Performance evaluation of iterative geometric fitting algorithms

Kenichi Kanatani ^{*,a}, Yasuyuki Sugaya ^{b}

[a] *Department of Computer Science, Okayama University, Okayama 700-8530, Japan*
[b] *Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Aichi 441-8580, Japan.*

Abstract

The convergence performance of typical numerical schemes for geometric fitting for computer vision applications is compared. First, the problem and the associated KCR lower bound are stated. Then, three well known fitting algorithms are described: FNS, HEIV, and renormalization. To these, we add a special variant of Gauss-Newton iterations. For initialization of iterations, random choice, least squares, and Taubin's method are tested. Simulation is conducted for fundamental matrix computation and ellipse fitting, which reveals different characteristics of each method.

Keywords: Geometric fitting; Ellipse fitting; Fundamental matrix; KCR lower bound; Convergence performance

1. Introduction

We consider the following class of problems, which we call geometric fitting: we fit a parameterized geometric model (a curve, a surface, or a relationship in high dimensions) expressed as an implicit equation

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

to N data $\mathbf{x}_\alpha$, $\alpha = 1, ..., N$, typically points in an image or point correspondences over multiple images (Kanatani, 1996). We use the term "geometric" to mean that the data $\mathbf{x}_\alpha$ are elements of some "geometric" space, which makes the problem very different from traditional numerical and statistical treatments (Björck, 1996). For example,

– There are no independent variables (inputs, controls, etc.). All we have is a set of observed data.
– There exists no explicit model which explains observations in terms of deterministic mechanisms and random noise. All descriptions are implicit.
– The underlying data space is homogeneous and isotropic with no inherent coordinate system.
– The estimation process should be invariant to changes of the coordinate system with respect to which the data are described.

The function $F(\mathbf{x};\mathbf{u})$, which may be a vector, is parameterized by vector $\mathbf{u}$. Each $\mathbf{x}_\alpha$ is assumed to be perturbed by independent noise from its true value $\bar{\mathbf{x}}_\alpha$ which strictly satisfies (1). By computing the parameter $\mathbf{u}$ of the

* Corresponding author.
E-mail addresses: kanatani@suri.it.okayama-u.ac.jp (K. Kanatani), sugaya@iim.ics.tut.ac.jp (Y. Sugaya).

0617-9473/$ - see front matter ©2007 Publised by Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2007.05.013

fitted equation, one can discern the underlying geometric structure (Kanatani, 1996). In this paper, we focus on problems for which (1) reduces to a linear form by changing variables. A large class of computer vision problems fall into this category (Kanatani, 1996).

To solve this problem, various algebraic methods were proposed in the past (Bookstein, 1979; Sampson, 1982; Taubin, 1991), but Kanatani (1996) pointed out that the problem can be regarded as statistical estimation and that maximum likelihood (ML) produces an optimal solution. To compute ML, Chojnacki et al. (2000) proposed a procedure called FNS, and Leedan and Meer (2000) presented a method called HEIV. In this paper, we add a special variant of Gauss-Newton iterations. These methods attain a theoretical accuracy bound (KCR lower bound) up to high order terms in noise (Chernov and Lesort, 2004; Kanatani, 1996). Kanatani's renormalization (Kanatani, 1993, 1996) also computes a solution nearly equivalent to them (Kanatani, 2007). However, all these are iterative methods with different convergence properties, which also depend on the choice of initial values. The purpose of this paper is to experimentally compare their convergence performance.

Sections 2 and 3 state the problem and the KCR lower bound. Section 4 describes the four algorithms: FNS, HEIV, renormalization, and a new scheme based on Gauss-Newton iterations. In Sect. 5, we list three types of initialization of the iterations: random choice, least squares, and Taubin's method. Then, we show numerical examples of two typical problems: fundamental matrix computation in Sect. 6 and ellipse fitting in Sect. 7. Section 8 concludes this paper.

2. KCR lower bound, least squares, and maximum likelihood

Kanatani (1996, 2007) proved that if noise in each datum $\mathbf{x}_\alpha$ is an independent Gaussian random variable with mean $\mathbf{0}$ and covariance matrix $V[\mathbf{x}_\alpha]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\mathbf{u}}$ of $\mathbf{u}$:

$$V[\hat{\mathbf{u}}] \succ \left(\sum_{\alpha=1}^{N} \frac{(\mathcal{P}_{\mathbf{u}} \nabla_{\mathbf{u}} \bar{F}_\alpha)(\mathcal{P}_{\mathbf{u}} \nabla_{\mathbf{u}} \bar{F}_\alpha)^\top}{(\nabla_{\mathbf{x}} \bar{F}_\alpha, V[\mathbf{x}_\alpha] \nabla_{\mathbf{x}} \bar{F}_\alpha)} \right)^{-}. \tag{2}$$

Here, $\succ$ means that the left-hand side minus the right is positive semidefinite, and the superscript $-$ denotes pseudoinverse. The symbols $\nabla_{\mathbf{x}} \bar{F}_\alpha$ and $\nabla_{\mathbf{u}} \bar{F}_\alpha$ denote the gradient of $F(\mathbf{x}; \mathbf{u})$ with respect to $\mathbf{x}$ and $\mathbf{u}$, respectively, evaluated at $\mathbf{x} = \bar{\mathbf{x}}_\alpha$. The symbol $\mathcal{P}_{\mathbf{u}}$ denotes projection onto the tangent space $T_{\mathbf{u}}(\mathcal{U})$ to the domain $\mathcal{U}$ of the parameter $\mathbf{u}$, which is generally a manifold in $\mathcal{R}^n$. Throughout this paper, we denote the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ by $(\mathbf{a}, \mathbf{b})$. Chernov and Lesort (2004) called the right-hand side of (2) the KCR (Kanatani-Cramer-Rao) lower bound and showed that it holds except for $O(\varepsilon^4)$ even if $\hat{\mathbf{u}}$ is not unbiased; it is sufficient that $\hat{\mathbf{u}}$ is "consistent" in the sense that $\hat{\mathbf{u}}$ converges to the true value $\mathbf{u}$ as the noise in the data decreases.

It is a common strategy to define an estimator through minimization or maximization of some cost function, although this is not always necessary. A widely used method is what is called least-squares estimation (LS) (and by many other names such as algebraic distance minimization), minimizing

$$J = \sum_{\alpha=1}^{N} F(\mathbf{x}_\alpha; \mathbf{u})^2. \tag{3}$$

A more sophisticated method is maximum likelihood (ML) (also called by many other names). We regard the data $\mathbf{x}_1$, ..., $\mathbf{x}_N$ as perturbed from their true values $\bar{\mathbf{x}}_1$, ..., $\bar{\mathbf{x}}_N$ by noise. The domain $\mathcal{X}$ of the data is generally a manifold in $\mathcal{R}^m$. Assuming that the noise is small, we view the noise as occurring in the tangent space $T_{\bar{\mathbf{x}}_\alpha}(\mathcal{X})$ to the domain $\mathcal{X}$ at each $\bar{\mathbf{x}}_\alpha$. Within that tangent space, the noise is assumed to be independent Gaussian with mean $\mathbf{0}$ and covariance matrix $V[\mathbf{x}_\alpha]$. Then, the likelihood of observing $\mathbf{x}_1$, ..., $\mathbf{x}_N$ is

$$C \prod_{\alpha=1}^{N} e^{-(\mathbf{x}_\alpha - \bar{\mathbf{x}}_\alpha, V[\mathbf{x}_\alpha]^{-}(\mathbf{x}_\alpha - \bar{\mathbf{x}}_\alpha))/2}, \tag{4}$$

where C is a normalization constant. The true values $\bar{\mathbf{x}}_1$, ..., $\bar{\mathbf{x}}_N$ are constrained by (6). Maximizing (4) is equivalent to minimizing the negative of its logarithm, which is written up to additive and multiplicative constants in the form

$$J = \sum_{\alpha=1}^{N} (\mathbf{x}_\alpha - \bar{\mathbf{x}}_\alpha, V[\mathbf{x}_\alpha]^{-}(\mathbf{x}_\alpha - \bar{\mathbf{x}}_\alpha)), \tag{5}$$

called the (square) Mahalanobis distance. This is to be minimized subject to

$$F(\bar{\mathbf{x}}_\alpha;\mathbf{u})=0, \qquad \alpha=1,...,N. \tag{6}$$

If $\bar{\mathbf{x}}_\alpha$ is decomposed into inputs (or controls) and outputs (or observations), and if (6) can be explicitly solved for the latter in terms of the former, we can eliminate this constraint and express (5) in terms of all unknowns. The resulting problem is called the generalized least-squares problem (Björck, 1996). In this framework, we need to regard the inputs, as well as the outputs, as noisy data, and we are minimizing the (Mahalanobis) distance from the model to both inputs and outputs. This interpretation is known as total least squares with an errors-in-variables model (Huffel and Lemmerling, 2002). This type of problem covers a large class of statistical problems in practice. In the present case, however, there are no such distinctions as inputs and outputs, because the data are points in a homogeneous and isotropic geometric space with no inherent coordinate system. Furthermore, we cannot express any part of the unknowns in terms of the rest. Most problems for computer vision applications belong to this type.

