Proc. 3rd Pacific Rim Symp. Image and Video Technology, January 2009,
Tokyo, Japan, pp. 179–190 (Best Paper Award).

Compact Fundamental Matrix Computation

Kenichi Kanatani[1] and Yasuyuki Sugaya[2]

[1] Department of Computer Science, Okayama University, Okayama 700-8530 Japan
[2] Department of Information and Computer Sciences,
Toyohashi University of Technology, Toyohashi, Aichi 441-8580 Japan
kanatani@suri.cs.okayama-u.ac.jp, sugaya@iim.ics.tut.ac.jp

Abstract. A very compact algorithm is presented for fundamental matrix computation from point correspondences over two images. The computation is based on the strict maximum likelihood (ML) principle, minimizing the reprojection error. The rank constraint is incorporated by the EFNS procedure. Although our algorithm produces the same solution as all existing ML-based methods, it is probably the most practical of all, being small and simple. By numerical experiments, we confirm that our algorithm behaves as expected.

1 Introduction

Computing the fundamental matrix from point correspondences is the first step of many vision applications including camera calibration, image rectification, structure from motion, and new view generation [6, 21]. Although its robustness is critical in practice, procedures for removing outlying matches heavily depend on computation for assumed inliers, e.g., RANSAC-type hypothesis-based computation followed by choosing the solution that has maximum support [6, 21]. In this paper, we focus on computation assuming inliers.

Since extracted feature points have uncertainty to some degree, we need statistical optimization, modeling the uncertainty as "noise" obeying a certain probability distribution. The standard model is independent Gaussian noise coupled with maximum likelihood (ML) estimation. This results in the minimization of the "reprojection error", also known as the "Gold Standard" [6].

Although all existing ML-based methods minimize the same function, vast differences exist in their computational processes. This is mainly due to the fact that the fundamental matrix is constrained to have rank 2. The strategies for incorporating this constraint are roughly classified into three categories:

A posteriori correction. The fundamental matrix is first computed without considering the rank constraint and is modified a posteriori so as to satisfy it (Fig. 1(a)). If the rank constraint is not considered, the computation is vastly simplified [6, 21]. The crudest method, yet widely used, is to minimize the square sum of the epipolar equation, called *least squares*, *algebraic distance minimization*, or *8-point algorithm* [5]. The *Taubin method* [20] incorporates the data covariance matrices in the simplest way. These two yield

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 179–190, 2009.

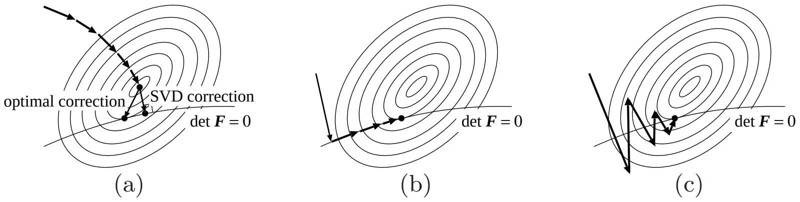


Fig. 1. (a) A posteriori correction. (b) Internal access. (c) External access.

the solution with simple algebraic manipulations [12, 9]. For incorporating the ML viewpoint, one needs iterations, for which many schemes exist including *FNS* [3], *HEIV* [15, 16], and the *projective Gauss-Newton iterations* [10]. For imposing the rank constraint, the most naive method, yet widely used, is to compute the SVD of the computed fundamental matrix and replace the smallest singular value by 0 [5]. A more sophisticated method is the *optimal correction* [8, 16]: the computed fundamental matrix is moved in the statistically mostly likely direction until it satisfies the rank constraint (Fig. 1(a)).

Internal access. The fundamental matrix is parameterized so that the rank constraint is identically satisfied and is optimized in the ("internal") parameter space (Fig. 1(b)). Many types of such parameterization have been proposed including algebraic elimination of the rank constraint and the expression in terms of epipoles [21, 17, 22]. Bartoli and Sturm [1] regarded the SVD of the fundamental matrix as its parameterization and do search in an augmented space. Sugaya and Kanatani [18] directly searched a 7-D space by the Levenberg-Marquardt (LM) method.

External access. We do iterations in the ("external") 9-D space of the fundamental matrix in such a way that an optimal solution that satisfies the rank constraint automatically results (Fig. 1(c)). This concept was first introduced by Chojnacki et al. [4], who presented a scheme called *CFNS*.

In this paper, we present a new method based on the external access principle. Its description is far more compact than any of existing ML-based methods. Although there is no accuracy gain, since all ML-based methods minimize the same function, the compactness of the algorithm is of great advantage. In fact, the reason why the non-optimal 8-point algorithm [5] is still in wide use is probably for fear of coding a complicated program and uneasiness at relying on "download". One algorithm is simple enough to code oneself[1], consisting only of vector and matrix operations in no higher than 9-D, just like the popular 8-point algorithm, yet producing an optimal solution.

We describe our algorithm in Sec. 2 and give a derivation in Sec. 3. In Sec. 4, we confirm its performance by numerical experiments. We conclude in Sec. 5 that our algorithm best suits practical use because of its compactness and good performance.

