Unified Computation of Strict Maximum Likelihood for Geometric Fitting

Kenichi KANATANI*
Department of Computer Science
Okayama University
Okayama 700-8530 Japan

Yasuyuki SUGAYA
Department of Information and Computer Sciences
Toyohashi University of Technology
Toyohashi, Aichi 441-8580 Japan

(Received December 14, 2009)

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.

1. INTRODUCTION

This paper presents a unified numerical scheme for computing strict maximum likelihood (ML) for a problem called *geometric fitting* [15] 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, we show that a globally optimal 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* [15], 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. 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 a problem of fitting to noisy vector data $\boldsymbol{x}_\alpha$, $\alpha = 1, ..., N$, an *implicit* equation in the form

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

parameterized by $\boldsymbol{\theta}$. To be specific, we want to estimate the parameter $\boldsymbol{\theta}$ in such a way that $F(\boldsymbol{x}_\alpha;\boldsymbol{\theta}) \approx 0$ for all α. Many computer vision problems are formulated in this way [14, 15]; one can infer the shapes and the positions of objects seen in images from the thus computed $\boldsymbol{\theta}$. 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 $\boldsymbol{x}$ in terms of $\boldsymbol{\theta}$ [9, 27].

The function $F(\boldsymbol{x};\boldsymbol{\theta})$ in (1) is generally nonlinear

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

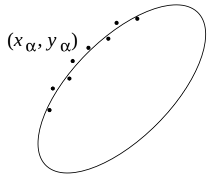

Figure 1: Fitting an ellipse to a point sequence.

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

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

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

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 $\boldsymbol{\xi}(x, y)$ and $\boldsymbol{\theta}$ by

$$\begin{aligned} &\boldsymbol{\xi}(x,y)(x^2 \;\; 2xy \;\; y^2 \;\; 2x \;\; 2y \;\; 1)^\top, \\ &\boldsymbol{\theta} = (A \;\; B \;\; C \;\; D \;\; E \;\; F)^\top, \end{aligned} \tag{4}$$

(3) has the form of (2) [21].

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 [14] (Fig. 2):

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

Here, $\boldsymbol{F}$ 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 $\boldsymbol{F}$ from point correspondences, we can reconstruct the 3-D shape of the scene and the camera positions [18]. If we define

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

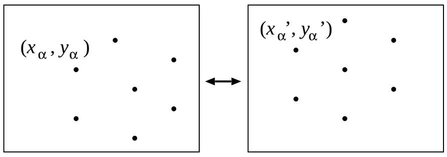

Figure 2: Computing the fundamental matrix from corresponding points between two images.

(5) has the form of (2) [19].

3. GAUSSIAN NOISE IN THE $\boldsymbol{\xi}$-SPACE

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

- 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 $\boldsymbol{x}_\alpha$ and Gaussian noise in the transformed data $\boldsymbol{\xi}_\alpha = \boldsymbol{\xi}(\boldsymbol{x}_\alpha)$. The covariance matrix $V[\boldsymbol{x}_\alpha]$ of $\boldsymbol{x}_\alpha$ and the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ of $\boldsymbol{\xi}_\alpha$ are related by

$$V[\boldsymbol{\xi}_\alpha] = \Big(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\Big)_\alpha V[\boldsymbol{x}_\alpha] \Big(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\Big)_\alpha^\top, \tag{7}$$

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

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 $V[\boldsymbol{\xi}_\alpha]$ is written as follows [21]:

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

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 $V[\boldsymbol{\xi}_\alpha]$ is written as follows [19]:

$V[\boldsymbol{\xi}_\alpha]$

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

$$= \sigma^2 \begin{pmatrix} x_\alpha^2+x_\alpha'^2 & x_\alpha' y_\alpha' & x_\alpha' & x_\alpha y_\alpha & 0 & 0 & x_\alpha & 0 & 0 \\ x_\alpha' y_\alpha' & x_\alpha^2+y_\alpha'^2 & y_\alpha' & 0 & x_\alpha y_\alpha & 0 & 0 & x_\alpha & 0 \\ x_\alpha' & y_\alpha' & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_\alpha y_\alpha & 0 & 0 & y_\alpha^2+x_\alpha'^2 & x_\alpha' y_\alpha' & x_\alpha' y_\alpha & 0 & 0 \\ 0 & x_\alpha y_\alpha & 0 & x_\alpha' y_\alpha' & y_\alpha^2+y_\alpha'^2 & y_\alpha' & 0 & y_\alpha & 0 \\ 0 & 0 & 0 & x_\alpha' & y_\alpha' & 1 & 0 & 0 & 0 \\ x_\alpha & 0 & 0 & y_\alpha & 0 & 0 & 1 & 0 & 0 \\ 0 & x_\alpha & 0 & 0 & y_\alpha & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{9}$$

4. ML IN THE $\boldsymbol{\xi}$-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, 15, 16] up to higher order noise terms.

