On the convergence of fitting algorithms in computer vision

N. Chernov
Department of Mathematics
University of Alabama at Birmingham
Birmingham, AL 35294
chernov@math.uab.edu

Abstract

We investigate several numerical schemes for estimating parameters in computer vision problems: HEIV, FNS, renormalization method, and others. We prove mathematically that these algorithms converge rapidly, provided the noise is small. In fact, in just 1-2 iterations they achieve maximum possible statistical accuracy. Our results are supported by a numerical experiment. We also discuss the performance of these algorithms when the noise increases and/or outliers are present.

1 Introduction

Fitting parametric models to digitized images is a central task in computer vision. Algebraic curve fitting, matching projections of stereo images, estimating coefficients of the epipolar equations [2, 3, 4, 6, 8, 10] fall into this category. In many cases, including the above, the principal equation describing the model takes form

$$\boldsymbol{\Theta}^T \mathbf{u}(\mathbf{x}) = 0. \tag{1}$$

Here $\boldsymbol{\Theta} = [\theta_1, \dots, \theta_l]^T$ is a vector representing unknown parameters; $\mathbf{x} = [x_1, \dots, x_k]^T$ is a vector representing an element of the data (for example, a point in the image); and $\mathbf{u}(\mathbf{x}) = [u_1(\mathbf{x}), \dots, u_l(\mathbf{x})]^T$ is a vector with the data transformed in a problem-dependent manner.

For example, fitting conics to scattered points on the xy plane falls into this scheme. In that case the conic is described by a quadratic equation

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \tag{2}$$

hence $\mathbf{\Theta} = [A, B, C, D, E, F]^T$ is our parameter vector; $\mathbf{x} = [x, y]^T$ is an element of the data; and $\mathbf{u}(\mathbf{x}) = [x^2, xy, y^2, x, y, 1]^T$ is the transformed data vector.

The estimation problem associated with (1) can be stated as follows. Given a collection $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of *image data* find the value of $\mathbf{\Theta} \neq \mathbf{0}$ that minimizes a certain *cost function.* It is commonly assumed that the experimental data $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is a random perturbation of some *true (but unknown) position vectors* $\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_n\}$, i.e.

$$\mathbf{x}_i = \bar{\mathbf{x}}_i + \delta\mathbf{x}_i, \qquad i = 1, \dots, n. \tag{3}$$

The true points $\bar{\mathbf{x}}_i$ satisfy $\bar{\mathbf{\Theta}}^T\mathbf{u}(\bar{\mathbf{x}}_i) = 0$ for $i = 1, \dots, n$, where $\bar{\mathbf{\Theta}}$ is the unknown parameter vector. For example, in the conic fitting problem, $\bar{\mathbf{x}}_i = [\bar{x}_i, \bar{y}_i]^T$ are some 'true' points on the unknown conic. Since the observed data $\mathbf{x}_i$ are corrupted by noise, we can only solve equation (1) approximately.

The noise vector $[\delta\mathbf{x}_1, \dots, \delta\mathbf{x}_n]^T$ is usually assumed to have independent normal components $\delta\mathbf{x}_i$ with zero mean. For simplicity, we let $\delta\mathbf{x}_i = N(\mathbf{0}, \sigma^2\mathbf{I})$, where $\mathbf{I}$ is the $k \times k$ identity matrix (generalization to arbitrary covariance matrices is straightforward). Then the *maximum likelihood* (ML) method consists of minimizing

$$\mathcal{F}_{\text{ML}} = \sum_{i=1}^{n} \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2 \to \min \tag{4}$$

subject to the constraint $\mathbf{\Theta}^T\mathbf{u}(\bar{\mathbf{x}}) = 0$. Thus the true vector $\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_n\}$ needs to be treated as an extra parameter vector (these are *nuisance parameters*). The ML method (4) is equivalent to the minimization of the sum of the orthogonal distances from the data points $\mathbf{x}_i \in \mathbb{R}^k$ to the parametric surface

$$\Sigma_{\mathbf{\Theta}} = \{\mathbf{y} \in \mathbb{R}^k \colon \mathbf{\Theta}^T\mathbf{u}(\mathbf{y}) = 0\},$$

i.e. the maximum likelihood estimate, $\hat{\mathbf{\Theta}}_{\text{ML}}$, can be obtained by solving the *orthogonal least squares problem*

$$\mathcal{F}_{\text{ML}} = \sum_{i=1}^{n} [\,\text{dist}(\mathbf{x}_i, \Sigma_{\mathbf{\Theta}})]^2 \to \min, \tag{5}$$

which allows us to eliminate the nuisance parameters from the picture.

The maximum likelihood method is rather impractical, and the following approximation is commonly used instead. The distances in (5) are of order σ, thus when the noise is small ($\sigma \approx 0$) we have

$$\text{(6)} \qquad \operatorname{dist}(\mathbf{x}_i, \Sigma_{\boldsymbol{\Theta}}) = \frac{|\boldsymbol{\Theta}^T \mathbf{u}(\mathbf{x}_i)|}{\|\boldsymbol{\Theta}^T \nabla_{\mathbf{x}} \mathbf{u}(\mathbf{x}_i)\|} + \mathcal{O}\big([\operatorname{dist}(\mathbf{x}_i, \Sigma_{\boldsymbol{\Theta}})]^2\big),$$

where $\nabla_{\mathbf{x}} \mathbf{u}(\mathbf{y})$ denotes the $l \times k$ matrix of the partial derivatives of the function $\mathbf{x} \mapsto \mathbf{u}(\mathbf{x})$ evaluated at $\mathbf{y}$. Thus, to the leading order in σ, the ML is equivalent to the minimization problem

$$\text{(7)} \qquad \mathcal{F}_{\text{AML}} = \sum_{i=1}^{n} \frac{\boldsymbol{\Theta}^T \mathbf{u}(\mathbf{x}_i) \mathbf{u}(\mathbf{x}_i)^T \boldsymbol{\Theta}}{\boldsymbol{\Theta}^T [\nabla_{\mathbf{x}} \mathbf{u}(\mathbf{x}_i)][\nabla_{\mathbf{x}} \mathbf{u}(\mathbf{x}_i)]^T \boldsymbol{\Theta}} \to \min,$$

which is referred to as *approximate maximum likelihood* (AML) method.

