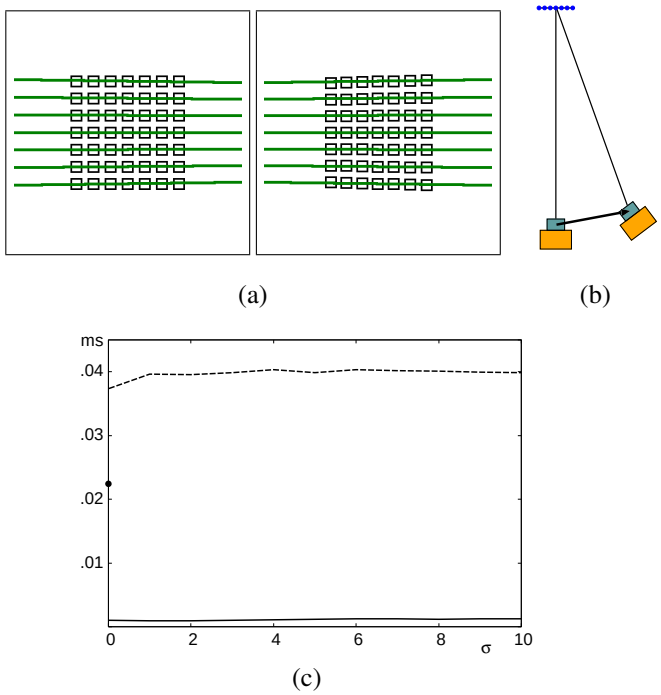


Fig. 9 (a) Stereo images of a planar grid. Some of the epipolar lines are drawn. (b) Top view of the camera configuration. (c) Computation time (ms) for triangulation per point (average over 1000 trials). The CPU is Intel Core2Duo E6700 2.66GHz. The horizontal axis is for the standard deviation σ of the added noise. Solid line: our method. Dashed line: The Hartley-Sturm method. The black dot is for the exceptional behavior for $\sigma = 0$.

identical to those by others such as Gander et al. [10] and Sturm and Gargallo [30].

– Global optimality: Our approach is to iteratively change the variables in the data space so that the Sampson error coincides with the reprojection error. Hence, the solution is globally optimal *provided* the Sampson error minimizer returns a global minimum. However, popular tools such as FNS and HEIV are not guaranteed to reach a global minimum. Of course, we could globally minimize the Sampson error in each iteration, using, say, branch and bound, but that would make the process extremely inefficient.
– Efficiency: We cannot draw a universal conclusion if our approach is more efficient than others, because the efficiency heavily depends on particular applications and implementations. For example, the efficiency of bundle adjustment can be greatly improved by appropriate preprocessing by considering the particularities of the problem. The efficiency of our approach, on the other hand, critically depends on the efficiency of the Sampson error minimizer we use. According to our experience, however, our approach is significantly faster than bundle adjustment in many examples. The computation is done in a low dimensional data space, arriving at the solution after one or two iterations.
– Convergence: It is difficult to give a mathematical proof of the convergence of our orthogonal projection, because it is a property of the hypersurface $\mathscr{S}$ defined by the constraint, rather than a property of the operation. The hypersurface $\mathscr{S}$ may have singularities, e.g., for the epipolar constraint, $\mathscr{S}$ is hyperbolic in 4-D with singularities at the epipoles. The iteration should converge if $\mathscr{S}$ is more or less flat and its supporting function is nearly linear, or equivalently, if the observations are sufficiently close to $\mathscr{S}$, i.e., if the noise is small. However, it is difficult to bound the noise level to insure convergence in terms of the differential characteristics of $\mathscr{S}$ such as the curvature. In our experience, however, the orthogonal projection converges after at most two or three iterations (mostly with one iteration). On the other hand, we have frequently observed that the inner Sampson error minimization failed to converge in the presence of large noise. Popular Sampson error minimizers such as FNS and HEIV are not guaranteed to converge. Thus, the convergence of our approach is practically dictated by the Sampson error minimizer we use.

12 Conclusions

This paper has presented a new numerical scheme for strict ML computation for geometric fitting problems. Our approach is orthogonal projection of observations $\mathbf{x}_\alpha$ onto a parameterized surface $\mathscr{S}$ defined by the constraint in the $\mathbf{x}$-space, where the orthogonality is defined with respect to the covariance matrix $V[\mathbf{x}_\alpha]$ of $\mathbf{x}_\alpha$. This approach has been adopted by many researchers for a general constraint, but we have shown that if the constraint is linearly separable, the optimization can be done by repeated Sampson error minimization *in the dynamically defined* $\boldsymbol{\xi}^*$*-space*. We illustrated our procedure by applying it to ellipse fitting and fundamental matrix computation. We have also shown that our theory encompasses optimal correction problems, demonstrating that compact schemes are obtained for computing perpendiculars to an ellipse and optimally triangulating stereo images.

