Proc. 21th Int. Conf. Image and Vision Computing New Zealand 2006
Great Barrier Island, New Zealand, November 2006, pp. 7–12

Performance Evaluation of Accurate Ellipse Fitting

Kenichi Kanatani

Department of Computer Science, Okayama University, Okayama, 700-8530 Japan.
Email: kanatani@suri.it.okayama-u.ac.jp

Abstract

This paper studies numerical schemes for fitting an ellipse to points in an image. First, the problem is posed as maximum likelihood, and the relationship to the KCR lower bound is stated. Then, the algorithms of FNS, HEIV, renormalization, and Gauss-Newton iterations are described. Using simulated and real image data, their convergence properties are compared, and their dependence on the shape of the arc to which an ellipse is to be fitted is revealed.

Keywords: ellipse fitting, KCR lower bound, FNS, HEIV, renormalization

1 Introduction

Circular objects in the scene are generally projected onto ellipses on the image plane, and their 3-D positions can be computed from their images [6]. For this reason, fitting ellipses to a point sequence is one of the first steps of various vision applications. In this paper, we concentrate on numerical aspects, assuming that outliers have already been removed, e.g., by the procedure described in [10].

Various algebraic fitting methods were proposed in the past [1, 13, 15, 16], but Kanatani [8] pointed out that ellipse fitting can be regarded as statistical estimation and that maximum likelihood (ML) produces an optimal solution. Since then, many numerical schemes have been proposed, e.g., FNS [4], the HEIV [14], and Gauss-Newton iterations [11]. These methods attain a theoretical accuracy bound (KCR lower bound [3, 8]) up to high order terms in noise. Kanatani's renormalization [7, 8, 12] also computes a solution nearly equivalent to them [9].

All these methods are iterative, and the convergence properties are different from method to method. The purpose of this paper is to experimentally compare their convergence behavior.

2 Ellipse Fitting

An ellipse is represented by

$$Ax^2+2Bxy+Cy^2+2f_0(Dx+Ey)+Ff_0^2=0, \quad (1)$$

where f_0 is an arbitrary scaling constant[1]. If we define

$$\boldsymbol{u}=(A,B,C,D,E,F)^\top, \quad (2)$$

$$\boldsymbol{\xi}=(x^2,2xy,y^2,2f_0x,2f_0y,f_0^2)^\top, \quad (3)$$

Eq. (1) is written as

$$(\boldsymbol{u},\boldsymbol{\xi})=0. \quad (4)$$

Throughout this paper, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a},\boldsymbol{b})$. Since the magnitude of the vector $\boldsymbol{u}$ is indeterminate, we adopt normalization $\|\boldsymbol{u}\|=1$.

Eq. (1) describes not necessarily an ellipse but also a parabola, a hyperbola, and their degeneracies (e.g., two lines) [6]. Even if the points (x_α,y_α) are sampled from an ellipse, the fitted equation may define a hyperbola or other curves in the presence of large noise, and a technique for preventing this has been proposed [13]. Here, however, we do not impose any constraints, assuming that noise is sufficiently small.

3 KCR Lower Bound

We write the data $\boldsymbol{\xi}_\alpha$ in the form $\boldsymbol{\xi}_\alpha=\bar{\boldsymbol{\xi}}_\alpha+\Delta\boldsymbol{\xi}_\alpha$, where $\bar{\boldsymbol{\xi}}_\alpha$ is the true value and $\Delta\boldsymbol{\xi}_\alpha$ the noise term. We define the covariance matrix of $\boldsymbol{\xi}_\alpha$ by

$$V[\boldsymbol{\xi}_\alpha]=E[\Delta\boldsymbol{\xi}_\alpha\Delta\boldsymbol{\xi}_\alpha^\top], \quad (5)$$

where $E[\,\cdot\,]$ denotes expectation over the noise distribution. If random noise of mean 0 and standard deviation σ is independently added to each coordinate of the points in the image, we can see from Eq. (3) that the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ has the form $4\sigma^2V_0[\boldsymbol{\xi}_\alpha]$ except for $O(\sigma^4)$, where $V_0[\boldsymbol{\xi}_\alpha]$ is

$$\begin{pmatrix} \bar{x}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & \bar{x}_\alpha^2+\bar{y}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & f_0\bar{y}_\alpha & f_0\bar{x}_\alpha & 0 \\ 0 & \bar{x}_\alpha\bar{y}_\alpha & \bar{y}_\alpha^2 & 0 & f_0\bar{y}_\alpha & 0 \\ f_0\bar{x}_\alpha & f_0\bar{y}_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0\bar{x}_\alpha & f_0\bar{y}_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6)$$