[1] But one can try ours if one wishes: `http://www.iim.ics.tut.ac.jp/~sugaya/public-e.html`

2 Optimal Fundamental Matrix Computation

Given two images of the same scene, suppose a point (x, y) in the first image corresponds to (x', y') in the second. We represent the corresponding points by 3-D vectors

$$\boldsymbol{x} = \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{x}' = \begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix}, \tag{1}$$

where f_0 is a scaling constant of the order of the image size[2]. As is well known, $\boldsymbol{x}$ and $\boldsymbol{x}'$ satisfy the *epipolar equation*,

$$(\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}') = 0, \tag{2}$$

where and hereafter we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$. The matrix $\boldsymbol{F}$ is of rank 2 and called the *fundamental matrix*. Since its scale is indeterminate, we normalize it to unit Frobenius norm $\|\boldsymbol{F}\| = 1$.

Suppose N correspondence pairs $\{\boldsymbol{x}_\alpha,\ \boldsymbol{x}'_\alpha\}_{\alpha=1}^N$ are detected. If the noise in their x- and y-coordinates is assumed to be independent, identical, and Gaussian, maximum likelihood (ML) is equivalent to minimizing the *reprojection error*

$$E = \sum_{\alpha=1}^{N} \Big(\|\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha\|^2 + \|\boldsymbol{x}'_\alpha - \bar{\boldsymbol{x}}'_\alpha\|^2 \Big), \tag{3}$$

with respect to $\bar{\boldsymbol{x}}_\alpha$, $\bar{\boldsymbol{x}}'_\alpha$, and $\boldsymbol{F}$ subject to

$$(\bar{\boldsymbol{x}}_\alpha, \boldsymbol{F}\bar{\boldsymbol{x}}'_\alpha) = 0, \qquad \alpha = 1, ..., N. \tag{4}$$

No simple procedure exists for minimizing (3) subject to (4) and the rank constraint on $\boldsymbol{F}$. Many researchers minimized the "Sampson error" (to be discussed later) that approximates (3) [6, 21]. Alternatively, the minimization is done in an "augmented" parameter space, as done by Bartoli and Sturm [1], computing a tentative 3-D reconstruction and adjusting the camera positions and the intrinsic parameters so that the resulting projection images are as close to the input images as possible. Such a strategy is called *bundle adjustment*. Search in a high dimensional space, in particular if one wants a globally optimal solution, requires a large amount of computation [7].

We now present a dramatically compact formulation: *we work in 9-D throughout*. Define 9-D vectors

$$\boldsymbol{u} = \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}, \qquad \boldsymbol{u}^\dagger \equiv \mathcal{N}[\begin{pmatrix} u_5u_9 - u_8u_6 \\ u_6u_7 - u_9u_4 \\ u_4u_8 - u_7u_5 \\ u_8u_3 - u_2u_9 \\ u_9u_1 - u_3u_7 \\ u_7u_2 - u_1u_8 \\ u_2u_6 - u_5u_3 \\ u_3u_4 - u_6u_1 \\ u_1u_5 - u_4u_2 \end{pmatrix}], \tag{5}$$

[2] This is for stabilizing numerical computation [5]. In our experiments, we set $f_0 =$ 600 pixels.

where $\mathcal{N}[\,\cdot\,]$ denotes normalization to unit norm. The vector $\boldsymbol{u}$ encodes the nine elements of the fundamental matrix $\boldsymbol{F}$. The normalization $\|\boldsymbol{F}\| = 1$ is equivalent to $\|\boldsymbol{u}\| = 1$. The vector $\boldsymbol{u}^\dagger$ encodes the nine elements of the cofactor $\boldsymbol{F}^\dagger$ of $\boldsymbol{F}$, so we call $\boldsymbol{u}^\dagger$ the "cofactor vector" of $\boldsymbol{u}$. We denote by "$\det \boldsymbol{u}$" the determinant of the matrix $\boldsymbol{F}$ corresponding to $\boldsymbol{u}$.

In order to emphasize the compactness of our algorithm, we state it first and then give its derivation, which is straightforward but rather lengthy. The main routine of our algorithm goes as follows:

main

```
1. Let $\boldsymbol{u}_0 = \boldsymbol{0}$, and initialize $\boldsymbol{u}$.
2. Let $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, $\hat{x}'_\alpha = x'_\alpha$, $\hat{y}'_\alpha = y'_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = \tilde{x}'_\alpha = \tilde{y}'_\alpha = 0$.
3. Compute the following 9-D vectors $\boldsymbol{\xi}_\alpha$ and the $9 \times 9$ matrices $V_0[\boldsymbol{\xi}_\alpha]$:

   $$\boldsymbol{\xi}_\alpha = \begin{pmatrix} \hat{x}_\alpha \hat{x}'_\alpha + \hat{x}'_\alpha \tilde{x}_\alpha + \hat{x}_\alpha \tilde{x}'_\alpha \\ \hat{x}_\alpha \hat{y}'_\alpha + \hat{y}'_\alpha \tilde{x}_\alpha + \hat{x}_\alpha \tilde{y}'_\alpha \\ f_0(\hat{x}_\alpha + \tilde{x}_\alpha) \\ \hat{y}_\alpha \hat{x}'_\alpha + \hat{x}'_\alpha \tilde{y}_\alpha + \hat{y}_\alpha \tilde{x}'_\alpha \\ \hat{y}_\alpha \hat{y}'_\alpha + \hat{y}'_\alpha \tilde{y}_\alpha + \hat{y}_\alpha \tilde{y}'_\alpha \\ f_0(\hat{y}_\alpha + \tilde{y}_\alpha) \\ f_0(\hat{x}'_\alpha + \tilde{x}'_\alpha) \\ f_0(\hat{y}'_\alpha + \tilde{y}'_\alpha) \\ f_0^2 \end{pmatrix}, \tag{6}$$

   $$V_0[\boldsymbol{\xi}_\alpha] = \begin{pmatrix} \hat{x}_\alpha^2 + \hat{x}_\alpha'^2 & \hat{x}'_\alpha \hat{y}'_\alpha & f_0 \hat{x}'_\alpha & \hat{x}_\alpha \hat{y}_\alpha & 0 & 0 & f_0 \hat{x}_\alpha & 0 & 0 \\ \hat{x}'_\alpha \hat{y}'_\alpha & \hat{x}_\alpha^2 + \hat{y}_\alpha'^2 & f_0 \hat{y}'_\alpha & 0 & \hat{x}_\alpha \hat{y}_\alpha & 0 & 0 & f_0 \hat{x}_\alpha & 0 \\ f_0 \hat{x}'_\alpha & f_0 \hat{y}'_\alpha & f_0^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{x}_\alpha \hat{y}_\alpha & 0 & 0 & \hat{y}_\alpha^2 + \hat{x}_\alpha'^2 & \hat{x}'_\alpha \hat{y}'_\alpha & f_0 \hat{x}'_\alpha & f_0 \hat{y}_\alpha & 0 & 0 \\ 0 & \hat{x}_\alpha \hat{y}_\alpha & 0 & \hat{x}'_\alpha \hat{y}'_\alpha & \hat{y}_\alpha^2 + \hat{y}_\alpha'^2 & f_0 \hat{y}'_\alpha & 0 & f_0 \hat{y}_\alpha & 0 \\ 0 & 0 & 0 & f_0 \hat{x}'_\alpha & f_0 \hat{y}'_\alpha & f_0^2 & 0 & 0 & 0 \\ f_0 \hat{x}_\alpha & 0 & 0 & f_0 \hat{y}_\alpha & 0 & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 \hat{x}_\alpha & 0 & 0 & f_0 \hat{y}_\alpha & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{7}$$

4. Call *EFNS* to update $\boldsymbol{u}$.
5. If $\boldsymbol{u} \approx \boldsymbol{u}_0$ up to sign, return $\boldsymbol{u}$ and stop. Else, update $\tilde{x}_\alpha$, $\tilde{y}_\alpha$, $\tilde{x}'_\alpha$, and $\tilde{y}'_\alpha$ by

   $$\begin{pmatrix} \tilde{x}_\alpha \\ \tilde{y}_\alpha \end{pmatrix} \leftarrow \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)}{(\boldsymbol{u}, V[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{u})} \begin{pmatrix} u_1 & u_2 & u_3 \\ u_4 & u_5 & u_6 \end{pmatrix} \begin{pmatrix} \hat{x}'_\alpha \\ \hat{y}'_\alpha \\ f_0 \end{pmatrix},$$
   $$\begin{pmatrix} \tilde{x}'_\alpha \\ \tilde{y}'_\alpha \end{pmatrix} \leftarrow \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)}{(\boldsymbol{u}, V[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{u})} \begin{pmatrix} u_1 & u_4 & u_7 \\ u_2 & u_5 & u_8 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ f_0 \end{pmatrix}. \tag{8}$$

6. Go back to Step 3 after updating $\boldsymbol{u}_0 \leftarrow \boldsymbol{u}$, $\hat{x}_\alpha \leftarrow x_\alpha - \tilde{x}_\alpha$, $\hat{y}_\alpha \leftarrow y_\alpha - \tilde{y}_\alpha$, $\hat{x}'_\alpha \leftarrow x'_\alpha - \tilde{x}'_\alpha$, and $\hat{y}'_\alpha \leftarrow y'_\alpha - \tilde{y}'_\alpha$.
```

The initialization in Step 1 can be done by the 8-point algorithm [5] or by the Taubin method [20] (also see [12, 9]). The EFNS routine in Step 4 goes as follows:

EFNS

1. Compute the following 9×9 matrices $\boldsymbol{M}$ and $\boldsymbol{L}$:

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \qquad \boldsymbol{L} = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}. \tag{9}$$

2. Compute the cofactor vector $\boldsymbol{u}^\dagger$ in (5) and the 9×9 projection matrix

$$\boldsymbol{P}_{\mathbf{u}^\dagger} \equiv \boldsymbol{I} - \boldsymbol{u}^\dagger \boldsymbol{u}^{\dagger\top}. \tag{10}$$

3. Compute the following 9×9 matrices:

$$\boldsymbol{X} = \boldsymbol{M} - \boldsymbol{L}, \qquad \boldsymbol{Y} = \boldsymbol{P}_{\mathbf{u}^\dagger} \boldsymbol{X} \boldsymbol{P}_{\mathbf{u}^\dagger}. \tag{11}$$

