Memoirs of the Faculty of Engineering, Okayama University, Vol.41, pp.63-72, January, 2007

Experimental Evaluation of Geometric Fitting Algorithms

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

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

(Received November 16, 2006)

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. Numerical simulations and real image experiments and conducted for fundamental matrix computation and ellipse fitting, which reveals different characteristics of each method.

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 in the form

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

to N data $\boldsymbol{x}_\alpha$, $\alpha = 1, ..., N$, typically points in an image or point correspondences over multiple images [9]. The function $F(\boldsymbol{x};\boldsymbol{u})$, which may be a vector function if the model is defined by multiple equations, is parameterized by vector $\boldsymbol{u}$. Each $\boldsymbol{x}_\alpha$ is assumed to be perturbed by independent noise from its true value $\bar{\boldsymbol{x}}_\alpha$ which strictly satisfies Eq. (1).

From the parameter $\boldsymbol{u}$ of the fitted equation, one can discern the underlying geometric structure [9]. In this paper, we focus on problems for which Eq. (1) reduces to a linear form by changing variables. A large class of computer vision problems fall into this category [9].

For this, various algebraic methods were proposed in the past, but Kanatani [9] 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. [3] proposed a procedure called FNS, and Leedan and Meer [13] 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 [2, 9]. Kanatani's renormalization [8, 9] also computes a solution nearly equivalent to them [10].

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.

Sect. 2 and 3 state the problem and the KCR lower bound. Sect. 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. We then show numerical and real image examples of two typical problems: fundamental matrix computation in Sect. 6 and ellipse fitting in Sect. 7. Section 8 concludes this paper.

2. Statistical Optimization

Kanatani [9, 10] proved that if each datum $\boldsymbol{x}_\alpha$ is an independent Gaussian random variable with mean $\bar{\boldsymbol{x}}_\alpha$ and covariance matrix $V[\boldsymbol{x}_\alpha]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\boldsymbol{u}}$ of $\boldsymbol{u}$:

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

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 the function $F(\boldsymbol{x};\boldsymbol{u})$ in Eq. (1) with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively, evaluated at $\boldsymbol{x} = \bar{\boldsymbol{x}}_\alpha$. 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 $\boldsymbol{u}$,

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

which is generally a manifold in $\mathcal{R}^n$. Throughout this paper, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$.

Chernov and Lesort [2] called the right-hand side of Eq. (2) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that it holds except for $O(\varepsilon^4)$ even if $\hat{\boldsymbol{u}}$ is not unbiased; it is sufficient that $\hat{\boldsymbol{u}}$ is "consistent" in the sense $\hat{\boldsymbol{u}}$ converges to the true value $\boldsymbol{u}$ 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

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

A more sophisticated method is *maximum likelihood* (*ML*) (also called by many other names). We regard the data $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_N$ as perturbed from their true values $\bar{\boldsymbol{x}}_1$, ..., $\bar{\boldsymbol{x}}_N$ 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 $\bar{\boldsymbol{x}}_\alpha$. Within that tangent space, the noise is assumed to be independent Gaussian with mean $\mathbf{0}$ and covariance matrix $V[\boldsymbol{x}_\alpha]$. Then, the likelihood of observing $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_N$ is

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

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

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

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

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

This constraint can be eliminated by introducing Lagrange multipliers and ignoring higher order terms in the noise level. The resulting form is

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

It can be shown that the covariance matrix $V[\hat{\boldsymbol{u}}]$ of the resulting estimator $\hat{\boldsymbol{u}}$ achieves the KCR lower bound except for $O(\varepsilon^4)$ [2, 9, 10].

3. Linearized Constraint

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

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

by changing variables. Since the data $\boldsymbol{x}_\alpha$ are m-dimensional vectors and the unknown parameter $\boldsymbol{u}$ 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 Eq. (8), we normalize $\boldsymbol{u}$ to $\|\boldsymbol{u}\| = 1$.

The KCR lower bound for the linearized constraint has the form

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

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

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

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 $\boldsymbol{x} = \bar{\boldsymbol{x}}_\alpha$. Note that in Eq. (9) we do not need the projection operator for the normalization constraint $\|\boldsymbol{u}\| = 1$, because $\bar{\boldsymbol{\xi}}_\alpha$ is orthogonal to $\boldsymbol{u}$ due to Eq. (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 [7]:

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

Here, $\boldsymbol{F}$ 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

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

Eq. (12) is linearized in the form of Eq. (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

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

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 $\boldsymbol{F}$ should also satisfy the constraint that $\det \boldsymbol{F} = 0$ [7]. If this constraint is taken into account, the KCR lower bound involves the corresponding projection operation [9, 11].