If Gaussian noise is assumed in the $\boldsymbol{\xi}$-space, ML reduces to minimization of the *Mahalanobis distance*

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

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

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

with respect to $\bar{\boldsymbol{\xi}}_\alpha$ and $\boldsymbol{\theta}$. Since the constraint is linear in $\bar{\boldsymbol{\xi}}_\alpha$, it can be eliminated by introducing Lagrange multipliers, reducing (10) to[2] [17].

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

which is called the *Sampson error*, originating from ellipse fitting by Sampson [28].

Various numerical schemes are available for minimizing (12) [17], including the *FNS* (*Fundamental Numerical Scheme*) of Chojnacki et al. [7], the *HEIV* (*Heteroscedastic Errors In Variables*) of Leedan and Meer [25], and the *projective Gauss-Newton iterations* of Kanatani and Sugaya [19, 21]. These apply when no special constraint (scale normalization aside) is imposed on $\boldsymbol{\theta}$. 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 [20].

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 $\boldsymbol{\xi}$-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 $\boldsymbol{x}$-space.

5. ML IN THE $\boldsymbol{x}$-SPACE

The preceding formulation suits numerical computation and accuracy analysis [17]. 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 $\boldsymbol{\xi}_\alpha$ 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 $\boldsymbol{\xi}_\alpha$ after the nonlinear transformation (6) is, strictly, no longer Gaussian. Whether we assume Gaussian noise in the $\boldsymbol{\xi}$-space or in the $\boldsymbol{x}$-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 $\boldsymbol{x}$-space, ML reduces to minimization of the Mahalanobis distances[4]

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

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

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

with respect to $\bar{\boldsymbol{x}}_\alpha$ and $\boldsymbol{\theta}$. In the following, we call the Mahalanobis distance (13) in the $\boldsymbol{x}$-space the *reprojection error* as opposed to the Sampson error (12), which equals the Mahalanobis distance (10) in the $\boldsymbol{\xi}$-space. Traditionally, the term "reprojection error" 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, and the 3-D shape with the smallest discrepancy is sought. Here, we slightly abuse this term to mean the discrepancy of actual observations and their guesses measured in the Mahalanobis distance *in the original $\boldsymbol{x}$-space*.

[2] If $\boldsymbol{\xi}_\alpha$ has constant components as in (4) and (6), the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ becomes singular as seen in (8) and (9). In such a case, we replace $V[\boldsymbol{\xi}_\alpha]^{-1}$ in (10) by the pseudoinverse, which means we focus only on those components of $\boldsymbol{\xi}_\alpha$ that can vary. Still, (12) holds [15].

[3] It was pointed out that CFNS does not necessarily compute a correct solution [20].

[4] The following argument holds if $V[\boldsymbol{x}_\alpha]$ is singular. All we need is to replace $V[\boldsymbol{x}_\alpha]^{-1}$ by its pseudoinverse and appropriately use projection operations [15].

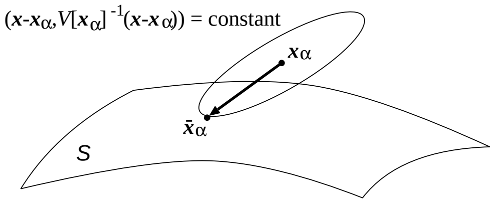
$(\boldsymbol{x}-\boldsymbol{x}_\alpha, V[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}-\boldsymbol{x}_\alpha))$ = constant
$\boldsymbol{x}_\alpha$
$\bar{\boldsymbol{x}}_\alpha$
S

Figure 3: Orthogonal projection of $\boldsymbol{x}_\alpha$ with respect to $V[\boldsymbol{x}_\alpha]$.

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 $F(\boldsymbol{x};\boldsymbol{\theta}) = 0$ defines a hypersurface $\mathcal{S}$ in the $\boldsymbol{x}$-space, and the problem of finding $\bar{\boldsymbol{x}}_\alpha$ that minimizes (13) can be interpreted as finding a point $\bar{\boldsymbol{x}}_\alpha \in \mathcal{S}$ closest to $\boldsymbol{x}_\alpha$ measured in (13). Hence, the solution is obtained by "orthogonally" projecting the observations $\boldsymbol{x}_\alpha$ onto $\mathcal{S}$, where orthogonality is defined with respect to $V[\boldsymbol{x}_\alpha]$: $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthogonal if $(\boldsymbol{a}, V[\boldsymbol{x}_\alpha]^{-1}\boldsymbol{b}) = 0$ (Fig. 3). An initial guess $\{\boldsymbol{x}_\alpha^{(0)}, \boldsymbol{\theta}^{(0)}\}$ ($\boldsymbol{x}_\alpha^{(0)}$ are the observations $\boldsymbol{x}_\alpha$ themselves) is assumed, and a sequence $\{\boldsymbol{x}_\alpha^{(0)}, \boldsymbol{\theta}^{(0)}\}$, $\{\boldsymbol{x}_\alpha^{(1)}, \boldsymbol{\theta}^{(1)}\}$, $\{\boldsymbol{x}_\alpha^{(2)}, \boldsymbol{\theta}^{(2)}\}$, ... is generated by computing $\{\boldsymbol{x}_\alpha^{(k)}, \boldsymbol{\theta}^{(k)}\}$ from $\{\boldsymbol{x}_\alpha^{(k-1)}, \boldsymbol{\theta}^{(k-1)}\}$ by various means [1, 2, 9, 27]. The resulting sequence is expected to converge to a locally optimal solution.