This difficulty can be overcome if noise is assumed to be small, which is usually the case in computer vision applications. On this assumption, we can eliminate the constraint (6) by introducing Lagrange multipliers, doing Taylor expansion, and ignoring higher order terms in noise. Then, we obtain the following form Kanatani (1996, 2007):

$$J=\sum_{\alpha=1}^{N}\frac{F(\mathbf{x}_\alpha;\mathbf{u})^2}{(\nabla_{\mathbf{x}}F_\alpha, V[\mathbf{x}_\alpha]\nabla_{\mathbf{x}}F_\alpha)}. \tag{7}$$

It can be shown that the covariance matrix $V[\hat{\mathbf{u}}]$ of the resulting estimator $\hat{\mathbf{u}}$ achieves the KCR lower bound except for $O(\varepsilon^4)$ (Chernov and Lesort, 2004; Kanatani, 1996, 2007).

3. Linearized constraint

We concentrate on a special subclass of geometric fitting problems in which (1) reduces to the linear form

$$(\boldsymbol{\xi}(\mathbf{x}_\alpha),\mathbf{u})=0, \tag{8}$$

by changing variables. Since the data $\mathbf{x}_\alpha$ are m-dimensional vectors and the unknown parameter $\mathbf{u}$ is a p-dimensional vector, $\boldsymbol{\xi}(\,\cdot\,)$ is a (generally nonlinear) embedding from $\mathcal{R}^m$ to $\mathcal{R}^p$. In order to remove scale indeterminacy of the form of (8), we normalize $\mathbf{u}$ to $\|\mathbf{u}\|=1$. The KCR lower bound for the linearized constraint has the form

$$V_{\rm KCR}[\hat{\mathbf{u}}]=\Big(\sum_{\alpha=1}^{N}\frac{\bar{\boldsymbol{\xi}}_\alpha\bar{\boldsymbol{\xi}}_\alpha^\top}{(\mathbf{u},V[\boldsymbol{\xi}_\alpha]\mathbf{u})}\Big)^-, \tag{9}$$

where $\bar{\boldsymbol{\xi}}_\alpha$ is an abbreviation for $\boldsymbol{\xi}(\bar{\mathbf{x}}_\alpha)$. The covariance matrix $V[\boldsymbol{\xi}_\alpha]$ of $\boldsymbol{\xi}(\mathbf{x}_\alpha)$ is given, except for higher order terms in noise, in the form

$$V[\boldsymbol{\xi}_\alpha]=\nabla_{\mathbf{x}}\bar{\boldsymbol{\xi}}_\alpha^\top V[\mathbf{x}_\alpha]\nabla_{\mathbf{x}}\bar{\boldsymbol{\xi}}_\alpha, \tag{10}$$

where $\nabla_{\mathbf{x}}\bar{\boldsymbol{\xi}}_\alpha$ is the $m\times p$ Jacobian matrix

$$\nabla_{\mathbf{x}}\boldsymbol{\xi}=\begin{pmatrix}\partial\xi_1/\partial x_1 & \cdots & \partial\xi_p/\partial x_1\\ \vdots & \ddots & \vdots\\ \partial\xi_1/\partial x_m & \cdots & \partial\xi_p/\partial x_m\end{pmatrix}, \tag{11}$$

evaluated at $\mathbf{x}=\bar{\mathbf{x}}_\alpha$. Note that in (9) we do not need the projection operator for the normalization constraint $\|\mathbf{u}\|$ = 1, because $\bar{\boldsymbol{\xi}}_\alpha$ is orthogonal to $\mathbf{u}$ due to (8); for the moment, we assume that no other internal constraints exist.

This subclass of geometric fitting problems covers a wide range of computer vision applications. The following are typical examples:

Example 1. Fundamental matrix computation

Suppose we have N corresponding points in two images of the same scene viewed from different positions. If point (x_α, y_α) in the first image corresponds to (x'_α, y'_α) in the second, the following epipolar equation holds (Hartley and Zisserman, 2000):

$$(\begin{pmatrix} x_\alpha \\ y_\alpha \\ 1 \end{pmatrix}, \mathbf{F}\begin{pmatrix} x'_\alpha \\ y'_\alpha \\ 1 \end{pmatrix}) = 0. \tag{12}$$

Here, $\mathbf{F}$ is a matrix, called the fundamental matrix, that depends only on the relative positions and orientations of the two cameras and their intrinsic parameters (e.g., their focal lengths) but not on the scene or the choice of the corresponding points. If we define

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

the constraint (12) is linearized in the form of (8). If independent Gaussian noise of mean 0 and standard deviation σ is added to each coordinates of (x_α, y_α) and (x'_α, y'_α), the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ has the form

$$V[\boldsymbol{\xi}_\alpha] = \sigma^2 \begin{pmatrix} \bar{x}_\alpha^2 + \bar{x}_\alpha'^2 & \bar{x}'_\alpha \bar{y}'_\alpha & \bar{x}'_\alpha & \bar{x}_\alpha \bar{y}_\alpha & 0 & 0 & \bar{x}_\alpha & 0 & 0 \\ \bar{x}'_\alpha \bar{y}'_\alpha & \bar{x}_\alpha^2 + \bar{y}_\alpha'^2 & \bar{y}'_\alpha & 0 & \bar{x}_\alpha \bar{y}_\alpha & 0 & 0 & \bar{x}_\alpha & 0 \\ \bar{x}'_\alpha & \bar{y}'_\alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{x}_\alpha \bar{y}_\alpha & 0 & 0 & \bar{y}_\alpha^2 + \bar{x}_\alpha'^2 & \bar{x}'_\alpha \bar{y}'_\alpha & \bar{x}'_\alpha & \bar{y}_\alpha & 0 & 0 \\ 0 & \bar{x}_\alpha \bar{y}_\alpha & 0 & \bar{x}'_\alpha \bar{y}'_\alpha & \bar{y}_\alpha^2 + \bar{y}_\alpha'^2 & \bar{y}'_\alpha & 0 & \bar{y}_\alpha & 0 \\ 0 & 0 & 0 & \bar{x}'_\alpha & \bar{y}'_\alpha & 1 & 0 & 0 & 0 \\ \bar{x}_\alpha & 0 & 0 & \bar{y}_\alpha & 0 & 0 & 1 & 0 & 0 \\ 0 & \bar{x}_\alpha & 0 & 0 & \bar{y}_\alpha & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{14}$$

except for $O(\sigma^4)$, where $(\bar{x}_\alpha, \bar{y}_\alpha)$ and $(\bar{x}'_\alpha, \bar{y}'_\alpha)$, are the true positions of (x_α, y_α) and (x'_α, y'_α), respectively. The fundamental matrix $\mathbf{F}$ should also satisfy the constraint that $\det \mathbf{F} = 0$ (Hartley and Zisserman, 2000). If this constraint is taken into account, the KCR lower bound involves the corresponding projection operation (Kanatani, 1996; Kanatani and Ohta, 2003).