4. Compute the two unit eigenvectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ of $\boldsymbol{Y}$ for the smallest eigenvalues in absolute value, and compute

$$\hat{\boldsymbol{u}} = (\boldsymbol{u}, \boldsymbol{v}_1)\boldsymbol{v}_1 + (\boldsymbol{u}, \boldsymbol{v}_2)\boldsymbol{v}_2. \tag{12}$$

5. Compute

$$\boldsymbol{u}' = \mathcal{N}[\boldsymbol{P}_{\mathbf{u}^\dagger} \hat{\boldsymbol{u}}]. \tag{13}$$

6. If $\boldsymbol{u}' \approx \boldsymbol{u}$ up to sign, return $\boldsymbol{u}'$ and stop. Else, let $\boldsymbol{u} \leftarrow \mathcal{N}[\boldsymbol{u} + \boldsymbol{u}']$ and go back to Step 1.

3 Derivation

3.1 Derivation of the Main Routine

First Approximation. We want to compute $\bar{\boldsymbol{x}}_\alpha$ and $\bar{\boldsymbol{x}}'_\alpha$ that minimize (3) subject to (4), but we may alternatively write

$$\bar{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - \Delta\boldsymbol{x}_\alpha, \qquad \bar{\boldsymbol{x}}'_\alpha = \boldsymbol{x}'_\alpha - \Delta\boldsymbol{x}'_\alpha, \tag{14}$$

and compute the correction terms $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$. Substituting (14) into (4), we have

$$E = \sum_{\alpha=1}^{N} \Big(\|\Delta\boldsymbol{x}_\alpha\|^2 + \|\Delta\boldsymbol{x}'_\alpha\|^2 \Big). \tag{15}$$

The epipolar equation (4) becomes

$$(\boldsymbol{x}_\alpha - \Delta\boldsymbol{x}_\alpha, \boldsymbol{F}(\boldsymbol{x}'_\alpha - \Delta\boldsymbol{x}'_\alpha)) = 0. \tag{16}$$

Ignoring the second order term in the correction terms, we obtain

$$(\boldsymbol{F}\boldsymbol{x}'_\alpha, \Delta\boldsymbol{x}_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \Delta\boldsymbol{x}'_\alpha) = (\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha). \tag{17}$$

Since the correction should be done in the image plane, we have the constraints

$$(\boldsymbol{k}, \Delta\boldsymbol{x}_\alpha) = 0, \qquad (\boldsymbol{k}, \Delta\boldsymbol{x}'_\alpha) = 0, \tag{18}$$

where we define $\boldsymbol{k} \equiv (0,0,1)^\top$. Introducing Lagrange multipliers for (17) and (18), we obtain $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$ that minimize (15) as follows (see [13] for the details):

$$\begin{aligned}\Delta\boldsymbol{x}_\alpha &= \frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)},\\ \Delta\boldsymbol{x}'_\alpha &= \frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}.\end{aligned} \tag{19}$$

Here, $\boldsymbol{P}_{\mathbf{k}}$ is the 3×3 projection matrix along $\boldsymbol{k}$:

$$\boldsymbol{P}_{\mathbf{k}} \equiv \boldsymbol{I} - \boldsymbol{k}\boldsymbol{k}^\top. \tag{20}$$

Substituting (19) into (15), we obtain (see [13] for the details)

$$E = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)^2}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}, \tag{21}$$

which is known as the *Sampson error* [6]. Suppose we have obtained the matrix $\boldsymbol{F}$ that minimizes (21) subject to $\det\boldsymbol{F} = 0$. Writing it as $\hat{\boldsymbol{F}}$ and substituting it into (14), we obtain

$$\begin{aligned}\hat{\boldsymbol{x}}_\alpha &= \boldsymbol{x}_\alpha - \frac{(\boldsymbol{x}_\alpha, \hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha}{(\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha) + (\hat{\boldsymbol{F}}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top\boldsymbol{x}_\alpha)},\\ \hat{\boldsymbol{x}}'_\alpha &= \boldsymbol{x}'_\alpha - \frac{(\boldsymbol{x}_\alpha, \hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top\boldsymbol{x}_\alpha}{(\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha) + (\hat{\boldsymbol{F}}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top\boldsymbol{x}_\alpha)}.\end{aligned} \tag{22}$$

Higher Order Correction. The solution (22) is only a first approximation. So, we estimate the true solution $\bar{\boldsymbol{x}}_\alpha$ and $\bar{\boldsymbol{x}}'_\alpha$ by writing, instead of (14),

$$\bar{\boldsymbol{x}}_\alpha = \hat{\boldsymbol{x}}_\alpha - \Delta\hat{\boldsymbol{x}}_\alpha, \qquad \bar{\boldsymbol{x}}'_\alpha = \hat{\boldsymbol{x}}'_\alpha - \Delta\hat{\boldsymbol{x}}'_\alpha, \tag{23}$$

and computing the correction terms $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$, which are small quantities of higher order than $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$. Substitution of (23) into (3) yields

$$E = \sum_{\alpha=1}^{N} \Big(\|\tilde{\boldsymbol{x}}_\alpha + \Delta\hat{\boldsymbol{x}}_\alpha\|^2 + \|\tilde{\boldsymbol{x}}'_\alpha + \Delta\hat{\boldsymbol{x}}'_\alpha\|^2\Big), \tag{24}$$

where we define

$$\tilde{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - \hat{\boldsymbol{x}}_\alpha, \qquad \tilde{\boldsymbol{x}}'_\alpha = \boldsymbol{x}'_\alpha - \hat{\boldsymbol{x}}'_\alpha. \tag{25}$$