[1] In our experiments, we set $f_0=600$. This is to make the coefficients have approximately the same magnitude for numerical stability. Theoretically, we can set $f_0=1$.

Here, $(\bar{x}_\alpha, \bar{y}_\alpha)$ is the true position of point (x_α, y_α). In actual computations, $(\bar{x}_\alpha, \bar{y}_\alpha)$ is approximated[2] by the data position (x_α, y_α).

We define the covariance matrix $V[\hat{\boldsymbol{u}}]$ of an estimate $\hat{\boldsymbol{u}}$ by

$$V[\hat{\boldsymbol{u}}] = E[(\boldsymbol{P}_{\mathbf{u}}\hat{\boldsymbol{u}})(\boldsymbol{P}_{\mathbf{u}}\hat{\boldsymbol{u}})^\top], \tag{7}$$

where $\boldsymbol{P}_{\mathbf{u}}$ is the projection matrix

$$\boldsymbol{P}_{\mathbf{u}} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top, \tag{8}$$

which projects $\hat{\boldsymbol{u}}$ onto the hyperplane orthogonal to $\boldsymbol{u}$ ($\boldsymbol{I}$ denotes the unit matrix). Since the parameter vector $\boldsymbol{u}$ is normalized to unit norm, its domain is the unit sphere $\mathcal{S}^5$ in $\mathcal{R}^6$. We focus on the asymptotic limit of small noise and evaluate the error after projecting $\hat{\boldsymbol{u}}$ onto the tangent space to $\mathcal{S}^5$ at $\boldsymbol{u}$ [8].

Kanatani [8, 9] proved that if $\boldsymbol{\xi}_\alpha$ is regarded as an independent Gaussian random variable of mean $\bar{\boldsymbol{\xi}}_\alpha$ and covariance matrix $V[\boldsymbol{\xi}_\alpha]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\boldsymbol{u}}$ of $\boldsymbol{u}$:

$$V[\hat{\boldsymbol{u}}] \succ 4\sigma^2 \Big(\sum_{\alpha=1}^{N} \frac{\bar{\boldsymbol{\xi}}_\alpha \bar{\boldsymbol{\xi}}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} \Big)^-_5. \tag{9}$$

Here, $\succ$ means that the left-hand side minus the right is positive semidefinite, and $(\,\cdot\,)^-_r$ means pseudoinverse of rank r.

Chernov and Lesort [3] called the right-hand side of Eq. (9) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that it holds except for terms of $O(\sigma^4)$ even if $\hat{\boldsymbol{u}}$ is not unbiased; it is sufficient that $\hat{\boldsymbol{u}}$ is "consistent" in the sense that $\hat{\boldsymbol{u}} \to \boldsymbol{u}$ as $\sigma \to 0$.

4 Maximum Likelihood (ML)

Maximum likelihood (*ML*) under Gaussian noise assumption is to minimize the sum of squared Mahalanobis distances

$$J = \frac{1}{2} \sum_{\alpha=1}^{N} (\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha, V_0[\boldsymbol{\xi}_\alpha]^-_2 (\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha)), \tag{10}$$

subject to the constraints $(\boldsymbol{u}, \bar{\boldsymbol{\xi}}_\alpha) = 0$, $\alpha = 1, ..., N$. Eliminating the constraints by introducing Lagrange multiplies, we can write Eq. (10) as follows [8, 9]:

$$J = \frac{1}{2} \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}. \tag{11}$$

It can be shown that the covariance matrix $V[\hat{\boldsymbol{u}}]$ of the resulting estimator $\hat{\boldsymbol{u}}$ agrees with the KCR lower bound except for terms of $O(\sigma^4)$ [8, 9].

