Statistical Optimization for Geometric Estimation: Minimization vs. Non-minimization

Kenichi Kanatani
Professor Emeritus, Okayama University, Okayama, 700-8530 Japan
Email: kanatani2013@yahoo.co.jp

***Abstract*—We overview techniques for optimal geometric estimation from noisy observations for computer vision applications. We first describe techniques based on minimization of a given cost function: least squares (LS), maximum likelihood (ML), and Sampson error minimization. We then summarize techniques not based on minimization: one solves a given matrix equation. Different choices of the matrices in it result in different methods: LS, iterative reweight, the Taubin method, renormalization, HyperLS, and hyper-renormalization. Doing statistical analysis and conducting numerical examples, we conclude that hyper-renormalization is the best method in terms of accuracy and efficiency.**

I. Introduction

One of the most important tasks of computer vision is to compute the 2-D and 3-D shapes of objects exploiting *geometric constraints*, by which we mean properties that can be described by relatively simple equations such as the objects being lines or planes, their being parallel or orthogonal, and the camera imaging geometry being perspective projection. We call such an inference *geometric estimation*. In the presence of noise, the assumed constraints do not exactly hold. This paper summarizes techniques for optimal geometric estimation in the presence of noise and reports the latest results.

II. Preliminaries

A. Definition of Geometric Estimation

The geometric estimation problem we consider here is defined as follows. We observe some quantity $\boldsymbol{x}$ (a vector), which is assumed to satisfy in the absence of noise an equation

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

parameterized by unknown vector $\boldsymbol{\theta}$. This equation is called the *geometric constraint*. Our task is to estimate the parameter $\boldsymbol{\theta}$ from noisy instances $\boldsymbol{x}_\alpha$, α = 1, ..., N, of $\boldsymbol{x}$. In many vision applications, we can reparameterize the problem so that the constraint is liner in the parameter $\boldsymbol{\theta}$ (but generally nonlinear in the data $\boldsymbol{x}$). Then, Eq. (1) has the form

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

where $\boldsymbol{\xi}(\boldsymbol{x})$ is a vector-valued nonlinear function of $\boldsymbol{x}$. In this paper, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$. Equation (2) implies that $\boldsymbol{\theta}$ in (2) has scale indeterminacy, so we normalize it to unit norm: $\|\boldsymbol{\theta}\|$ = 1.

Example 1: (Line fitting) To a given point sequence (x_α, y_α), α = 1, ..., N, we want to fit a line

$$Ax + By + C = 0. \tag{3}$$

(Fig. 1(a).) If we define

$$\boldsymbol{\xi}(x,y) \equiv (x,\, y,\, 1)^\top, \qquad \boldsymbol{\theta} \equiv (A,\, B,\, C)^\top, \tag{4}$$

the line equation is written as

$$(\boldsymbol{\xi}(x,y), \boldsymbol{\theta}) = 0. \tag{5}$$

Example 2: (Ellipse fitting) To a given point sequence (x_α, y_α), α = 1, ..., N, we want to fit an ellipse

$$Ax^2 + 2Bxy + Cy^2 + 2(Dx + Ey) + F = 0. \tag{6}$$

(Fig. 1(b).) If we define

$$\begin{aligned} &\boldsymbol{\xi}(x,y) \equiv (x^2,\, 2xy,\, y^2,\, 2x,\, 2y,\, 1)^\top, \\ &\boldsymbol{\theta} \equiv (A,\, B,\, C,\, D,\, E,\, F)^\top, \end{aligned} \tag{7}$$

the ellipse equation is written as

$$(\boldsymbol{\xi}(x,y), \boldsymbol{\theta}) = 0. \tag{8}$$

Example 3: (Fundamental matrix computation) Corresponding points (x, y) and (x', y') in two images of the same 3-D scene taken from different positions satisfy the *epipolar equation* [6]

$$(\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \boldsymbol{F}\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}) = 0, \tag{9}$$

where $\boldsymbol{F}$ is called the *fundamental matrix*, from which we can compute the camera positions and the 3-D structure of the scene [6] (Fig. 1(c)). If we define

$$\boldsymbol{\xi}(x,y,x',y') \equiv (xx', xy', x, yx', yy', y, x', y', 1)^\top, \tag{10}$$

$$\boldsymbol{\theta} \equiv (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top, \tag{11}$$

the epipolar equation is written as

$$(\boldsymbol{\xi}(x,y,x',y'), \boldsymbol{\theta}) = 0. \tag{12}$$

B. Modeling of Noise

In the context of image analysis, "noise" means *uncertainty of image processing operations*, rather than random fluctuations over time or space as commonly understood in physics and communications. Standard image processing operations such as feature extraction and edge detection are not perfect and do not necessarily output exactly what we are looking for. We model this uncertainty in statistical terms: the observed value $\boldsymbol{x}_\alpha$ is regarded as a perturbation from its true value $\bar{\boldsymbol{x}}_\alpha$ by an independent random Gaussian variable $\Delta\boldsymbol{x}_\alpha$ of mean $\boldsymbol{0}$ and covariance matrix $V[\boldsymbol{x}_\alpha]$. Furthermore, $V[\boldsymbol{x}_\alpha]$ is assumed to be known *up to scale*. Namely, we write it as

$$V[\boldsymbol{x}_\alpha] = \sigma^2 V_0[\boldsymbol{x}_\alpha] \tag{13}$$

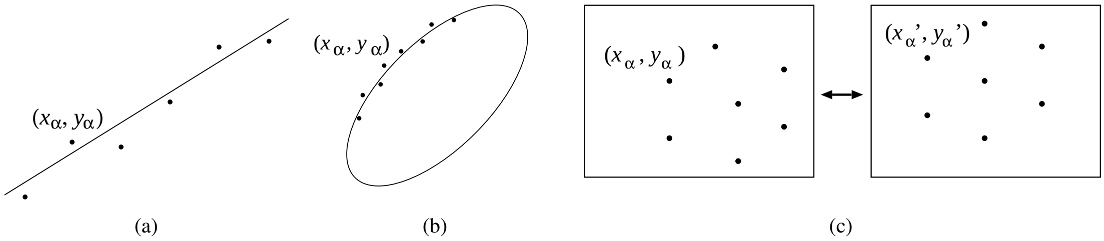


Fig. 1. (a) Line fitting. (b) Ellipse fitting. (c) Fundamental matrix computation.

for some unknown constant σ, which we call the *noise level*. The matrix $V_0[\boldsymbol{x}_\alpha]$, which we call the *normalized covariance matrix*, describes the orientation dependence of uncertainty in relative terms and is assumed to be known.

