Fundamental Matrix Computation: Theory and Practice

Kenichi Kanatani† and Yasuyuki Sugaya‡
† *Department of Computer Science, Okayama University, Okayama, Japan 700-8530*
kanatani@suri.it.okayama-u.ac.jp
‡ Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Aichi 441-8580 Japan
sugaya@iim.ics.tut.ac.jp

Abstract

We classify and review existing algorithms for computing the fundamental matrix from point correspondences and propose new effective schemes: 7-parameter Levenberg-Marquardt (LM) search, extended FNS, and EFNS-based bundle adjustment. Doing experimental comparison, we show that EFNS and the 7-parameter LM search exhibit the best performance and that additional bundle adjustment does not increase the accuracy to any noticeable degree.

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 [7]. This problem has attracted a special attention because of the following two characteristics:

1. Feature points are extracted by an image processing operation [8, 15, 18, 21]. As s result, the detected locations invariably have uncertainty to some degree.
2. Detected points are matched by comparing surrounding regions in respective images, using various measures of similarity and correlation [13, 17, 24]. Hence, mismatches are unavoidable to some degree.

The first problem has been dealt with by *statistical optimization* [9]: we model the uncertainty as "noise" obeying a certain probability distribution and compute a fundamental matrix such that its deviation from the true value is as small as possible in expectation. The second problem has been coped with by *robust estimation* [19], which can be viewed as hypothesis testing: we compute a tentative fundamental matrix as a hypothesis and check how many points support it. Those points regarded as "abnormal" according to the hypothesis are called *outliers*, otherwise *inliers*, and we look for a fundamental matrix that has as many inliers as possible.

Thus, the two problems are inseparably interwoven. In this paper, we focus on the first problem, assuming that all corresponding points are inliers. Such a study is indispensable for any robust estimation technique to work successfully.

However, there is an additional compounding element in doing statistical optimization of the fundamental matrix: it is constrained to have rank 2, i.e., its determinant is 0. This rank constraint has been incorporated in various ways. Here, we categorize them into the following three approaches:

A posteriori correction. The fundamental matrix is optimally computed without considering the rank constraint and is modified in an optimal manner so that the constraint is satisfied (Fig. 1(a)).

Internal access. The fundamental matrix is minimally parameterized so that the rank constraint is identically satisfied and is optimized in the reduced ("internal") parameter space (Fig. 1(b)).

External access. We do iterations in the redundant ("external") parameter space in such a way that an optimal solution that satisfies the rank constraint automatically results (Fig. 1(c)).

The purpose of this paper is to review existing methods in this framework and propose new improved methods. In particular, this paper contains the following three technically new results:

1. We present a new internal access method [1].
2. We present a new external access method [2].
3. We present a new bundle adjustment algorithm[3].

Then, we experimentally compare their performance, using simulated and real images.

In Sect. 2, we summarize the mathematical background. In Sect. 3, we study the a posteriori correction approach. We review two correction schemes (SVD correction and optimal correction), three unconstrained optimization techniques (FNS, HEIV, projective Gauss-Newton iterations),

[1] A preliminary version was presented in our conference paper [22].
[2] A preliminary version was presented in our conference paper [12].
[3] This has not been presented anywhere else.

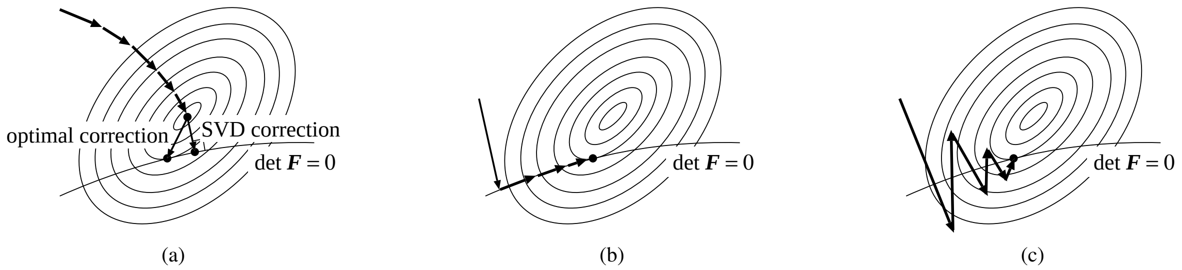


Figure 1. (a) A posteriori correction. (b) Internal access. (c) External access.

and two initialization methods (least squares (LS) and the Taubin method).

In Sect. 4, we focus on the internal access approach and present a new compact scheme for doing 7-parameter Levenberg-Marquardt (LM) search. In Sect. 5, we investigate the external access approach and point out that the CFNS of Chojnacki et al. [4], a pioneering external access method, does not necessarily converge to a correct solution. To complement this, we present a new method, called EFNS, and demonstrate that it always converges to an optimal value; a mathematical justification is given to this. In Sect. 6, we compare the accuracy of all the methods and conclude that our EFNS and the 7-parameter LM search started from optimally corrected ML exhibit the best performance.

In Sect. 7, we study the bundle adjustment (Gold Standard) approach and present a new efficient computational scheme for it. In Sect. 8, we experimentally test the effect of this approach and conclude that additional bundle adjustment does not increase the accuracy to any noticeable degree. Sect. 9 concludes this paper.

2. Mathematical Fundamentals

Fundamental matrix. We are given two images of the same scene. We take the image origin $(0,0)$ is at the frame center. Suppose a point (x,y) in the first image and the corresponding point (x',y') in the second. We represent them by 3-D vectors

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

where f_0 is a scaling constant of the order of the image size[4]. Then, the following the *epipolar equation* is satisfied [7]:

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

where and throughout this paper we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a},\boldsymbol{b})$. The matrix $\boldsymbol{F} = (F_{ij})$ in Eq. (2) is of rank 2 and called the *fundamental matrix*; it depends on the relative positions and orientations of the two cameras and their intrinsic parameters (e.g., their focal lengths) but not on the scene or the choice of the corresponding points.