[2] We have confirmed that this does not cause any noticeable changes in the final results.

Eq. (11) is minimized by solving

$$\begin{aligned} \nabla_{\mathbf{u}} J &= \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)\boldsymbol{\xi}_\alpha}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} - \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u}}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2} \\ &= (\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u} = \boldsymbol{0}, \end{aligned} \tag{12}$$

where the 6×6 matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ are defined by

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \tag{13}$$

$$\boldsymbol{L} = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}. \tag{14}$$

FNS

The *FNS* (*fundamental numerical scheme*) of Chojnacki et al. [4] solves Eq. (12) by the following iterations:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ in Eqs. (13) and (14).
3. Solve the eigenvalue problem
$$(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u}' = \lambda \boldsymbol{u}', \tag{15}$$
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

$$(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u} = \lambda \boldsymbol{u} \tag{16}$$

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

$$(\boldsymbol{u}, \boldsymbol{M}\boldsymbol{u}) - (\boldsymbol{u}, \boldsymbol{L}\boldsymbol{u}) = \lambda. \tag{17}$$

On the other hand, Eqs. (13) and (14) imply that $(\boldsymbol{u}, \boldsymbol{M}\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{L}\boldsymbol{u})$ identically, meaning $\lambda = 0$.

HEIV

Let

$$\boldsymbol{\xi}_\alpha = \begin{pmatrix} \boldsymbol{z}_\alpha \\ f_0^2 \end{pmatrix}, \qquad \boldsymbol{u} = \begin{pmatrix} \boldsymbol{v} \\ F \end{pmatrix}, \tag{18}$$

$$V_0[\boldsymbol{\xi}_\alpha] = \begin{pmatrix} V_0[\boldsymbol{z}_\alpha] & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix}. \tag{19}$$

Define 5×5 matrices $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{L}}$ by

$$\tilde{\boldsymbol{M}} = \sum_{\alpha=1}^{N} \frac{\tilde{\boldsymbol{z}}_\alpha \tilde{\boldsymbol{z}}_\alpha^\top}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})}, \tag{20}$$

$$\tilde{\boldsymbol{L}} = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{v}, \tilde{\boldsymbol{z}}_\alpha)^2 V_0[\boldsymbol{z}_\alpha]}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})^2}, \tag{21}$$

where we put

$$\tilde{\boldsymbol{z}}_\alpha = \boldsymbol{z}_\alpha - \bar{\boldsymbol{z}}, \tag{22}$$

$$\bar{\boldsymbol{z}} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{z}_\alpha}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})} \Bigg/ \sum_{\beta=1}^{N} \frac{1}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\beta]\boldsymbol{v})} \ . \tag{23}$$

Then, Eq. (12) splits into the following two equations [5]:

$$\tilde{\boldsymbol{M}}\boldsymbol{v} = \tilde{\boldsymbol{L}}\boldsymbol{v}, \qquad (\boldsymbol{v}, \bar{\boldsymbol{z}}) + f_0^2 F = 0. \tag{24}$$

If we determine a 5-D unit vector $\boldsymbol{v}$ that satisfies the first equation, the value of F is determined from the second, and we obtain $\boldsymbol{u}$ in the form

$$\boldsymbol{u} = N[\begin{pmatrix} \boldsymbol{v} \\ F \end{pmatrix}], \tag{25}$$

where $N[\,\cdot\,]$ denotes normalization to unit norm. The *HEIV* (*heteroscedastic errors-in-variables*) method of Leedan and Meer [14] computes the vector $\boldsymbol{v}$ by the following iterations:

1. Initialize $\boldsymbol{v}$.
2. Compute the matrices $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{L}}$ in Eqs. (20) and (21).
3. Solve the generalized eigenvalue problem

   $$\tilde{\boldsymbol{M}}\boldsymbol{v}' = \lambda\tilde{\boldsymbol{L}}\boldsymbol{v}', \tag{26}$$

   and compute the unit eigenvector $\boldsymbol{v}'$ for the 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 [14] pointed out that choosing in Step 3 not the eigenvalue closest to 1 but the smallest one improves the convergence performance. We call the above procedure the *original HEIV* and the one using the smallest eigenvalue the *modified HEIV*.