If the observation $\boldsymbol{x}_\alpha$ is regarded as a random variable, its nonlinear mapping $\boldsymbol{\xi}(\boldsymbol{x}_\alpha)$, which we write $\boldsymbol{\xi}_\alpha$ for short, is also a random variable. Its covariance matrix $V[\boldsymbol{\xi}_\alpha] = \sigma^2 V_0[\boldsymbol{\xi}_\alpha]$ is evaluated to a first approximation in terms of the Jacobi matrix $\partial\boldsymbol{\xi}/\partial\boldsymbol{x}$ of the mapping $\boldsymbol{\xi}(\boldsymbol{x})$ as follows:

$$V_0[\boldsymbol{\xi}_\alpha] = \left.\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}_\alpha} V_0[\boldsymbol{x}_\alpha] \left.\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{x}}\right|^{\top}_{\boldsymbol{x}=\bar{\boldsymbol{x}}_\alpha}. \tag{14}$$

This expression contains the true value $\bar{\boldsymbol{x}}_\alpha$, which is replaced by the observation $\boldsymbol{x}_\alpha$ in actual computation. It has been confirmed by experiments that this replacement does not practically affect the final result. It has also been confirmed that upgrading the first approximation to higher orders does not have any practical effect.

C. Geometric Models for Geometric Estimation

One of the most prominent distinctions of the geometric estimation from the traditional statistical estimation is that the starting equation, Eq. (1) (or Eq. (2)), which we call the *geometric model*, only specifies the necessary constraint and does not explain the mechanism as to how the data $\boldsymbol{x}_\alpha$ are generated. Hence, we cannot express $\boldsymbol{x}_\alpha$ in terms of the parameter $\boldsymbol{\theta}$ as an explicit function. In spite of this differences, two approaches exist in both statistical and geometric estimation domains:

Minimization approach

We choose the value $\boldsymbol{\theta}$ that minimizes a specified cost function. This is regarded as the standard for computer vision applications.

Non-minimization approach

We compute the value $\boldsymbol{\theta}$ by solving a set of equations, called *estimating equations* [4]; the solution does not necessarily minimize any cost function. In traditional statistical estimation domains, this approach is regarded as more general and more flexible with a possibility of yielding better solutions than the minimization approach. However, this is not so widely recognized in computer vision research.

D. KCR Lower Bound

For minimization or non-minimization approaches, there exists a theoretical accuracy limit. If $\boldsymbol{\theta}$ is estimated from noisy observation $\{\boldsymbol{\xi}_\alpha\}_{\alpha=1}^N$ by some means, the resulting estimate $\hat{\boldsymbol{\theta}}$ is as a function $\hat{\boldsymbol{\theta}}(\{\boldsymbol{\xi}_\alpha\}_{\alpha=1}^N)$, called an *estimator* of $\boldsymbol{\theta}$. Let $\Delta\boldsymbol{\theta}$ be its error in $\hat{\boldsymbol{\theta}}$. The covariance matrix of $\hat{\boldsymbol{\theta}}$ is defined by

$$V[\hat{\boldsymbol{\theta}}] = E[\Delta\boldsymbol{\theta}\Delta\boldsymbol{\theta}^\top], \tag{15}$$

where $E[\,\cdot\,]$ denotes expectation over data uncertainty. If we can assume that

1) each $\boldsymbol{\xi}_\alpha$ is perturbed from its true value $\bar{\boldsymbol{\xi}}_\alpha$ by independent Gaussian noise of mean $\boldsymbol{0}$ and covariance matrix $V[\boldsymbol{\xi}_\alpha] = \sigma^2 V_0[\boldsymbol{\xi}_\alpha]$, and
2) the function $\hat{\boldsymbol{\theta}}(\{\boldsymbol{\xi}_\alpha\}_{\alpha=1}^N)$ is an *unbiased estimator*, i.e., $E[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}$ identically holds for all $\boldsymbol{\theta}$,

then the following inequality holds [2], [9], [10], [12].

$$V[\hat{\boldsymbol{\theta}}] \succ \frac{\sigma^2}{N}\Big(\frac{1}{N}\sum_{\alpha=1}^N \frac{\bar{\boldsymbol{\xi}}_\alpha\bar{\boldsymbol{\xi}}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}\Big)^-. \tag{16}$$

Here, $\boldsymbol{A} \succ \boldsymbol{B}$ means that $\boldsymbol{A} - \boldsymbol{B}$ is a positive semidefinite symmetric matrix, and $(\,\cdot\,)^-$ denotes the pseudo inverse. Chernov and Lesort [2] called the right side Eq. (16) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound*.

III. Minimization Approach

First, we overview some popular techniques based on minimization.

A. Least Squares (LS)

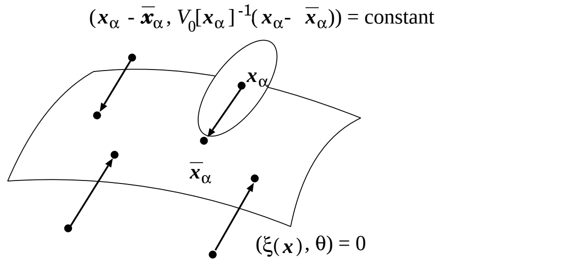
Since the true values $\bar{\boldsymbol{\xi}}_\alpha$ of the observations $\boldsymbol{\xi}_\alpha$ satisfy $(\bar{\boldsymbol{\xi}}_\alpha, \boldsymbol{\theta}) = 0$, we choose the value $\boldsymbol{\theta}$ that minimizes

$$J = \frac{1}{N}\sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 \tag{17}$$

for noisy observations $\boldsymbol{\xi}_\alpha$ subject to the constraint $\|\boldsymbol{\theta}\| = 1$. This can also be viewed as minimizing $\sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2/\|\boldsymbol{\theta}\|^2$. Equation (17) can be rewritten in the form

$$\begin{aligned} J &= \frac{1}{N}\sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 = \frac{1}{N}\sum_{\alpha=1}^N \boldsymbol{\theta}^\top\boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top\boldsymbol{\theta} \\ &= (\boldsymbol{\theta}, \underbrace{\frac{1}{N}\sum_{\alpha=1}^N \boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top}_{\equiv \boldsymbol{M}}\boldsymbol{\theta}) = (\boldsymbol{\theta}, \boldsymbol{M}\boldsymbol{\theta}). \end{aligned} \tag{18}$$

Thee unit vector $\boldsymbol{\theta}$ that minimizes this quadratic form is given by the unit eigenvector of $\boldsymbol{M}$ for the smallest eigenvalue.



$(x_\alpha - \bar{x}_\alpha, V_0[x_\alpha]^{-1}(x_\alpha - \bar{x}_\alpha))$ = constant
x_α
$\bar{x}_\alpha$
$(\xi(x), \theta) = 0$

Fig. 2. Fitting a hypersurface $(\boldsymbol{\xi}(\boldsymbol{x}), \boldsymbol{\theta}) = 0$ to points $\boldsymbol{x}_\alpha$ in the data space.