The $l \times l$ matrices

$$\mathbf{A}_i = \mathbf{u}(\mathbf{x}_i)\mathbf{u}(\mathbf{x}_i)^T, \qquad \mathbf{B}_i = [\nabla_{\mathbf{x}} \mathbf{u}(\mathbf{x}_i)][\nabla_{\mathbf{x}} \mathbf{u}(\mathbf{x}_i)]^T$$

do not depend on the parameter $\boldsymbol{\Theta}$, they can be evaluated in advance, so the minimization problem can be written as

$$\text{(8)} \qquad \mathcal{F}_{\text{AML}} = \sum_{i=1}^{n} \frac{\boldsymbol{\Theta}^T \mathbf{A}_i \boldsymbol{\Theta}}{\boldsymbol{\Theta}^T \mathbf{B}_i \boldsymbol{\Theta}} \to \min$$

Note that both numerator and denominator are homogeneous quadratic polynomials in $\boldsymbol{\Theta}$, hence $\mathcal{F}_{\text{AML}}$ is unaffected by rescaling $\boldsymbol{\Theta} \mapsto c\boldsymbol{\Theta}$, so the minimization can be restricted to the unit sphere $\|\boldsymbol{\Theta}\| = 1$. Several efficient numerical algorithms have been developed in the last decade for solving the minimization problem (8), see Section 3.

In general statistics, maximum likelihood estimates are asymptotically efficient (optimal) as $n \to \infty$, in the sense that their variance approaches the Rao-Cramer lower bound. In computer vision, however, it is more appropriate to assume that n stays fixed and characterize estimates in the small noise limit $\sigma \to 0$, see extensive discussions in [8, 9, 10]. Under these conditions all reasonable estimates (including all mentioned in this paper) satisfy $\hat{\boldsymbol{\Theta}} = \bar{\boldsymbol{\Theta}} + \mathcal{O}(\sigma)$, hence their variance-covariance matrix satisfies $\operatorname{Cov}(\hat{\boldsymbol{\Theta}}) = \mathcal{O}(\sigma^2)$. Our estimates are usually biased, but their bias is of the second order $\mathbb{E}(\hat{\mathbf{\Theta}}) - \bar{\mathbf{\Theta}} = \mathcal{O}(\sigma^2)$, where $\mathbb{E}$ denotes the expectation, see a proof in [1]. Therefore the mean square error

$$\mathbb{E}\big[(\hat{\mathbf{\Theta}} - \bar{\mathbf{\Theta}})(\hat{\mathbf{\Theta}} - \bar{\mathbf{\Theta}})^T\big] = \operatorname{Cov}(\hat{\mathbf{\Theta}}) + [\,\operatorname{bias}(\hat{\mathbf{\Theta}})][\,\operatorname{bias}(\hat{\mathbf{\Theta}})]^T$$

is primarily determined by the covariance matrix, and the bias only has a second order effect. For the covariance matrix, a lower bound (an analogue of Cramer-Rao lower bound) is obtained by Kanatani [6, 7] for unbiased estimates and extended to all consistent estimates (including all mentioned here) by Chernov and Lesort [1]. Kanatani-Cramer-Rao lower bound (KCR bound) can be stated as

$$\sigma^{-2}\operatorname{Cov}(\hat{\mathbf{\Theta}}) \geq \operatorname{Cov}_{\min} + \mathcal{O}(\sigma^2), \tag{9}$$

where $\operatorname{Cov}_{\min}$ is a positive semidefinite matrix, for which explicit formulas, in terms of $\bar{\mathbf{\Theta}}$ and $\mathbf{u}(\bar{\mathbf{x}}_i)$, are available [6, 7, 1].

Both ML (4) and AML (8) estimates are statistically optimal in the sense that they satisfy the KCR bound (9). In fact, they are statistically equivalent as their covariance matrices coincide, to the leading order:

$$\operatorname{Cov}(\hat{\mathbf{\Theta}}_{\mathrm{ML}}) = \operatorname{Cov}(\hat{\mathbf{\Theta}}_{\mathrm{AML}}) + \mathcal{O}(\sigma^4).$$

In this paper we investigate practical aspects of numerical schemes for solving the minimization problem (8). We investigate their rates of convergence at small noise ($\sigma \approx 0$) and their stability at large noise.

2 Robust versions of the AML

First, the approximation (6) is only good at small noise and becomes dangerously inaccurate when $\operatorname{dist}(\mathbf{x}_i, \Sigma_{\mathbf{\Theta}})$ is of order one. This may happen when either the noise is large or the data are contaminated by a few outliers. To see the danger, let us examine the matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ in (8).

Both $\mathbf{A}_i$ and $\mathbf{B}_i$ are symmetric and positive semi-definite matrices, but they are usually singular. In fact, $\operatorname{rank}\mathbf{A}_i = 1$ and $\operatorname{rank}\mathbf{B}_i \leq \min\{k, l\}$. For example, in the conic fitting problem $\operatorname{rank}\mathbf{B}_i \leq 2$, while $\mathbf{B}_i$ is a 6×6 matrix. More precisely, the denominator $\mathbf{\Theta}^T\mathbf{B}_i\mathbf{\Theta}$ in (8) vanishes when the point $\mathbf{x}_i$ lies at the center of the conic $\mathbf{\Theta}^T\mathbf{u}(\mathbf{x}) = 0$ (because the quadratic function (2) has either an extremum or a saddle at the conic center).

Practically, a data point $\mathbf{x}_i$ may occur near the conic center if either the noise is large or an outlier is present. Then the corresponding term in (8) ‘blows up’ and becomes dominant, which distracts the minimization procedure. Moreover, it is possible, even at small noise and without outliers, that during an iterative minimization of $\mathcal{F}_{\mathrm{AML}}$ a conic comes up whose center happens to be near one of the data points (we observed such phenomena experimentally), then a similar breakdown occurs.