If we define[5]

$$\boldsymbol{u} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top, \quad (3)$$

$$\boldsymbol{\xi} = (xx', xy', xf_0, yx', yy', yf_0, f_0x', f_0y', f_0^2)^\top, \quad (4)$$

Equation (2) can be rewritten as

$$(\boldsymbol{u}, \boldsymbol{\xi}) = 0. \qquad (5)$$

The magnitude of $\boldsymbol{u}$ is indeterminate, so we normalize it to $\|\boldsymbol{u}\| = 1$, which is equivalent to scaling $\boldsymbol{F}$ so that $\|\boldsymbol{F}\| = 1$. With a slight abuse of symbolism, we hereafter denote by $\det \boldsymbol{u}$ the determinant of the matrix $\boldsymbol{F}$ defined by $\boldsymbol{u}$.

If we write N observed noisy correspondence pairs as 9-D vectors $\{\boldsymbol{\xi}_\alpha\}$ in the form of Eq. (4), our task is to estimate from $\{\boldsymbol{\xi}_\alpha\}$ a 9-D vector $\boldsymbol{u}$ that satisfies Eq. (5) subject to the constraints $\|\boldsymbol{u}\| = 1$ and $\det \boldsymbol{u} = 0$.

Covariance matrices. Let us write $\boldsymbol{\xi}_\alpha = \bar{\boldsymbol{\xi}}_\alpha + \Delta\boldsymbol{\xi}_\alpha$, where $\bar{\boldsymbol{\xi}}_\alpha$ is the true value and $\Delta\boldsymbol{\xi}_\alpha$ the noise term. The covariance matrix of $\boldsymbol{\xi}_\alpha$ is defined by

$$V[\boldsymbol{\xi}_\alpha] = E[\Delta\boldsymbol{\xi}_\alpha \Delta\boldsymbol{\xi}_\alpha^\top], \qquad (6)$$

where $E[\,\cdot\,]$ denotes expectation over the noise distribution. If the noise in the x- and y-coordinates is independent and of mean 0 and standard deviation σ, the covariance matrix of $\boldsymbol{\xi}_\alpha$ has the form $V[\boldsymbol{\xi}_\alpha] = \sigma^2 V_0[\boldsymbol{\xi}_\alpha]$ up to $O(\sigma^4)$, where

$$V_0[\boldsymbol{\xi}_\alpha] = \left(\begin{array}{cccc} \bar{x}_\alpha^2 + \bar{x}_\alpha'^2 & \bar{x}_\alpha'\bar{y}_\alpha' & f_0\bar{x}_\alpha' & \bar{x}_\alpha\bar{y}_\alpha \\ \bar{x}_\alpha'\bar{y}_\alpha' & \bar{x}_\alpha^2 + \bar{y}_\alpha'^2 & f_0\bar{y}_\alpha' & 0 \\ f_0\bar{x}_\alpha' & f_0\bar{y}_\alpha' & f_0^2 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & \bar{y}_\alpha^2 + \bar{x}_\alpha'^2 \\ 0 & \bar{x}_\alpha\bar{y}_\alpha & 0 & \bar{x}_\alpha'\bar{y}_\alpha' \\ 0 & 0 & 0 & f_0\bar{x}_\alpha' \\ f_0\bar{x}_\alpha & 0 & 0 & f_0\bar{y}_\alpha \\ 0 & f_0\bar{x}_\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right.$$

[4]This is for stabilizing numerical computation [6]. In our experiments, we set f_0 = 600 pixels.

[5]The vector $\boldsymbol{\xi}$ is known as the "Kronecker product" of the vectors $(x,y,f_0)^\top$ and $(x',y',f_0)^\top$.

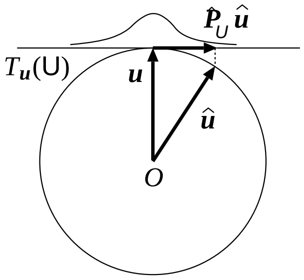


Figure 2. The deviation is projected onto the tangent space, with which we identify the noise domain.

$$\begin{matrix} 0 & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & f_0\bar{x}_\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bar{x}'_\alpha\bar{y}'_\alpha & f_0\bar{x}'_\alpha & f_0\bar{y}_\alpha & 0 & 0 \\ \bar{y}^2_\alpha + \bar{y}'^2_\alpha & f_0\bar{y}'_\alpha & 0 & f_0\bar{y}_\alpha & 0 \\ f_0\bar{y}'_\alpha & f_0^2 & 0 & 0 & 0 \\ 0 & 0 & f_0^2 & 0 & 0 \\ f_0\bar{y}_\alpha & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}\right), \qquad (7)$$

In actual computations, the true positions $(\bar{x}_\alpha, \bar{y}_\alpha)$ and $(\bar{x}'_\alpha, \bar{y}'_\alpha)$ are replaced by their data (x_α, y_α) and (x'_α, y'_α), respectively[6].

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

$$V[\hat{\boldsymbol{u}}] = E[(\boldsymbol{P}_\mathcal{U}\hat{\boldsymbol{u}})(\boldsymbol{P}_\mathcal{U}\hat{\boldsymbol{u}})^\top], \qquad (8)$$

where $\boldsymbol{P}_\mathcal{U}$ is the linear operator projecting $\mathcal{R}^9$ onto the domain $\mathcal{U}$ of $\boldsymbol{u}$ defined by the constraints $\|\boldsymbol{u}\| = 1$ and $\det \boldsymbol{u} = 0$; we evaluate the error of $\hat{\boldsymbol{u}}$ by projecting it onto the tangent space $T_\mathbf{u}(\mathcal{U})$ to $\mathcal{U}$ at $\boldsymbol{u}$ (Fig. 2) [9].