Example 2. Conic fitting

Suppose we want to fit a quadratic curve (circle, ellipse, parabola, hyperbola, or their degeneracy), or a conic, to N points (x_α, y_α), $\alpha = 1, ..., N$, in the plane. The constraint has the form

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2(Dx_\alpha + Ey_\alpha) + F = 0. \tag{15}$$

If we define

$$\boldsymbol{\xi}(x, y) = \begin{pmatrix} x^2 & 2xy & y^2 & 2x & 2y & 1 \end{pmatrix}^\top, \qquad \mathbf{u} = \begin{pmatrix} A & B & C & D & E & F \end{pmatrix}^\top, \tag{16}$$

the constraint (15) is linearized in the form of (8). If independent Gaussian noise of mean 0 and standard deviation σ is added to each coordinates of (x_α, y_α), the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ has the form

$$V[\boldsymbol{\xi}_\alpha] = 4\sigma^2 \begin{pmatrix} \bar{x}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & 0 & \bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & \bar{x}_\alpha^2 + \bar{y}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & \bar{y}_\alpha & \bar{x}_\alpha & 0 \\ 0 & \bar{x}_\alpha\bar{y}_\alpha & \bar{y}_\alpha^2 & 0 & \bar{y}_\alpha & 0 \\ \bar{x}_\alpha & \bar{y}_\alpha & 0 & 1 & 0 & 0 \\ 0 & \bar{x}_\alpha & \bar{y}_\alpha & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{17}$$

except for $O(\sigma^4)$, where $(\bar{x}_\alpha, \bar{y}_\alpha)$ is the true position of (x_α, y_α).

In traditional domains of statistics where noise is large with a small number of point samples, the standard approach is to express the conic in an explicit parametric form and minimize the sum of the square distances of the sample points to the conic to be fitted by doing nonlinear minimization; see, e.g., Gander et al. (1994). In image processing and computer vision applications, on the other hand, noise is small (often in the subpixel level) but a vast number of data are processed. For this reason, the implicit form of (15) as is has been used (Bookstein, 1979; Sampson, 1982; Taubin, 1991), but the statistical characteristics of noise has not been fully incorporated. The following formulation provides a statistically optimal fitting algorithms based on the implicit form of (15).

4. Numerical computation of ML

As we can see from (17) and (14), the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ of $\boldsymbol{\xi}_\alpha$ in many practical problems is factored into the form

$$V[\boldsymbol{\xi}_\alpha] = \varepsilon^2 V_0[\boldsymbol{\xi}_\alpha], \tag{18}$$

where ε is a constant that characterizes the noise and $V_0[\boldsymbol{\xi}_\alpha]$ is a matrix that depends only on the true data values. Hereafter, we call ε the noise level and $V_0[\boldsymbol{\xi}_\alpha]$ the normalized covariance matrix.

For the constraint in the form of (8), the function (7) reduces to

$$J = \frac{1}{2}\sum_{\alpha=1}^{N} \frac{(\mathbf{u}, \boldsymbol{\xi}_\alpha)^2}{(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})}. \tag{19}$$

The covariance matrix $V[\boldsymbol{\xi}_\alpha]$ can be replaced by $V_0[\boldsymbol{\xi}_\alpha]$, because multiplication of J by a positive constant does not affect its minimization. The ML solution is obtained by solving $\nabla_{\mathbf{u}} J = \mathbf{0}$. From (19), we have

$$\nabla_{\mathbf{u}} J = \sum_{\alpha=1}^{N} \frac{(\mathbf{u}, \boldsymbol{\xi}_\alpha)\boldsymbol{\xi}_\alpha}{(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})} - \sum_{\alpha=1}^{N} \frac{(\mathbf{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]\mathbf{u}}{(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})^2}, \tag{20}$$

where we define

$$\mathbf{M} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})}, \qquad \mathbf{L} = \sum_{\alpha=1}^{N} \frac{(\mathbf{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]}{(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})^2}. \tag{21}$$

Note that J in (19) is a homogeneous function of degree 0 in $\mathbf{u}$, i.e., multiplication of $\mathbf{u}$ by any nonzero number does not change the value of J. Hence, the convention $\|\mathbf{u}\| = 1$, or any other scale normalization, does not affect the solution that minimizes (19). If we were to minimize $\sum_{\alpha=1}^{N}(\mathbf{u}, \boldsymbol{\xi}_\alpha)^2$, for instance, the solution would depend on which scale normalization is used, e.g., the constraint $\|\mathbf{u}\| = 1$ and the constraint $u_1 = 1$ would result in different solutions (Gander et al., 1994). To mend this, Bookstein (1979) introduced a scale constraint that is invariant to Euclidean coordinate transformations so that the particular solution arising from that constraint would consistently be obtained after coordinate changes.

The fact that any scale constraint insures the invariance to coordinate changes in our formulation is a natural consequence of the definition of J in (19). Since it is based on the likelihood, which has the same value whichever coordinate system is used, it is invariant to similarity transformations (rotations, translations, and scale changes) of the coordinate system. In fact, if we switch to a new coordinate system with respect to which $\boldsymbol{\xi}_\alpha$ is expressed as $\tilde{\boldsymbol{\xi}}_\alpha$, then $\mathbf{u}$ is transformed into $\tilde{\mathbf{u}}$ in such a way that $(\tilde{\mathbf{u}}, \tilde{\boldsymbol{\xi}}_\alpha) = C(\mathbf{u}, \boldsymbol{\xi}_\alpha)$ for some nonzero constant C, and the covariance matrix $V_0[\boldsymbol{\xi}_\alpha]$ is transformed to $V_0[\tilde{\boldsymbol{\xi}}_\alpha]$ in such a way that $(\tilde{\mathbf{u}}, V_0[\tilde{\boldsymbol{\xi}}_\alpha]\tilde{\mathbf{u}}) = C(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})$. As a result, the ML solution is invariant to similarity. This invariance would be lost if we adopted a pure algebraic formulation, such as Bookstein (1979) and Gander et al. (1994), without introducing the covariance matrix $V_0[\boldsymbol{\xi}_\alpha]$.

4.1. *Fundamental numerical scheme (FNS)*

The procedure called FNS (fundamental numerical scheme) of Chojnacki et al. (2000) for solving (20) is described as follows:

(i) Initialize $\mathbf{u}$.
(ii) Compute the matrices $\mathbf{M}$ and $\mathbf{L}$ in (21).
(iii) Solve the eigenvalue problem

$$(\mathbf{M} - \mathbf{L})\mathbf{u}' = \lambda \mathbf{u}', \tag{22}$$

and compute the unit eigenvector $\mathbf{u}'$ for the eigenvalue λ closest to 0.
(iv) If $\mathbf{u}' \approx \mathbf{u}$ except for sign, return $\mathbf{u}'$ and stop. Else, let $\mathbf{u} \leftarrow \mathbf{u}'$ and go back to Step 2.

Later, Chojnacki et al. (2005) pointed out that convergence performance improves if we choose in Step 3 not the eigenvalue closest to 0 but the smallest one. We call the above procedure the original FNS and the one using the smallest eigenvalue the modified FNS. Whichever eigenvalue is chosen for λ, we have $\lambda = 0$ after convergence. In fact, convergence means

$$(\mathbf{M} - \mathbf{L})\mathbf{u} = \lambda \mathbf{u} \tag{23}$$

for some $\mathbf{u}$. Computing the inner product with $\mathbf{u}$ on both sides, we have

$$(\mathbf{u}, \mathbf{M}\mathbf{u}) - (\mathbf{u}, \mathbf{L}\mathbf{u}) = \lambda. \tag{24}$$

On the other hand, the definition (21) implies that $(\mathbf{u}, \mathbf{M}\mathbf{u}) = (\mathbf{u}, \mathbf{L}\mathbf{u})$ identically in $\mathbf{u}$, meaning $\lambda = 0$.

4.2. *Heteroscedastic errors-in-variables (HEIV)*

The relation (20) can be rewritten as

$$\mathbf{M}\mathbf{u} = \mathbf{L}\mathbf{u}. \tag{25}$$

The HEIV (heteroscedastic errors-in-variables) method of Leedan and Meer (2000) is to solve the generalized eigenvalue problem $\mathbf{M}\mathbf{u} = \lambda \mathbf{L}\mathbf{u}$ iteratively. In many applications, the matrix $\mathbf{L}$ is not positive definite, so we cannot directly solve this generalized eigenvalue problem. For a wide range of problems, however, the vectors $\boldsymbol{\xi}_\alpha$ and $\mathbf{u}$ and the normalized covariance matrix $V_0[\boldsymbol{\xi}_\alpha]$ have the following form:

$$\boldsymbol{\xi}_\alpha = \begin{pmatrix} \mathbf{z}_\alpha \\ C \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} \mathbf{v} \\ a \end{pmatrix}, \qquad V_0[\boldsymbol{\xi}_\alpha] = \begin{pmatrix} V_0[\mathbf{z}_\alpha] & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix}. \tag{26}$$

For example, $C = 1$ and $a = F_{33}$ for fundamental matrix computation (see (13) and (14)), and $C = 1$ and $a = F$ for conic fitting (see (16) and (17)).