As a remedy, we propose to modify the function (8) as follows:

$$\mathcal{F}_{\mathrm{AML},\gamma} = \sum_{i=1}^{n} \frac{\mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta}}{\mathbf{\Theta}^T (\mathbf{B}_i + \gamma \mathbf{A}_i)\mathbf{\Theta}} \to \min \tag{10}$$

where $\gamma > 0$ is a properly chosen constant. It is easy to see that this modification gives an estimate, $\hat{\mathbf{\Theta}}_{\mathrm{AML},\gamma}$, statistically equivalent to the AML at small noise, to the leading order. Indeed, as $\mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta} = [\mathbf{\Theta}^T \mathbf{u}(\mathbf{x}_i)]^2 = \mathcal{O}(\sigma^2)$, we obtain

$$\frac{\mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta}}{\mathbf{\Theta}^T (\mathbf{B}_i + \gamma \mathbf{A}_i)\mathbf{\Theta}} = \frac{\mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta}}{\mathbf{\Theta}^T \mathbf{B}_i \mathbf{\Theta}} + \mathcal{O}(\sigma^4)$$

thus $\hat{\mathbf{\Theta}}_{\mathrm{AML},\gamma} = \hat{\mathbf{\Theta}}_{\mathrm{AML}} + \mathcal{O}(\sigma^2)$ and $\mathrm{Cov}(\hat{\mathbf{\Theta}}_{\mathrm{AML},\gamma}) = \mathrm{Cov}(\hat{\mathbf{\Theta}}_{\mathrm{AML}}) + \mathcal{O}(\sigma^4)$. In particular, $\hat{\mathbf{\Theta}}_{\mathrm{AML},\gamma}$ is also statistically optimal (it attains the KCR lower bound), to the leading order in σ.

On the other hand, the modified terms in (10) are uniformly bounded:

$$\frac{\mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta}}{\mathbf{\Theta}^T (\mathbf{B}_i + \gamma \mathbf{A}_i)\mathbf{\Theta}} \leq \frac{1}{\gamma}$$

for *all* $\mathbf{\Theta} \neq \mathbf{0}$, thus the danger of exploding is eliminated.

Our modification can be viewed from a different perspective. The problem (10) is, in fact, a weighted orthogonal least squares procedure

$$\sum_{i=1}^{n} w_i [\,\mathrm{dist}(\mathbf{x}_i, \Sigma_{\mathbf{\Theta}})]^2 \to \min$$

with weights

$$w_i = \frac{\mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta}}{[\,\mathrm{dist}(\mathbf{x}_i, \Sigma_{\mathbf{\Theta}})]^2 \, \mathbf{\Theta}^T (\mathbf{B}_i + \gamma \mathbf{A}_i)\mathbf{\Theta}}.$$

We know, due to (6), that $w_i \approx 1$ whenever $\mathrm{dist}(\mathbf{x}_i, \Sigma_{\mathbf{\Theta}}) \approx 0$, but the behavior of w_i for points $\mathbf{x}_i$ farther away from $\Sigma_{\mathbf{\Theta}}$ is more complicated.

We will illustrate the behavior of w by a simple example. Consider a circle $x^2 + y^2 = R^2$. The distance from a point $\mathbf{x} = (x, y)$ to the circle is $d = \sqrt{x^2 + y^2} - R$ (note that it is positive outside and negative inside). Then, after some elementary calculations, we arrive at

$$w = \frac{(R + d/2)^2}{(R + d)^2 + \gamma d^2 (R + d/2)^2}$$

Figure 1 shows the graph of $w(d)$ for three different values of γ. We see that for the original AML (8), corresponding to $\gamma = 0$, the function $w(d)$ monotonically decreases. It somewhat suppresses points outside the circle (as $w < 1$) and favors those inside it (as $w > 1$). Far away points ($d \to \infty$) are given small but positive weights $w \approx 1/4$. Points near the circle center ($d \approx -R$) are given weights approaching infinity (this is the singularity observed earlier).

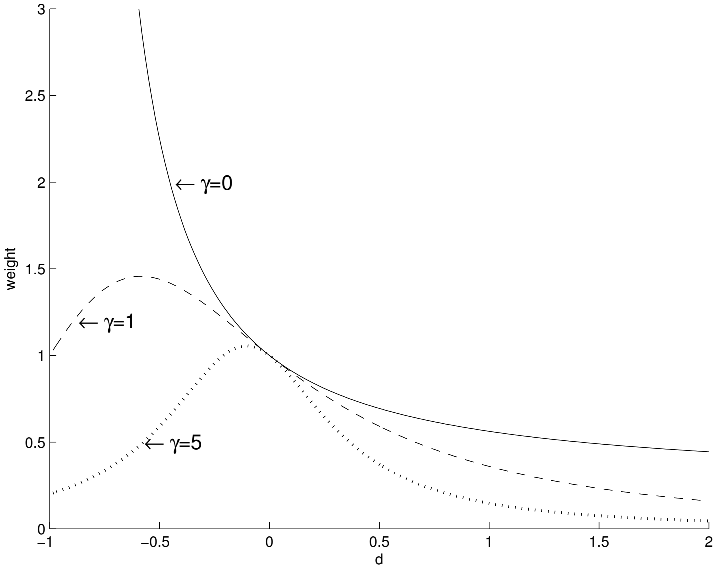

Figure 1: Weight $w(d)$ for $\gamma = 0$, 1, 5. Here we set $R = 1$.

On the contrary, for the modified AML the weight function is bounded and has no singularities. It suppresses distant points both inside and outside the circle. For outside points, the weight function $w(d)$ drops rapidly with d and even vanishes as $d \to \infty$. Points near the circle center also get small weights $w \approx 1/(\gamma R^2)$.

Thus the modified AML (10) has features of a *robust* fit that is able to suppress (filter out) faraway points. This is an unexpected benefit of our attempt to eliminate the dangerous singularity of the original AML.

In the next section we discuss numerical algorithms for computing the AML (8). Here we note that all of them can be used to compute the modified AML (10) as well, since all we need is to redefine the matrix $\mathbf{B}_i$ (which is independent of $\mathbf{\Theta}$).