Bundle adjustment. Introducing auxiliary variables, we solve the constraint $F(\bar{\boldsymbol{x}}_\alpha;\boldsymbol{\theta}) = 0$ for each $\bar{\boldsymbol{x}}_\alpha$ in terms of $\boldsymbol{\theta}$ and the auxiliary variables. The resulting expressions are substituted into (13), resulting an explicit function E of $\boldsymbol{\theta}$ 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 [12]. 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 and passing through the reconstructed 3-D points in such a way that all rays pierce the image plane as closely to their observed positions as possible [30].

The purpose of this paper is to show that if the constraint has the special form of (14) we can exploit the close relationship between the ML in the $\boldsymbol{\xi}$-space ((10), (11), and (12)) and the ML in the $\boldsymbol{x}$-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 and the orientations of the major and minor axes, and the angles of individual points seen from the center are introduced, and the resulting high dimensional parameter space is searched by various numerical schemes [4, 10, 11, 29]. 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 of these auxiliary variables is searched so that the image positions obtained by reprojecting the reconstructed 3-D points are as close to the observed points as possible [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 we are considering. Also, the search space of $\boldsymbol{\theta}$ and the auxiliary variables 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 [26], but we explicitly take advantage of the correspondence between ML in the $\boldsymbol{x}$-space and ML in the $\boldsymbol{\xi}$-space.

Note that minimization of (10) is not affected by multiplication of $V[\boldsymbol{\xi}_\alpha]$ by any positive constant. Similarly, $V[\boldsymbol{x}_\alpha]$ in (13) can be multiplied by any positive constant. Hence, the scales of $V[\boldsymbol{\xi}_\alpha]$ and $V[\boldsymbol{x}_\alpha]$ 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 $V[\boldsymbol{\xi}_\alpha]$ and $V[\boldsymbol{x}_\alpha]$ up to a *common* scale.

7. ORTHOGONAL PROJECTION

As opposed to the traditional orthogonal projection approach, we regard $\boldsymbol{\theta}$ throughout as a *free variable* whose value is yet to be specified, and minimize

(13) subject to (14). We identify the hypersurface $\mathcal{S}$ defined by (14) in the $\boldsymbol{x}$-space with a "level set" of $(\boldsymbol{\xi}(\boldsymbol{x}), \boldsymbol{\theta}) = c$ for $c = 0$ and view the $\boldsymbol{x}$-space as filled with continuously varying level sets with varying c (*foliation* in mathematical terms). The observation $\boldsymbol{x}_\alpha$ is on the surface (or *leaf*) $\mathcal{S}_\alpha$: $(\boldsymbol{\xi}(\boldsymbol{x}), \boldsymbol{\theta}) = c_\alpha$, where $c_\alpha = (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})$. We first project $\boldsymbol{x}_\alpha$ onto $\mathcal{S}$ in the direction orthogonal to $\mathcal{S}_\alpha$ at $\boldsymbol{x}_\alpha$ with respect to $V[\boldsymbol{x}_\alpha]$. For short, let us call that direction the $V[\boldsymbol{x}_\alpha]$-*normal* of $\mathcal{S}_\alpha$ at $\boldsymbol{x}_\alpha$. It has the following form (See Appendix):

$$\boldsymbol{n}_\alpha = V[\boldsymbol{x}_\alpha]\Big(\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{x}}\Big)_\alpha^\top\boldsymbol{\theta}. \tag{15}$$

We compute the projection $\hat{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - \lambda_\alpha\boldsymbol{n}_\alpha$ by determining the magnitude λ_α so as to satisfy

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

Recall that $\boldsymbol{\theta}$ is a *free variable*. If it were a fixed value, we could determine λ_α by numerical line search. If each component of $\boldsymbol{\xi}(\boldsymbol{\xi})$ is a polynomial of $\boldsymbol{\xi}$, we might even be able to compute λ_α analytically. However, what we need is to obtain λ_α as an explicit function of the free variable $\boldsymbol{\theta}$ to be optimized later. So, assuming that λ_α is small and using the Taylor expansion