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

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

Eq. (15) is linearized in the form of Eq. (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).

4. Numerical Computation of ML

As we can see from Eqs. (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 Eq. (8), Eq. (7) reduces to

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

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 maximum likelihood solution is obtained by solving

$$\begin{aligned} \nabla_{\mathbf{u}} J &= \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)\boldsymbol{\xi}_\alpha}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} - \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u}}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2} \\ &= (\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u} = \boldsymbol{0}, \end{aligned} \tag{20}$$

where we define

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

4.1 Fundamental Numerical Scheme (FNS)

The procedure called *FNS* (*fundamental numerical scheme*) of Chojnacki et al. [3] for solving Eq. (20) is described as follows:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ in Eqs. (21).
3. Solve the eigenvalue problem
$$(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u}' = \lambda \boldsymbol{u}', \tag{22}$$
and compute the unit eigenvector $\boldsymbol{u}'$ for the eigenvalue λ closest to 0.
4. If $\boldsymbol{u}' \approx \boldsymbol{u}$ except for sign, return $\boldsymbol{u}'$ and stop. Else, let $\boldsymbol{u} \leftarrow \boldsymbol{u}'$ and go back to Step 2.

Later, Chojnacki et al. [5] 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

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

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

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

On the other hand, Eqs. (21) imply that $(\boldsymbol{u}, \boldsymbol{M}\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{L}\boldsymbol{u})$ identically, meaning $\lambda = 0$.

4.2 Heteroscedastic Errors-in-Variables (HEIV)

Equation (20) can be rewritten as

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

The *HEIV* (*heteroscedastic errors-in-variables*) method of Leedan and Meer [13] is to iteratively solve the generalized eigenvalue problem $\boldsymbol{M}\boldsymbol{u} = \lambda\boldsymbol{L}\boldsymbol{u}$. In many applications, the matrix $\boldsymbol{L}$ 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 $\boldsymbol{u}$ and the normalized covariance matrix $V_0[\boldsymbol{\xi}_\alpha]$ have the following form:

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

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

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

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

where we put

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

Then, Eq. (25) splits into the following two equations [5, 13]:

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

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

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

where $N[\,\cdot\,]$ denotes normalization to unit norm. The vector $\boldsymbol{u}$ that satisfies the first of Eqs. (29) is computed by the following iterations [5, 13]:

1. Initialize $\boldsymbol{v}$.
2. Compute the matrices $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{L}}$ in Eqs. (27).
3. Solve the generalized eigenvalue problem
$$\tilde{\boldsymbol{M}}\boldsymbol{v}' = \lambda\tilde{\boldsymbol{L}}\boldsymbol{v}', \tag{31}$$
and compute the unit generalized eigenvector $\boldsymbol{v}'$ for the generalized eigenvalue λ closest to 1.
4. If $\boldsymbol{v}' \approx \boldsymbol{v}$ except for sign, return $\boldsymbol{v}'$ and stop. Else, let $\boldsymbol{v} \leftarrow \boldsymbol{v}'$ and go back to Step 2.

However, Leedan and Meer [13] 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

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

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

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

On the other hand, Eqs. (27) imply that $(\boldsymbol{v}, \tilde{\boldsymbol{M}}\boldsymbol{v}) = (\boldsymbol{v}, \tilde{\boldsymbol{L}}\boldsymbol{v})$ identically, meaning $\lambda = 1$.

4.3 Renormalization

Kanatani's *renormalization* [8, 9] is to approximate the matrix $\boldsymbol{L}$ in Eqs. (21) in the form

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

The constant c is determined so that $\boldsymbol{M} - c\boldsymbol{N}$ has eigenvalue 0. This is done by the following iterations [9]:

1. Initialize $\boldsymbol{u}$ and let $c = 0$.
2. Compute the matrix $\boldsymbol{M}$ in Eqs. (21) and the matrix $\boldsymbol{N}$ in Eqs. (34).
3. Solve the eigenvalue problem
$$(\boldsymbol{M} - c\boldsymbol{N})\boldsymbol{u}' = \lambda\boldsymbol{u}', \tag{35}$$
and compute the unit eigenvector $\boldsymbol{u}'$ for the eigenvalue λ closest to 0.
4. If $\lambda \approx 0$, return $\boldsymbol{u}'$ and stop. Else, let
$$c \leftarrow c + \frac{\lambda}{(\boldsymbol{u}', \boldsymbol{N}\boldsymbol{u}')}, \quad \boldsymbol{u} \leftarrow \boldsymbol{u}' \tag{36}$$
and go back to Step 2.