3 Popular numerical schemes

Several efficient numerical algorithms have been developed in the last decade for computing the AML (8). All of them attempt to solve the *variational equation* $\nabla_{\mathbf{\Theta}}\mathcal{F}_{\mathrm{AML}} = \mathbf{0}$; and direct differentiation of (8) gives

$$\mathbf{M}_{\mathbf{\Theta}}\mathbf{\Theta} - \mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta} = \mathbf{0} \tag{11}$$

where

$$\mathbf{M}_{\mathbf{\Theta}} = \sum_{i=1}^{n} \frac{\mathbf{A}_i}{\mathbf{\Theta}^T\mathbf{B}_i\mathbf{\Theta}}, \qquad \mathbf{L}_{\mathbf{\Theta}} = \sum_{i=1}^{n} \frac{\mathbf{\Theta}^T\mathbf{A}_i\mathbf{\Theta}}{(\mathbf{\Theta}^T\mathbf{B}_i\mathbf{\Theta})^2}\mathbf{B}_i$$

Since the equation (11) has no closed form solution, all algorithms are necessarily iterative: given $\mathbf{\Theta}_m$, they compute $\mathbf{\Theta}_{m+1}$ until the procedure stabilizes (i.e., until $\mathbf{\Theta}_{m+1}$ gets sufficiently close to $\mathbf{\Theta}_m$). We present three most popular algorithms below.

1. Heteroscedastic errors-in-variables (HEIV) method by Leedan and others [13, 14]. In its basic form (see [3, 4]), it computes $\mathbf{\Theta}_{m+1}$ as the unit eigenvector of the generalized eigenvalue problem

$$\mathbf{M}_{\mathbf{\Theta}_m}\mathbf{\Theta} = \lambda_m\mathbf{L}_{\mathbf{\Theta}_m}\mathbf{\Theta} \tag{12}$$

corresponding to the smallest (closest to zero) eigenvalue.[1]

2. Fundamental numerical scheme (FNS) by Chojnacki et al [2, 3, 4] computes $\mathbf{\Theta}_{m+1}$ as the unit eigenvector of the eigenvalue problem

$$(\mathbf{M}_{\mathbf{\Theta}_m} - \mathbf{L}_{\mathbf{\Theta}_m})\mathbf{\Theta} = \lambda_m\mathbf{\Theta} \tag{13}$$

[1] It is known that $\lambda_m \to 1$ provided the algorithm converges at all, see [12], and for this reason some authors suggest to choose the eigenvector $\mathbf{\Theta}_{m+1}$ corresponding to the eigenvalue closets to one (rather than zero); but this may cause divergence, see [12].

corresponding to the smallest (closest to zero) eigenvalue.

3. Renormalization procedure by Kanatani [5, 6, 10], see also [2], computes $\mathbf{\Theta}_{m+1}$ as the unit eigenvector of the eigenvalue problem

$$(\mathbf{M}_{\mathbf{\Theta}_m} - \mu_m \mathbf{N}_{\mathbf{\Theta}_m})\mathbf{\Theta} = \lambda_m \mathbf{\Theta}, \qquad \mathbf{N}_{\mathbf{\Theta}} = \sum_{i=1}^{n} \frac{\mathbf{B}_i}{\mathbf{\Theta}^T \mathbf{B}_i \mathbf{\Theta}} \tag{14}$$

corresponding to the smallest (closest to zero) eigenvalue. Here μ_m is an additional parameter that must be updated at every iteration by

$$\mu_{m+1} = \mu_m + \frac{\lambda_m}{\mathbf{\Theta}_{m+1}^T \mathbf{N}_{\mathbf{\Theta}_m} \mathbf{\Theta}_{m+1}}.$$

All these methods require an initial guess $\mathbf{\Theta}_0$, which is usually supplied by a simple *algebraic fit*

$$\mathcal{F}_{\text{ALG}} = \sum_{i=1}^{n} \frac{\mathbf{\Theta}^T \mathbf{u}(\mathbf{x}_i)\mathbf{u}(\mathbf{x}_i)^T \mathbf{\Theta}}{\mathbf{\Theta}^T \mathbf{\Theta}} = \sum_{i=1}^{n} \frac{\mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta}}{\mathbf{\Theta}^T \mathbf{\Theta}} \to \min \tag{15}$$

The solution of (15) is the unit eigenvector of the matrix $\sum_{i=1}^n \mathbf{A}_i$ corresponding to its smallest eigenvalue. A slightly more accurate initial guess can be obtained by Taubin's fit [16]

$$\mathcal{F}_{\text{TAU}} = \frac{\sum_{i=1}^n \mathbf{\Theta}^T \mathbf{A}_i \mathbf{\Theta}}{\sum_{i=1}^n \mathbf{\Theta}^T \mathbf{B}_i \mathbf{\Theta}} \to \min\,. \tag{16}$$

Note that this is a modification of (8). Taubin's fit is invariant under Euclidean transformations (parallel translations and rotations) while the algebraic fit (15) is not. The solution of (16) is the unit eigenvector of the generalized eigenvalue problem

$$\Big(\sum_{i=1}^{n} \mathbf{A}_i\Big)\mathbf{\Theta} = \lambda_m \Big(\sum_{i=1}^{n} \mathbf{B}_i\Big)\mathbf{\Theta} \tag{17}$$

corresponding to the smallest (closest to zero) eigenvalue. Kanatani's renormalization procedure also requires setting $\mu_0 = 0$.

A very clear exposition of Algorithms 1–3 and their variations, including background ideas and comparison, is given in [2, 3, 4].

Lately a new competing method emerged [10]:

4. Reduced scheme computes $\mathbf{\Theta}_{m+1}$ as the unit eigenvector of the eigenvalue problem

$$\mathbf{M}_{\mathbf{\Theta}_m}\mathbf{\Theta} = \lambda_m \mathbf{\Theta} \tag{18}$$

corresponding to the smallest eigenvalue. Note that the matrices $\mathbf{L}_{\mathbf{\Theta}}$ and $\mathbf{N}_{\mathbf{\Theta}}$ are not used at all and need not be computed.