Let us define $(p-1) \times (p-1)$ matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{L}}$ by

$$\tilde{\mathbf{M}} = \sum_{\alpha=1}^{N} \frac{\tilde{\mathbf{z}}_\alpha \tilde{\mathbf{z}}_\alpha^\top}{(\mathbf{v}, V_0[\mathbf{z}_\alpha]\mathbf{v})}, \qquad \tilde{\mathbf{L}} = \sum_{\alpha=1}^{N} \frac{(\mathbf{v}, \tilde{\mathbf{z}}_\alpha)^2 V_0[\mathbf{z}_\alpha]}{(\mathbf{v}, V_0[\mathbf{z}_\alpha]\mathbf{v})^2}, \tag{27}$$

where we put

$$\tilde{\mathbf{z}}_\alpha = \mathbf{z}_\alpha - \bar{\mathbf{z}}, \qquad \bar{\mathbf{z}} = \sum_{\alpha=1}^{N} \frac{\mathbf{z}_\alpha}{(\mathbf{v}, V_0[\mathbf{z}_\alpha]\mathbf{v})} \Bigg/ \sum_{\beta=1}^{N} \frac{1}{(\mathbf{v}, V_0[\mathbf{z}_\beta]\mathbf{v})}\ . \tag{28}$$

Then, the equation (25) splits into two (Chojnacki et al., 2005; Leedan and Meer, 2000):

$$\tilde{\mathbf{M}}\mathbf{v} = \tilde{\mathbf{L}}\mathbf{v}, \qquad (\mathbf{v}, \bar{\mathbf{z}}) + Ca = 0. \tag{29}$$

If we compute a $(p-1)$-dimensional unit vector $\mathbf{v}$ that satisfies the first equation, the second gives a. Hence, we obtain

$$\mathbf{u} = N[\begin{pmatrix} \mathbf{v} \\ a \end{pmatrix}], \tag{30}$$

where $N[\,\cdot\,]$ denotes normalization to unit norm. The vector $\mathbf{u}$ that satisfies the first equation in (29) is computed by the following iterations (Chojnacki et al., 2005; Leedan and Meer, 2000):

(i) Initialize $\mathbf{v}$.

(ii) Compute the matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{L}}$ in (27).

(iii) Solve the generalized eigenvalue problem

$$\tilde{\mathbf{M}}\mathbf{v}' = \lambda\tilde{\mathbf{L}}\mathbf{v}', \tag{31}$$

and compute the unit generalized eigenvector $\mathbf{v}'$ for the generalized eigenvalue λ closest to 1.

(iv) If $\mathbf{v}' \approx \mathbf{v}$ except for sign, return $\mathbf{v}'$ and stop. Else, let $\mathbf{v} \leftarrow \mathbf{v}'$ and go back to Step 2.

However, Leedan and Meer (2000) pointed out that choosing in Step 3 not the generalized eigenvalue closest to 1 but the smallest one improves the convergence performance. Here, we call the above procedure the original HEIV and the one using the smallest generalized eigenvalue the modified HEIV. Whichever generalized eigenvalue is chosen for λ, we have $\lambda = 1$ after convergence. In fact, convergence means

$$\tilde{\mathbf{M}}\mathbf{v} = \lambda\tilde{\mathbf{L}}\mathbf{v} \tag{32}$$

for some $\mathbf{v}$. Computing the inner product with $\mathbf{v}$ on both sides, we have

$$(\mathbf{v}, \tilde{\mathbf{M}}\mathbf{v}) = \lambda(\mathbf{v}, \tilde{\mathbf{L}}\mathbf{v}). \tag{33}$$

On the other hand, the definition (27) implies that $(\mathbf{v}, \tilde{\mathbf{M}}\mathbf{v}) = (\mathbf{v}, \tilde{\mathbf{L}}\mathbf{v})$ identically in $\mathbf{v}$, meaning $\lambda = 1$.

4.3. *Renormalization*

Kanatani's renormalization (Kanatani, 1993, 1996) is to approximate the matrix $\mathbf{L}$ in (21) in the form

$$\mathbf{L} \approx c\mathbf{N}, \qquad \mathbf{N} = \sum_{\alpha=1}^{N} \frac{V_0[\boldsymbol{\xi}_\alpha]}{(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})}. \tag{34}$$

The constant c is determined so that $\mathbf{M} - c\mathbf{N}$ has eigenvalue 0. This is done by the following iterations (Kanatani, 1996):

(i) Initialize $\mathbf{u}$ and let $c = 0$.

(ii) Compute the matrix $\mathbf{M}$ in (21) and the matrix $\mathbf{N}$ in (34).

(iii) Solve the eigenvalue problem

$$(\mathbf{M} - c\mathbf{N})\mathbf{u}' = \lambda\mathbf{u}', \tag{35}$$

and compute the unit eigenvector $\mathbf{u}'$ for the eigenvalue λ closest to 0.

(iv) If $\lambda \approx 0$, return $\mathbf{u}'$ and stop. Else, let

$$c \leftarrow c + \frac{\lambda}{(\mathbf{u}', \mathbf{N}\mathbf{u}')}, \qquad \mathbf{u} \leftarrow \mathbf{u}' \tag{36}$$

and go back to Step 2.

4.4. *Projective Gauss-Newton iterations*

Since the gradient $\nabla_{\mathbf{u}} J$ is given by (20), we can minimize the function (19) by Newton iterations. If we evaluate the Hessian $\nabla^2_{\mathbf{u}} J$, the increment $\Delta\mathbf{u}$ in $\mathbf{u}$ is determined by solving

$$(\nabla^2_{\mathbf{u}} J)\Delta\mathbf{u} = -\nabla_{\mathbf{u}} J. \tag{37}$$

Since $\nabla^2_{\mathbf{u}} J$ is singular (recall that J is constant in the direction of $\mathbf{u}$), the solution is indeterminate. However, if we use pseudoinverse and compute

$$\Delta\mathbf{u} = -(\nabla^2_{\mathbf{u}} J)^{-}\nabla_{\mathbf{u}} J, \tag{38}$$

we obtain a solution orthogonal to $\mathbf{u}$. Differentiating (20) and introducing Gauss-Newton approximation (i.e., ignoring terms that contain $(\mathbf{u}, \boldsymbol{\xi}_\alpha)$), we see that the Hessian is nothing but the matrix $\mathbf{M}$ in (21). We enforce $\mathbf{M}$ to have eigenvalue 0 for $\mathbf{u}$, using the projection matrix $\mathbf{P}_{\mathbf{u}}$. The iteration procedure goes as follows:

(i) Initialize $\mathbf{u}$.

(ii) Compute the matrices $\mathbf{M}$ and $\mathbf{L}$ in (21), and let

$$\mathbf{u}' = N[\mathbf{u} - (\mathbf{P}_{\mathbf{u}}\mathbf{M}\mathbf{P}_{\mathbf{u}})^{-}(\mathbf{M} - \mathbf{L})\mathbf{u}], \qquad \mathbf{P}_{\mathbf{u}} = \mathbf{I} - \mathbf{u}\mathbf{u}^\top. \tag{39}$$

($\mathbf{I}$ is the unit matrix.)

(iii) If $\mathbf{u}' \approx \mathbf{u}$, return $\mathbf{u}'$ and stop. Else, let $\mathbf{u} \leftarrow \mathbf{u}'$ and go back to Step 2.

5. Initialization

For initialization of the iterations, we test the following three:

5.1. *Random choice*

We generate nine independent Gaussian random numbers of mean 0 and standard deviation 1 and normalize the vector consisting of them into unit norm.

5.2. *Least squares (LS)*

Approximating the denominators in (19) by a constant, we minimize

$$J_{\mathrm{LS}} = \frac{1}{2}\sum_{\alpha=1}^{N}(\mathbf{u}, \boldsymbol{\xi}_\alpha)^2 = \frac{1}{2}(\mathbf{u}, \mathbf{M}_{\mathrm{LS}}\mathbf{u}), \qquad \mathbf{M}_{\mathrm{LS}} = \sum_{\alpha=1}^{N}\boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top. \tag{40}$$

The function (40) is minimized by the unit eigenvector $\mathbf{u}$ of $\mathbf{M}_{\mathrm{LS}}$ for the smallest eigenvalue. This can be efficiently computed using singular value decomposition (SVD): $\mathbf{u}$ is the left singular vector for the smallest singular value of the $p \times N$ matrix having $\boldsymbol{\xi}_1$, ..., $\boldsymbol{\xi}_N$ as columns.