4.4 Gauss-Newton Iterations

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

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

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

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

we obtain a solution orthogonal to $\boldsymbol{u}$.

Differentiating Eq. (20) and introducing Gauss-Newton approximation (i.e., ignoring terms that contain $(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)$), we see that the Hessian is nothing but the matrix $\boldsymbol{M}$ in Eqs. (21). We enforce $\boldsymbol{M}$ to have eigenvalue 0 for $\boldsymbol{u}$, using the projection matrix

$$\boldsymbol{P}_{\mathbf{u}} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top, \tag{39}$$

where $\boldsymbol{I}$ is the unit matrix. The iteration procedure goes as follows:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ in Eqs. (21).
3. Let
$$\boldsymbol{u}' = N[\boldsymbol{u} - (\boldsymbol{P}_{\mathbf{u}}\boldsymbol{M}\boldsymbol{P}_{\mathbf{u}})^-(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u}]. \tag{40}$$
4. If $\boldsymbol{u}' \approx \boldsymbol{u}$, return $\boldsymbol{u}'$ and stop. Else, let $\boldsymbol{u} \leftarrow \boldsymbol{u}'$ 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 Eq. (19) by a constant, we minimize

$$J_{\mathrm{LS}} = \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 = \frac{1}{2}(\boldsymbol{u}, \boldsymbol{M}_{\mathrm{LS}}\boldsymbol{u}),$$
$$\boldsymbol{M}_{\mathrm{LS}} = \sum_{\alpha=1}^{N}\boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top. \tag{41}$$

Equation (41) is minimized by the unit eigenvalue $\boldsymbol{u}$ of $\boldsymbol{M}_{\mathrm{LS}}$ for the smallest eigenvalue.

5.3 Taubin's Method

Replacing the denominators in Eq. (19) by their average, we minimize the following function[1] [14]:

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

[1] Taubin [14] did not take the covariance matrix into account. This is a modification of his method.

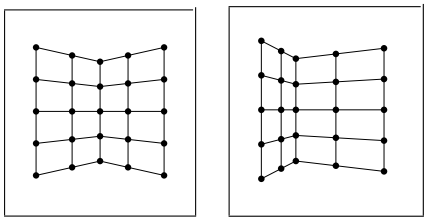


Figure 1: Simulated images of planar grid surfaces.

$$\boldsymbol{N}_{\mathrm{TB}} = \sum_{\alpha=1}^{N}V_0[\boldsymbol{\xi}_\alpha]. \tag{42}$$

Equation (42) is minimized by solving the generalized eigenvalue problem

$$\boldsymbol{M}_{\mathrm{LS}}\boldsymbol{u} = \lambda\boldsymbol{N}_{\mathrm{TB}}\boldsymbol{u}, \tag{43}$$

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

$$\tilde{\boldsymbol{M}}_{\mathrm{LS}} = \sum_{\alpha=1}^{N}\tilde{\boldsymbol{z}}_\alpha\tilde{\boldsymbol{z}}_\alpha^\top, \qquad \tilde{\boldsymbol{N}}_{\mathrm{TB}} = \sum_{\alpha=1}^{N}V_0[\boldsymbol{z}_\alpha], \tag{44}$$

where

$$\tilde{\boldsymbol{z}}_\alpha = \boldsymbol{z}_\alpha - \bar{\boldsymbol{z}}, \qquad \bar{\boldsymbol{z}} = \frac{1}{N}\sum_{\alpha=1}^{N}\boldsymbol{z}_\alpha. \tag{45}$$

Then, Eq. (43) splits into two equations

$$\tilde{\boldsymbol{M}}_{\mathrm{LS}}\boldsymbol{v} = \lambda\tilde{\boldsymbol{N}}_{\mathrm{TB}}\boldsymbol{v}, \qquad (\boldsymbol{v}, \bar{\boldsymbol{z}}) + Ca = 0. \tag{46}$$