Technically, the methods 3 and 4 do not solve the variational equation (11). The renormalization procedure 3 computes an approximate solution to (11) that nonetheless satisfies the KCR lower bound [10, Section 15]. The reduced scheme 4 solves the 'reduced' equation

$$\mathbf{M}_{\mathbf{\Theta}}\mathbf{\Theta} = \mathbf{0} \tag{19}$$

whose solution differs from that of (11) by $\mathcal{O}(\sigma^2)$. Indeed, when $\mathbf{\Theta}$ is near either solution, then, as we noted earlier, $\mathbf{\Theta}^T\mathbf{A}_i\mathbf{\Theta} = [\mathbf{\Theta}^T\mathbf{u}(\mathbf{x}_i)]^2 = \mathcal{O}(\sigma^2)$, hence $\mathbf{L}_{\mathbf{\Theta}} = \mathcal{O}(\sigma^2)$ while $\mathbf{M}_{\mathbf{\Theta}} = \mathcal{O}(1)$. Thus Algorithm 4 is statistically equivalent to the AML and satisfies the KCR bound, see also [1, Section 3].

All the above estimates – the ML, AML, and the solution of (19) – are known to have a small bias $\mathcal{O}(\sigma^2)$. While such small bias cannot affect the statistical efficiency of these estimates, certain steps can be taken to reduce it. The full version of Algorithm 1 includes such steps [13, 15], see also [4]. Similar steps can be added to Algorithm 2, see [4]. Algorithm 3 already incorporates the bias removal, and recently Kanatani proposed [10, 11] corrections to Algorithms 2 and 4 that reduce the bias to $\mathcal{O}(\sigma^6)$.

4 Convergence of numerical schemes

The algorithms described in the previous section have been tested by various authors and their excellent performance is well documented [2, 3, 4, 5, 6, 13]. However, there seem to be no attempts made to determine their convergence rates theoretically. One reason for that omission is, probably, a very intricate matrix-type relation between $\mathbf{\Theta}_m$ and $\mathbf{\Theta}_{m+1}$ that obscures the process. Nonetheless, a sort of theoretical analysis is possible and we do it below.

It is not hard to see, to begin with, that all the algorithms in the previous section are variants of the fixed-point scheme. In general, the fixed-point scheme only converge linearly, if at all. However, at small noise all our algorithms converge extremely fast for the reasons explained below.

We first analyze the simplest Algorithm 4. Denote by $\boldsymbol{\Theta}$ the solution of equation (19) and put $\boldsymbol{\Theta}_m = \boldsymbol{\Theta} + \delta\boldsymbol{\Theta}_m$. Then Algorithm 4 consists of solving the eigenvalue problem

$$\mathbf{M}_{\boldsymbol{\Theta}+\delta\boldsymbol{\Theta}_m}(\boldsymbol{\Theta} + \delta\boldsymbol{\Theta}_{m+1}) = \lambda_m(\boldsymbol{\Theta} + \delta\boldsymbol{\Theta}_{m+1}).$$

Taylor expansion of the matrix $\mathbf{M}_{\boldsymbol{\Theta}+\delta\boldsymbol{\Theta}_m}$ gives

$$(\mathbf{M}_{\boldsymbol{\Theta}} + \delta\mathbf{M}_m)(\boldsymbol{\Theta} + \delta\boldsymbol{\Theta}_{m+1}) = \lambda_m(\boldsymbol{\Theta} + \delta\boldsymbol{\Theta}_{m+1}), \tag{20}$$

where

$$\delta\mathbf{M}_m = -2\sum_i \frac{\mathbf{A}_i(\boldsymbol{\Theta}^T\mathbf{B}_i\,\delta\boldsymbol{\Theta}_m)}{(\boldsymbol{\Theta}^T\mathbf{B}_i\boldsymbol{\Theta})^2} + \mathcal{O}(\|\delta\boldsymbol{\Theta}_m\|^2).$$

To the leading order, we get

$$\mathbf{M}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_{m+1} - \mathbf{K}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_m = \lambda_m\boldsymbol{\Theta}, \qquad \mathbf{K}_{\boldsymbol{\Theta}} = 2\sum_i \frac{\mathbf{A}_i\boldsymbol{\Theta}\boldsymbol{\Theta}^T\mathbf{B}_i}{(\boldsymbol{\Theta}^T\mathbf{B}_i\boldsymbol{\Theta})^2}. \tag{21}$$

Recall that $|\boldsymbol{\Theta}^T\mathbf{u}(\mathbf{x}_i)| = \mathcal{O}(\sigma)$, hence $\|\mathbf{A}_i\boldsymbol{\Theta}\| = \mathcal{O}(\sigma)$. This implies $\mathbf{K}_{\boldsymbol{\Theta}} = \mathcal{O}(\sigma)$. Denote by $\boldsymbol{\Pi}_{\|} = \boldsymbol{\Theta}\boldsymbol{\Theta}^T$ the projection onto the line span$\{\boldsymbol{\Theta}\}$ and by $\boldsymbol{\Pi}_{\perp} = \mathbf{I} - \boldsymbol{\Theta}\boldsymbol{\Theta}^T$ the projection onto the orthogonal hyperplane. Then (21) can be rewritten as

$$\mathbf{M}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_{m+1} - \boldsymbol{\Pi}_{\perp}\mathbf{K}_{\boldsymbol{\Theta}}(\delta\boldsymbol{\Theta}_m) - \boldsymbol{\Pi}_{\|}\mathbf{K}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_m = \lambda_m\boldsymbol{\Theta}.$$

The first two vectors are orthogonal to $\boldsymbol{\Theta}$, while the last two are parallel to $\boldsymbol{\Theta}$. Therefore $-\boldsymbol{\Pi}_{\|}\mathbf{K}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_m = \lambda_m\boldsymbol{\Theta}$ and $\mathbf{M}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_{m+1} = \boldsymbol{\Pi}_{\perp}\mathbf{K}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_m$, thus

$$\delta\boldsymbol{\Theta}_{m+1} = \mathbf{M}_{\boldsymbol{\Theta}}^{-}\mathbf{K}_{\boldsymbol{\Theta}}\,\delta\boldsymbol{\Theta}_m, \tag{22}$$

where $\mathbf{M}_{\boldsymbol{\Theta}}^{-}$ denotes the generalized (Moore-Penrose) inverse of $\mathbf{M}_{\boldsymbol{\Theta}}$ (we can omit the matrix $\boldsymbol{\Pi}_{\perp}$ because it is absorbed by $\mathbf{M}_{\boldsymbol{\Theta}}^{-}$).[2]

Now equation (22) implies a striking fact:

$$\|\delta\boldsymbol{\Theta}_{m+1}\| = \mathcal{O}\big(\sigma\|\delta\boldsymbol{\Theta}_m\|\big). \tag{23}$$

That is, at every iteration the error gets smaller by a factor of $\mathcal{O}(\sigma)$. This is consistent with the linear convergence pattern, of course, but the small factor here makes the convergence remarkably swift.