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

we express λ_α to a first approximation in the form

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

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

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

which is nothing but the Sampson error (10). Now, we choose the value of $\boldsymbol{\theta}$, which has so far been unspecified, to be the value $\hat{\boldsymbol{\theta}}$ that minimizes (19) *over the entire domain of* $\boldsymbol{\theta}$, using an existing Sampson error minimization method such as FNS or HEIV. Then, the orthogonal projection is given by

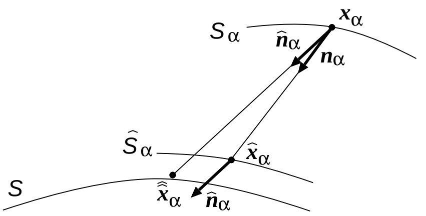

Figure 4: Successive projection along the $V[\boldsymbol{x}_\alpha]$-normal of the level set.

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

8. STRICT ORTHOGONAL PROJECTION

Now that we know the nature of the Sampson error, we modify it so that it coincides with the reprojection error. In the preceding computation, we projected $\boldsymbol{x}_\alpha$ along the $V[\boldsymbol{x}_\alpha]$-normal of the "level set $\mathcal{S}_\alpha$" at $\boldsymbol{x}_\alpha$, while our goal is to project $\boldsymbol{x}_\alpha$ along the $V[\boldsymbol{x}_\alpha]$-normal of the "hypersurface $\mathcal{S}$" itself at $\bar{\boldsymbol{x}}_\alpha$, which is yet unknown. We now project $\boldsymbol{x}_\alpha$ along the $V[\boldsymbol{x}_\alpha]$-normal of the "level set $\hat{\mathcal{S}}_\alpha$" at $\hat{\boldsymbol{x}}_\alpha$ that we have just computed (Fig. 4). To do this, we again view $\boldsymbol{\theta}$ as a free variable. Namely, once we have computed $\hat{\boldsymbol{x}}_\alpha$ by (20), we forget the value $\hat{\boldsymbol{\theta}}$ and regard $\boldsymbol{\theta}$ as unspecified. The $V[\boldsymbol{x}_\alpha]$-normal of $\hat{\mathcal{S}}_\alpha$ at $\hat{\boldsymbol{x}}_\alpha$ is

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

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

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

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

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

where we define

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

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

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

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

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

Here, we define

$$\hat{\boldsymbol{\xi}}^*_\alpha \equiv \hat{\boldsymbol{\xi}}_\alpha + \Big(\frac{\partial\hat{\boldsymbol{\xi}}}{\partial\boldsymbol{x}}\Big)_\alpha \tilde{\boldsymbol{x}}_\alpha, \quad V[\hat{\boldsymbol{\xi}}_\alpha] \equiv \Big(\frac{\partial\hat{\boldsymbol{\xi}}}{\partial\boldsymbol{x}}\Big)_\alpha V[\boldsymbol{x}_\alpha] \Big(\frac{\partial\hat{\boldsymbol{\xi}}}{\partial\boldsymbol{x}}\Big)^\top_\alpha, \tag{27}$$

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

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

Again, this has the same form as (12). So, we choose the value of $\boldsymbol{\theta}$, so far unspecified, to the value $\hat{\hat{\boldsymbol{\theta}}}$ that minimizes (28) *over the entire domain of* $\boldsymbol{\theta}$, using an existing Sampson error minimization method such as FNS or HEIV. The true value $\bar{\boldsymbol{x}}$ is estimated to be

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

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

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

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], where we remove the scale indeterminacy of the ellipse parameter $\boldsymbol{\theta}$ in (4) by normalizing it to $\|\boldsymbol{\theta}\| = 1$:

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 $V[\hat{\boldsymbol{\xi}}_\alpha]$ 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 $\boldsymbol{\xi}^*_\alpha$, $\alpha = 1, ..., N$:
$$\boldsymbol{\xi}^*_\alpha = \begin{pmatrix} \hat{x}^2_\alpha + 2\hat{x}_\alpha\tilde{x}_\alpha \\ 2(\hat{x}_\alpha\hat{y}_\alpha + \hat{y}_\alpha\tilde{x}_\alpha + \hat{x}_\alpha\tilde{y}_\alpha) \\ \hat{y}^2_\alpha + 2\hat{y}_\alpha\tilde{y}_\alpha \\ 2(\hat{x}_\alpha + \tilde{x}_\alpha) \\ 2(\hat{y}_\alpha + \tilde{y}_\alpha) \\ 1 \end{pmatrix}. \tag{30}$$