5.3. *Taubin's method*

Replacing the denominators in (19) by their average, we minimize the following function:

$$J_{\mathrm{TB}} = \frac{1}{2}\frac{\sum_{\alpha=1}^{N}(\mathbf{u}, \boldsymbol{\xi}_\alpha)^2}{\sum_{\alpha=1}^{N}(\mathbf{u}, V_0[\boldsymbol{\xi}_\alpha]\mathbf{u})} = \frac{1}{2}\frac{(\mathbf{u}, \mathbf{M}_{\mathrm{LS}}\mathbf{u})}{(\mathbf{u}, \mathbf{N}_{\mathrm{TB}}\mathbf{u})}, \qquad \mathbf{N}_{\mathrm{TB}} = \sum_{\alpha=1}^{N} V_0[\boldsymbol{\xi}_\alpha]. \tag{41}$$

This is a modification of the method by Taubin (1991) (he did not take the covariance matrix into account). The function (41) is minimized by solving the generalized eigenvalue problem

$$\mathbf{M}_{\mathrm{LS}}\mathbf{u} = \lambda \mathbf{N}_{\mathrm{TB}}\mathbf{u}, \tag{42}$$

for the smallest generalized eigenvalue. However, the matrix $\mathbf{N}_{\mathrm{TB}}$ is often not positive definite, so we decompose $\boldsymbol{\xi}_\alpha$, $\mathbf{u}$, and $V_0[\boldsymbol{\xi}_\alpha]$ in the form of (26) and define $(p-1)\times(p-1)$ matrices $\tilde{\mathbf{M}}_{\mathrm{LS}}$ and $\tilde{\mathbf{N}}_{\mathrm{TB}}$ by

$$\tilde{\mathbf{M}}_{\mathrm{LS}} = \sum_{\alpha=1}^{N} \tilde{\mathbf{z}}_\alpha \tilde{\mathbf{z}}_\alpha^\top, \qquad \tilde{\mathbf{N}}_{\mathrm{TB}} = \sum_{\alpha=1}^{N} V_0[\mathbf{z}_\alpha], \tag{43}$$

where

$$\tilde{\mathbf{z}}_\alpha = \mathbf{z}_\alpha - \bar{\mathbf{z}}, \qquad \bar{\mathbf{z}} = \frac{1}{N}\sum_{\alpha=1}^{N} \mathbf{z}_\alpha. \tag{44}$$

Then, the equation (42) splits into two:

$$\tilde{\mathbf{M}}_{\mathrm{LS}}\mathbf{v} = \lambda \tilde{\mathbf{N}}_{\mathrm{TB}}\mathbf{v}, \qquad (\mathbf{v}, \bar{\mathbf{z}}) + Ca = 0. \tag{45}$$

If we compute the unit generalized eigenvector $\mathbf{v}$ of the first equation for the smallest generalized eigenvalue λ, the second gives a. Hence, $\mathbf{u}$ is given by (30).

6. Fundamental matrix computation

Figure 1(a) shows two simulated images of two planar grid planes joined at angle 60°. The image size is 600×600 (pixels), and the focal length is 1200 (pixels). We added random Gaussian noise of mean 0 and standard deviation σ (pixels) to the image coordinates of each grid point independently and estimated the fundamental matrix by FNS, HEIV, renormalization, and projective Gauss-Newton iterations.

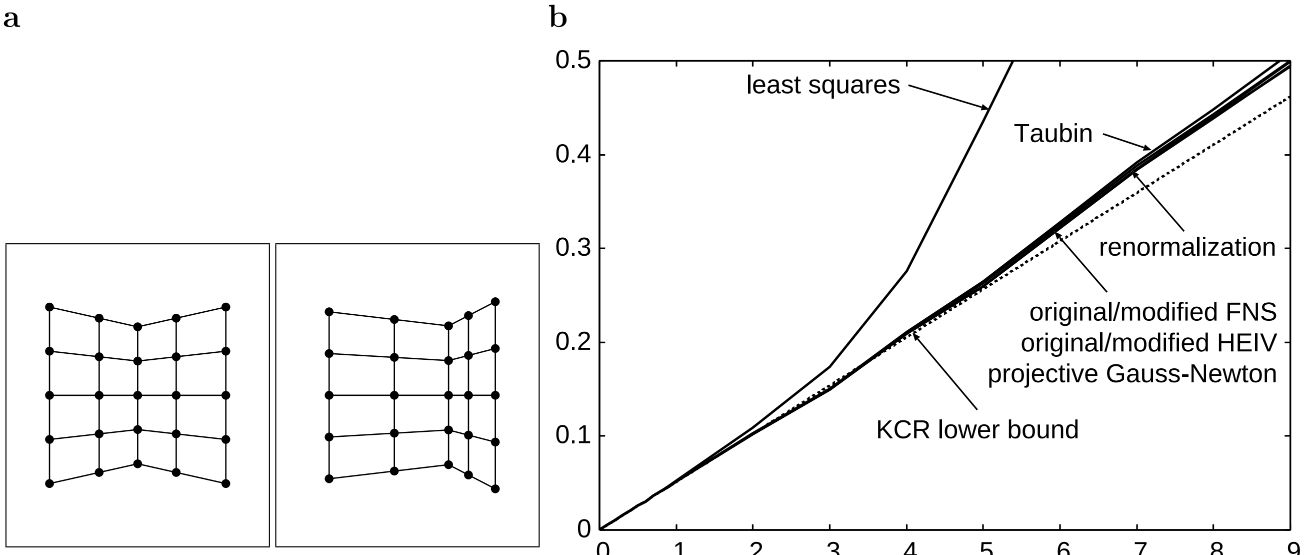


Fig. 1. (a) Simulated images of planar grid surfaces. (b) RMS error vs. noise level.

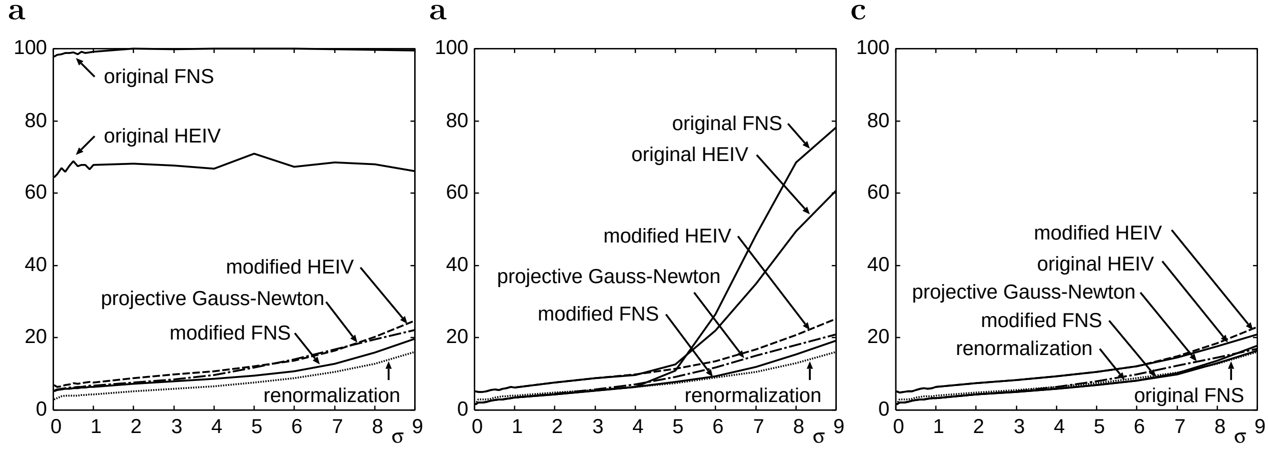


Fig. 2. Average number of iterations vs. noise level. (a) Random initialization. (b) LS initialization. (c) Taubin initialization.