[2] Kanatani informed me that (22) may also be derived from a general perturbation theorem, see [6, Section2.2.6].

Some further conclusions can be drawn. Since $|\mathbf{\Theta}^T\mathbf{A}_i\mathbf{\Theta}| = \mathcal{O}(\sigma^2)$, the simple algebraic fit (15) supplies a value $\mathbf{\Theta}_0$ satisfying $\mathbf{\Theta}_0^T\mathbf{A}_i\mathbf{\Theta}_0 = \mathcal{O}(\sigma^2)$, therefore $\delta\mathbf{\Theta}_0 = \mathcal{O}(\sigma)$. Consequently, $\delta\mathbf{\Theta}_1 = \mathcal{O}(\sigma^2)$, thus the first iteration $\mathbf{\Theta}_1$ already produces an estimate statistically equivalent to the AML and hence statistically optimal (i.e. attaining Kanatani-Cramer-Rao lower bound, to the leading order).

Furthermore, if one aims at the removal of a small bias to achieve a 'hyperaccurate' estimate, see the previous section, then one needs just one more iteration, because $\delta\mathbf{\Theta}_2 = \mathcal{O}(\sigma^3)$.

The analysis of Algorithms 1 and 2 goes along the same lines, only a few extra terms appear in some formulas.

In the case of Algorithm 2, the $\mathbf{K}_{\mathbf{\Theta}}$ matrix takes form

$$\mathbf{K}_{\mathbf{\Theta}} = 2\sum_i \frac{\mathbf{A}_i\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{B}_i + \mathbf{B}_i\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{A}_i}{(\mathbf{\Theta}^T\mathbf{B}_i\mathbf{\Theta})^2} - 4\sum_i \frac{\mathbf{B}_i\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{A}_i\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{B}_i}{(\mathbf{\Theta}^T\mathbf{B}_i\mathbf{\Theta})^3}. \tag{24}$$

This is a longer expression than (21), but the key property $\mathbf{K}_{\mathbf{\Theta}} = \mathcal{O}(\sigma)$ remains valid. So the main conclusion (23) holds.

In Algorithm 1, the eigenvalue λ_m is actually close to 1, rather than 0, and we replace it by $\lambda_m = 1 + \varepsilon_m$. Then we obtain, to the leading order,

$$(\mathbf{M}_{\mathbf{\Theta}} - \mathbf{L}_{\mathbf{\Theta}})\delta\mathbf{\Theta}_{m+1} = \mathbf{K}_{\mathbf{\Theta}}\delta\mathbf{\Theta}_m + \varepsilon_m\mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta}, \tag{25}$$

where $\mathbf{K}_{\mathbf{\Theta}}$ is again given by (24). Let $V_\perp$ denote the hyperplane in $\mathbb{R}^l$ orthogonal to $\mathbf{\Theta}$. Let $\mathbf{\Pi}_\| = \mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta}\mathbf{\Theta}^T/\mathbf{\Theta}^T\mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta}$ denote the projection onto the line span$\{\mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta}\}$ parallel to $V_\perp$ and $\mathbf{\Pi}_\perp = \mathbf{I} - \mathbf{\Pi}_\|$ the projection onto $V_\perp$ parallel to $\mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta}$. Then (25) can be rewritten as

$$(\mathbf{M}_{\mathbf{\Theta}} - \mathbf{L}_{\mathbf{\Theta}})\,\delta\mathbf{\Theta}_{m+1} = \mathbf{\Pi}_\perp\mathbf{K}_{\mathbf{\Theta}}\,\delta\mathbf{\Theta}_m + \mathbf{\Pi}_\|\mathbf{K}_{\mathbf{\Theta}}\,\delta\mathbf{\Theta}_m + \varepsilon_m\mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta}.$$

The first two vectors here lie in $V_\perp$ and the other two are parallel to $\mathbf{L}_{\mathbf{\Theta}}\mathbf{\Theta}$. Therefore, $(\mathbf{M}_{\mathbf{\Theta}} - \mathbf{L}_{\mathbf{\Theta}})\,\delta\mathbf{\Theta}_{m+1} = \mathbf{\Pi}_\perp\mathbf{K}_{\mathbf{\Theta}}\,\delta\mathbf{\Theta}_m$, hence

$$\delta\mathbf{\Theta}_{m+1} = (\mathbf{M}_{\mathbf{\Theta}} - \mathbf{L}_{\mathbf{\Theta}})^{-}\mathbf{\Pi}_\perp\mathbf{K}_{\mathbf{\Theta}}\,\delta\mathbf{\Theta}_m.$$

We again arrive at the main formula (23).

Algorithm 3 is somewhat more involved, because of the extra parameter μ_m, so we leave it out.

Our analysis demonstrates that all the practical algorithms for computing the AML (or its modified version (18)) enjoy 'ballistic' convergence, provided the noise is small; in fact, after just one or two iterations they achieve maximal possible statistical accuracy. This perhaps explains phenomenal success of these numerical schemes.

Lastly we make a remark on the global aspects of our numerical algorithms. Suppose the noise is still small ($\sigma \sim 0$), but the initial guess is chosen poorly, so that $\|\delta\mathbf{\Theta}_0\| \gg \sigma$ (this may happen due to an outlier, for example). In some extreme cases, the initial guess may be picked randomly, resulting in $\|\delta\mathbf{\Theta}_0\| = \mathcal{O}(1)$.