4. Compute the 6-D unit vector $\boldsymbol{\theta} = (\theta_i)$ that minimizes the following function (e.g., by FNS or HEIV):
$$E(\boldsymbol{\theta}) = \sum_{\alpha=1}^N \frac{(\boldsymbol{\theta}, \boldsymbol{\xi}^*_\alpha)^2}{(\boldsymbol{\theta}, V[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})}. \tag{31}$$

5. Update $\tilde{x}_\alpha$, $\tilde{y}_\alpha$, $\hat{x}_\alpha$, and $\hat{y}_\alpha$ in the form
$$\begin{pmatrix} \tilde{x}_\alpha \\ \tilde{y}_\alpha \end{pmatrix} \leftarrow \frac{2(\boldsymbol{\theta}, \boldsymbol{\xi}^*_\alpha)}{(\boldsymbol{\theta}, V[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})} \begin{pmatrix} \theta_1 & \theta_2 & \theta_4 \\ \theta_2 & \theta_3 & \theta_5 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ 1 \end{pmatrix}, \tag{32}$$
$$\hat{x}_\alpha \leftarrow x_\alpha - \tilde{x}_\alpha, \qquad \hat{y}_\alpha \leftarrow y_\alpha - \tilde{y}_\alpha. \tag{33}$$

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 $\boldsymbol{\theta}$ and stop. Else, let $E_0 \leftarrow E$ and go back to Step 2.

This scheme was obtained by Kanatani and Sugaya [22] 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 [22], 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 10 points on an ellipse, and Fig. 5(b) shows the RMS

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

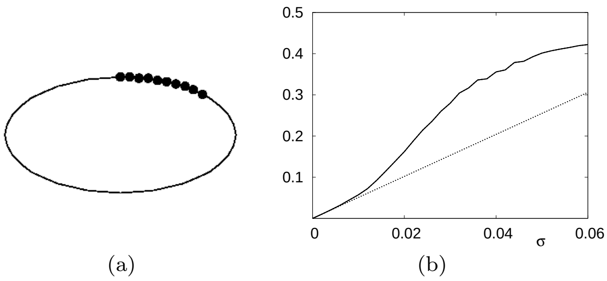

Figure 5: (a) 10 points on an ellipse. (b) RMS error of the fitted ellipse over 1000 trials vs. the noise level σ. The plots of ML in the $\boldsymbol{\xi}$-space and ML in the $\boldsymbol{x}$-space completely overlap (the solid line). The dotted line shows the KCR lower bound.

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 horizontal axis is extended to an unrealistically large value of σ for the sake of comparison. The plots of of ML in the $\boldsymbol{\xi}$-space and ML in the $\boldsymbol{x}$-space completely overlap (the solid line in the figure). The dotted line shows the theoretical accuracy limit (*KCR lower bound* [6, 15, 16]).

Example 6 (Fundamental matrix computation) The vector $\boldsymbol{\theta}$ in (6) that encodes the fundamental matrix $\boldsymbol{F}$ of rank 2 is computed from corresponding points (x_α, y_α) and (x'_α, y'_α), $\alpha = 1, ..., N$, as follows^5, where we remove the scale indeterminacy of $\boldsymbol{\theta}$ by normalizing it to $\|\boldsymbol{\theta}\| = 1$:

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 $V[\hat{\boldsymbol{\xi}}_\alpha]$ 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 $\boldsymbol{\xi}^*_\alpha$, $\alpha = 1, ..., N$:

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

4. Compute the 9-D unit vector $\boldsymbol{\theta} = (\theta_i)$ that minimizes the following function subject to the constraint that the resulting fundamental matrix $\boldsymbol{F}$ has rank 2 (e.g., by EFNS [20]):

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

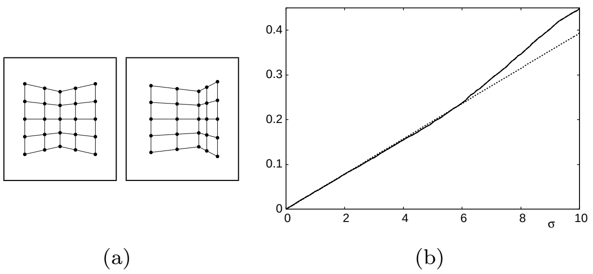

Figure 6: (a) Planer grids viewed from two angles. (b) RMS error of the fitted fundamental matrix over 1000 trials vs. the noise level σ. The plots of ML in the $\boldsymbol{\xi}$-space and ML in the $\boldsymbol{x}$-space completely overlap (the solid line). The dotted line shows the KCR lower bound.

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

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

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

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

6. Compute the reprojection error

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

7. If $E \approx E_0$, return $\boldsymbol{\theta}$ 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 fundamental matrix computation. 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 6(a) shows two views of a planar grid viewed from two angles, and Fig. 6(b) shows 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. The horizontal axis is extended to an unrealistically large value of σ for the sake of comparison. Again, the plots of ML in the $\boldsymbol{\xi}$-space and ML in the $\boldsymbol{x}$-space completely overlap (the solid line in the figure). The dotted line shows the KCR lower bound.