This method is called *least squares* (*LS*) or *algebraic distance minimization*. Because the solution is directly obtained without any search, LS is widely used in many applications. However, it has been observed that the solution has a large statistical bias. For ellipse fitting in *Example 2*, for instance, the fitted ellipse is almost always smaller than the true shape.

B. Maximum Likelihood (ML)

If the noise in each $\boldsymbol{x}_\alpha$ is an independent Gaussian variable of mean $\mathbf{0}$ and covariance matrix $V[\boldsymbol{x}_\alpha] = \sigma^2 V_0[\boldsymbol{x}_\alpha]$, the *Mahalanobis distance* of the observations $\{\boldsymbol{x}_\alpha\}$ from their true values $\{\bar{\boldsymbol{x}}_\alpha\}$ is

$$J = \frac{1}{N}\sum_{\alpha=1}^{N}(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha)), \qquad (19)$$

and *maximum likelihood* (*ML*) minimizes this subject

$$(\boldsymbol{\xi}(\bar{\boldsymbol{x}}_\alpha), \boldsymbol{\theta}) = 0. \qquad (20)$$

Geometrically, ML can be interpreted to be fitting to N points $\boldsymbol{x}_\alpha$ in the data space the parameterized hypersurface $(\boldsymbol{\xi}(\boldsymbol{x}), \boldsymbol{\theta})$ = 0 by adjusting $\boldsymbol{\theta}$ (Fig. 2), where the discrepancy of the points from the surface is measured not by the Euclid distance but by the Mahalanobis distance of Eq. (19) inversely weighted by the covariances, thereby imposing heavier penalties on the points with higher certainty.

If the noise is homogeneous (i.e., independent of α) and isotropic (i.e., independent of orientation), we can write $V_0[\boldsymbol{x}_\alpha]$ = $\boldsymbol{I}$ (the identity), which reduces Eq. (19) to the *geometric distance*

$$J = \frac{1}{N}\sum_{\alpha=1}^{N}\|\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha\|^2. \qquad (21)$$

Minimizing this subject to Eq. (20) is called *geometric distance minimization* or *total least squares* (*TLS*). It is also called *reprojection error minimization* for 3-D reconstruction, where $\bar{\boldsymbol{x}}_\alpha$ represents the projection of the assumed 3-D structure onto the image plane and $\boldsymbol{x}_\alpha$ is its actually observed positions. For computer vision applications, this approach is widely regarded as the ultimate method and called the *Gold Standard* [6]. However, this is a highly nonlinear optimization and difficult to solve by a direct means. The difficulty stems from the fact that the parameter $\boldsymbol{\theta}$ to be optimized is not contained in the cost function J but only linked to the constraint of Eq. (20), which cannot be solved for $\boldsymbol{\theta}$.

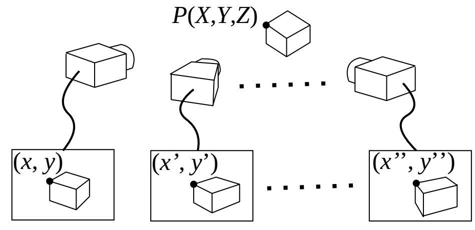


Fig. 3. 3-D reconstruction by bundle adjustment

C. Bundle Adjustment

The standard technique for minimizing Eq. (19) subject to Eq. (20) is to introduce a problem-dependent auxiliary variable to each $\boldsymbol{X}_\alpha$ and express $\bar{\boldsymbol{x}}_\alpha$ in terms of $\boldsymbol{X}_\alpha$ and $\boldsymbol{\theta}$ in the form

$$\bar{\boldsymbol{x}}_\alpha = \bar{\boldsymbol{x}}_\alpha(\boldsymbol{X}_\alpha, \boldsymbol{\theta}). \qquad (22)$$

Substituting this into Eq. (19), we minimize

$$J(\{\boldsymbol{X}_\alpha\}_{\alpha=1}^{N}, \boldsymbol{\theta}) = \frac{1}{N}\sum_{\alpha=1}^{N}(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha(\boldsymbol{X}_\alpha, \boldsymbol{\theta}), V_0[\boldsymbol{x}_\alpha]^{-1}(\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha(\boldsymbol{X}_\alpha, \boldsymbol{\theta}))), \qquad (23)$$

over the joint parameter space of $\{\boldsymbol{X}_\alpha\}_{\alpha=1}^{N}$ and $\boldsymbol{\theta}$.

A typical example of this approach is 3-D reconstruction from multiple images (Fig. 3), for which $\boldsymbol{x}_\alpha$ has the form of $(x_\alpha, y_\alpha, x'_\alpha, y'_\alpha, ..., x''_\alpha, y''_\alpha)$, concatenating the projections (x_α, y_α), (x'_α, y'_α), ..., (x''_α, y''_α) of the αth point in the scene onto the images. The parameter $\boldsymbol{\theta}$ specifies the state of all the cameras, consisting of the extrinsic and intrinsic parameters. If we introduce the 3-D position $\boldsymbol{X}_\alpha = (X_\alpha, Y_\alpha, Z_\alpha)$ of each point as an auxiliary variable, the true value $\bar{\boldsymbol{x}}_\alpha$ of $\boldsymbol{x}_\alpha$ can be explicitly expressed in the form $\bar{\boldsymbol{x}}_\alpha(\boldsymbol{X}_\alpha, \boldsymbol{\theta})$, which describes the image positions of the 3-D point $\boldsymbol{X}_\alpha$ that should be observed if the cameras have the parameter $\boldsymbol{\theta}$. Then, the discrepancy of the observed projections $\boldsymbol{x}_\alpha$ from the predicted projections $\bar{\boldsymbol{x}}_\alpha(\boldsymbol{X}_\alpha, \boldsymbol{\theta})$, i.e., the "reprojection" error, is minimized over the joint parameter space of $\{\boldsymbol{X}_\alpha\}_{\alpha=1}^{N}$ and $\boldsymbol{\theta}$. This process is called *bundle adjustment* [21], [26]. The dimension of the parameter space is $3N$ + (the dimension of $\boldsymbol{\theta}$), which becomes very large when many points are observed.

The standard numerical search technique is the *Levenberg-Marquardt* (*LM*) *method* [23], but, depending on the initial value of the iterations, the search may fall into a local minimum. Various global optimization techniques have also been studied [5]. A typical method is *branch and bound*, which introduces a function that gives a lower bound of J over a given region and divides the parameter space into small cells; those cells which have lower bounds that are above the tested values are removed, and other cells are recursively subdivided [5], [7]. However, the evaluation of the lower bound involves a complicated technique, and searching the entire space requires a significant amount of computational time.