The fundamental matrix $\mathbf{F}$ should satisfy the constraint $\det \mathbf{F} = 0$ (Hartley and Zisserman, 2000), and Chojnacki et al. (2004) presented a FNS-like procedure to incorporate this constraint. However, once the solution $\hat{\mathbf{u}}$ of (20) is obtained, it can be easily corrected so as to satisfy $\det \mathbf{F} = 0$ in such a way that the accuracy is equivalent to the constrained minimization of (19) subject to $\det \mathbf{F} = 0$ except for higher order terms in σ (see Appendix A for the procedure) (Kanatani, 1996; Kanatani and Ohta, 2003). So, we consider here the computation prior to this correction.

Figure 1(b) plots for each σ the RMS (root-mean-squares) of $\|\mathbf{P}_{\mathbf{u}}\hat{\mathbf{u}}\|$ over 1000 independent trials. We compared LS, Taubin's method, and the four iterative methods starting from the Taubin solution and confirmed that for each method the final solution does not depend on the initial value as long as the iterations converge. The dotted line indicates the KCR lower bound implied by (9). We can see that Taubin's method is considerably better than LS. The four iterative methods indeed improve the Taubin solution, but the improvement is rather small. All the solutions nearly agree with the KCR lower bound when noise is small and gradually deviate from it as noise increases. Since FNS, HEIV, and projective Gauss-Newton minimize the same function, the resulting solution is virtually the same. The renormalization solution is nearly equivalent to them.

Figure 2 shows the average number of iterations of each method for 1000 trials. We stopped when the increment in $\mathbf{u}$ was less than 10^{-6} in norm (the sign of the eigenvector was chosen so that the orientation aligns with the previous solution). Figure 2(a) is for random initialization. The original FNS did not converge for about 99% of the trials after 100 iterations; the original HEIV not for about 40%. We stopped after 100 iterations and set the iteration count to 100. We can see from Fig. 2(a) that the modified FNS/HEIV converge much more quickly than the original FNS/HEIV. This can be explained as follows. If the computed $\mathbf{u}'$ is close to the true value $\mathbf{u}$, the matrix $\mathbf{L}$ in (21) and the matrix $\tilde{\mathbf{L}}$ in (27) are both close to $\mathbf{O}$. Initially, however, they may be very different from $\mathbf{O}$ when the initial value is randomly chosen. The relations (22) and (31) are written, respectively, as

$$(\mathbf{M} - \mathbf{L} - \lambda \mathbf{I})\mathbf{u}' = \mathbf{0}, \qquad (\tilde{\mathbf{M}} - \lambda \tilde{\mathbf{L}})\mathbf{v}' = \mathbf{0}. \tag{46}$$

Note that $\mathbf{L}$ and $\tilde{\mathbf{L}}$ are both positive definite. In order to cancel their effects, we need to choose λ to be negative in the first equation and smaller than 1 in the second.

As predicted from this explanation, the difference between the original FNS/HEIV and the modified FNS/HEIV shrinks as we use better initial values, as seen from Fig. 2(b), (c). We also see that the (original or modified) FNS is more efficient than (original or modified) HEIV. Another finding is that, for random initialization, renormalization is the most efficient. This is because we start solving (35) with $c = 0$, canceling the effect of $\mathbf{N}$ whatever it is, and the resulting $\mathbf{u}'$ is close to the LS solution. In contrast, FNS and HEIV may produce a solution very different from the true value when initially the matrices $\mathbf{L}$ and $\tilde{\mathbf{L}}$ are very different from $\mathbf{O}$.

As Fig. 2(b), (c) shows, however, the convergence performance of FNS and HEIV improves as we use better initial values. Naturally, projective Gauss-Newton iterations converge faster when started from better initial values. In contrast, renormalization behaves almost independently of initialization, confirming the above explanation. Overall, Taubin-initialized (original or modified) FNS shows the best convergence performance. Here, we focus only on the number of iterations, because the complexity of one iteration is nearly the same for all methods: we solve the eigenvalue problem of a 9×9 matrix or the generalized eigenvalue problem of an 8×8 matrix or compute the pseudoinverse of a 9×9 matrix. These computations take, for example, $1 \sim 3$ms, $2 \sim 3$ms, and $1 \sim 3$, respectively, for 100 corresponding pairs on Pentium 4 3.4GHz.

7. Ellipse fitting

7.1. *Fitting to a short arc*

Figure 3(a) shows 20 equidistant points $(\bar{x}_\alpha, \bar{y}_\alpha)$ on an ellipse. We added Gaussian noise of mean 0 and standard deviation σ to the x and y coordinates of each point independently and fitted an ellipse by different methods. The equation (15) does not necessarily describe an ellipse. Even if the points are sampled from an ellipse, the fitted equation may define a hyperbola or other curves in the presence of large noise, and Fitzgibbon et al. (1999) presented a technique for preventing this. Here, however, we do not impose any constraints to prevent non-ellipses, assuming that noise is sufficiently small. Recently, Kanatani (2006) showed that a "hyperaccurate" ellipse fitting method exists. However, this is a correction to the ML solution, so the ML solution must be obtained first. Here, we focus on the efficiency of ML computation.

Figure 3(b) plots for each σ the RMS of $\|\mathbf{P}_\mathbf{u}\hat{\mathbf{u}}\|$ computed over 1000 independent trials starting the Taubin solution. As in the case of fundamental matrix computation, the final solution does not depend on the initial value as long as the iterations converge, and the solutions of FNS, FNS, HEIV, and projective Gauss-Newton are virtually the same. Renormalization also produces solutions very close to them, and their accuracy is close to the KCR lower bound (dotted line).

For each σ, we computed the average number of iterations over 1000 independent trials (Fig. 4). We stopped when the increment in $\mathbf{u}$ is less than 10^{-6} in norm as before. As in the case of fundamental matrix computation, the modified FNS/HEIV always converge faster than the original FNS/HEIV. This is most apparent for random initialization, for which the original FNS/HEIV did not converge for 16% and 49%, respectively, of the trials. We can also see that the difference between the original FNS/HEIV and the modified FNS/HEIV shrinks as we use better initial values. The behavior of renormalization, on the other hand, is almost unchanged, as before. Overall, the most efficient method is the modified HEIV for whichever initialization. However, there is no difference between (original or modified) FNS/HEIV if initialized by Taubin's method.

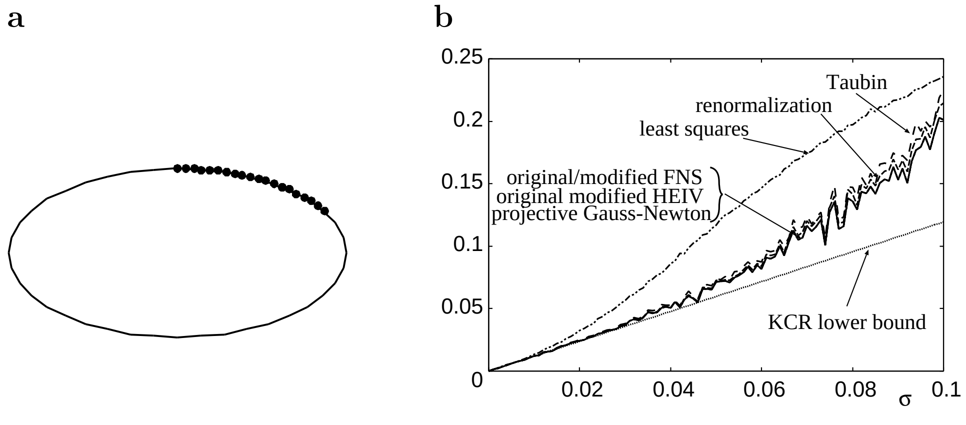


Fig. 3. (a) Twenty points on elliptic arcs on a short arc. (b) RMS error of ellipse fitting vs. noise level.