Here, we should note the following two issues in actual computation.

- In the iterations, the reprojection error E generally *increases*. By definition, E is the sum of square Mahalanobis (Euclidean in the above examples) distances between the observation $\boldsymbol{x}_\alpha$ and our guess $\hat{\boldsymbol{x}}_\alpha$. We start by regarding $\boldsymbol{x}_\alpha$ as our guess $\hat{\boldsymbol{x}}_\alpha$, so initially E is 0. Since the guess does not satisfy the constraint, we modify $\hat{\boldsymbol{x}}_\alpha$ so as to satisfy it. Maximum likelihood means that we do this by increasing E *by a minimal amount*. Geometrically, this is equivalent to orthogonal projection. In actual iterations, the increase of E is not always monotonic; it may oscillate when convergence is near. In each step of this process, on the other hand, we *minimize* E *with respect to* $\boldsymbol{\theta}$. In general, optimization with constraints involve both minimization and maximization [5].
- In the Step 7 of ellipse fitting and fundamental matrix computation, we stop the iteration when $E \approx E_0$. We could stop when the current value $\boldsymbol{\theta}_{\rm cur}$ of $\boldsymbol{\theta}$ is close to the previous value $\boldsymbol{\theta}_{\rm pre}$, but then care must be taken. First, the Sampson error $E(\boldsymbol{\theta})$ in the form of (31) and (36) do not depend on the sign of $\boldsymbol{\theta}$, so it may happen that $\boldsymbol{\theta}_{\rm cur}$ has an opposite direction to $\boldsymbol{\theta}_{\rm pre}$. Hence, we must align the orientations of $\boldsymbol{\theta}_{\rm cur}$ and $\boldsymbol{\theta}_{\rm pre}$ before comparing them. However, a potentially more serious problem exists: the interference of the nested iteration loops. Minimizing (31) and (36) usually involves iterations, so we must stop if the update of $\boldsymbol{\theta}$ is sufficiently small. If we use the same threshold for $\boldsymbol{\theta}$ to stop the inner loop for minimizing $E(\theta)$ and the outer loop for computing E, the inner loop may stop but the outer loop may not, or vice versa. If we demand higher accuracy of $\boldsymbol{\theta}$ for the outer loop than can be afforded by the internal Sampson error minimization, the iterations may continue forever. No such interferences can occur if we stop the inner loop by $\boldsymbol{\theta}$ and the outer loop by E.

10. OPTIMAL CORRECTION

As a byproduct, our strict ML leads to a new numerical scheme for *optimal correction* [15]: we optimally correct a given $\boldsymbol{x}$ so as to satisfy the constraint $(\boldsymbol{\xi}(\boldsymbol{x}), \boldsymbol{\theta}) = 0$, where the parameter $\boldsymbol{\theta}$ is given and fixed. The problem is stated as reprojection error minimization: Minimize

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

subject to

$$(\boldsymbol{\xi}(\bar{\boldsymbol{x}}), \boldsymbol{\theta}) = 0, \qquad (41)$$

for a given $\boldsymbol{\theta}$. We only need to remove the computation of $\boldsymbol{\theta}$ 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 ($\boldsymbol{\theta}$ is defined by (4)):

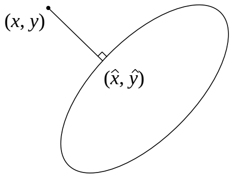

Figure 7: Drawing a perpendicular to an ellipse.

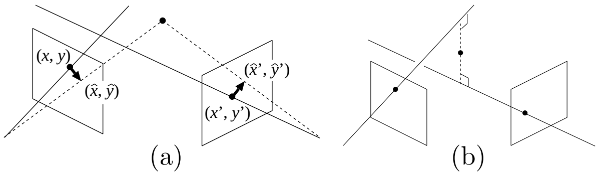

Figure 8: Computing the 3-D position from noisy correspondence pair. (a) The mid-point method. (b) Optimal triangulation.