Geometry of the constraint. The unit normal to the hypersurface defined by $\det \boldsymbol{u} = 0$ is $\nabla_\mathbf{u} \det \boldsymbol{u}$. After normalization, it has the form

$$\boldsymbol{u}^\dagger \equiv N[\begin{pmatrix} u_5u_9 - u_8u_6 \\ u_6u_7 - u_9u_4 \\ u_4u_8 - u_7u_5 \\ u_8u_3 - u_2u_9 \\ u_9u_1 - u_3u_7 \\ u_7u_2 - u_1u_8 \\ u_2u_6 - u_5u_3 \\ u_3u_4 - u_6u_1 \\ u_1u_5 - u_4u_2 \end{pmatrix}], \qquad (9)$$

where $N[\,\cdot\,]$ denotes normalization into unit norm[7]. It is easily seen that the rank constraint $\det \boldsymbol{u} = 0$ is equivalently written as

$$(\boldsymbol{u}^\dagger, \boldsymbol{u}) = 0. \qquad (10)$$

Since the domain $\mathcal{U}$ is included in the unit sphere $\mathcal{S}^8 \subset \mathcal{R}^9$, the vector $\boldsymbol{u}$ is everywhere orthogonal to $\mathcal{U}$. Hence, $\{\boldsymbol{u}, \boldsymbol{u}^\dagger\}$ is an orthonormal basis of the orthogonal complement of the tangent space $T_\mathbf{u}(\mathcal{U})$. It follows that the projection operator $\boldsymbol{P}_\mathcal{U}$ in Eq. (8) has the following matrix representation ($\boldsymbol{I}$ denotes the unit matrix):

$$\boldsymbol{P}_\mathcal{U} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top - \boldsymbol{u}^\dagger\boldsymbol{u}^{\dagger\top}. \qquad (11)$$

KCR lower bound. If the noise in $\{\boldsymbol{\xi}_\alpha\}$ is independent and Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma^2 V_0[\boldsymbol{\xi}_\alpha]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\boldsymbol{u}}$ of $\boldsymbol{u}$ [9]:

$$V[\hat{\boldsymbol{u}}] \succ \sigma^2 \Big(\sum_{\alpha=1}^N \frac{(\boldsymbol{P}_\mathcal{U}\bar{\boldsymbol{\xi}}_\alpha)(\boldsymbol{P}_\mathcal{U}\bar{\boldsymbol{\xi}}_\alpha)^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} \Big)^-_8. \qquad (12)$$

Here, $\succ$ means that the left-hand side minus the right is positive semidefinite, and $(\,\cdot\,)^-_r$ denotes the pseudoinverse of rank r. Chernov and Lesort [2] called the right-hand side of Eq. (12) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that Eq. (12) holds up to $O(\sigma^4)$ even if $\hat{\boldsymbol{u}}$ is not unbiased; it is sufficient that $\hat{\boldsymbol{u}} \to \boldsymbol{u}$ as $\sigma \to 0$.

Maximum likelihood. If the noise in $\{\boldsymbol{\xi}_\alpha\}$ is independent and Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma^2 V_0[\boldsymbol{\xi}_\alpha]$, *maximum likelihood* (ML) estimation of $\boldsymbol{u}$ is to minimize the sum of square Mahalanobis distances

$$J = \sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha, V_0[\boldsymbol{\xi}_\alpha]^-_4 (\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha)), \qquad (13)$$

subject to $(\boldsymbol{u}, \bar{\boldsymbol{\xi}}_\alpha) = 0$, $\alpha = 1, ..., N$. Geometrically, we are fitting a hyperplane $(\boldsymbol{u}, \boldsymbol{\xi}) = 0$ in the $\boldsymbol{\xi}$-space to N points $\{\boldsymbol{\xi}_\alpha\}$ as closely as possible; the closeness is measured not in the Euclidean sense but in the Mahalanobis distance inversely weighted by the covariance matrix $V_0[\boldsymbol{\xi}_\alpha]$ representing the uncertainty of each datum.

Eliminating the constraints $(\boldsymbol{u}, \bar{\boldsymbol{\xi}}_\alpha) = 0$ by using Lagrange multipliers, we obtain [9]

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

The ML estimator $\hat{\boldsymbol{u}}$ minimizes this subject to the normalization $\|\boldsymbol{u}\| = 1$ and the rank constraint $(\boldsymbol{u}^\dagger, \boldsymbol{u}) = 0$.

3. A Posteriori Correction

3.1. Correction schemes

The a posteriori correction approach first minimizes Eq. (14) without considering the rank constraint and

[6] Experiments have confirmed that this does not noticeable changes in final results.

[7] The inside of $N[\,\cdot\,]$ represents the "cofactor" of $\boldsymbol{F}$ in the vector form.

then modifies the resulting solution $\tilde{\boldsymbol{u}}$ so as to satisfy it (Fig. 1(a)).

SVD correction. A naive idea is to compute the singular value decomposition (SVD) of the computed fundamental matrix and replace the smallest singular value by 0, resulting in a matrix of rank 2 "closest" to the original one in Frobenius norm [6]. We call this *SVD correction*.