Recall that $\mathbf{\Theta}^T\mathbf{A}_i\mathbf{\Theta} = \mathcal{O}(\sigma^2)$. Thus for any choice of $\mathbf{\Theta}_0$ we have $\mathbf{\Theta}^T\mathbf{M}_{\mathbf{\Theta}_0}\mathbf{\Theta} = \mathcal{O}(\sigma^2)$, hence the symmetric positive definite matrix $\mathbf{M}_{\mathbf{\Theta}_0}$ always has a small eigenvalue (of order σ^2), and the corresponding eigenvector must be within the distance $\mathcal{O}(\sigma)$ from $\mathbf{\Theta}$. But the matrix $\mathbf{L}_{\mathbf{\Theta}_0}$ need not be small, all its eigenvalues may be of order one.

Our algorithms react to this new challenge quite differently. Algorithm 4 will surely find the eigenvector of $\mathbf{M}_{\mathbf{\Theta}_0}$ corresponding to the smallest eigenvalue, resulting in $\|\delta\mathbf{\Theta}_1\| = \mathcal{O}(\sigma)$ (and subsequently it will converge in 1-2 iterations). Algorithms 1 and 3 will also find that vector, approximately, as choosing λ small will allow them to suppress the 'bad' matrix $\mathbf{L}_{\mathbf{\Theta}_0}$ (we also remind the reader that $\mu_0 = 0$ in Algorithm 3).

On the contrary, Algorithm 2 may be distracted, as it does not suppress the 'bad' matrix $\mathbf{L}_{\mathbf{\Theta}_0}$; in fact subtracting it from the 'good' matrix $\mathbf{M}_{\mathbf{\Theta}_0}$ may destroy the latter. We indeed observed, experimentally, that Algorithm 2 tends to wander aimlessly for 50-100 iterations or more when seeded with a random vector $\mathbf{\Theta}_0$. Algorithms 1 and 4, on the other hand, always converge within 3-4 iterations, for whatever choice of $\mathbf{\Theta}_0$.

For a similar reason, Algorithm 2 becomes somewhat unreliable when the noise is not-so-small. While both matrices $\mathbf{M}_{\mathbf{\Theta}}$ and $\mathbf{L}_{\mathbf{\Theta}}$ are positive semi-definite, the difference $\mathbf{M}_{\mathbf{\Theta}} - \mathbf{L}_{\mathbf{\Theta}}$ may have negative eigenvalues. It may happen that the smallest eigenvalue of $\mathbf{M}_{\mathbf{\Theta}}$ (representing the correct fit) may be transformed into a negative eigenvalue of $\mathbf{M}_{\mathbf{\Theta}} - \mathbf{L}_{\mathbf{\Theta}}$, while some larger positive eigenvalue of $\mathbf{M}_{\mathbf{\Theta}}$ (which is of no significance for the fit) may be transformed into a small positive eigenvalue of $\mathbf{M}_{\mathbf{\Theta}} - \mathbf{L}_{\mathbf{\Theta}}$. Then Algorithm 2 will pick the latter and miss the former.

To fix this problem, the authors of Algorithm 2 recently proposed [4, Section 5.6] a *stable version of FNS*, where at every iteration the *smallest* eigenvalue is chosen, instead of the one closest to zero. This modification significantly improves the performance of Algorithm 2 at a not-so-small noise, see the next section.

5 Experiment

To compare the performance of different numerical algorithms we test them on a conic fitting problem (2). As a 'true' curve, we chose an ellipse, $(x/5)^2 + y^2 = 1$. The 'true' $n = 20$ points were equally spaced on the right upper quarter ($x > 0, y > 0$) of this ellipse.

To test the numerical schemes, we added a random 2D gaussian noise $N(\mathbf{0}, \sigma^2\mathbf{I})$ to the true points, and varied σ from 0 to 0.005. For each value of σ, we generated $M = 10000$ random samples and for each algorithm we computed the root-mean-square error

$$\text{RMSE} = \left[\frac{1}{M}\sum_{t=1}^{M} \|\hat{\mathbf{\Theta}}_t - \mathbf{\Theta}\|^2\right]^{1/2}$$

where $\hat{\mathbf{\Theta}}_t$ denotes the estimate obtained in the tth realization of the noise. Since RMSE is asymptotically proportional to σ, we plot the ratio RMSE/σ in Figure 2. Labels are assigned as follows:

A – Algebraic fit, Eq. (15)
T – Taubin's fit, Eq. (16)
H – HEIV method, Eq. (12)
F – FNS, Eq. (13)
S – Stable version of the FNS, see Section 4
N – Renormalization scheme, Eq. (14)
R – Reduced scheme, Eq. (18)

We observed that all the iterative schemes H, F, S, N, R stabilize after 1-2 iterations, and further iterations do not change their mean errors in any significant way (this is in perfect agreement with our conclusions in the previous section).

Now we comment on the graphs shown in Figure 2. Asymptotically, as $\sigma \to 0$, all the iterative schemes converge the the same value 0.40, which corresponds to the Kanatani-Cramer-Rao lower bound. The algebraic methods A and T converge to a higher value 0.53, demonstrating their consistent inefficiency.

As the noise increases, Taubin's fit and the HEIV algorithm display a remarkable durability, while other methods slacken sooner or later. The advantage of the stable version of the FNS over its original version is obvious.