D. Gaussian Approximation of Noise in the $\boldsymbol{\xi}$-Space

If the noise in the observation $\boldsymbol{x}_\alpha$ is Gaussian, the noise in its nonlinear transformation $\boldsymbol{\xi}_\alpha = \boldsymbol{\xi}(\boldsymbol{x}_\alpha)$ is not strictly Gaussian, although it is expected to be Gaussian-like for small

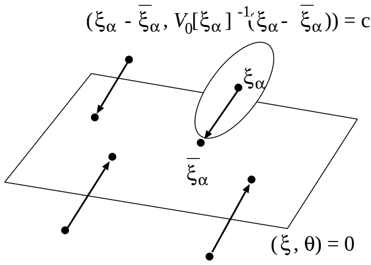

$(\xi_\alpha - \bar{\xi}_\alpha, V_0[\xi_\alpha]^{-1}(\xi_\alpha - \bar{\xi}_\alpha)) = \text{constant}$
ξ_α
$\bar{\xi}_\alpha$
$(\xi, \theta) = 0$

Fig. 4. Fitting a hyperplane $(\boldsymbol{\xi}, \boldsymbol{\theta}) = 0$ to points $\boldsymbol{\xi}_\alpha$ in the $\boldsymbol{\xi}$-space.

noise. If it is approximated to be Gaussian, the optimization computation becomes much simpler. Suppose $\boldsymbol{\xi}_\alpha$ has noise of mean $\mathbf{0}$ with the covariance matrix $V[\boldsymbol{\xi}_\alpha] = \sigma^2 V_0[\boldsymbol{\xi}_\alpha]$ evaluated by Eq. (14). Then, the ML computation reduces to minimizing the Mahalanobis distance

$$J = \frac{1}{N}\sum_{\alpha=1}^{N}(\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha, V_0[\boldsymbol{\xi}_\alpha]^{-1}(\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha)) \tag{24}$$

in the $\boldsymbol{\xi}$-space subject to the *linear* constraint

$$(\bar{\boldsymbol{\xi}}_\alpha, \boldsymbol{\theta}) = 0. \tag{25}$$

Geometrically, this is interpreted to be fitting to N points $\boldsymbol{\xi}_\alpha$ in the $\boldsymbol{\xi}$-space the parameterized "hyperplane" $(\boldsymbol{\xi}, \boldsymbol{\theta}) = 0$ by adjusting $\boldsymbol{\theta}$, where the discrepancy of the points from the plane is measured by the Mahalanobis distance of Eq. (24) inversely weighted by the covariances of the data in the $\boldsymbol{\xi}$-space (Fig. 4).

The constraint of Eq. (25) can be eliminated using Lagrange multipliers, leading to unconstrained minimization of

$$J = \frac{1}{N}\sum_{\alpha=1}^{N}\frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \tag{26}$$

which is known as the *Sampson error* [6].

E. Sampson Error Minimization

Various numerical techniques have been proposed for minimizing the Sampson error of Eq. (26). The best known is the *FNS* (*Fundamental Numerical Scheme*) of Chojnacki et al. [3], which goes as follows:

1) Let $W_\alpha = 1$, $\alpha = 1, ..., N$, and $\boldsymbol{\theta}_0 = \mathbf{0}$.
2) Comput the matrices

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top,$$
$$\boldsymbol{L} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha^2 (\boldsymbol{\theta}_0, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]. \tag{27}$$

3) Solve the eigenvalue problem $(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{\theta} = \lambda\boldsymbol{\theta}$, and compute the unit eigenvector $\boldsymbol{\theta}$ for the smallest eigenvalue λ.
4) If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, let

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}, \tag{28}$$

and go back to Step 2.

Other methods exist, including the *HEIV* (*Heteroscedastic Errors-in-Variables*) of Leedan and Meer [20] and Matei and Meer [22], and the *projective Gauss-Newton iterations* of Kanatani and Sugaya [16].

F. Computation of the Exact ML Solution

Since the Sampson error of Eq. (26) is obtained by approximating the non-Gaussian noise distribution in the $\boldsymbol{\xi}$-space by a Gaussian distribution, the solution does not necessarily coincide with the ML solution that minimizes the Mahalanobis distance of Eq. (19). However, once we have obtained the solution $\boldsymbol{\theta}$ that minimizes Eq. (26), we can iteratively modify Eq. (26) by using that $\boldsymbol{\theta}$ so that Eq. (26) coincides with Eq. (19) in the end, meaning that we obtain the exact ML solution. The procedure goes as follows [19]:

1) Let $J_0^* = \infty$ (a sufficiently large number), $\hat{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha$, and $\tilde{\boldsymbol{x}}_\alpha = \mathbf{0}$, $\alpha = 1, ..., N$.
2) Evaluate the normalized covariance matrices $V_0[\hat{\boldsymbol{\xi}}_\alpha]$ by replacing $\boldsymbol{x}_\alpha$ by $\hat{\boldsymbol{x}}_\alpha$ in their definition.
3) Compute the following $\boldsymbol{\xi}_\alpha^*$:

$$\boldsymbol{\xi}_\alpha^* = \boldsymbol{\xi}_\alpha + \left.\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_\alpha} \tilde{\boldsymbol{x}}_\alpha. \tag{29}$$

4) Compute the value $\boldsymbol{\theta}$ that minimizes the *modified Sampson error*

$$J^* = \frac{1}{N}\sum_{\alpha=1}^{N}\frac{(\boldsymbol{\xi}_\alpha^*, \boldsymbol{\theta})^2}{(\boldsymbol{\theta}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})}. \tag{30}$$

5) Update $\tilde{\boldsymbol{x}}_\alpha$ and $\hat{\boldsymbol{x}}_\alpha$ as follows:

$$\tilde{\boldsymbol{x}}_\alpha \leftarrow \frac{(\boldsymbol{\xi}_\alpha^*, \boldsymbol{\theta}) V_0[\boldsymbol{x}_\alpha]}{(\boldsymbol{\theta}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{\theta})} \left.\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_\alpha}^\top \boldsymbol{\theta}, \quad \hat{\boldsymbol{x}}_\alpha \leftarrow \boldsymbol{x}_\alpha - \tilde{\boldsymbol{x}}_\alpha. \tag{31}$$

6) Evaluate J^* by

$$J^* = \frac{1}{N}\sum_{\alpha}^{N}(\tilde{\boldsymbol{x}}_\alpha, V_0[\boldsymbol{x}_\alpha]\tilde{\boldsymbol{x}}_\alpha). \tag{32}$$

If $J^* \approx J_0$, return $\boldsymbol{\theta}$ and stop. Else, let $J_0 \leftarrow J^*$ and go back to Step 2.