Optimal correction. A more sophisticated method is the *optimal correction* [9, 16]. According to the statistical optimization theory [9], the covariance matrix $V[\tilde{\boldsymbol{u}}]$ of the rank unconstrained solution $\tilde{\boldsymbol{u}}$ can be evaluated, so $\tilde{\boldsymbol{u}}$ is moved in the direction of the mostly likely fluctuation implied by $V[\tilde{\boldsymbol{u}}]$ until it satisfies the rank constraint (Fig. 1(a)). The procedure goes as follows [9]:

1. Compute the following 9×9 matrix $\tilde{\boldsymbol{M}}$:

$$\tilde{\boldsymbol{M}} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\tilde{\boldsymbol{u}}, V_0[\boldsymbol{\xi}_\alpha]\tilde{\boldsymbol{u}})}. \tag{15}$$

2. Compute the matrix $V_0[\tilde{\boldsymbol{u}}]$ as follows:

$$V_0[\tilde{\boldsymbol{u}}] = (\boldsymbol{P}_{\tilde{\mathbf{u}}} \tilde{\boldsymbol{M}} \boldsymbol{P}_{\tilde{\mathbf{u}}})_8^-, \tag{16}$$

where

$$\boldsymbol{P}_{\tilde{\mathbf{u}}} = \boldsymbol{I} - \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}^\top. \tag{17}$$

3. Update the solution $\tilde{\boldsymbol{u}}$ as follows ($\tilde{\boldsymbol{u}}^\dagger$ is defined by Eq. (9) for $\tilde{\boldsymbol{u}}$):

$$\tilde{\boldsymbol{u}} \leftarrow N[\tilde{\boldsymbol{u}} - \frac{1}{3}\frac{(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{u}}^\dagger)V_0[\tilde{\boldsymbol{u}}]\tilde{\boldsymbol{u}}^\dagger}{(\tilde{\boldsymbol{u}}^\dagger, V_0[\tilde{\boldsymbol{u}}]\tilde{\boldsymbol{u}}^\dagger)}]. \tag{18}$$

4. If $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{u}}^\dagger) \approx 0$, return $\tilde{\boldsymbol{u}}$ and stop. Else, update $\boldsymbol{P}_{\tilde{\mathbf{u}}}$ and $V_0[\tilde{\boldsymbol{u}}]$ in the form

$$\boldsymbol{P}_{\tilde{\mathbf{u}}} \leftarrow \boldsymbol{I} - \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}^\top, \qquad V_0[\tilde{\boldsymbol{u}}] \leftarrow \boldsymbol{P}_{\tilde{\mathbf{u}}} V_0[\tilde{\boldsymbol{u}}] \boldsymbol{P}_{\tilde{\mathbf{u}}}, \tag{19}$$

and go back to Step 3.

Explanation. Since $\tilde{\boldsymbol{u}}$ is a unit vector, its endpoint is on the unit sphere $\mathcal{S}^8$ in $\mathcal{R}^9$. Eq. (18) is essentially the Newton iteration formula for displacing $\tilde{\boldsymbol{u}}$ in the direction in the tangent space $T_{\tilde{\mathbf{u}}}(\mathcal{S}^8)$ along which J is least increased so that $(\tilde{\boldsymbol{u}}^\dagger, \tilde{\boldsymbol{u}})$ = 0 is satisfied. However, $\tilde{\boldsymbol{u}}$ deviates from $\mathcal{S}^8$ by a small distance of high order as it proceeds in $T_{\tilde{\mathbf{u}}}(\mathcal{S}^8)$, so we pull it back onto $\mathcal{S}^8$ using the operator $N[\,\cdot\,]$. From that point, the same procedure is repeated until $(\tilde{\boldsymbol{u}}^\dagger, \tilde{\boldsymbol{u}})$ = 0. However, the normalized covariance matrix $V_0[\tilde{\boldsymbol{u}}]$ is defined in the tangent space $T_{\tilde{\mathbf{u}}}(\mathcal{S}^8)$, which changes as $\tilde{\boldsymbol{u}}$ moves. Eq. (19) corrects it so that $V_0[\tilde{\boldsymbol{u}}]$ has the domain $T_{\tilde{\mathbf{u}}}(\mathcal{S}^8)$ at the displaced point $\tilde{\boldsymbol{u}}$.

3.2. Unconstrained ML

Before imposing the rank constraint, we need to solve unconstrained minimization of Eq. (14), for which many method exist including FNS [3], HEIV [14], and the projective Gauss-Newton iterations [11]. Their convergence properties were studies in [11].

FNS. The *FNS* (*Fundamental Numerical Scheme*) of Chojnacki et al. [3] is based on the fact that the derivative of Eq. (14) with respect to $\boldsymbol{u}$ has the form

$$\nabla_{\mathbf{u}} J = 2\boldsymbol{X}\boldsymbol{u}, \tag{20}$$

where $\boldsymbol{X}$ has the following form [3]:

$$\boldsymbol{X} = \boldsymbol{M} - \boldsymbol{L}, \tag{21}$$

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \quad \boldsymbol{L} = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}. \tag{22}$$

The FNS solves

$$\boldsymbol{X}\boldsymbol{u} = \boldsymbol{0}. \tag{23}$$

by the following iterations [3, 11]:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrix $\boldsymbol{X}$ in Eq. (21).
3. Solve the eigenvalue problem

$$\boldsymbol{X}\boldsymbol{u}' = \lambda \boldsymbol{u}', \tag{24}$$

and compute the unit eigenvector $\boldsymbol{u}'$ for the smallest eigenvalue λ.

4. If $\boldsymbol{u}' \approx \boldsymbol{u}$ up to sign, return $\boldsymbol{u}'$ and stop. Else, let $\boldsymbol{u} \leftarrow \boldsymbol{u}'$ and go back to Step 2.

Originally, the eigenvalue closest to 0 was chosen [3] in Step 3. Later, Chojnacki, et al. [5] pointed out that the choice of the smallest eigenvalue improves the convergence. This was also confirmed by the experiments of Kanatani and Sugaya [11].

Whichever eigenvalue is chosen for λ, we have λ = 0 after convergence. In fact, convergence means

$$\boldsymbol{X}\boldsymbol{u} = \lambda \boldsymbol{u} \tag{25}$$

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

$$(\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) = \lambda, \tag{26}$$

but from Eq. (30) we have the identity $(\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u})$ = 0 in $\boldsymbol{u}$. Hence, λ = 0, and $\boldsymbol{u}$ is the desired solution.