Whichever eigenvalue is chosen for λ, we have $\lambda = 1$ after convergence. In fact, convergence means

$$\tilde{\boldsymbol{M}}\boldsymbol{v} = \lambda\tilde{\boldsymbol{L}}\boldsymbol{v} \tag{27}$$

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

$$(\boldsymbol{v}, \tilde{\boldsymbol{M}}\boldsymbol{v}) = \lambda(\boldsymbol{v}, \tilde{\boldsymbol{L}}\boldsymbol{v}). \tag{28}$$

On the other hand, Eqs. (20) and (21) imply that $(\boldsymbol{v}, \tilde{\boldsymbol{M}}\boldsymbol{v}) = (\boldsymbol{v}, \tilde{\boldsymbol{L}}\boldsymbol{v})$ identically, meaning $\lambda = 1$.

Renormalization

The *renormalization* of Kanatani [8] is to approximate the matrix $\boldsymbol{L}$ in Eq. (14) in the form

$$\boldsymbol{L} \approx c\boldsymbol{N}, \qquad \boldsymbol{N} = \sum_{\alpha=1}^{N} \frac{V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}. \tag{29}$$

The constant c is determined so that $\boldsymbol{M} - c\boldsymbol{N}$ has eigenvalue 0. This is done by the following iterations [8]:

1. Initialize $\boldsymbol{u}$ and let $c = 0$.
2. Compute the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ in Eqs. (13) and (29).
3. Solve the eigenvalue problem

   $$(\boldsymbol{M} - c\boldsymbol{N})\boldsymbol{u}' = \lambda\boldsymbol{u}', \tag{30}$$

   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

   $$c \leftarrow c + \frac{\lambda}{(\boldsymbol{u}', \boldsymbol{N}\boldsymbol{u}')}, \qquad \boldsymbol{u} \leftarrow \boldsymbol{u}' \tag{31}$$

   and go back to Step 2.

Gauss-Newton Iterations (GN)

Kanatani and Sugaya [11] proposed to minimize Eq. (11) directly by Gauss-Newton iterations. Differentiating Eq. (12) and introducing Gauss-Newton approximation (i.e., ignoring terms that contain $(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)$), we see that the Hessian is simply the matrix $\boldsymbol{M}$ in Eq. (13). In order to enforce the normalization constraint $\|\boldsymbol{u}\| = 1$ in a differential form, we enforce $\boldsymbol{M}$ to have eigenvalue 0 by the projection matrix $\boldsymbol{P}_{\boldsymbol{u}}$ of Eq. (8) and compute pseudoinverse. The procedure goes as follows:

1. Initialize $\boldsymbol{u}$.
2. Compute

   $$\boldsymbol{u}' = N[\boldsymbol{u} - (\boldsymbol{P}_{\boldsymbol{u}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{u}})_5^-(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u}]. \tag{32}$$

3. 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:

Random Choice

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

Least Squares (LS)

Approximating the denominators in Eq. (11) by a constant, we minimize

$$J_{\rm LS} = \frac{1}{2}\sum_{\alpha=1}^{N}(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 = \frac{1}{2}(\boldsymbol{u}, \boldsymbol{M}_{\rm LS}\boldsymbol{u}), \qquad (33)$$

where we define

$$\boldsymbol{M}_{\rm LS} = \sum_{\alpha=1}^{N}\boldsymbol{\xi}_\alpha\boldsymbol{\xi}_\alpha^\top. \qquad (34)$$

Eq. (33) is minimized by the unit eigenvalue $\boldsymbol{u}$ of $\boldsymbol{M}_{\rm LS}$ for the smallest eigenvalue.