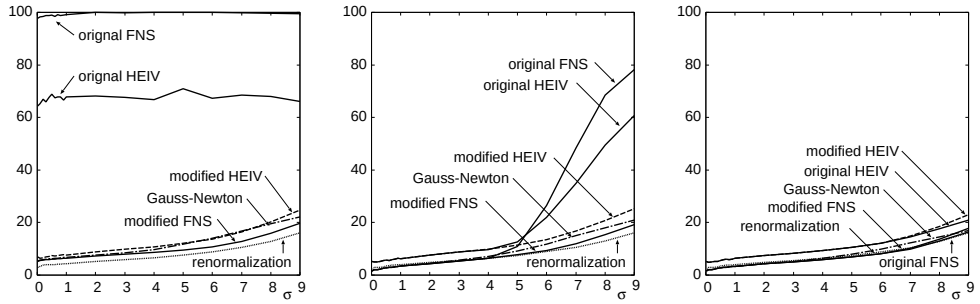
If we compute the unit generalized eigenvector $\boldsymbol{v}$ of the first equation for the smallest generalized eigenvalue λ, the second gives a. Hence, $\boldsymbol{u}$ is given by Eq. (30).

6. Fundamental Matrix Computation

6.1 Simulated Images

Figure 1 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 different methods.

The fundamental matrix $\boldsymbol{F}$ should satisfy the constraint $\det\boldsymbol{F} = 0$ [7], and Chojnacki et al. [4] presented a FNS-like procedure to incorporate this constraint. However, once the solution $\hat{\boldsymbol{u}}$ of Eq. (20) is obtained, it can be easily corrected so as to satisfy $\det\boldsymbol{F} = 0$ in such a way that the accuracy is equivalent to the constrained minimization of Eq. (19) subject to $\det\boldsymbol{F} = 0$ except for higher order terms in σ



(a) (b) (c)

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

(see Appendix A for the procedure) [9, 11]. So, we consider here the computation prior to this correction.

Figure 2 shows the average number of iterations of each method for 1000 trials. We stopped when the increment in $\boldsymbol{u}$ 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.

Figure 2(a) shows that the modified FNS/HEIV converge much more quickly than the original FNS/HEIV. This can be explained as follows. If the computed $\boldsymbol{u}'$ is close to the true value $\boldsymbol{u}$, the matrix $\boldsymbol{L}$ in Eqs. (21) and the matrix $\tilde{\boldsymbol{L}}$ in Eqs. (27) are both close to $\boldsymbol{O}$. Initially, however, they may be very different from $\boldsymbol{O}$ when the initial value is randomly chosen. Equations (22) and (31) are written, respectively, as

$$(\boldsymbol{M} - \boldsymbol{L} - \lambda \boldsymbol{I})\boldsymbol{u}' = \boldsymbol{0}, \quad (\tilde{\boldsymbol{M}} - \lambda \tilde{\boldsymbol{L}})\boldsymbol{v}' = \boldsymbol{0}. \quad (47)$$

Note that $\boldsymbol{L}$ and $\tilde{\boldsymbol{L}}$ 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 Eq. (35) with $c = 0$, canceling the effect of $\boldsymbol{N}$ whatever it is, and the resulting $\boldsymbol{u}'$ 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 $\boldsymbol{L}$ and $\tilde{\boldsymbol{L}}$ are very different from $\boldsymbol{O}$.

As Fig. 2(b), (c) shows, however, the convergence performance of FNS and HEIV improves as we use better initial values. Naturally, 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.

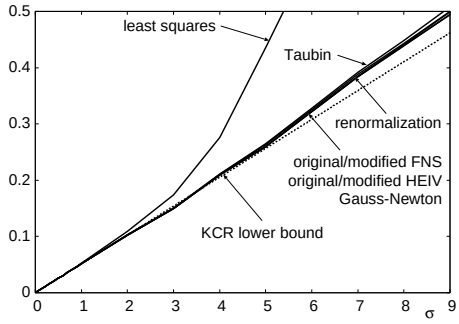


Figure 3: RMS error vs. noise level.

Figure 3 plots for each σ the RMS (root-mean-squares) of $\|\boldsymbol{P}_{\mathbf{u}}\hat{\boldsymbol{u}}\|$ 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 Eq. (9).

We can see that Taubin's method is considerably better[2] 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 Gauss-Newton minimize the same function, the resulting solution is virtually the same. The renormalization solution is nearly equivalent to them.

[2]The mechanism of the superiority of Taubin's method over LS is analyzed in detail in [10].