HEIV. Equation (23) is rewritten as

$$\boldsymbol{M}\boldsymbol{u} = \boldsymbol{L}\boldsymbol{u}. \tag{27}$$

We introduce a new 8-D vector $\boldsymbol{v}$, 8-D data vectors $\boldsymbol{z}_\alpha$, and their 8×8 normalized covariance matrices $V|0[\boldsymbol{z}_\alpha]$ by

$$\boldsymbol{\xi}_\alpha = \begin{pmatrix} \boldsymbol{z}_\alpha \\ f_0^2 \end{pmatrix}, \qquad \boldsymbol{u} = \begin{pmatrix} \boldsymbol{v} \\ F_{33} \end{pmatrix},$$
$$V_0[\boldsymbol{\xi}_\alpha] = \begin{pmatrix} V_0[\boldsymbol{z}_\alpha] & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix}, \tag{28}$$

and define 8×8 matrices

$$\tilde{\boldsymbol{M}} = \sum_{\alpha=1}^{N} \frac{\tilde{\boldsymbol{z}}_\alpha \tilde{\boldsymbol{z}}_\alpha^\top}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})}, \qquad \tilde{\boldsymbol{L}} = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{v}, \tilde{\boldsymbol{z}}_\alpha)^2 V_0[\boldsymbol{z}_\alpha]}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})^2}, \tag{29}$$

where we put

$$\tilde{\boldsymbol{z}}_\alpha = \boldsymbol{z}_\alpha - \bar{\boldsymbol{z}},$$
$$\bar{\boldsymbol{z}} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{z}_\alpha}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})} \Bigg/ \sum_{\beta=1}^{N} \frac{1}{(\boldsymbol{v}, V_0[\boldsymbol{z}_\beta]\boldsymbol{v})}. \tag{30}$$

Then, Eq. (27) splits into the following two equations [5, 14]:

$$\tilde{\boldsymbol{M}}\boldsymbol{v} = \tilde{\boldsymbol{L}}\boldsymbol{v}, \qquad (\boldsymbol{v}, \bar{\boldsymbol{z}}) + f_0^2 F_{33} = 0. \tag{31}$$

Hence, if an 8-D vector $\boldsymbol{v}$ that satisfies the first equation is computed, the second equation gives F_{33}, and we obtain

$$\boldsymbol{u} = N\Big[\begin{pmatrix} \boldsymbol{v} \\ F_{33} \end{pmatrix}\Big]. \tag{32}$$

The *HEIV* (*Heteroscedastic Errors-in-Variable*) of Leedan and Meer [14] computes the vector $\boldsymbol{v}$ that satisfies the first of Eqs. (31) by the following iterations [5, 14]:

1. Initialize $\boldsymbol{v}$.
2. Compute the matrices $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{L}}$ in Eqs. (29).
3. Solve the generalized eigenvalue problem

$$\tilde{\boldsymbol{M}}\boldsymbol{v}' = \lambda \tilde{\boldsymbol{L}}\boldsymbol{v}', \tag{33}$$

   and compute the unit generalized eigenvector $\boldsymbol{v}'$ for the smallest generalized eigenvalue λ.
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.

In order to reach the solution of Eqs. (31), it appears natural to choose the generalized eigenvalue λ in Eq. (33) to be the one closest 1. However, Leedan and Meer [14] observed that choosing the smallest one improves the convergence performance. This was also confirmed by the experiments of Kanatani and Sugaya [11].

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

$$\tilde{\boldsymbol{M}}\boldsymbol{v} = \lambda \tilde{\boldsymbol{L}}\boldsymbol{v} \tag{34}$$

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

$$(\boldsymbol{v}, \tilde{\boldsymbol{M}}\boldsymbol{v}) = \lambda(\boldsymbol{v}, \tilde{\boldsymbol{L}}\boldsymbol{v}), \tag{35}$$

but from Eqs. (29) we have the identity $(\boldsymbol{v}, \tilde{\boldsymbol{M}}\boldsymbol{v}) = (\boldsymbol{v}, \tilde{\boldsymbol{L}}\boldsymbol{v})$ in $\boldsymbol{v}$. Hence, $\lambda = 1$, and $\boldsymbol{u}$ is the desired solution.

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

$$(\nabla_{\mathbf{u}}^2 J)\Delta \boldsymbol{u} = -\nabla_{\mathbf{u}} J. \tag{36}$$

However, $\nabla_{\mathbf{u}}^2 J$ is singular, since J is constant in the direction of $\boldsymbol{u}$ (see Eq. (14)). Hence, the solution is indeterminate. However, if we use pseudoinverse and compute

$$\Delta \boldsymbol{u} = -(\nabla_{\mathbf{u}}^2 J)_8^- \nabla_{\mathbf{u}} J, \tag{37}$$

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

Differentiating Eq. (20) and introducing Gauss-Newton approximation (i.e., ignoring terms that contain $(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)$), we see that the Hessian is nothing but the matrix $2\boldsymbol{M}$ in Eqs. (22). We enforce $\boldsymbol{M}$ to have eigenvalue 0 for $\boldsymbol{u}$, using the projection matrix

$$\boldsymbol{P}_{\mathbf{u}} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top \tag{38}$$

onto the direction orthogonal to $\boldsymbol{u}$. The iteration procedure goes as follows:

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

$$\boldsymbol{u}' = N[\boldsymbol{u} - (\boldsymbol{P}_{\mathbf{u}}\boldsymbol{M}\boldsymbol{P}_{\mathbf{u}})_8^-(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u}]. \tag{39}$$

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.

3.3. Initialization

The FNS, the HEIV, and the projective Gauss-Newton are all iterative method, so they require initial values. The best known non-iterative procedures are the least squares and the Taubin method.