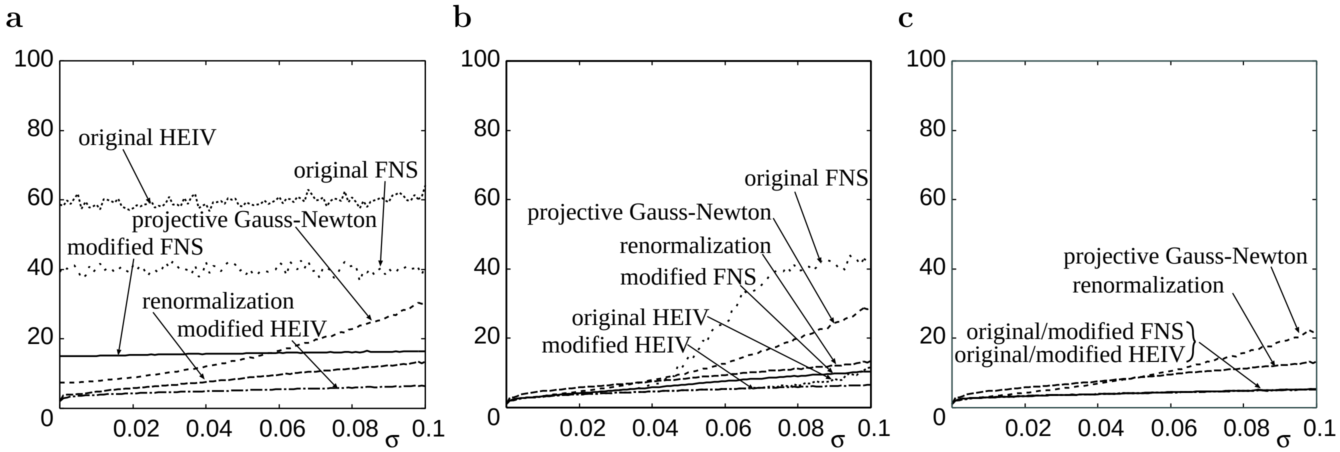


Fig. 4. Average number of iterations for ellipse fitting vs. noise level. (a) Random initialization. (b) LS initialization. (c) Taubin initialization.

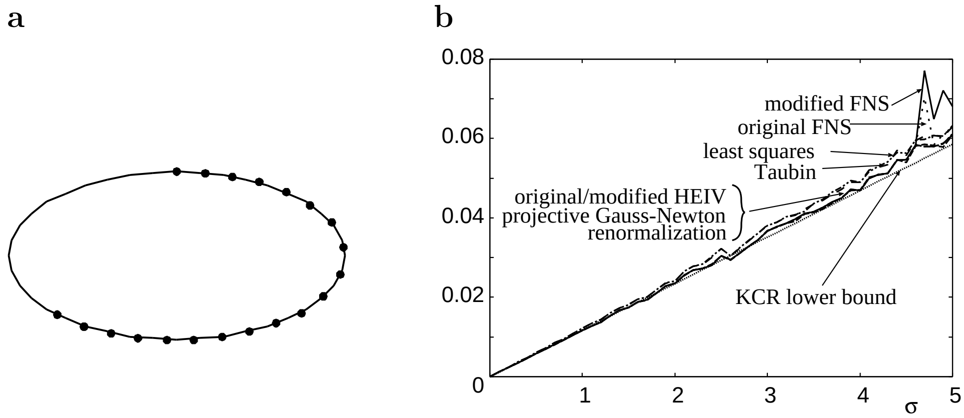
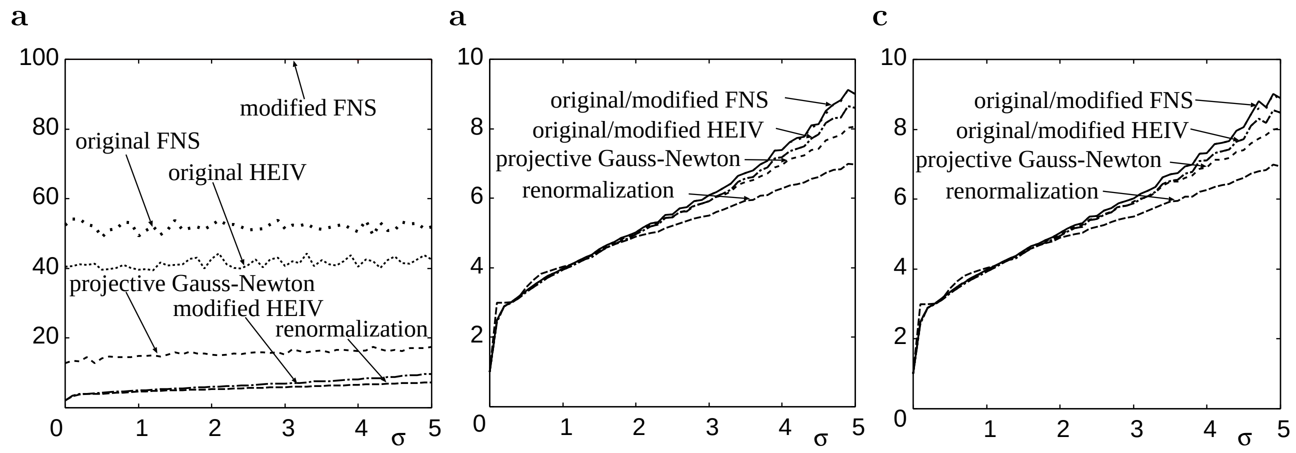


Fig. 5. (a) Twenty points on elliptic arcs on a long arc. (b) RMS error of ellipse fitting vs. noise level.

7.2. *Fitting to a long arc*

The comparative behavior of each method for points distributed over a short elliptic arc in Fig. 3(a) is more or less similar to the case of fundamental matrix computation. However, the behavior is very different when a long elliptic arc shown in Fig. 5(a) is used. Figure 5(b) shows the RMS error corresponding to Fig. 3(b). Now, the LS solution, which is prone to statistical bias, is as accurate as Taubin's method. This is because bias is less likely to arise for a long arc (no bias would arise for the entire ellipse due to the symmetry). All solutions have the accuracy close to the KCR lower bound.

Figure 6 shows the number of iterations corresponding to Fig. 4. This time, all methods converged within 10 iterations when initialized by LS or Taubin's method, so the vertical axis is restricted over that range. The most unexpected is the fact that the modified FNS is worse than the original FNS. For random initialization, the modified FNS did not converge after 100 iterations for all 1000 trials, while the original FNS did not converge for 24% of the trials. This is related to the singularity of ellipse fitting (Chernov, 2007) (see Appendix B): Some of the terms on the right-hand side of (19) diverge to $\pm\infty$. This happens when a data point exists near the center of the current candidate fit, which is more likely to occur when the data points are distributed over a long arc. As can be seen from Fig. 6, renormalization is the most stable for whichever initialization. As we noted earlier, this is because the iterations start from $c = 0$. Projective Gauss-Newton iterations are also stable.



(a) (b) (c)

Fig. 6. Average number of iterations for ellipse fitting to the points in Fig. 1(b) vs. noise level. (a) Random initialization. (b) LS initialization. (c) Taubin initialization.

The above observations are based on the number of iterations, but as in the case of fundamental matrix computation the complexity of each iteration is nearly the same for all methods. For 148 points (short arc) extracted from a real image, for example, one iteration takes 1 ∼ 2ms for all methods, and for 414 points (long arc), it takes 3 ∼ 5ms on Pentium 4 3.4GHz.

8. Conclusions

We have compared the convergence performance of different numerical schemes for geometric fitting. First, we stated the problem and the associated KCR lower bound. Then, we described the algorithms of FNS, HEIV, and renormalization, to which we added projective Gauss-Newton iterations. For initial values, we tested random choice, LS, and Taubin's method. Numerical experiments of fundamental matrix computation and ellipse fitting revealed different characteristics of each method. Overall, FNS exhibited the best convergence performance if initialized by Taubin's method.

Acknowledgments

The author thanks Nikolai Chernov of the University of Alabama, U.S.A., Wojciech Chojnacki of the University of Adelaide, Australia, and Peter Meer of the University of Rutgers, U.S.A., for helpful discussions.