The epipolar equation (4) now becomes

$$(\hat{\boldsymbol{x}}_\alpha - \Delta\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}(\hat{\boldsymbol{x}}'_\alpha - \Delta\hat{\boldsymbol{x}}'_\alpha)) = 0. \tag{26}$$

Ignoring second order term in $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$, we have

$$(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \Delta\hat{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \Delta\hat{\boldsymbol{x}}'_\alpha) = (\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha). \tag{27}$$

This is a higher order approximation of (4) than (17). Introducing Lagrange multipliers to (27) and the constraints

$$(\boldsymbol{k}, \Delta\hat{\boldsymbol{x}}_\alpha) = 0, \qquad (\boldsymbol{k}, \Delta\hat{\boldsymbol{x}}'_\alpha) = 0, \tag{28}$$

we obtain $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$ as follows (see [13] for the details):

$$\begin{aligned} \Delta\hat{\boldsymbol{x}}_\alpha &= \frac{\Big((\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha)\Big)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)} - \tilde{\boldsymbol{x}}_\alpha, \\ \Delta\hat{\boldsymbol{x}}'_\alpha &= \frac{\Big((\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha)\Big)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)} - \tilde{\boldsymbol{x}}'_\alpha. \end{aligned} \tag{29}$$

The reprojection error (24) now has the form (see [13] for the details)

$$E = \sum_{\alpha=1}^{N} \frac{\Big((\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha)\Big)^2}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)}. \tag{30}$$

Suppose we have obtained the matrix $\boldsymbol{F}$ that minimizes this subject to $\det \boldsymbol{F} = 0$. Writing it as $\hat{\boldsymbol{F}}$ and substituting it into (29), we obtain from (25) the solution

$$\begin{aligned} \hat{\hat{\boldsymbol{x}}}_\alpha &= \boldsymbol{x}_\alpha - \frac{\Big((\hat{\boldsymbol{x}}_\alpha, \hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha) + (\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha)\Big)\boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha}{(\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha) + (\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha)}, \\ \hat{\hat{\boldsymbol{x}}}'_\alpha &= \boldsymbol{x}'_\alpha - \frac{\Big((\hat{\boldsymbol{x}}_\alpha, \hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha) + (\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha)\Big)\boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha}{(\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha) + (\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha)}. \end{aligned} \tag{31}$$

The resulting $\{\hat{\hat{\boldsymbol{x}}}_\alpha, \hat{\hat{\boldsymbol{x}}}'_\alpha\}$ are a better approximation than $\{\hat{\boldsymbol{x}}_\alpha, \hat{\boldsymbol{x}}'_\alpha\}$. Rewriting $\{\hat{\hat{\boldsymbol{x}}}_\alpha, \hat{\hat{\boldsymbol{x}}}'_\alpha\}$ as $\{\hat{\boldsymbol{x}}_\alpha, \hat{\boldsymbol{x}}'_\alpha\}$, we repeat this until the iterations converge. In the end, $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$ in (26) become $\mathbf{0}$, and the epipolar equation is exactly satisfied.

Compact 9-D Description. The above algorithm is greatly simplified by using the 9-D vector encoding of (5). The definition of $\boldsymbol{\xi}_\alpha$ in (6) and $V_0[\boldsymbol{\xi}_\alpha]$ in (7) implies the following identities:

$$(\hat{\boldsymbol{x}}_\alpha, \hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha) + (\hat{\boldsymbol{F}}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\hat{\boldsymbol{F}}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha) = \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)}{f_0^2}, \tag{32}$$

$$(\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha) + (\hat{\boldsymbol{F}}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top\boldsymbol{x}_\alpha) = \frac{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}{f_0^2}. \tag{33}$$

Since we define $\tilde{\boldsymbol{x}}_\alpha$ and $\tilde{\boldsymbol{x}}'_\alpha$ by (25), we obtain from (31) the update form in (8). If we let $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, $\hat{x}'_\alpha = x'_\alpha$, $\hat{y}'_\alpha = y'_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = \tilde{x}'_\alpha = \tilde{y}'_\alpha = 0$, as in the Step 2 of the main routine, the update form (8) is equivalent to (22). Thus, the main routine is completed except Step 4, where we need to minimize (21) and (30) subject to $\det \boldsymbol{F} = 0$.

3.2 Derivation of EFNS

Problem. Using the identities (32) and (33), we can rewrite (30) as

$$E = \frac{1}{f_0^2}\sum_{\alpha=1}^{N}\frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}. \tag{34}$$

If we let $\hat{x}_\alpha = x_\alpha$, $\hat{y}_\alpha = y_\alpha$, $\hat{x}'_\alpha = x'_\alpha$, $\hat{y}'_\alpha = y'_\alpha$, and $\tilde{x}_\alpha = \tilde{y}_\alpha = \tilde{x}'_\alpha = \tilde{y}'_\alpha = 0$, as in the Step 2 of the main routine, this reduces to the Sampson error in (21). The problem is to minimize (34) subject to $\det \boldsymbol{u} = 0$.