Least squares (LS). This is the most popular method, also known as the *algebraic distance minimization* or the *8-point algorithm* [6]. Approximating the denominators in Eq. (14) by a constant, we minimize

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

where we define

$$\boldsymbol{M}_{\mathrm{LS}} = \sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top. \tag{41}$$

Equation (40) 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. (14) by their average, we minimize the following function[8] [23]:

$$J_{\rm TB} = \frac{\sum_{\alpha=1}^{N}(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{\sum_{\alpha=1}^{N}(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} = \frac{(\boldsymbol{u}, \boldsymbol{M}_{\rm LS}\boldsymbol{u})}{(\boldsymbol{u}, \boldsymbol{N}_{\rm TB}\boldsymbol{u})}. \tag{42}$$

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

$$\boldsymbol{N}_{\rm TB} = \sum_{\alpha=1}^{N} V_0[\boldsymbol{\xi}_\alpha]. \tag{43}$$

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

$$\boldsymbol{M}_{\rm LS}\boldsymbol{u} = \lambda \boldsymbol{N}_{\rm TB}\boldsymbol{u} \tag{44}$$

for the smallest generalized eigenvalue. However, we cannot directly solve this, because $\boldsymbol{N}_{\rm TB}$ is not positive definite. So, we decompose $\boldsymbol{\xi}_\alpha$, $\boldsymbol{u}$, and $V_0[\boldsymbol{\xi}_\alpha]$ in the form of Eqs. (28) and define 8×8 matrices $\tilde{\boldsymbol{M}}_{\rm LS}$ and $\tilde{\boldsymbol{N}}_{\rm TB}$ by

$$\tilde{\boldsymbol{M}}_{\rm LS} = \sum_{\alpha=1}^{N} \tilde{\boldsymbol{z}}_\alpha \tilde{\boldsymbol{z}}_\alpha^\top, \qquad \tilde{\boldsymbol{N}}_{\rm LS} = \sum_{\alpha=1}^{N} V_0[\boldsymbol{z}_\alpha], \tag{45}$$

where

$$\tilde{\boldsymbol{z}}_\alpha = \boldsymbol{z}_\alpha - \bar{\boldsymbol{z}}, \qquad \bar{\boldsymbol{z}} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{z}_\alpha. \tag{46}$$

Then, Eq. (44) splits into two equations

$$\tilde{\boldsymbol{M}}_{\rm LS}\boldsymbol{v} = \lambda \tilde{\boldsymbol{N}}_{\rm TB}\boldsymbol{v}, \qquad (\boldsymbol{v}, \bar{\boldsymbol{z}}) + f_0^2 F_{33} = 0. \tag{47}$$

We compute the unit generalized eigenvector $\boldsymbol{v}$ of the first equation for the smallest generalized eigenvalue λ. The second equation gives F_{33}, and $\boldsymbol{u}$ is given in the form of Eq. (32). It has been shown that Taubin's method produces a very accurate close to the unconstrained ML solution [11].

4. Internal Access

The fundamental matrix $\boldsymbol{F}$ has nine elements, on which the normalization $\|\boldsymbol{F}\| = 1$ and the rank constraint $\det \boldsymbol{u} = 0$ are imposed. Hence, it has seven degrees of freedom. The internal access minimizes Eq. (14) by searching the reduced 7-D parameter space (Fig. 1(b)).

Many types of 7-degree parameterizations have been obtained, e.g., by algebraic elimination of the rank constraint or by expressing the fundamental matrix in terms of epipoles [20, 25], but the resulting expressions are complicated, and the geometric meaning of the individual unknowns are not clear. This was overcome by Bartoli and Sturm [1], who regarded the SVD of $\boldsymbol{F}$ as its parameterization. Their expression is compact, and each parameter has its geometric meaning. However, they included, in addition to $\boldsymbol{F}$, the tentatively reconstructed 3-D positions of the observed feature points, the relative positions of the two cameras, and their intrinsic parameters as unknowns and minimized the reprojection error; such an approach is known as *bundle adjustment*. Since the tentative 3-D reconstruction from two images is indeterminate, they chose the one for which the first camera matrix is in a particular form ("canonical form").

Here, we point out that we can avoid this complication by directly minimizing Eq. (14) by the Levenberg-Marquardt (LM) method, using the parameterization of Bartoli and Sturm [1] (a preliminary version was presented in our conference paper [22]).

The fundamental matrix $\boldsymbol{F}$ has rank 2, so its SVD has the form

$$\boldsymbol{F} = \boldsymbol{U}{\rm diag}(\sigma_1, \sigma_2, 0)\boldsymbol{V}^\top, \tag{48}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, and σ_1 and σ_2 are the singular values. Since the normalization $\|\boldsymbol{F}\|^2 = 1$ is equivalent to $\sigma_1^2 + \sigma_2^2 = 1$ (Appendix A), we adopt the following parameterization[9]:

$$\sigma_1 = \cos\theta, \qquad \sigma_2 = \sin\theta. \tag{49}$$

The orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ have three degrees of freedom each, so they and θ constitute the seven degrees of freedom. However, the analysis becomes complicated if $\boldsymbol{U}$ and $\boldsymbol{V}$ are directly expressed in three parameters each (e.g., the Euler angles or the rotations around each coordinate axis). Following Bartoli and Sturm [1], we adopt the "Lie algebraic method": we represent the "increment" in $\boldsymbol{U}$ and $\boldsymbol{V}$ by three parameters each. Let ω_1, ω_2, and ω_3 represent the increment in $\boldsymbol{U}$, and ω_1', ω_2', and ω_3' in $\boldsymbol{V}$. The derivatives of Eq. (14) with respect to them are as follows (Appendix A):

$$\nabla_\omega J = 2\boldsymbol{F}_U^\top \boldsymbol{X}\boldsymbol{u}, \qquad \nabla_{\omega'} J = 2\boldsymbol{F}_V^\top \boldsymbol{X}\boldsymbol{u}. \tag{50}$$

Here, $\boldsymbol{X}$ is the matrix in Eq. (21), and $\boldsymbol{F}_U$, and $\boldsymbol{F}_V$ are defined by

$$\boldsymbol{F}_U = \begin{pmatrix} 0 & F_{31} & -F_{21} \\ 0 & F_{32} & -F_{22} \\ 0 & F_{33} & -F_{23} \\ -F_{31} & 0 & F_{11} \\ -F_{32} & 0 & F_{12} \\ -F_{33} & 0 & F_{13} \\ F_{21} & -F_{11} & 0 \\ F_{22} & -F_{12} & 0 \\ F_{23} & -F_{13} & 0 \end{pmatrix},$$

[8] Taubin [23] did not take the covariance matrix into account. This is a modification of his method.