Appendix A. Rank constraint optimization

The computed fundamental matrix $\mathbf{F}$ can be optimally corrected so as to satisfy $\det \mathbf{F} = 0$ as follows (Kanatani, 1996; Kanatani and Ohta, 2003). Let $\hat{\mathbf{u}}$ be the 9-dimensional vector representation of the ML estimate $\hat{\mathbf{F}}$ of the fundamental matrix $\mathbf{F}$ computed without the constraint $\det \mathbf{F} = 0$. Compute

$$\tilde{\mathbf{M}} = \sum_{\alpha=1}^{N} \frac{(\mathbf{P}_{\hat{\mathbf{u}}}\boldsymbol{\xi}_\alpha)(\mathbf{P}_{\hat{\mathbf{u}}}\boldsymbol{\xi}_\alpha)^\top}{(\hat{\mathbf{u}}, V_0[\boldsymbol{\xi}_\alpha]\hat{\mathbf{u}})}, \tag{A.1}$$

where $\mathbf{P}_{\hat{\mathbf{u}}}$ is the projection matrix defined in (39). Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_9$ $(= 0)$ be the eigenvalues of $\tilde{\mathbf{M}}$, and $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_9$ $(= \hat{\mathbf{u}})$ the corresponding orthonormal system of eigenvectors. The covariance matrix $V[\hat{\mathbf{u}}]$ of $\hat{\mathbf{u}}$ is estimated to be $V_0[\hat{\mathbf{u}}]$ up to a positive multiplier in the following form (Kanatani, 1996):

$$V_0[\hat{\mathbf{u}}] = \frac{\mathbf{u}_1\mathbf{u}_1^\top}{\lambda_1} + \cdots + \frac{\mathbf{u}_8\mathbf{u}_8^\top}{\lambda_8}. \tag{A.2}$$

For numerical computation, we multiply this expression by λ_8 to make it $O(1)$ to prevent numerical instability. We update $\hat{\mathbf{u}}$ and $V_0[\hat{\mathbf{u}}]$ iteratively until they converge as follows:

$$\hat{\mathbf{u}} \leftarrow N[\hat{\mathbf{u}} - \frac{(\det \mathbf{F})V_0[\hat{\mathbf{u}}]\hat{\mathbf{u}}^\dagger}{(\hat{\mathbf{u}}^\dagger, V_0[\hat{\mathbf{u}}]\hat{\mathbf{u}}^\dagger)}], \qquad V_0[\hat{\mathbf{u}}] \leftarrow \mathbf{P}_{\hat{\mathbf{u}}} V_0[\hat{\mathbf{u}}] \mathbf{P}_{\hat{\mathbf{u}}}. \tag{A.3}$$

Here, $\hat{\mathbf{u}}^\dagger$ denotes the following transformation of the vector $\hat{\mathbf{u}}$, corresponding to the cofactor $\mathbf{F}^\dagger$ of $\hat{\mathbf{F}}$:

$$\hat{\mathbf{u}}^\dagger = \begin{pmatrix} \hat{u}_5\hat{u}_9 - \hat{u}_8\hat{u}_6 \\ \hat{u}_6\hat{u}_7 - \hat{u}_9\hat{u}_4 \\ \hat{u}_4\hat{u}_8 - \hat{u}_7\hat{u}_5 \\ \hat{u}_8\hat{u}_3 - \hat{u}_2\hat{u}_6 \\ \hat{u}_9\hat{u}_1 - \hat{u}_3\hat{u}_7 \\ \hat{u}_7\hat{u}_2 - \hat{u}_1\hat{u}_8 \\ \hat{u}_2\hat{u}_6 - \hat{u}_5\hat{u}_3 \\ \hat{u}_3\hat{u}_4 - \hat{u}_6\hat{u}_1 \\ \hat{u}_1\hat{u}_5 - \hat{u}_4\hat{u}_2 \end{pmatrix}. \tag{A.4}$$

Appendix B. Singularity of ellipse fitting

The error term $\Delta\boldsymbol{\xi}_\alpha$ in (8) is written to a first approximation in the form

$$\Delta\boldsymbol{\xi}_\alpha = \frac{\partial\boldsymbol{\xi}}{\partial x}\Big|_{x=x_\alpha, y=y_\alpha} \Delta x_\alpha + \frac{\partial\boldsymbol{\xi}}{\partial y}\Big|_{x=x_\alpha, y=y_\alpha} \Delta y_\alpha. \tag{B.1}$$

By our assumption, we have $E[\Delta x_\alpha^2] = E[\Delta y_\alpha^2] = \sigma^2$ and $E[\Delta x_\alpha \Delta y_\alpha] = E[\Delta x_\alpha]E[\Delta y_\alpha] = 0$. Hence the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ in (10) is written as

$$V[\boldsymbol{\xi}_\alpha] = \sigma^2 \Big(\frac{\partial\boldsymbol{\xi}}{\partial x}\frac{\partial\boldsymbol{\xi}}{\partial x}^\top + \frac{\partial\boldsymbol{\xi}}{\partial y}\frac{\partial\boldsymbol{\xi}}{\partial y}^\top \Big)\Big|_{x=x_\alpha, y=y_\alpha} \tag{B.2}$$

The function (19) has a singularity if there is some α for which

$$(\mathbf{u}, V[\boldsymbol{\xi}_\alpha]\mathbf{u}) = \Big((\mathbf{u}, \frac{\partial\boldsymbol{\xi}}{\partial x})^2 + (\mathbf{u}, \frac{\partial\boldsymbol{\xi}}{\partial y})^2 \Big)\Big|_{x=x_\alpha, y=y_\alpha} = 0. \tag{B.3}$$

This occurs when there is a point (x_α, y_α) such that

$$(\mathbf{u}, \frac{\partial\boldsymbol{\xi}}{\partial x}) = (\mathbf{u}, \frac{\partial\boldsymbol{\xi}}{\partial y}) = 0. \tag{B.4}$$

Since the vector $\boldsymbol{\xi}$ is a function of x and y in the form of (16), the equation $z = (\mathbf{u}, \boldsymbol{\xi}(x, y))$ defines a convex surface in the xyz space. The equation (15) describes its cross section with the xy plane. Since this surface takes its minimum at the center of the ellipse, we have

$$\frac{\partial(\mathbf{u}, \boldsymbol{\xi})}{\partial x} = (\mathbf{u}, \frac{\partial\boldsymbol{\xi}}{\partial x}) = 0, \qquad \frac{\partial(\mathbf{u}, \boldsymbol{\xi})}{\partial y} = (\mathbf{u}, \frac{\partial\boldsymbol{\xi}}{\partial y}) = 0, \tag{B.5}$$

there. Hence, the function (19) diverges to ±∞ if one of the data points (x_α, y_α) is at the center of the ellipse represented by $\mathbf{u}$.

References

Björck, A., 1996. Numerical Methods for Least Squares Problems, SIAM, Philadelphia, U.S.A.

Bookstein, F.L., 1979. Fitting conic sections to scattered data, Comput. Gr. Image Process., 9, 56–71.

Chernov, N., 2007. On the convergence of fitting algorithms in computer vision, J. Math. Imaging. Vision, 27, 231-239.

Chernov, N., Lesort, C., 2004. Statistical efficiency of curve fitting algorithms, Comput. Stat. Data Anal., 47, 713–728.

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

Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D., 2004. A new constrained parameter estimator for computer vision applications, Image Vis. Comput., 22, 85–91.

Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D., 2005. FNS, CFNS and HEIV: A unifying approach, J. Math. Imaging Vis., 23, 175–183.

Fitzgibbon, A., Pilu, M., Fisher, R.B., 1999. Direct least square fitting of ellipses, IEEE Trans. Patt. Anal. Mach. Intell., 21, 476–480.

Gander, W., Golub, H., Strebel, R., 1994. Least-squares fitting of circles and ellipses, BIT, 34, 558–578.

Hartley, R., Zisserman, A., 2000. Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, U.K.

Huffel, S.V., Lemmerling, P., 2002. Total Least Squares and Errors-in-Variable Modeling, Kluwer, Dordrecht, the Netherlands.

Kanatani, K., 1993. Renormalization for unbiased estimation, Proc. 4th Int. Conf. Comput. Vision, 599–606.

Kanatani, K., 1996. Statistical Optimization for Geometric Computation: Theory and Practice, Elsevier, Amsterdam, the Netherlands; reprinted, 2005, Dover, New York, U.S.A.

Kanatani, K., 2006 Ellipse fitting with hyperaccuracy, IEICE Trans. Inf. Sys., E89-D, 2653–2660.

Kanatani, K., Statistical optimization for geometric fitting: Theoretical accuracy analysis and high order error analysis, Mem. Fac. Eng. Okayama Univ., 41, 73–92.

Kanatani, K., Ohta, N., 2003. Comparing optimal three-dimensional reconstruction for finite motion and optical flow, J. Elec. Imaging, 12, 478–488.

Leedan, Y., Meer, P., 2000. Heteroscedastic regression in computer vision: Problems with bilinear constraint, Int. J. Comput. Vis., 37, 127–150.

Sampson, P.D., 1982. Fitting conic sections to “very scattered data: An iterative refinement of the Bookstein algorithms, Comput. Gr. Image Process., 18, 97–108.

Taubin, G., 1991. 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, 1115–1138.