Geometry. The necessary and sufficient condition for E to be stationary at a point $\boldsymbol{u}$ on the 8-D unit sphere $\mathcal{S}^8$ in $\mathcal{R}^9$ is that its gradient $\nabla_{\mathbf{u}}E$ is orthogonal to the hypersurface defined by $\det \boldsymbol{u} = 0$. Direct manipulation shows

$$\boldsymbol{u}^\dagger = \mathcal{N}[\nabla_{\mathbf{u}} \det \boldsymbol{u}]. \tag{35}$$

In other words, $\boldsymbol{u}^\dagger$ is the unit surface normal to the hypersurface defined by $\det \boldsymbol{u} = 0$. It follows that $\nabla_{\mathbf{u}}E$ should be parallel to the cofactor vector $\boldsymbol{u}^\dagger$ at the stationary point. Differentiating (34) with respect to $\boldsymbol{u}$, we see that

$$\nabla_{\mathbf{u}}E = \frac{2}{f_0^2}\boldsymbol{X}\boldsymbol{u}, \tag{36}$$

where $\boldsymbol{X}$ is the matrix in (11). Using the projection matrix $\boldsymbol{P}_{\mathbf{u}^\dagger}$ in (10), we can express the parallelism of $\nabla_{\mathbf{u}}E$ and $\boldsymbol{u}^\dagger$ as

$$\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{X}\boldsymbol{u} = \boldsymbol{0}. \tag{37}$$

The rank constraint $\det \boldsymbol{u} = 0$ is equivalently written as

$$(\boldsymbol{u}^\dagger, \boldsymbol{u}) = 0, \tag{38}$$

which is a direct consequence of the identity $\boldsymbol{F}^\dagger\boldsymbol{F} = (\det \boldsymbol{F})\boldsymbol{I}$. In terms of the projection matrix $\boldsymbol{P}_{\mathbf{u}^\dagger}$, the rank constraint (38) is equivalently written as

$$\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{u} = \boldsymbol{u}. \tag{39}$$

It follows that the stationarity condition (37) is written as

$$\boldsymbol{Y}\boldsymbol{u} = \boldsymbol{0}, \tag{40}$$

where $\boldsymbol{Y}$ is the matrix defined in (11). Our task is to compute the solution $\boldsymbol{u}$ that satisfies the stationarity condition (40) and the rank constraint (39).

Justification of the Procedure. We now show that the desired solution can be obtained by the EFNS routine in Sec. 2. To see this, we show that when the iterations have converged, the eigenvectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ of $\boldsymbol{Y}$ both have eigenvalue 0. From the definition of $\boldsymbol{Y}$ in (11) and $\boldsymbol{P}_{\mathbf{u}^\dagger}$ in (10), the cofactor vector $\boldsymbol{u}^\dagger$ is always an eigenvector of $\boldsymbol{Y}$ with eigenvalue 0. This means that either $\boldsymbol{v}_1$ or $\boldsymbol{v}_2$ has eigenvalue 0. Suppose one, say $\boldsymbol{v}_1$, has nonzero eigenvalue λ ($\neq 0$). Then, $\boldsymbol{v}_2 = \pm\boldsymbol{u}^\dagger$. By construction, the vector $\hat{\boldsymbol{u}}$ in (12) belongs to the linear span of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ ($= \pm\boldsymbol{u}^\dagger$), which are mutually orthogonal, and the vector $\boldsymbol{u}'$ in (13) is a projection of $\hat{\boldsymbol{u}}$ within that linear span onto the direction orthogonal to $\boldsymbol{u}^\dagger$. Hence, $\boldsymbol{u}'$ should coincide with $\pm\boldsymbol{v}_1$. After the iterations have converged, we have $\boldsymbol{u} = \boldsymbol{u}'$ ($= \pm\boldsymbol{v}_1$), so $\boldsymbol{u}$ is an eigenvector of $\boldsymbol{Y}$ with eigenvalue λ, i.e., $\boldsymbol{Y}\boldsymbol{u} = \lambda\boldsymbol{u}$. Taking the inner product with $\boldsymbol{u}$ on both sides, we have

$$(\boldsymbol{u}, \boldsymbol{Y}\boldsymbol{u}) = \lambda. \tag{41}$$

On the other hand, $\boldsymbol{u}$ ($= \pm\boldsymbol{v}_1$) is orthogonal to the cofactor vector $\boldsymbol{u}^\dagger$ ($= \pm\boldsymbol{v}_2$), so $\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{u} = \boldsymbol{u}$. Hence,

$$(\boldsymbol{u}, \boldsymbol{Y}\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{X}\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) = 0, \tag{42}$$

because from the definition of $\boldsymbol{X}$ in (11) we see that $(\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) = 0$ is an identity in $\boldsymbol{u}$. In fact, we can confirm from the definition of $\boldsymbol{M}$ and $\boldsymbol{L}$ in (9) that $(\boldsymbol{u}, \boldsymbol{M}\boldsymbol{u})$ $= (\boldsymbol{u}, \boldsymbol{L}\boldsymbol{u})$ holds identically in $\boldsymbol{u}$. Since (41) and (42) contradict our assumption that $\lambda \neq 0$, $\boldsymbol{v}_1$ is also an eigenvector of $\boldsymbol{Y}$ with eigenvalue 0. Thus, (39) and (40) both hold, so $\boldsymbol{u}$ is the desired solution.