Since the modified Sampson error in Eq. (30) has the same form as the Sampson error in Eq. (26), we can minimize it by FNS (or other methods). According to numerical experiments, this modification converges after four or five rounds, yet in many practical problems the first four or five effective digits remain unchanged [17], [18]. In this sense, we can practically identify the Sampson error minimization with the ML computation.

G. Hyperaccurate Correction of ML

It is known that the ML solution, with which the Sampson error minimization solution can be identified, has statistical bias of $O(\sigma^2)$. Hence, the accuracy is further improved by subtracting the expected bias. This process is called *hyperaccurate correction* and goes as follows [11], [12]:

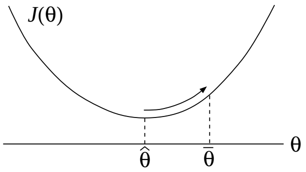


Fig. 5. If the value $\hat{\boldsymbol{\theta}}$ that minimizes the cost function $J(\boldsymbol{\theta})$ is biased from the true value $\bar{\boldsymbol{\theta}}$, we correct $\hat{\boldsymbol{\theta}}$ so that it approaches $\bar{\boldsymbol{\theta}}$ in expectation. As a result, the value of $J(\boldsymbol{\theta})$ increases.

1) Estimate the square noise level σ^2 from the ML solution $\hat{\boldsymbol{\theta}}$ and the corresponding matrix $\boldsymbol{M}$ in Eq. (27) by
$$\hat{\sigma}^2 = \frac{(\hat{\boldsymbol{\theta}}, \boldsymbol{M}\hat{\boldsymbol{\theta}})}{1-(n-1)/N}, \tag{33}$$
where n is the dimension of the vector $\hat{\boldsymbol{\theta}}$.
2) Compute the correction term
$$\Delta_c\boldsymbol{\theta} = -\frac{\hat{\sigma}^2}{N}\boldsymbol{M}^-\sum_{\alpha=1}^{N} W_\alpha(\boldsymbol{e}_\alpha, \hat{\boldsymbol{\theta}})\boldsymbol{\xi}_\alpha + \frac{\hat{\sigma}^2}{N^2}\boldsymbol{M}^-\sum_{\alpha=1}^{N} W_\alpha^2(\boldsymbol{\xi}_\alpha, \boldsymbol{M}^- V_0[\boldsymbol{\xi}_\alpha]\hat{\boldsymbol{\theta}})\boldsymbol{\xi}_\alpha, \tag{34}$$
where $\boldsymbol{e}_\alpha$ is a vector that depends on individual problems, and $\boldsymbol{M}^-$ is the truncated pseudoinverse (the smallest eigenvalue is replaced by 0 in its spectral decomposition).
3) Correct $\hat{\boldsymbol{\theta}}$ in the form
$$\hat{\boldsymbol{\theta}} \leftarrow \mathcal{N}[\hat{\boldsymbol{\theta}} - \Delta_c\boldsymbol{\theta}], \tag{35}$$
where $\mathcal{N}[\,\cdot\,]$ is the normalization operator into unit norm ($\mathcal{N}[\boldsymbol{a}] \equiv \boldsymbol{a}/\|\boldsymbol{a}\|$).

The vector $\boldsymbol{e}_\alpha$ is $\boldsymbol{0}$ for many problems including line fitting (*Example 1*) and fundamental matrix computation (*Example 3*). It is generally $\boldsymbol{0}$ if multiple images are involved. A typical problem of nonzero $\boldsymbol{e}_\alpha$ is ellipse fitting (*Example 2*), for which $\boldsymbol{e}_\alpha = (1,0,1,0,0,0)^\top$. However, the effect is negligibly small, and the solution is practically the same if $\boldsymbol{e}_\alpha$ is replaced by $\boldsymbol{0}$.

However, bias correction is a departure from the minimization principle. Namely, we correct the computed $\hat{\boldsymbol{\theta}}$ that minimizes the cost function so that it approaches the true value $\bar{\boldsymbol{\theta}}$ in expectation. As a result, *the value of the cost function increases* (Fig. 5). Then, why not directly compute, by some means, that optimal value from the beginning? This is the motivation of the "non-minimization approach", directly solving some equation rather than minimizing some cost function.

IV. Non-minimization Approach

A. Principle

We reiterate our goal: appropriately assuming the statistical properties of the noise, we infer from noisy observations the values that would satisfy given constraints in the absence of noise. This problem could be reduced to minimization of some cost function $J(\boldsymbol{\theta})$, but *we need not necessarily do so.*

The minimization approach relies on the knowledge that the value $\boldsymbol{\theta}$ that minimizes $J(\boldsymbol{\theta})$ is expected, either theoretically or experimentally, to be close to the true value $\bar{\boldsymbol{\theta}}$. However, how close it is is determined by the function $J(\boldsymbol{\theta})$ and is not of our control. In contrast, the non-minimization approach directly gives a procedure for computing $\boldsymbol{\theta}$, for which the following three stages are involved:

1) Devise a scheme for computing $\bar{\boldsymbol{\theta}}$ from the data $\boldsymbol{x}_\alpha$, assuming that they have noiseless values $\bar{\boldsymbol{x}}_\alpha$.
2) Substitute $\boldsymbol{x}_\alpha = \bar{\boldsymbol{x}}_\alpha + \Delta\boldsymbol{x}_\alpha$ in the scheme and evaluate the error $\Delta\boldsymbol{\theta}$ of the computed value $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} + \Delta\boldsymbol{\theta}$ in terms of the noise terms $\Delta\boldsymbol{x}_\alpha$.
3) Modify the scheme so that $\Delta\boldsymbol{\theta}$ is minimized in some sense, e.g., reducing its RMS and/or bias.

The core is Stage 1, which makes the non-minimization approach possible, because *geometry is exact if the observation is exact*. Namely, we can compute the true geometry for exact data. This is a big contrast to the traditional statistical domains, such as agriculture, pharmaceutics, and economics, in which what is going on is uncertain from the beginning, hence statistical inference is called for. As a result, such concepts as "exact observation" and "exact solution" do not make sense.

Stage 2 can be done by the well kwon perturbation analysis, also called "error propagation", because *we know the underlying exact geometry*, although usually tedious and messy calculus is required. The perturbation analysis is done by expanding quantities in the noise level σ and omitting higher order terms in σ. This is justified because we are focusing on the uncertainty behavior in a small noise range.

Stage 3 is not obvious and is the most difficult part; we need to introduce ingenious and clever tricks. For example, if the scheme contains a constant that does not affect the final result if all the data are exact *yet influences the computation in the presence of noise*, its value is optimally chosen so that the highest accuracy is achieved.

One side effect is this non-minimization formalism based on error analysis is that when the final computational procedure is presented, it is often difficult to grasp its intuitive meaning, as we will see in the following. Perhaps, this is the main reason that this approach has been not so popular or widely accepted in the domain of computer vision.