[9] Bartoli and Sturm [1] took the ratio $\gamma = \sigma_2/\sigma_1$ as a variable. Here, we adopt the angle θ for the symmetry. As is well known, it has the value $\pi/4$ (i.e., $\sigma_1 = \sigma_2$) if the principal point is at the origin $(0, 0)$ and if there are no image distortions [7, 9].

$$
\boldsymbol{F}_V = \begin{pmatrix} 0 & F_{13} & -F_{12} \\ -F_{13} & 0 & F_{11} \\ F_{12} & -F_{11} & 0 \\ 0 & F_{23} & -F_{22} \\ -F_{23} & 0 & F_{21} \\ F_{22} & -F_{21} & 0 \\ 0 & F_{33} & -F_{32} \\ -F_{33} & 0 & F_{31} \\ F_{32} & -F_{31} & 0 \end{pmatrix}. \tag{51}
$$

The derivative of Eq. (14) with respect to θ has the form (Appendix A)

$$
\frac{\partial J}{\partial \theta} = 2(\boldsymbol{u}_\theta, \boldsymbol{X}\boldsymbol{u}), \tag{52}
$$

where we define

$$
\boldsymbol{u}_\theta = \begin{pmatrix} U_{12}V_{12}\cos\theta - U_{11}V_{11}\sin\theta \\ U_{12}V_{22}\cos\theta - U_{11}V_{21}\sin\theta \\ U_{12}V_{32}\cos\theta - U_{11}V_{31}\sin\theta \\ U_{22}V_{12}\cos\theta - U_{21}V_{11}\sin\theta \\ U_{22}V_{22}\cos\theta - U_{21}V_{21}\sin\theta \\ U_{22}V_{32}\cos\theta - U_{21}V_{31}\sin\theta \\ U_{32}V_{12}\cos\theta - U_{31}V_{11}\sin\theta \\ U_{32}V_{22}\cos\theta - U_{31}V_{21}\sin\theta \\ U_{32}V_{32}\cos\theta - U_{31}V_{31}\sin\theta \end{pmatrix}. \tag{53}
$$

Adopting Gauss-Newton approximation, which amounts to ignoring terms involving $(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)$, we obtain the second derivatives as follows (Appendix A):

$$
\nabla^2_\omega J = 2\boldsymbol{F}_U^\top \boldsymbol{M}\boldsymbol{F}_U, \qquad \nabla^2_{\omega'} J = 2\boldsymbol{F}_V^\top \boldsymbol{M}\boldsymbol{F}_V,
$$
$$
\nabla_{\omega\omega'} J = 2\boldsymbol{F}_U^\top \boldsymbol{M}\boldsymbol{F}_V, \qquad 2\frac{\partial J^2}{\partial\theta^2} = (\boldsymbol{u}_\theta, \boldsymbol{M}\boldsymbol{u}_\theta),
$$
$$
\frac{\partial \nabla_\omega J}{\partial\theta} = 2\boldsymbol{F}_U^\top \boldsymbol{M}\boldsymbol{u}_\theta, \qquad \frac{\partial \nabla_{\omega'} J}{\partial\theta} = 2\boldsymbol{F}_V^\top \boldsymbol{M}\boldsymbol{u}_\theta. \tag{54}
$$

The 7-parameter LM search goes as follows:

1. Initialize $\boldsymbol{F} = \boldsymbol{U}\mathrm{diag}(\cos\theta, \sin\theta, 0)\boldsymbol{V}^\top$.
2. Compute J in Eq. (14), and let c = 0.0001.
3. Compute $\boldsymbol{F}_U$, $\boldsymbol{F}_V$, and $\boldsymbol{u}_\theta$ in Eqs. (51) and (53).
4. Compute $\boldsymbol{X}$ in Eq. (21), the first derivatives in Eqs. (50) an (52), and the second derivatives in Eqs. (54).
5. Compute the following matrix $\boldsymbol{H}$:
$$
\boldsymbol{H} = \begin{pmatrix} \nabla^2_\omega J & \nabla_{\omega\omega'} J & \partial\nabla_\omega J/\partial\theta \\ (\nabla_{\omega\omega'} J)^\top & \nabla^2_{\omega'} J & \partial\nabla_{\omega'} J/\partial\theta \\ (\partial\nabla_\omega J/\partial\theta)^\top & (\partial\nabla_{\omega'} J/\partial\theta)^\top & \partial J^2/\partial\theta^2 \end{pmatrix}. \tag{55}
$$
6. Solve the 7-D simultaneous linear equations
$$
(\boldsymbol{H} + cD[\boldsymbol{H}]) \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\omega}' \\ \Delta\theta \end{pmatrix} = -\begin{pmatrix} \nabla_\omega J \\ \nabla_{\omega'} J \\ \partial J/\partial\theta \end{pmatrix}, \tag{56}
$$
for $\boldsymbol{\omega}$, $\boldsymbol{\omega}'$, and $\Delta\theta$, where $D[\,\cdot\,]$ denotes the diagonal matrix obtained by taking out only the diagonal elements.
7. Update $\boldsymbol{U}$, $\boldsymbol{V}$, and θ by
$$
\boldsymbol{U}' = \mathcal{R}(\boldsymbol{\omega})\boldsymbol{U}, \quad \boldsymbol{V}' = \mathcal{R}(\boldsymbol{\omega}')\boldsymbol{V}, \quad \theta' = \theta + \Delta\theta, \tag{57}
$$
where $\mathcal{R}(\boldsymbol{\omega})$ denotes rotation around $N[\boldsymbol{\omega}]$ by angle $\|\boldsymbol{\omega}\|$.
8. Update $\boldsymbol{F}$ as follows:
$$
\boldsymbol{F}' = \boldsymbol{U}'\mathrm{diag}(\cos\theta', \sin\theta', 0)\boldsymbol{V}'^\top. \tag{58}
$$
9. Let J' be the value of Eq. (14) for $\boldsymbol{F}'$.
10. Unless $J' < J$ or $J' \approx J$, let $c \leftarrow 10c$, and go back to Step 6.
11. If $\boldsymbol{F}' \approx \boldsymbol{F}$, return $\boldsymbol{F}'$ and stop. Else, let $\boldsymbol{F} \leftarrow \boldsymbol{F}'$, $\boldsymbol{U} \leftarrow \boldsymbol{U}'$, $\boldsymbol{V} \leftarrow \boldsymbol{V}'$, $\theta \leftarrow \theta'$, and $c \leftarrow c/10$, and go back to Step 3.