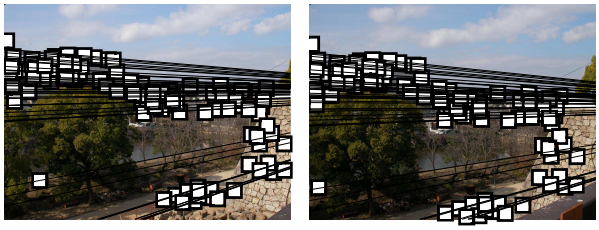


Figure 4: Corresponding points in real images and computed epipolar lines.

Table 1: Number of iterations and the computation time (sec) for different methods and different initializations.

	random		LS		Taubin	
original FNS	94.3	.168s	5	.009s	5	.012s
modified FNS	12.0	.030s	5	.010s	5	.013s
original HEIV	74.6	.264s	7	.019s	7	.025s
modified HEIV	9.1	.037s	7	.020s	7	.026s
renormalization	7.0	.022s	7	.013s	7	.017s
Gauss-Newton	10.3	.038s	5	.009s	6	.017s

6.2 Real Images

Figure 4 shows two images of the same scene. We manually chose corresponding 100 points as marked there and computed the fundamental matrix by different methods. The solution is the same whichever is used, and the computed epipolar lines are drawn in the images.

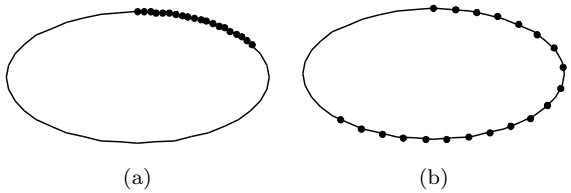
Table 1 lists the number of iterations and the computation time (sec) for each method. For random initialization, we computed the average over 100 independent trials. We used Pentium 4 3.4GHz for the CPU with 2GB main memory and Linux for the OS.

We observe that for whichever initialization, FNS is always better than HEIV. For both, the choice of the eigenvalue is irrelevant if the iterations are initialized by LS or Taubin' method; for random initialization, the original FNS/HEIV do not converge in most of the trials (recall that 100 means nonconvergence). As predicted, the number of iterations of renormalization does not depend on initialization.

The difference in computation time between LS and Taubin initializations is due to the initialization computation: 0.0009s for LS vs. 0.0015s for Taubin. Overall, LS initialized (original or modified) FNS shows the best convergence performance.

7. Ellipse Fitting

Figure 5 shows two examples of 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



(a) (b)

Figure 5: 20 points on elliptic arcs. (a) Short arch. (b) Long arc

fitted an ellipse by different methods.

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. [6] presented a technique for preventing this Here, however, we do not impose any constraints to prevent non-ellipses, assuming that noise is sufficiently small.

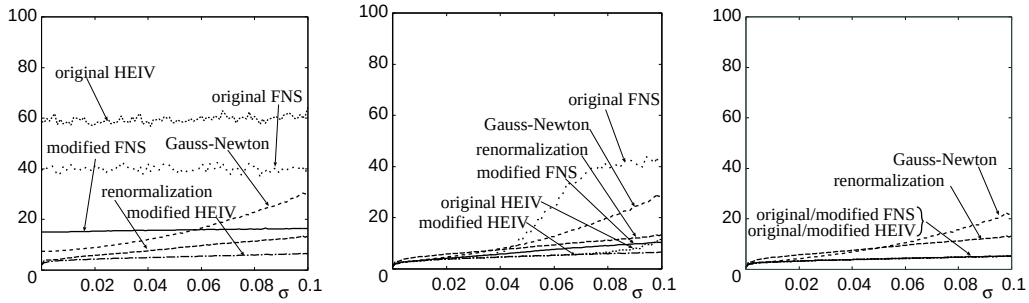
Doing numerical experiments, we have found that the convergence performance differs greatly, depending on whether we use points on a short elliptic arc as in Fig. 5(a) or on a long elliptic arc as in Fig. 5(b). So, we study the two cases separately.

7.1 Fitting to a Short Arc

For each σ, we computed the average number of iterations over 1000 independent trials (Fig. 6). We stopped when the increment in $\boldsymbol{u}$ 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.

Figure 7 plots for each σ the RMS of $\|\boldsymbol{P}_{\mathbf{u}}\hat{\boldsymbol{u}}\|$ 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 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).