Observations. The EFNS was first introduced by Kanatani and Sugaya [11] as a general constrained parameter estimation in abstract terms. It is a straightforward extension of the *FNS* of Chojnacki et al. [3] to include an arbitrary number of additional constraints. In fact, if we replace $\boldsymbol{P}_{\mathbf{u}^\dagger}$ in (11) by the identity $\boldsymbol{I}$, the resulting procedure is identical to FNS. For this reason, Kanatani and Sugaya [11] called it *EFNS* (*Extended FNS*). They applied it to minimization of the Sampson error (21) and pointed out that the CFNS of Chojnacki et al. [4] does not necessarily converge to a correct solution while EFNS does. Our new finding here is that it can also be used for *strict ML* (minimization of the reprojection error) if we introduce the new *intermediate variables* $\boldsymbol{\xi}_\alpha$ and $V_0[\boldsymbol{\xi}_\alpha]$ as in (6) and (7).

The justification described earlier relies on the premise that the iterations converge. As pointed in [11], if we let $\boldsymbol{u} \leftarrow \boldsymbol{u}'$ in the Step 6 of the EFNS routine, the next value of $\boldsymbol{u}'$ computed in Step 5 often reverts to the former value of $\boldsymbol{u}$, falling in infinite looping. So, the "midpoint" $(\boldsymbol{u}' + \boldsymbol{u})/2$ is normalized to a unit vector $\mathcal{N}[\boldsymbol{u}' + \boldsymbol{u}]$. This greatly improves convergence. In fact, we have confirmed that this technique also improves the convergence of FNS, which sometimes oscillates in the presence of very large noise.

Theoretically speaking, our algorithm may not produce a global minimum of the reprojection error (3). The problem is not the main routine, for which one need not worry about local minima, as argued in the optimal triangulation case [14]. However, the EFNS routine is not theoretically guaranteed to reach the absolute minimum of E in (34), although we have never experienced the contrary in all our experiments.

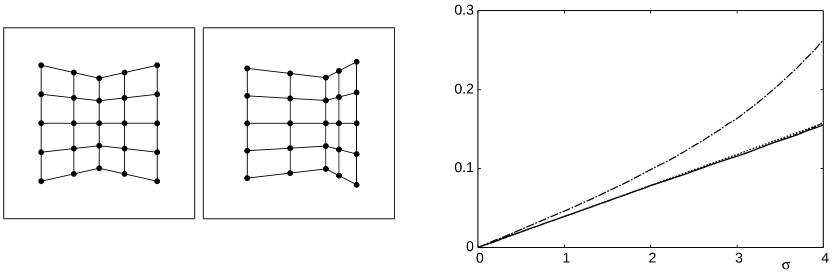
4 Performance Confirmation

Figure 2(a) shows simulated images of two planar grid surfaces. The image size is 600 × 600 pixels with 1200 pixel focal length. We added random Gaussian noise of mean 0 and standard deviation σ to the x- and y-coordinates of each grid point independently and from them computed the fundamental matrix.

Since all existing ML-based methods minimize the same reprojection error, their mutual accuracy comparison does not make much sense. Rather, our concern is if our algorithm really converges to a correct solution. To see this, we compare our algorithm with a carefully tuned alternative method. We compute an initial solution by least squares, from which we start the FNS of Chojnacki et al. [3], and the resulting solution is optimally corrected to satisfy the rank constraint. From it, we start a direct 7-D search, using the Levenberg-Marquardt (LM) method [18].

Figure 2(b) plots, for each σ, the RMS of $\|\boldsymbol{P}_{\mathcal{U}}\hat{\boldsymbol{u}}\|$ for the computed solution $\hat{\boldsymbol{u}}$ over 10,000 independent trials with different noise, where $\boldsymbol{P}_{\mathcal{U}}$ ($\equiv \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top - \boldsymbol{u}^\dagger\boldsymbol{u}^{\dagger\top}$) denotes projection onto the space of deviations from the true solution $\boldsymbol{u}$ and the rank constraint $\det\boldsymbol{u} = 0$. Our algorithm was initialized by least squares. As a reference, the chained line shows the corresponding result of the 8-point algorithm (least squares followed by SVD rank correction) [5], and the dotted line indicates the theoretical accuracy limit (*KCR lower bound*) [2, 8].

From Fig. 2(b), we see that the solid line (our algorithm) and the dashed line (the alternative method) completely coincide, indicating that the same solution is reached although their paths of approach may be very different (Fig. 1). We also see that the accuracy almost coincides with the theoretical limit, so *no further improvement is hoped for*. As predicted, the 8-point algorithm performs poorly. Doing many experiments (not all shown here), we observed the following:



(a) (b)

Fig. 2. (a) Simulated images of planar grid surfaces. (b) The RMS error vs. noise level σ. Solid line: our algorithm. Dashed line: the alternative method. Chained line: the 8-point algorithm. Dotted line: KCR lower bound.