1. Let $E_0 = \infty$ (a sufficiently large number), $\hat{x} = x$, $\hat{y} = y$, and $\tilde{x} = \tilde{y} = 0$.
2. Let $V[\hat{\boldsymbol{\xi}}]$ be the matrix obtained by substituting $\hat{x}$ and $\hat{y}$ for x_α and y_α, respectively, in (8).
3. Compute the following $\boldsymbol{\xi}^*$:
$$\boldsymbol{\xi}^* = \begin{pmatrix} \hat{x}^2 + 2\hat{x}\tilde{x} \\ 2(\hat{x}\hat{y} + \hat{y}\tilde{x} + \hat{x}\tilde{y}) \\ \hat{y}^2 + 2\hat{y}\tilde{y} \\ 2(\hat{x} + \tilde{x}) \\ 2(\hat{y} + \tilde{y}) \\ 1 \end{pmatrix}. \qquad (42)$$
4. Update $\tilde{x}$, $\tilde{y}$, $\hat{x}$, and $\hat{y}$ in the form
$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \leftarrow \frac{2(\boldsymbol{\theta}, \boldsymbol{\xi}^*)}{(\boldsymbol{\theta}, V[\hat{\boldsymbol{\xi}}]\boldsymbol{\theta})} \begin{pmatrix} \theta_1 & \theta_2 & \theta_4 \\ \theta_2 & \theta_3 & \theta_5 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix}, \qquad (43)$$
$$\hat{x} \leftarrow x - \tilde{x}, \quad \hat{y} \leftarrow y - \tilde{y}. \qquad (44)$$
5. Compute the reprojection error
$$E = \tilde{x}^2 + \tilde{y}^2. \qquad (45)$$
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 the above simple procedure has not been known. Usually, the computation converges after three or four iterations, but even the first solution has sufficient accuracy for practical use.

Example 8 (Triangulation) When the fundamental matrix $\boldsymbol{F}$ 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) determined by $\boldsymbol{F}$ (Fig. 8(a)). This is because the lines of sight determined by points (x, y) and (x', y') can intersect in the scene if and only if the epipolar equation in (5) holds [14]. 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 ($\boldsymbol{\theta}$ 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 $V[\hat{\boldsymbol{\xi}}]$ 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).

3. Compute the following $\boldsymbol{\xi}^*$:

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

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

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \leftarrow \frac{(\boldsymbol{\theta}, \boldsymbol{\xi}^*)}{(\boldsymbol{\theta}, V[\hat{\boldsymbol{\xi}}]\boldsymbol{\theta})} \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{pmatrix} \begin{pmatrix} \hat{x}' \\ \hat{y}' \\ 1 \end{pmatrix},$$
$$\begin{pmatrix} \tilde{x}' \\ \tilde{y}' \end{pmatrix} \leftarrow \frac{(\boldsymbol{\theta}, \boldsymbol{\xi}^*)}{(\boldsymbol{\theta}, V[\hat{\boldsymbol{\xi}}]\boldsymbol{\theta})} \begin{pmatrix} \theta_1 & \theta_4 & \theta_7 \\ \theta_2 & \theta_5 & \theta_8 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix}, \qquad (47)$$

$$\hat{x} \leftarrow x - \tilde{x}, \qquad \hat{y} \leftarrow y - \tilde{y},$$
$$\hat{x}' \leftarrow x' - \tilde{x}', \qquad \hat{y}' \leftarrow y' - \tilde{y}'. \qquad (48)$$

5. Compute the reprojection error

$$E = \tilde{x}^2 + \tilde{y}^2 + \tilde{x}'^2 + \tilde{y}'^2. \qquad (49)$$

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.

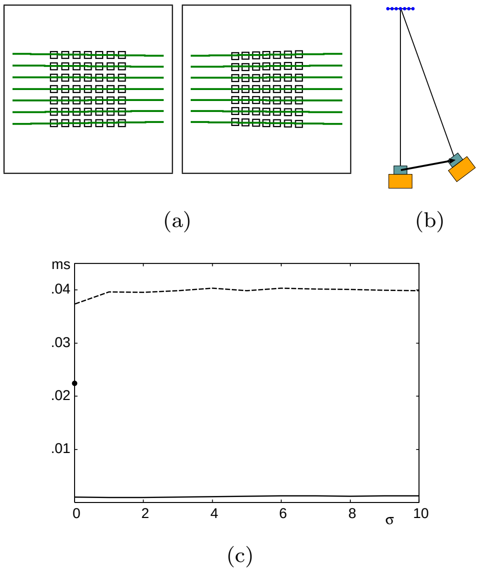

Figure 9: (a) Stereo images of a planar grid. Some of the epipolar lines are drawn. (b) Top view of the camera configuration. (c) Computation time (ms) for triangulation per point (average over 1000 trials). The CPU is Intel Core2Duo E6700 2.66GHz. The horizontal axis is for the standard deviation σ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

This procedure is nothing but the optimal stereo triangulation of Kanatani et al. [24], 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 [13], who determined the epipolar lines of the corresponding points by algebraically solving a 6-degree polynomial. Kanatani et al. [24] experimentally confirmed that their solution is identical to that of Hartley and Sturm [13] 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 [14]. Indeed, we have confirmed that our ellipse fitting solution is identical to those by others such as Gander et al. [10] and Sturm and Gargallo [29].