(a) (b) (c)

Figure 6: Average number of iterations for ellipse fitting vs. noise level. (a) Random initialization. (b) LS initialization. (c) Taubin initialization.

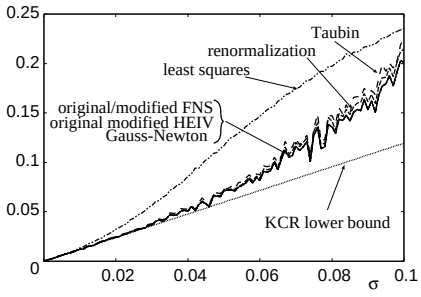


Figure 7: RMS error of ellipse fitting to the points in Fig. 5(a) vs. noise level.

7.2 Fitting to a Long Arc

The comparative behavior of each method for points distributed over a short elliptic arc in Fig. 5(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(b) is used.

Figure 8 shows the number of iterations corresponding to Fig. 6. 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 [1] (see Appendix B): Some of the terms on the right-hand side of Eq. (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. 8, renormalization is the most stable for whichever initialization. As we noted earlier, this is because the iterations start from $c = 0$. Gauss-Newton iterations are also stable.

Figure 9 shows the RMS error corresponding to Fig. 7. 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.

7.3 Real Images

The left image in Fig. 10 shows edges extracted from a real image of a circular object occluded to a large extend. For edge detection, we used the method of Kanatani and Ohta [12]. We fitted an ellipse to 148 points that constitute an elliptic arc by different methods. All iterations are initialized Taubin's method. The right image shows the resulting ellipses overlaid on the original image .

The solutions obtained by (original and modified) FNS/HEIV, renormalization, and Gauss-Newton iterations are indistinguishable when displayed as ellipses. We can see that the accuracy of LS is very much lower than Taubin's method, compared to which the accuracy gain by ML is very small.

Figure 11 shows corresponding results for a less occluded circular object. In this case, the LS fit is sufficiently accurate, producing practically the same solution as the other methods.

Table 2 lists the number of iterations and computation time (sec) for different methods. The computation time for Taubin initialization is 0.00063s for Fig. 10 and 0.00166s for Fig. 11.

Since the initial Taubin solution is sufficiently accurate already, all the methods converge after the same number of iterations except renormalization. This exceptional behavior of renormalization can be explained as follows. Because renormalization always starts from $c = 0$ irrespective of the accuracy gain by the initial Taubin solution, it requires one or two additional iterations to arrive at the same accuracy.

8. Conclusions

We have compared the convergence performance of different numerical schemes for geometric fitting.

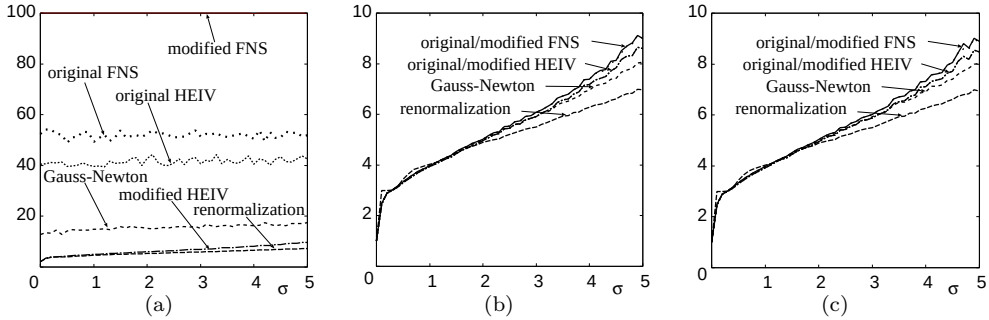


Figure 8: Average number of iterations for ellipse fitting to the points in Fig. 3 vs. noise level. (a) Random initialization. (b) LS initialization. (c) Taubin initialization.

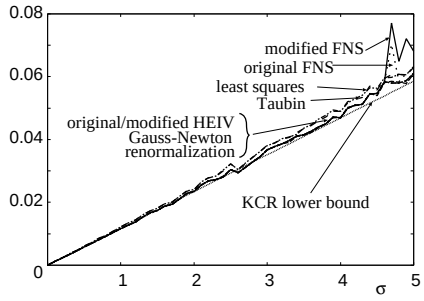


Figure 9: RMS error of ellipse fitting to the points in Fig. 5(b) vs. noise level.