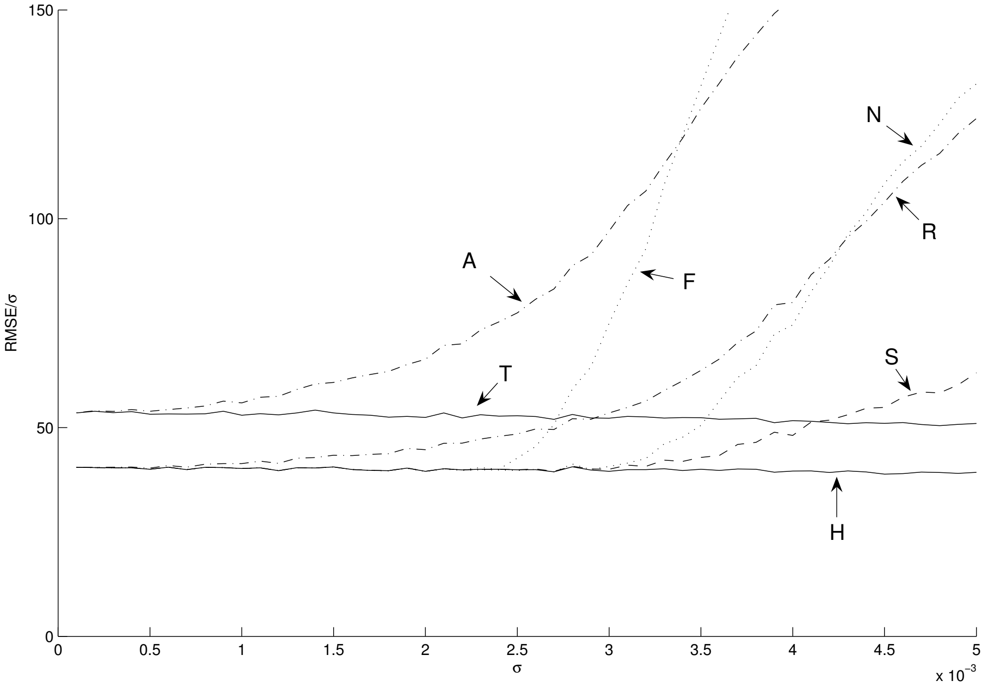

Figure 2: The root-mean-square error RMSE (normalized by σ) versus the noise level σ.

We seeded the iterative algorithms H, F, S, N, R with the estimate obtained by the algebraic fit A, as it is common in such experiments and recommended in other papers [2, 3, 4, 10]. However, when we seeded them with the more reliable Taubin's fit, then all of them performed almost identically to the HEIV (the flat line H in Figure 2), deviating from it by 10% at most. Since Taubin's fit is just as easily computable as the algebraic fit, we believe it should be used as a seed.

We have not used the 'robust' modification of the $\mathbf{B}_i$ matrices, which was described Section 2. There was no need for that, indeed, as our samples were not contaminated by outliers. When we tried the robust version of the $\mathbf{B}_i$ matrices (with $\gamma = 5$), the improvement was insignificant (except the stable version of the FNS became 5-10% more accurate at large noise).

Overall, our experimental results agreed well with our theoretical conclusions. A rather unexpected (and unexplained) phenomenon was a somewhat poor performance of the renormalization and reduced schemes, (14) and (18), at large noise. Interestingly, both methods avoid using the matrix $\mathbf{L}_{\boldsymbol{\Theta}}$ and their first iteration $\boldsymbol{\Theta}_1$ is identical (indeed, recall that $\mu_0 = 0$ in (14)). To investigate their failures, we looked closely at several examples and observed that the matrix $\mathbf{M}_{\boldsymbol{\Theta}_0}$ had two nearly equal small eigenvalues, one corresponding to the correct fit and the other totally wrong. In these cases the reduced and renormalization methods were confused and often picked the wrong one, while the HEIV algorithm and the FNS managed to pick the right one. Perhaps the matrix $\mathbf{L}_{\boldsymbol{\Theta}}$ serves as a 'guide' in such ambiguous cases.

Acknowledgement. The author benefited from electronic discussions with K. Kanatani, W. Chojnacki, and M. J. Brooks. The author was partially supported by NSF grant DMS-0354775.

References

[1] N. Chernov and C. Lesort, "Statistical efficiency of curve fitting algorithms", *Computational Statistics and Data Analysis*, Vol. 47, pp. 713–728, 2004.

[2] W. Chojnacki, M. J. Brooks, and A. van den Hengel, "Rationalising the renormalisation method of Kanatani", *Journal of Mathematical Imaging and Vision*, Vol. 14, pp. 21–38, 2001.

[3] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley, "From FNS to HEIV: A link between two vision parameter estimation methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, pp. 264–268, 2004.

[4] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley, "FNS, CFNS and HEIV: A unifying approach", *Journal of Mathematical Imaging and Vision*, Vol. 23, pp. 175–183, 2005.

[5] K. Kanatani, "Statistical bias of conic fitting and renormalization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 320–326, 1994.

[6] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, 1996.

[7] K. Kanatani, "Cramer-Rao lower bounds for curve fitting", *Graphical Models and Image Processing*, Vol. 60, pp. 93–99, 1998.

[8] K. Kanatani, "Uncertainty modeling and model selection for geometric inference", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, pp. 1307–1319, 2004.

[9] K. Kanatani, "For geometric inference from images, what kind of statistical model is necessary?" *Systems and Computers in Japan*, Vol. 35, pp. 1–9, 2004.

[10] K. Kanatani, "Further improving geometric fitting", In *Proceedings 5th International Conference on 3-D Digital Imaging Modeling*, 2005, Ottawa, Canada, pp. 2–13.

[11] K. Kanatani, private communication.

[12] Y. Leedan, Statistical analysis of quadratic problems in computer vision, Ph.D. thesis, ECE Department, Rutgers Univ., May 1997.

[13] Y. Leedan and P. Meer, "Heteroscedastic regression in computer vision: Problems with bilinear constraint", *Intern. J. Comp. Vision*, Vol. 37, pp. 127–150, 2000.

[14] B. Matei and P. Meer, "A General Method for Errors-in-Variables Problems in Computer Vision", In *Proceedings, CVPR 2000, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, 2000, IEEE Computer Society Press: Los Alamitos, CA 2000, Vol. 2, 18–25.

[15] B. Matei and P. Meer, "Reduction of Bias in Maximum Likelihood Ellipse Fitting", In *Proceedings, 15th International Conference on Computer Vision and Pattern Recognition*, 2000, Barcelona, Spain, Vol. 3, 802–806.

[16] G. Taubin, "Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined by Implicit Equations, with Applications to Edge and Range Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 1115–1138, 1991.