- Global optimality: *Theoretically*, our approach can produces a globally optimal solution as opposed to traditional orthogonal projection approaches. By "theoretically", we mean that this depends on the performance of the Sampson error minimizer we use. Our approach is to iteratively change the variables in the data space so that the Sampson error coincides with the reprojection error. Thus, the solution is globally optimal if 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.
- Convergence: It is difficult to give a mathematical proof of the convergence of our orthogonal projection, because the convergence is a property of the hypersurface $\mathcal{S}$ defined by the constraint, rather than a property of the operation. The hypersurface $\mathcal{S}$ may have singularities, e.g., for the epipolar constraint, $\mathcal{S}$ is hyperbolic in 4-D with singularities at the epipoles. The iterations should converge if $\mathcal{S}$ is more or less flat and its supporting function is nearly linear, or equivalently, if the observations are sufficiently close to $\mathcal{S}$, 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 $\mathcal{S}$ such as the curvature. In our experience, the orthogonal projection converges after at most three to four iterations; we have never encountered nonconvergence. On the other hand, we have frequently observed that the internal 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 $\boldsymbol{x}_\alpha$ onto a parameterized surface $\mathcal{S}$ defined by the constraint in the $\boldsymbol{x}$-space, where the orthogonality is defined with respect to the covariance matrix $V[\boldsymbol{x}_\alpha]$ of $\boldsymbol{x}_\alpha$. 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 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 (No. 21500172).

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), 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), 277–296.

[3] A. Bartoli and P. Sturm: Nonlinear estimation of fundamental matrix with minimal parameters, *IEEE Trans. Patt. Anal. Mach. Intell.*, 26 (2004), 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), 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), 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), 85–91.

[9] W. Föstner: On weighting and choosing constraints for optimally reconstructing the geometry of image triplets, *Proc. 6th Euro. Conf. Computer Vision*, Dublin, Ireland, Vol. 2, June/July (2000), 669–701.

[10] W. Gander, H. Golub, and R. Strebel: Least-squares fitting of circles and ellipses, *BIT*, 34 (1994), 558–578.

[11] M. Harker and P. O'Leary: First order geometric distance (The myth of Sampsonus), *Proc. 17th Brit. Mach. Vision Conf.*, Edinburgh, U.K., Vol. 1, September (2006), 87–96.

[12] R. Hartley and F. Kahl: Optimal algorithms in multiview geometry, *Proc. 8th Asian Conf. Computer Vision*, Tokyo, Japan, Vol. 1, November (2007), 13–34.

[13] R. I. Hartley and P. Sturm: triangulation, *Comp. Vision Image Understand.*, 68 (1997), 46–157.

[14] R. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., (2000).

[15] K. Kanatani: *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, The Netherlands (1996); reprinted Dover, New York, U.S.A., (2005).

[16] K. Kanatani: Cramer-Rao lower bounds for curve fitting, *Graphical Models Image Process.*, 60 (1998), 93–99.

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

[18] K. Kanatani and N. Ohta: Comparing optimal three-dimensional reconstruction for finite motion and optical flow, *J. Electr. Imaging*, 12 (2003), 478–488.

[19] K. Kanatani and Y. Sugaya: High accuracy fundamental matrix computation and its performance evaluation, *IEICE Trans. Inf. Syst.*, E90-D (2007), 579–585.

[20] K. Kanatani and Y. Sugaya: Extended FNS for constrained parameter estimation, *Proc. Meeting Image Recognition and Understanding*, Hiroshima, Japan, July–August (2007), 219–226.

[21] K. Kanatani and Y. Sugaya: Performance evaluation of iterative geometric fitting algorithms, *Comp. Stat. Data Anal.*, 52 (2007), 1208–1222.

[22] K. Kanatani and Y. Sugaya: Compact algorithm for strictly ML ellipse fitting, *Proc. 19th Int. Conf. Pattern Recognition*, Tampa, FL, U.S.A., TuBCT8.52, December (2008), 1–4.

[23] K. Kanatani and Y. Sugaya: Compact fundamental matrix computation, *Proc. 3rd Pacific Rim Symp. Image and Video Technology*, Tokyo, Japan, January (2009), 179–190.

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

[25] Y. Leedan and P. Meer: Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comp. Vision*, 37 (2000), 127–150.

[26] 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), 1537–1552.

[27] E. M. Mikhail and F. Ackermann: *Observations and Least Squares*, University Press of America, Lanham, MD., U.S.A. (1976).

[28] P. D. Sampson: Fitting conic sections to "very scattered" data: An iterative refinement of the Bookstein algorithm, *Comp. Graphics Image Process.*, 18 (1982), 97–108.

[29] P. Sturm and P. Gargallo: Conic fitting using the geometric distance, *Proc. 8th Asian Conf. Computer Vision*, Tokyo, Japan, Vol. 2, November (2007), 784–795.

[30] 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), 298–375.

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

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

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

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

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

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