B. Estimating equation

Existing geometric estimation methods without minimizing any cost function have different motivations and derivations. Yet, all have a similarity: they solve a matrix equation in the form
$$\boldsymbol{M}\boldsymbol{\theta} = \lambda\boldsymbol{N}\boldsymbol{\theta} \tag{36}$$
where $\boldsymbol{M}$ and $\boldsymbol{N}$ are matrices defined in terms of the observations $\boldsymbol{\xi}_\alpha$ and the parameter $\boldsymbol{\theta}$. Equation (36) plays the role of the *estimating equation*, and different choices of the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ result in different techniques:

Least Squares (LS)

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \qquad \boldsymbol{N} = \boldsymbol{I}. \tag{37}$$

Taubin method [25]

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \quad \boldsymbol{N} = \sum_{1} N V_0[\boldsymbol{\xi}_\alpha]. \tag{38}$$

HyperLS [14], [15], [24]

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top.$$

$$\boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N}\Big(V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[\boldsymbol{\xi}_\alpha \boldsymbol{e}_\alpha^\top]\Big) - \frac{1}{N^2}\sum_{\alpha=1}^{N}\Big((\boldsymbol{\xi}_\alpha, \boldsymbol{M}^- \boldsymbol{\xi}_\alpha) V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{M}^- \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top]\Big). \tag{39}$$

Iterative reweight

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \qquad \boldsymbol{N} = \boldsymbol{I}. \tag{40}$$

Renormalization [8], [9]

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \;\; \boldsymbol{N} = \frac{1}{N}\sum_{1}^{N} \frac{V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{\theta}. V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}. \tag{41}$$

Hyper-renormalization [13]

$$\boldsymbol{M} = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]]\boldsymbol{\theta})},$$

$$\boldsymbol{N} = \frac{1}{N}\sum_{\alpha=1}^{N} \frac{V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[\boldsymbol{\xi}_\alpha \boldsymbol{e}_\alpha^\top]}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})} - \frac{1}{N^2}\sum_{\alpha=1}^{N} \frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{M}^- \boldsymbol{\xi}_\alpha) V_0[\boldsymbol{\xi}_\alpha] + 2\mathcal{S}[V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{M}^- \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top]}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})^2}. \tag{42}$$

C. Algebraic solution

Since the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ do not contain $\boldsymbol{\theta}$ for LS, the Taubin method, and HyperLS, the solution $\boldsymbol{\theta}$ is immediately obtained by solving the generalized eigenvalue problem of Eq. (36), computing the unit eigenvector $\boldsymbol{\theta}$ for the smallest absolute eigenvalue λ. Standard numerical tools assume that $\boldsymbol{N}$ is positive definite, but this is not necessarily the case for the above forms of $\boldsymbol{N}$. This causes no trouble, because Eq. (36) can be rewritten as

$$\boldsymbol{N}\boldsymbol{\theta} = \frac{1}{\lambda}\boldsymbol{M}\boldsymbol{\theta}. \tag{43}$$

The matrix $\boldsymbol{M}$ is positive definite for noisy data, so we can use a standard tool to compute the unit eigenvector $\boldsymbol{\theta}$ for the lagest absolute eigenvalue $1/\lambda$. If the matrix $\boldsymbol{M}$ happens to have eigenvalue 0, it indicates that the data are all exact, so the corresponding eigenvector is the exact solution.

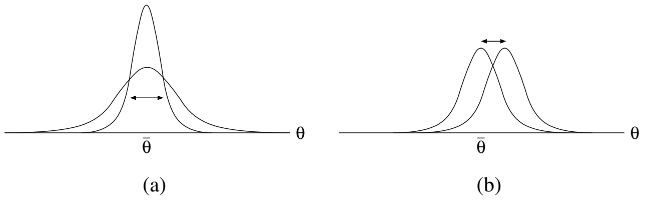


Fig. 6. (a) The matrix $\boldsymbol{M}$ determines the covariance of the solution. (b) The matrix $\boldsymbol{N}$ controls the bias of the solution.

D. Iterative procedure

For iterative reweight, renormalization, and hyper-renormalization, the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ contain the unknown $\boldsymbol{\theta}$. We can solve Eq. (36) by the following iterative procedure:

1) Let $\boldsymbol{\theta}_0 = \boldsymbol{0}$ and compute the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$, where $(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})$ is replaced by 1.
2) Solve the generalized eigenvalue problem of Eq. (36) and compute the unit eigenvector $\boldsymbol{\theta}$ for the smallest absolute eigenvalue λ.
3) If $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ up to sign, return $\boldsymbol{\theta}$ and stop. Else, recompute $\boldsymbol{M}$ and $\boldsymbol{N}$ using the computed $\boldsymbol{\theta}$, let $\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}$, and go back to Step 2.

We can use standard numerical tools by converging the generalized eigenvalue problem in the form of Eq. (43).

E. Analysis of Covariance and Bias

Since we regard the observations $\boldsymbol{\xi}_\alpha$, α = 1, ..., N as random variables, the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ in Eq. (36) constructed using them are random variables, too. Hence, the computed $\boldsymbol{\theta}$ is also a random variable, so it has a probability distribution $p(\boldsymbol{\theta})$. We can apply the perturbation technique of Kanatani [12] to analyze the statistical properties of the compute $\boldsymbol{\theta}$. It turns out that:

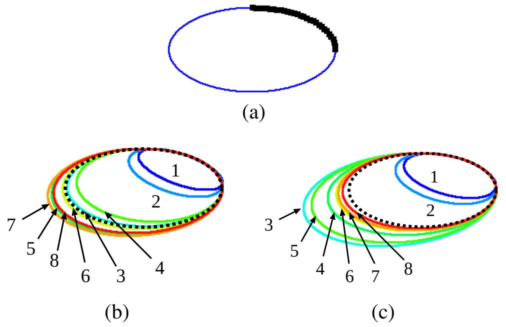
- The matrix $\boldsymbol{M}$ determines the covariance of the computed $\boldsymbol{\theta}$ (Fig. 6(a)).
- The matrix $\boldsymbol{N}$ controls the bias of the computed $\boldsymbol{\theta}$ (Fig. 6(b)).

Detailed analysis reveals the following facts [1], [12], [13]:

- For iterative reweight, renormalization, and hyper-renormalization, the first order covariance matrix $V[\boldsymbol{\theta}]$ reaches the KCR lower bound of Eq. (16) up to $O(\sigma^4)$.
- For LS, iterative reweight, the Taubin method, and renormalization, the solution has bias of $O(\sigma^2)$.
- The solution of HyperLS and hyper-renormalization has no bias up to $O(\sigma^4)$.