Our approach is problem-independent in the sense that the computation is based solely only the linearly separable constraint; apparently different problems, such as ellipse fitting and fundamental matrix computation, can be solved by the same procedure. This contrasts to bundle adjustment, for which we need to derive an explicit cost function from particularities of the problem and a high-dimensional space needs to be searched.

Since our approach is ML, the accuracy is the same as all other ML-based method. It is difficult to obtain a universal conclusion for efficiency, because it depends on applications and implementation. The efficiency and convergence of our

approach are practically determined by the performance of the Sampson error minimizer we use.

In this paper, Sampson error minimization is treated as a black box; we are not proposing any new Sampson error minimizer. Our finding here directs us to focus on improving Sampson error minimization. Once a good Sampson error minimizer is discovered, it is automatically upgraded to a good reprojection error minimizer by our theory. To have established this fact is the main contribution of this paper.

Acknowledgments

The authors thanks Mike Brooks and Wojciech Chojnacki of the University Adelaide, Australia, Nikolai Chernov of the University of Alabama at Birmingham, U.S.A., and Wolfgang Förstner of the University of Bonn, for helpful discussions. This work was supported in part by the Ministry of Education, Culture, Sports, Science, and Technology, Japan, under a Grant in Aid for Scientific Research (C 21500172).

Appendix. $V[\mathbf{x}_\alpha]$-normal

Define the orthogonality of vectors $\mathbf{a}$ and $\mathbf{b}$ by $(\mathbf{a}, \mathbf{G}\mathbf{b}) = 0$ for a positive definite symmetric matrix $\mathbf{G}$. The gradient of a surface $\mathscr{S}$: $F(\mathbf{x}) = 0$ is $\nabla_{\mathbf{x}} F$, and $(\mathbf{t}, \nabla_{\mathbf{x}} F) = 0$ holds for any tangent vector to $\mathscr{S}_\alpha$ at $\mathbf{x}$. If we let

$$\mathbf{n} = \mathbf{G}^{-1} \nabla_{\mathbf{x}} F, \tag{50}$$

we see that for any tangent vector $\mathbf{t}$ to $\mathscr{S}$ at $\mathbf{x}$

$$(\mathbf{t}, \mathbf{G}\mathbf{n}) = (\mathbf{t}, \mathbf{G}\mathbf{G}^{-1} \nabla_{\mathbf{x}} F) = (\mathbf{t}, \nabla_{\mathbf{x}} F) = 0. \tag{51}$$

Thus, $\mathbf{n}$ is orthogonal to any tangent vector $\mathbf{t}$ to $\mathscr{S}$ at $\mathbf{x}$. In our problem, $\mathbf{G} = V[\mathbf{x}_\alpha]^{-1}$ and $F = (\xi(\mathbf{x}), \theta) - c_\alpha$, so the gradient is $\nabla_{\mathbf{x}} F(\mathbf{x}_\alpha) = (\partial \xi / \partial \mathbf{x})_\alpha^\top \theta$ at $\mathbf{x}_\alpha$, and the $V[\mathbf{x}_\alpha]$-normal at $\mathbf{x}_\alpha$ is given by (15).