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 a special variant of Gauss-Newton iterations. For initial values, we tested random choice, LS, and Taubin's method. Numerical and real image 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. This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C (No. 17500112).

References

[1] N. Chernov, On the convergence of numerical schemes in computer vision, *J. Math. Imaging. Vision*, **25** (2007), to appear.

[2] N. Chernov and C. Lesort, Statistical efficiency of curve fitting algorithms, *Comput. Stat. Data Anal.*, **47**-4 (2004-11), 713–728.

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

[4] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley, A new constrained parameter estimator for computer vision applications, *Image Vis. Comput.*, **22**-2 (2004-2), 85–91.

[5] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D Gawley, FNS, CFNS and HEIV: A unifying approach, *J. Math. Imaging Vis.*, **23**-2 (2005-9), 175–183.

[6] A. Fitzgibbon, M. Pilu, and R. B. Fisher, Direct least square fitting of ellipses, *IEEE Trans. Patt. Anal. Mach. Intell.*, **21**-5 (1999-5), 476–480.

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

[8] K. Kanatani, Renormalization for unbiased estimation, *Proc. 4th Int. Conf. Comput. Vision* May 1993, Berlin, Germany, pp. 599–606.

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

[10] K. Kanatani, Hyperaccuracy for geometric fitting, *4th Int. Workshop Total Least Squares and Errors-in-Variables Modelling*, Leuven, Belgium, August 2006.

[11] K. Kanatani and N. Ohta, Comparing optimal three-dimensional reconstruction for finite motion and optical flow, *J. Elec. Imaging*, **12**-3 (2003-7), 478–488.

[12] K. Kanatani and N. Ohta, Automatic detection of circular objects by ellipse growing, *Int. J. Image Graphics*, **4**-1 (2004-1), 35–50.

[13] Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comput. Vis.*, **37**-2 (2000-6), 127–150.

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

Appendix

A Rank Constraint Optimization

The computed fundamental matrix $\boldsymbol{F}$ can be optimally corrected so as to satisfy $\det \boldsymbol{F} = 0$ as follows [9, 11].

Let $\hat{\boldsymbol{u}}$ be the 9-dimensional vector representation of the ML estimate $\hat{\boldsymbol{F}}$ of the fundamental matrix $\boldsymbol{F}$

Figure 10: Left: Edge image of a circular object occluded to a large extend. Right: Ellipses fitted to the 148 edge points overlaid on the original image. The thin line is for LS, the white line is for Taubin's method, and the thick line is for the ML solution.

Figure 11: Left: Edge image of a circular object occluded to a small extend. Right: Ellipses fitted to the 414 edge points overlaid on the original image. The thin line is for LS, the white line is for Taubin's method, and the thick line is for the ML solution.

computed without the constraint $\det \boldsymbol{F} = 0$. Compute

$$\tilde{\boldsymbol{M}} = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{P}_{\hat{\boldsymbol{u}}}\boldsymbol{\xi}_\alpha)(\boldsymbol{P}_{\hat{\boldsymbol{u}}}\boldsymbol{\xi}_\alpha)^\top}{(\hat{\boldsymbol{u}}, V_0[\boldsymbol{\xi}_\alpha]\hat{\boldsymbol{u}})}, \tag{48}$$

where $\boldsymbol{P}_{\hat{\boldsymbol{u}}}$ is the projection matrix defined in Eqs. (40). Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_9 \ (= 0)$ be the eigenvalues of $\tilde{\boldsymbol{M}}$, and $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, ..., $\boldsymbol{u}_9 \ (= \hat{\boldsymbol{u}})$ the corresponding orthonormal system of eigenvectors. The covariance matrix $V[\hat{\boldsymbol{u}}]$ of $\hat{\boldsymbol{u}}$ is estimated to be $V_0[\hat{\boldsymbol{u}}]$ up to a positive multiplier in the following form[3] [9]:

$$V_0[\hat{\boldsymbol{u}}] = \frac{\boldsymbol{u}_1\boldsymbol{u}_1^\top}{\lambda_1} + \cdots + \frac{\boldsymbol{u}_8\boldsymbol{u}_8^\top}{\lambda_8}. \tag{49}$$