Thus, we conclude that

- Hyper-renormalization is optimal in terms of both covariance and bias. Its covariance matrix achieves the KCR lower bound up to $O(\sigma^4)$ and has no bias up to $O(\sigma^4)$.



method		2	4	6	7/8
number of iterations	(a)	4	4	4	9
	(b)	4	4	4	8

Fig. 7. (a) Thirty points on an ellipse. (b), (c) Fitted ellipses ($\sigma = 0.5$ pixels). 1) LS, 2) iterative reweight, 3) Taubin, 4) renormalization, 5) HyperLS, 6) hyper-renormalization, 7) ML, 8) ML with hyperaccurate correction. The dotted lines indicate the true shape. The table lists the number of iterations for methods 2, 4, 6, and 7/8 (methods 1, 3, and 5 are noniterative, and methods 7 and 8 have the same complexity).

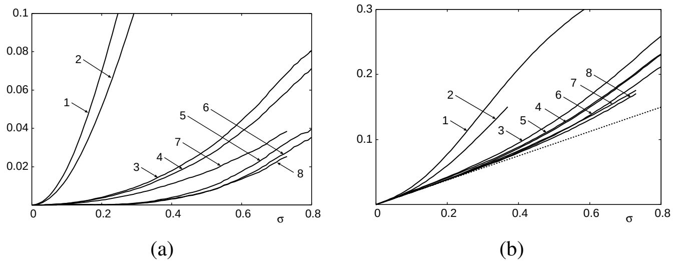


Fig. 8. The bias (a) and the RMS error (b) of the fitted ellipse for the standard deviation σ of the added noise over 10000 independent trials. 1) LS, 2) iterative reweight, 3) the Taubin method, 4) renormalization, 5) HyperLS, 6) hyper-renormalization, 7) ML, 8) hyperaccurate correction of ML. The dotted line in (c) indicates the KCR lower bound.

V. Numerical Examples

We define 30 equidistant points on the ellipse shown in Fig. 7(a). The major and minor axis are set to 100 and 50 pixels, respectively. We added random Gaussian noise of mean 0 and standard deviation σ to the x and y coordinates of each point independently and fit an ellipse, using: 1) LS, 2) iterative reweight, 3) the Taubin method, 4) renormalization, 5) HyperLS, 6) hyper-renormalization, 7) ML, and 8) ML with hyperaccurate correction.

Fig. 7(b), (c) show fitted ellipses for $\sigma = 0.5$ pixels; although the noise magnitude is the same, the resulting ellipses are different for different noise. The true shape is indicated by dotted lines. We can see that LS and iterative reweight have large bias, producing much smaller ellipses than the true shape. The closest ellipse is given by hyper-renormalization in Fig. 7(b) and by ML with hyperaccurate correction in Fig. 7(c).

The number of iterations for each method is also shown there. We see that ML (we used FNS) with/without hyperaccurate correction requires about twice as many iterations for convergence as iterative reweight, renormalization, and hyper-renormalization; Taubin and HyperLS are noniterative algebraic methods, while hyperaccurate correction is an analytical procedure after ML has converged. The fast convergence of hyper-renormalization is a result of its initialization by HyperLS, while FNS for ML starts from LS.

Note that all the ellipses in Fig. 7(b),(c) fit fairly well to the data points, meaning that not much difference exists among their geometric errors, i.e., the sums of the square distances of the data points from the fitted ellipse. However, the deviation is large in the part where *no data points exist*. Since $\boldsymbol{\theta}$ expresses the coefficients of the ellipse equation, the error $\Delta\boldsymbol{\theta}$ evaluates how the "ellipse equation", i.e., the ellipse itself, differs. This implies that the geometric error is not a good measure of ellipse fitting; we need to evaluate the error in $\boldsymbol{\theta}$.

We evaluated the accuracy of $\boldsymbol{\theta}$ in statical terms. Since the computed $\boldsymbol{\theta}$ and its true value $\bar{\boldsymbol{\theta}}$ are both unit vectors, we measure the discrepancy between them by the orthogonal component $\Delta^\perp\boldsymbol{\theta} = \boldsymbol{P}_{\bar{\boldsymbol{\theta}}}\boldsymbol{\theta}$, where $\boldsymbol{P}_{\bar{\boldsymbol{\theta}}}$ ($\equiv \boldsymbol{I} - \bar{\boldsymbol{\theta}}\bar{\boldsymbol{\theta}}^\top$) is the projection matrix along $\bar{\boldsymbol{\theta}}$. We generated 10000 independent noise instances and evaluated the bias B (Fig. 8(b)) and the RMS (root-mean-square) error D (Fig. 8(c)) defined by

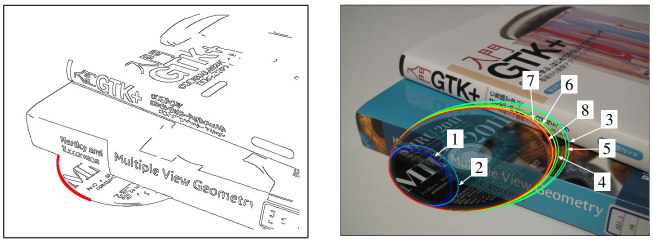
$$B = \left\| \frac{1}{10000}\sum_{a=1}^{10000} \Delta^\perp\boldsymbol{\theta}^{(a)} \right\|, \qquad D = \sqrt{\frac{1}{10000}\sum_{a=1}^{10000} \|\Delta^\perp\boldsymbol{\theta}^{(a)}\|^2}, \tag{44}$$

where $\boldsymbol{\theta}^{(a)}$ is the solution in the ath trial. The dotted line in Fig. 8(c) indicates the KCR lower bound.

The interrupted plots in Fig. 8(a) for iterative reweight and ML (we used FNS) with/without hyperaccurate correction indicate that the iterations did not converge beyond that noise level. Our convergence criterion is $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\| < 10^{-6}$ for the current value $\boldsymbol{\theta}$ and the value $\boldsymbol{\theta}_0$ in the preceding iteration; their signs are adjusted before subtraction. If this criterion is not satisfied after 100 iterations, we stopped. For each σ, we regarded the iterations as not convergent if any among the 10000 trials does not converge.

We can see from Fig. 8(a) that LS and iterative reweight have very large bias, in contrast to which the bias is very small for the Taubin method and renormalization. The bias of HyperLS and hyper-renormalization is still smaller and even smaller than ML. Note that the RMS error D of Eq. (44) is a measure of the distance from the true value $\bar{\boldsymbol{\theta}}$, while the first order covariance matrix $V[\boldsymbol{\theta}]$ measure the "width" of the bell-shaped distribution from its center (see Fig. 6(a)), which is common to iterative reweight, renormalization, and hyper-renormalization. Hence, the difference in the RMS error in Fig. 8(b) is a direct consequence of the bias difference shown in Fig. 8(a).