References

1. S. J. Ahn, W. Rauh, H. S. Cho, and H.-J. Warecke, Orthogonal distance fitting of implicit curves and surfaces, *IEEE Trans. Patt. Anal. Mach. Intell.*, 24 (2002), pp. 620–638.
2. A. Atieg and G. A. Watson, A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances, *J. Comp. Appl. Math.*, 158 (2003), pp. 277–296.
3. A. Bartoli and P. Sturm, Nonlinear estimation of fundamental matrix with minimal parameters, *IEEE Trans. Patt. Anal. Mach. Intell.*, 26 (2004), pp. 426–432.
4. A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, U.S.A., 1996.
5. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, U.K., 2004.
6. N. Chernov and C. Lesort, Statistical efficiency of curve fitting algorithms, *Comp. Stat. Data Anal.*, 47 (2004), pp. 713–728.
7. W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley, On the fitting of surfaces to data with covariances, *IEEE Trans. Patt. Anal. Mach. Intell.*, 22 (2000), pp. 1294–1303.
8. W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley, A new constrained parameter estimator for computer vision applications, *Image Vision Comp.*, 22 (2004), pp. 85–91.
9. W. Föstner, On weighting and choosing constraints for optimally reconstructing the geometry of image triplets, *Proc. 6th Euro. Conf. Comput. Vision*, June/July 2000, Dublin, Ireland, Vol. 2, pp. 669–701.
10. W. Gander, H. Golub, and R. Strebel, Least-squares fitting of circles and ellipses, *BIT*, 34 (1994), pp. 558–578.
11. M. Harker and P. O'Leary, First order geometric distance (The myth of Sampsonus), *Proc. 17th Brit. Mach. Vision Conf.*, September 2006, Edinburgh, U.K., Vol. 1, pp. 87–96.
12. R. I. Hartley, In defense of the eight-point algorithm, *IEEE Trans. Patt. Anal. Mach. Intell.*, 19 (1997), pp. 580–593.
13. R. Hartley and F. Kahl, Optimal algorithms in multiview geometry, *Proc. 8th Asian Conf. Comput. Vision*, November 2007, Tokyo, Japan, Vol. 1, pp. 13–34.
14. R. I. Hartley and P. Sturm, triangulation, *Comp. Vision Image Understand.*, 68 (1997), pp. 146–157.
15. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, Cambridge, U.K., 2004.
16. K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, The Netherlands, 1996; reprinted Dover, New York, U.S.A., 2005.
17. K. Kanatani, Cramer-Rao lower bounds for curve fitting, *Graphical Models Image Process.*, 60 (1998), pp. 93–99.
18. K. Kanatani, Statistical optimization for geometric fitting: Theoretical accuracy analysis and high order error analysis, *Int. J. Comp. Vision*, 80 (2008), pp. 167–188.
19. K. Kanatani and N. Ohta, Comparing optimal three-dimensional reconstruction for finite motion and optical flow, *J. Electr. Imaging*, 12 (2003), pp. 478–488.
20. K. Kanatani and Y. Sugaya, High accuracy fundamental matrix computation and its performance evaluation, *IEICE Trans. Inf. Syst.*, E90-D (2007), pp. 579–585.
21. K. Kanatani and Y. Sugaya, Extended FNS for constrained parameter estimation, *Proc. Meeting Image Recognition and Understanding*, July–August 2007, Hiroshima, Japan, pp. 219–226.
22. K. Kanatani and Y. Sugaya, Performance evaluation of iterative geometric fitting algorithms, *Comp. Stat. Data Anal.*, 52 (2007), pp. 1208–1222.
23. K. Kanatani and Y. Sugaya, Compact algorithm for strictly ML ellipse fitting, *Proc. 19th Int. Conf. Pattern Recog.*, December 2008, Tampa, FL, U.S.A.
24. K. Kanatani and Y. Sugaya, Compact fundamental matrix computation, *IPSJ Trans. Comput. Vision Appl.*, 2 (2010), pp. 59–70.
25. K. Kanatani, Y. Sugaya, and H. Niitsuma, Triangulation from two views revisited: Hartley-Sturm vs. optimal correction, *Proc. 19th British Machine Vision Conf.*, September 2008, Leeds, U.K., pp. 173–182.
26. Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comp. Vision*, 37 (2000), pp. 127–150.
27. B. C. Matei and P. Meer, Estimation of nonlinear errors-in-variables models for computer vision applications, *IEEE Trans. Patt. Anal. Mach. Intell.*, 28 (2006), pp. 1537–1552.
28. E. M. Mikhail and F. Ackermann, *Observations and Least Squares*, University Press of America, Lanham, MD., U.S.A., 1976.
29. P. D. Sampson, Fitting conic sections to "very scattered" data: An iterative refinement of the Bookstein algorithm, *Comp. Graphics Image Process.*, 18 (1982), pp. 97–108.
30. P. Sturm and P. Gargallo, Conic fitting using the geometric distance, *Proc. 8th Asian Conf. Comput. Vision*, November 2007, Tokyo, Japan, Vol. 2, pp 784–795.
31. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. Fitzgibbon, Bundle adjustment—A modern synthesis, in B. Triggs, A. Zisserman, and R. Szeliski, eds., *Vision Algorithms: Theory and Practice*, Springer, Berlin, Germany, 2000, pp. 298–375.

Kenichi Kanatani received his B.E., M.S., and Ph.D. in applied mathematics from the University of Tokyo in 1972, 1974 and 1979, respectively. After serving as Professor of computer science at Gunma University, Gunma, Japan, he is currently Professor of computer science at Okayama University, Okayama, Japan. He is the author of many books on computer vision and received many awards including the best paper awards from IPSJ (1987) and IEICE (2005). He is an IEEE Fellow.

Yasuyuki Sugaya received his B.E., M.S., and Ph.D. in computer science from the University of Tsukuba, Ibaraki, Japan, in 1996, 1998, and 2001, respectively. From 2001 to 2006, he was Assistant Professor of computer science at Okayama University, Okayama, Japan. Currently, he is Associate Professor of information and computer sciences at Toyohashi University of Technology, Toyohashi, Aichi, Japan. His research interests include image processing and computer vision. He received the IEICE best paper award in 2005.