Taubin's Method

Replacing the denominators in Eq. (11) by their average, we minimize the following function[3] [16]:

$$J_{\rm TB} = \frac{1}{2}\frac{\sum_{\alpha=1}^{N}(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{\sum_{\alpha=1}^{N}(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} = \frac{1}{2}\frac{(\boldsymbol{u}, \boldsymbol{M}_{\rm LS}\boldsymbol{u})}{(\boldsymbol{u}, \boldsymbol{N}_{\rm TB}\boldsymbol{u})}. \qquad (35)$$

The matrix $\boldsymbol{N}_{\rm TB}$ has the form

$$\boldsymbol{N}_{\rm TB} = \sum_{\alpha=1}^{N} V_0[\boldsymbol{\xi}_\alpha]. \qquad (36)$$

Eq. (35) is minimized by solving the generalized eigenvalue problem

$$\boldsymbol{M}_{\rm LS}\boldsymbol{u} = \lambda\boldsymbol{N}_{\rm TB}\boldsymbol{u} \qquad (37)$$

for the smallest eigenvalue. Since $\boldsymbol{N}_{\rm TB}$ is not positive definite, we decompose $\boldsymbol{\xi}_\alpha$, $\boldsymbol{u}$, and $V_0[\boldsymbol{\xi}_\alpha]$ in the form of Eqs. (19) and define 5×5 matrices $\tilde{\boldsymbol{M}}_{\rm LS}$ and $\tilde{\boldsymbol{N}}_{\rm TB}$ by

$$\tilde{\boldsymbol{M}}_{\rm LS} = \sum_{\alpha=1}^{N}\tilde{\boldsymbol{z}}_\alpha\tilde{\boldsymbol{z}}_\alpha^\top, \qquad \tilde{\boldsymbol{N}}_{\rm TB} = \sum_{\alpha=1}^{N} V_0[\boldsymbol{z}_\alpha], \qquad (38)$$

where

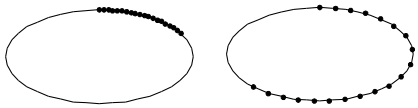
$$\tilde{\boldsymbol{z}}_\alpha = \boldsymbol{z}_\alpha - \bar{\boldsymbol{z}}, \qquad \bar{\boldsymbol{z}} = \frac{1}{N}\sum_{\alpha=1}^{N}\boldsymbol{z}_\alpha. \qquad (39)$$

Then, Eq. (37) splits into two equations

$$\tilde{\boldsymbol{M}}_{\rm LS}\boldsymbol{v} = \lambda\tilde{\boldsymbol{N}}_{\rm TB}\boldsymbol{v}, \qquad (\boldsymbol{v}, \bar{\boldsymbol{z}}) + f_0^2 F_{33} = 0. \qquad (40)$$

We compute the unit eigenvector $\boldsymbol{v}$ of the first equation for the smallest eigenvalue λ. The second equation gives F_{33}, and $\boldsymbol{u}$ is given by Eq. (25).

[3] Taubin [16] did not take the covariance matrix into account. This is a modification of his method.



(a) (b)

Figure 1: 20 points on elliptic arcs. (a) Short arch. (b) Long arc

6 Numerical Examples

Fig. 1 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 fitted an ellipse by FNS, HEIV, renormalization, and GN. For each σ, we plotted the average number of iterations over 1000 independent trials. We stopped when the new value $\boldsymbol{u}'$ differs from the previous value[4] $\boldsymbol{u}$ by $\|\boldsymbol{u}' - \boldsymbol{u}\| < 10^{-6}$.

Doing numerical experiments, we have found that the convergence performance significantly differs depending on whether we use points on a short elliptic arc or on a long elliptic arc.

Fitting to a Short Arc

Figure 2 plots the number of iterations for the short arc in Fig. 1(a). When the iterations did not converge after 100 iterations, we stopped and set the iteration count to 100. We can see that 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.