We update $\hat{\boldsymbol{u}}$ and $V_0[\hat{\boldsymbol{u}}]$ iteratively until they converge as follows:

$$\hat{\boldsymbol{u}} \leftarrow N[\hat{\boldsymbol{u}} - \frac{(\det \boldsymbol{F})V_0[\hat{\boldsymbol{u}}]\hat{\boldsymbol{u}}^\dagger}{(\hat{\boldsymbol{u}}^\dagger, V_0[\hat{\boldsymbol{u}}]\hat{\boldsymbol{u}}^\dagger)}], \qquad V_0[\hat{\boldsymbol{u}}] \leftarrow \boldsymbol{P}_{\hat{\boldsymbol{u}}}V_0[\hat{\boldsymbol{u}}]\boldsymbol{P}_{\hat{\boldsymbol{u}}}. \tag{50}$$

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

$$\hat{\boldsymbol{u}}^\dagger = \begin{pmatrix} \hat{u}_5\hat{u}_9 - \hat{u}_8\hat{u}_6 \\ \hat{u}_6\hat{u}_7 - \hat{u}_9\hat{u}_4 \\ \hat{u}_4\hat{u}_8 - \hat{u}_7\hat{u}_5 \\ \hat{u}_8\hat{u}_3 - \hat{u}_2\hat{u}_6 \\ \hat{u}_9\hat{u}_1 - \hat{u}_3\hat{u}_7 \\ \hat{u}_7\hat{u}_2 - \hat{u}_1\hat{u}_8 \\ \hat{u}_2\hat{u}_6 - \hat{u}_5\hat{u}_3 \\ \hat{u}_3\hat{u}_4 - \hat{u}_6\hat{u}_1 \\ \hat{u}_1\hat{u}_5 - \hat{u}_4\hat{u}_2 \end{pmatrix}. \tag{51}$$

B Singularity of Ellipse Fitting

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

$$\Delta\boldsymbol{\xi}_\alpha = \frac{\partial\boldsymbol{\xi}}{\partial x}\Big|_{x=x_\alpha, y=y_\alpha} \Delta x_\alpha + \frac{\partial\boldsymbol{\xi}}{\partial y}\Big|_{x=x_\alpha, y=y_\alpha} \Delta y_\alpha. \tag{52}$$

[3] For numerical computation, we multiply this expression by λ_8 to make it $O(1)$ to prevent numerical instability.

Table 2: Number of iterations and the computation time (sec) for fitting an ellipse to the data in Figs. 10 and 11.

	Fig. 10		Fig. 11	
original FNS	5	.000s	3	.000s
modified FNS	5	.008s	3	.013s
original HEIV	5	.006s	3	.010s
modified HEIV	5	.006s	3	.010s
Gauss-Newton	5	.009s	3	.015s
renormalization	7	.008s	4	.012s

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 Eq. (10) is written as

$$V[\boldsymbol{\xi}_\alpha] = \sigma^2\Big(\frac{\partial\boldsymbol{\xi}}{\partial x}\frac{\partial\boldsymbol{\xi}}{\partial x}^\top + \frac{\partial\boldsymbol{\xi}}{\partial y}\frac{\partial\boldsymbol{\xi}}{\partial y}^\top\Big)\Big|_{x=x_\alpha, y=y_\alpha} \tag{53}$$

The function J in Eq. (19) has a singularity if there is some α for which

$$(\boldsymbol{u}, V[\boldsymbol{\xi}_\alpha]\boldsymbol{u}) = \Big((\boldsymbol{u}, \frac{\partial\boldsymbol{\xi}}{\partial x})^2 + (\boldsymbol{u}, \frac{\partial\boldsymbol{\xi}}{\partial y})^2\Big)\Big|_{x=x_\alpha, y=y_\alpha} = 0. \tag{54}$$

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

$$(\boldsymbol{u}, \frac{\partial\boldsymbol{\xi}}{\partial x}) = (\boldsymbol{u}, \frac{\partial\boldsymbol{\xi}}{\partial y}) = 0. \tag{55}$$

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

$$\frac{\partial(\boldsymbol{u}, \boldsymbol{\xi})}{\partial x} = (\boldsymbol{u}, \frac{\partial\boldsymbol{\xi}}{\partial x}) = 0, \qquad \frac{\partial(\boldsymbol{u}, \boldsymbol{\xi})}{\partial y} = (\boldsymbol{u}, \frac{\partial\boldsymbol{\xi}}{\partial y}) = 0, \tag{56}$$

there. Hence, the function J in Eq. (19) diverges to $\pm\infty$ if one of the data points (x_α, y_α) is at the center of the ellipse represented by $\boldsymbol{u}$.