Since the hyper-renormalization solution does not have bias up to $O(\sigma^4)$, a close examination of the small σ part reveals that hyper-renormalization outperforms ML. The highest accuracy is achieved, although the difference is very small, by ML with hyperaccurate correction. However, it first requires the ML solution, and the FNS iterations for its computation may not converge above a certain noise level, as shown in Figs. 8(a), (b). On the other hand, hyper-renormalization is not so sensitive to noise. This is because the first iteration step coincides with HyperLS. We conclude that hyper-renormalization is the



(a)

(b)

method	2	4	6	7/8
# of iter.	4	3	3	6

Fig. 9. (a) An edge image of a scene with a circular object. An ellipse is fitted to the 160 edge points indicated. (b) Fitted ellipses superimposed on the original image. The occluded part is artificially composed for visual ease. 1) LS, 2) iterative reweight, 3) Taubin, 4) renormalization, 5) HyperLS, 6) hyper-renormalization, 7) ML, 8) ML with hyperaccurate correction.

best method in terms of accuracy and efficiency for practical computation.

Figure 9(a) is an edge image of a scene with a circular object. We fitted an ellipse to the 160 edge points indicated there, using various methods. Figure 9(b) shows the fitted ellipses superimposed on the original image, where the occluded part is artificially composed for visual ease. We can see that LS and iterative reweight produce much smaller ellipses than the true shape as in Fig. 7(b),(c). All other fits are very close to the true ellipse, and ML gives the best fit in this particular instance. The number of iterations before convergence for each method is also shown in Fig. 9. Again, FNS for ML with/without hyperaccurate correction required about twice as many iterations as other methods.

VI. Concluding Remarks

We have overviewed techniques for optimal geometric estimation from noisy observations for computer vision applications. We first described approaches based on minimization: LS, ML, and Sampson error minimization. We then summarized the non-minimization approach of solving a matrix equation. Different choices of the matrices in it result in different methods: LS, iterative reweight, the Taubin method, renormalization, HyperLS, and hyper-renormalization. Doing statistical analysis and conducting numerical examples, we conclude that hyper-renormalization is the best method in terms of accuracy and efficiency.

Acknowledgments: A major part of this paper resulted from the author's collaboration with Prasanna Rangarajan of Southern Methodist University, U.S.A., Ali Al-Sharadqah of East Caroline University, U.S.A., Nikolai Chernov of the University of Alabama at Birmingham, U.S.A., and Yasuyuki Sugaya of Toyohashi University of Technology, Japan.

References

[1] Al-Sharadqah, A., Chernov, N.: A doubly optimal ellipse fit. Comp. Stat. Data Anal. 56(9), 2771–2781 (2012)

[2] Chernov, N., Lesort, C.: Statistical efficiency of curve fitting algorithms. Comp. Stat. Data Anal. 47(4), 713–728 (2004)

[3] Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D.: On the fitting of surfaces to data with covariances. IEEE Trans. Patt. Anal. Mach. Intell. 22(11), 1294–1303 (2000)

[4] Godambe, V.P. (ed.): Estimating Functions. Oxford University Press, New York (1991)

[5] Hartley, R., Kahl, F.: Optimal algorithms in multiview geometry. In: Proc. 8th Asian Conf. Comput. Vis., Tokyo, Japan, vol. 1, pp. 13–34 (November 2007)

[6] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)

[7] Kahl, F., Agarwal, S., Chandraker, M.K., Kriegman, D., Belongie, S.: Practical global optimization for multiview geometry. Int. J. Comput. Vis. 79(3), 271–284 (2008)

[8] Kanatani, K.: Renormalization for unbiased estimation. In: Proc. 4th Int. Conf. Comput. Vis., Berlin, Germany, pp. 599–606 (May 1993)

[9] Kanatani, K.: Statistical Optimization for Geometric Computation: Theory and Practice. Elsevier, Amsterdam (1996); reprinted, Dover, New York (2005)

[10] Kanatani, K.: Cramer-Rao lower bounds for curve fitting. Graphical Models Image Process. 60(2), 93–99 (1998)

[11] Kanatani, K.: Ellipse fitting with hyperaccuracy. IEICE Trans. Inf. & Syst. E89-D(10), 2653–2660 (2006)

[12] Kanatani, K.: Statistical optimization for geometric fitting: Theoretical accuracy analysis and high order error analysis. Int. J. Comput. Vis. 80(2), 167–188 (2008)

[13] Kanatani, K., Al-Sharadqah, A., Chernov, N., Sugaya, Y.: Renormalization Returns: Hyper-renormalization and Its Applications. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmidt. C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 348–397. Springer, Heidelberg (2012).

[14] Kanatani K., Rangarajan, P.: Hyper least squares fitting of circles and ellipses. Comput. Stat. Data Anal. 55(6), 2197–2208 (2011)

[15] Kanatani, K., Rangarajan, P., Sugaya, Y., Niitsuma, H.: HyperLS and its applications. IPSJ Trans. Comput. Vis. Appl. 3, 80–94 (2011)

[16] Kanatani, K., Sugaya, Y.: Performance evaluation of iterative geometric fitting algorithms. Comp. Stat. Data Anal. 52(2), 1208–1222 (2007)

[17] Kanatani, K., Sugaya, Y.: Compact algorithm for strictly ML ellipse fitting. In: Proc. 19th Int. Conf. Patt. Recog., Tampa, FL (December 2008)

[18] Kanatani, K., Sugaya, Y.: Compact fundamental matrix computation. IPSJ Tran. Comput. Vis. Appl. 2, 59–70 (2010)

[19] Kanatani, K., Sugaya, Y.: Unified computation of strict maximum likelihood for geometric fitting. J. Math. Imaging Vis. 38(1), 1–13 (2010)

[20] Leedan, Y., Meer, P.: Heteroscedastic regression in computer vision: Problems with bilinear constraint. Int. J. Comput. Vis. 37(2), 127-150 (2000)

[21] Lourakis, M.I.A., Argyros, A.A.: SBA: A software package for generic sparse bundle adjustment. ACM Trans. Math. Software 36(1), 2, 1–30 (2009)

[22] Matei, J., Meer, P.: Estimation of nonlinear errors-in-variables models for computer vision applications. IEEE Trans. Patt. Anal. Mach. Intell. 28(10), 1537–1552 (2006)

[23] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing, 2nd edn. Cambridge University Press, Cambridge (1992)

[24] Rangarajan, P., Kanatani, K.: Improved algebraic methods for circle fitting. Electronic J. Stat. 3, 1075–1082 (2009)

[25] Taubin, G.: Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans. Patt. Anal. Mach. Intell. 13(11), 1115–1138 (1991)

[26] Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.: Bundle Adjustment—A Modern Synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.), ICCV-WS 1999. LNCS, vol. 1883, pp. 298–375. Vision Algorithms: Theory and Practice, Springer, Heidelberg (2000)