1. The main routine converges after a few (at most four) iterations.
2. If we stop at Step 4 in the initial round without doing any further iterations, we obtain the Sampson solution. Yet, it coincides with the final (strict ML) solution up to three to four decimal places. The high accuracy of the Sampson solution was also noted by Zhang [21].
3. If initialized by least squares, the 7-D search does not necessarily arrive at the true minimum of the reprojection error, being trapped to local minima, as reported in [19]. After a careful tuning as described above, the solution coincides with our algorithm, which directly arrives at the same solution without any such tuning.

5 Concluding Remarks

We have presented a very compact algorithm for computing the fundamental matrix from point correspondences over two images based on the strict ML principle using the EFNS procedure. The computation consists only of vector and matrix operations in no higher than 9-D just like the 8-point algorithm, yet producing an optimal solution. By numerical experiments, we have confirmed that our algorithm behaves satisfactorily. Because of its compactness and good performance, we expect it to be a standard tool for fundamental matrix computation.

References

1. Bartoli, A., Sturm, P.: Nonlinear estimation of fundamental matrix with minimal parameters, IEEE Trans. Patt. Anal. Mach. Intell. 26(3), 426–432 (2004)
2. Chernov, N., Lesort, C.: Statistical efficiency of curve fitting algorithms, Comp. Stat. Data Anal. 47(4), 713–728 (2004)
3. Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D.: On the fitting of surfaces to data with covariances, IEEE Trans. Patt. Anal. Mach. Intell. 22(11), 1294–1303 (2000)
4. Chojnacki, W., Brooks, M.J., van den Hengel, A., Gawley, D.: A new constrained parameter estimator for computer vision applications, Image Vis. Comput. 22(2), 85–91 (2004)
5. Hartley, R.I.: In defense of the eight-point algorithm, IEEE Trans. Patt. Anal. Mach. Intell. 19(6), 580–593 (1997)
6. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
7. Kahl, F., Henrion, D.: Globally optimal estimates for geometric reconstruction problems, Int. J. Comput. Vis. 74(1), 3–15 (2007)
8. Kanatani, K.: Statistical Optimization for Geometric Computation: Theory and Practice. Elsevier, Amsterdam (1996); reprinted. Dover, New York (2005)
9. Kanatani, K.: Statistical optimization for geometric fitting: Theoretical accuracy analysis and high order error analysis, Int. J. Compt. Vis. 80(2), 167–188 (2008)
10. Kanatani, K., Sugaya, Y.: High accuracy fundamental matrix computation and its performance evaluation, IEICE Trans. Information and Systems, E90-D(2), 579–585 (2007)

11. Kanatani, K., Sugaya, Y.: Extended FNS for constrained parameter estimation. In: Proc. 10th Meeting Image Recognition Understanding, Hiroshima, Japan, July 2007, pp. 219–226 (2007)
12. Kanatani, K., Sugaya, Y.: Performance evaluation of iterative geometric fitting algorithms, Comp. Stat. Data Anal. 52(2), 1208–1222 (2007)
13. Kanatani, K., Sugaya, Y.: Small algorithm for fundamental matrix computation. In: Proc. Meeting Image Recognition and Understanding, Karuizawa, Japan, July 2008, pp. 947–954 (2008)
14. Kanatani, K., Sugaya, Y., Niitsuma, H.: Triangulation from two views revisited: Hartley-Sturm vs. optimal correction. In: Proc. 19th British Mach. Vis. Conf., September 2008, pp. 173–182 (2008)
15. Leedan, Y., Meer, P.: Heteroscedastic regression in computer vision: Problems with bilinear constraint, Int. J. Comput. Vis. 37(2), 127–150 (2000)
16. Matei, J., Meer, P.: Estimation of nonlinear errors-in-variables models for computer vision applications, IEEE Trans. Patt. Anal. Mach. Intell. 28(10), 1537–1552 (2006)
17. Migita, T., Shakunaga, T.: One-dimensional search for reliable epipole estimation, In: Proc. IEEE Pacific Rim Symp. Image Video Tech., Hsinchu, Taiwan, December 2006, pp. 1215–1224 (2006)
18. Sugaya, Y., Kanatani, K.: High accuracy computation of rank-constrained fundamental matrix. In: Proc. 18th British Mach. Vis. Conf., September 2007, vol. 1, pp. 282–291 (2007)
19. Sugaya, Y., Kanatani, K.: Highest accuracy fundamental matrix computation. In: Proc. 8th Asian Conf. Comput. Vis., Tokyo, Japan, November 2008, vol. 2, pp. 311–321 (2008)
20. Taubin, G.: Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation, IEEE Trans. Patt. Anal. Mach. Intell. 13(11), 1115–1138 (1991)
21. Zhang, Z.: Determining the epipolar geometry and its uncertainty: A review, Int. J. Comput. Vis. 27(2), 161–195 (1998)
22. Zhang, Z., Loop, C.: Estimating the fundamental matrix by transforming image points in projective space, Comput. Vis. Image Understand. 82(2), 174–180 (2001)