This can be explained as follows. If the computed $\boldsymbol{u}'$ is close to the true value $\boldsymbol{u}$, the matrix $\boldsymbol{L}$ in Eq. (14) and the matrix $\tilde{\boldsymbol{L}}$ in Eq. (21) are both close to $\boldsymbol{O}$. Initially, however, they may be very different from $\boldsymbol{O}$. Eqs. (15) and (26) are written, respectively, as

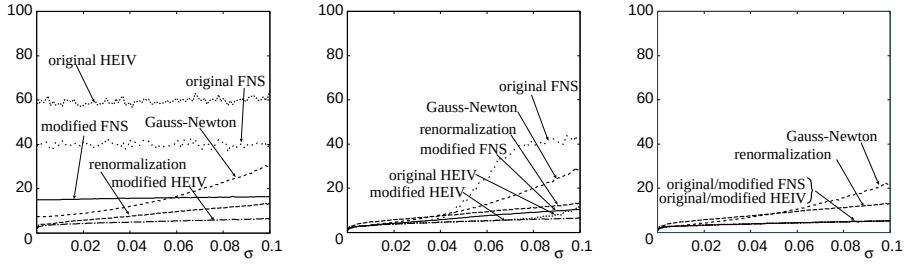
$$(\boldsymbol{M} - \boldsymbol{L} - \lambda\boldsymbol{I})\boldsymbol{u}' = \boldsymbol{0}, \qquad (\tilde{\boldsymbol{M}} - \lambda\tilde{\boldsymbol{L}})\boldsymbol{v}' = \boldsymbol{0}. \quad (41)$$

The matrices $\boldsymbol{L}$ and $\tilde{\boldsymbol{L}}$ are both positive definite. In order that their effects be canceled, 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.

Another finding is that although FNS, HEIV and GN converges faster as we use better initial values, the behavior of renormalization is almost unchanged. This is because we start solving Eq. (30)

[4] Since $\boldsymbol{u}$ and $-\boldsymbol{u}$ represent the same ellipse, we computed the smaller of the two values $\|\boldsymbol{u}' \pm \boldsymbol{u}\|$.



(a) Random initialization (b) LS initialization (c) Taubin initialization

Figure 2: Average number of iterations for ellipse fitting to the points in Fig. 1(a) vs. noise level.

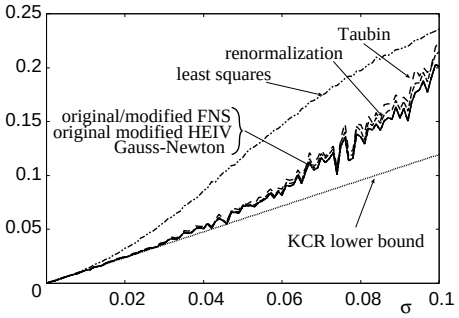


Figure 3: Root-mean-squares error of ellipse fitting to the points in Fig. 1(a) vs. noise level.

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 $\boldsymbol{O}$. Naturally, GN converges faster if started from better initial values.

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 plots for each σ the root-mean-squares of $\|\boldsymbol{P}_{\boldsymbol{u}}\hat{\boldsymbol{u}}\|$ over 1000 independent trials. We compared LS, Taubin's method, and the four iterative methods starting from the Taubin solution. We 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).

From Fig. 3, we can see that Taubin's method is considerably better[5] 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; as noise increases, they gradually deviate from it. Since FNS, HEIV, and GN minimize the same function, the resulting solution is virtually the same. The accuracy of renormalization is also very close to them.

[5]The mechanism of the superiority of Taubin's method over LS is analyzed in detail in [9].

Fitting to a Long Arc

Fig. 4 shows the number of iterations for the long arc in Fig. 1(b). In this case, 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, as compared with Fig. 2, 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 failed to converge only for 24% of the trials.

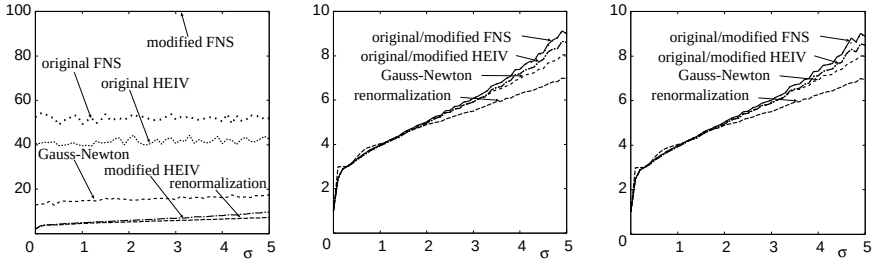
This is related to the singularity of ellipse fitting [2]: Some of the terms on the right-hand side of Eq. (11) diverge to $\pm\infty$. This happens when a data point exists near the center of the current candidate fit, which is more likely to occur when the data points are distributed over a long arc.

As we can see from Fig. 4, renormalization is the most stable for whichever initialization. As we noted earlier, this is because the iterations start from $c = 0$; Eq. (30) yields a value $\boldsymbol{u}'$ close to the LS solution, which is already fairly accurate for a long arc. GN is also stable, because the solution continuously changes in the course of the iterations, while FNS and HEIV may compute oscillating eigenvectors.

Figure 5 compares the accuracy of all the methods in the same way as Fig. 3. As expected, the LS solution, which is usually prone to statistical bias, is as accurate as Taubin's method, because bias is less likely to arise for a long arc. Also, the improvement by the (original or modified) FNS/HEIV, renormalization, and GN is very small. All yields practically the same solution very close to the KCR lower bound.

7 Conclusions

We have studied the convergence behavior of typical iterative numerical schemes for maximal likelihood (ML) of ellipse fitting. After posing the problem in relation to the KCR lower bound, we



(a) Random initialization (b) LS initialization (c) Taubin initialization

Figure 4: Average number of iterations for ellipse fitting to the points in Fig. 1(b) vs. noise level.

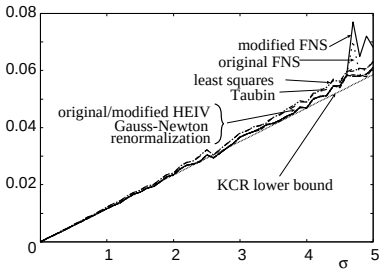


Figure 5: Root-mean-squares error of ellipse fitting to the points in Fig. 1(b) vs. noise level.

described the algorithms of FNS, HEIV, renormalization, and Gauss-Newton iterations (GN). Using simulated image data, we compared their convergence performance.

For a short arc, the modified FNS/HEIV have better convergence properties than the original FNS/HEIV. The convergence of renormalization is little affected by the choice of the initial value. Overall, the modified HEIV is the most efficient.

For a long arc, however, the modified FNS is worse than the original FNS if randomly initialized, and the renormalization is the most efficient. If the iterations converge, however, the fitting accuracy is far higher than for a short arc whichever method is used.

Acknowledgments. The authors thanks Nikolai Chernov of the University of Alabama, U.S.A., and Wojciech Chojnacki of the University of Adelaide, Australia, for helpful discussions. He also thank Yasuyuki Sugaya of Toyohashi University of Technology, Japan, and Junpei Yamada of Okayama University, Japan, for participating in numerical experiments. 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] F.J. Bookstein, Fitting conic sections to scattered data, *Comput. Graphics Image Process.*, **9** (1979), 56–71.

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

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

[4] 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.

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

[6] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.

[7] K. Kanatani, Statistical bias of conic fitting and renormalization, *IEEE Tran. Patt. Anal. Mach. Intell.*, **16**-3 (1994-3), 320–326.

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

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

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

[11] K. Kanatani and Y. Sugaya, High accuracy fundamental matrix computation and its performance evaluation, *Proc. 17th British Machine Vision Conf.*, Sept. 2006, Edinburgh, U.K., Vol. 1, pp. 217–226.

[12] Y. Kanazawa and K. Kanatani, Optimal conic fitting and reliability evaluation, *IEICE Trans. Inf. & Sys.*, **E79-D**-9 (1996-9), 1323–1328.

[13] A. Fitzgibbon, M. Pilu and R.B. Fisher, Direct least square fitting of ellipses, *IEEE Trans. Patt. Anal. Mach. Intell.*, **21**-5 (1999-3), 476–480.

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

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

[16] 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.