5. External Access

The external access approach does iterations in the 9-D $\boldsymbol{u}$-space in such a way that an optimal solution satisfying the rank constraint automatically results (Fig. 1(c)). The concept dates back to such heuristics as introducing penalties to the violation of the constraints or projecting the solution onto the surface of the constraints in the course of iterations, but it is Chojnacki et al. [4] that first presented a systematic scheme, which they called CFNS.

Stationarity Condition. According to the variational principle, the necessary and sufficient condition for the function J to be stationary at a point $\boldsymbol{u}$ in $\mathcal{S}^8$ in $\mathcal{R}^9$ is that its gradient $\nabla_{\mathbf{u}} J$ is orthogonal to the hypersurface defined by $\det \boldsymbol{u} = 0$ or by Eq. (10), and its surface normal is given by $\boldsymbol{u}^\dagger$ in Eq. (9). However, $\nabla_{\mathbf{u}} J = \boldsymbol{X}\boldsymbol{u}$ is always tangent to $\mathcal{S}^8$, because of the identity $(\boldsymbol{u}, \nabla_{\mathbf{u}} J) = (\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) = 0$ in $\boldsymbol{u}$. Hence, $\nabla_{\mathbf{u}} J$ should be parallel to $\boldsymbol{u}^\dagger$. This means that if we define the projection matrix

$$
\boldsymbol{P}_{\mathbf{u}^\dagger} = \boldsymbol{I} - \boldsymbol{u}^\dagger \boldsymbol{u}^{\dagger\top} \tag{59}
$$

onto the direction orthogonal to $\boldsymbol{u}^\dagger$, the stationarity condition is written as

$$
\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{X}\boldsymbol{u} = \boldsymbol{0}. \tag{60}
$$

The rank constraint of Eq. (10) is written as $\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{u} = \boldsymbol{u}$. Combined with Eq. (60), the desired solution should be such that

$$
\boldsymbol{Y}\boldsymbol{u} = \boldsymbol{0}, \qquad \boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{u} = \boldsymbol{u}, \tag{61}
$$

where we define

$$
\boldsymbol{Y} = \boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{X}\boldsymbol{P}_{\mathbf{u}^\dagger}. \tag{62}
$$

CFNS. Chojnacki et al. [4] showed that the stationarity condition of Eqs. (61) is written as a single equation in the form

$$
\boldsymbol{Q}\boldsymbol{u} = \boldsymbol{0}, \tag{63}
$$

where $\boldsymbol{Q}$ is a rather complicated symmetric matrix. They proposed to solve this by iterations in the same form as their FNS and called it *CFNS* (*Constrained FNS*):

1. Initialize $\boldsymbol{u}$.
2. Compute the matrix $\boldsymbol{Q}$.
3. Solve the eigenvalue problem

$$\boldsymbol{Q}\boldsymbol{u}' = \lambda \boldsymbol{u}', \tag{64}$$

   and compute the unit eigenvector $\boldsymbol{u}'$ for the eigenvalue λ closest to 0.
4. If $\boldsymbol{u}' \approx \boldsymbol{u}$ up to sign, return $\boldsymbol{u}'$ and stop. Else, let $\boldsymbol{u} \leftarrow \boldsymbol{u}'$, and go back to Step 2.

Infinitely many candidates exist for the matrix $\boldsymbol{Q}$ with which the problem is written as Eq. (13), but not all of them allow the above iterations to converge. Chojnacki et al. [4] gave one (see Appendix B), but the derivation is not written in their paper. Later, we show that CFNS does not necessarily converge to a correct solution.

EFNS. We now present a new iterative scheme, which we call *EFNS* (*Extended FNS*), for solving Eqs. (61) (a preliminary version was presented in a more abstract form in our conference paper [12]). The procedure goes as follows:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrix $\boldsymbol{X}$ in Eq. (21).
3. Computer the projection matrix $\boldsymbol{P}_{\mathbf{u}^\dagger}$ ($\boldsymbol{u}^\dagger$ is defined by Eq. (9)):

$$\boldsymbol{P}_{\mathbf{u}^\dagger} = \boldsymbol{I} - \boldsymbol{u}^\dagger \boldsymbol{u}^{\dagger\top}. \tag{65}$$

4. Compute the matrix $\boldsymbol{Y}$ in Eq. (62).
5. Solve the eigenvalue problem

$$\boldsymbol{Y}\boldsymbol{v} = \lambda \boldsymbol{v}, \tag{66}$$

   and compute the two unit eigenvectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ for the smallest eigenvalues in absolute terms.
6. Compute the following vector $\hat{\boldsymbol{u}}$:

$$\hat{\boldsymbol{u}} = (\boldsymbol{u}, \boldsymbol{v}_1)\boldsymbol{v}_1 + (\boldsymbol{u}, \boldsymbol{v}_2)\boldsymbol{v}_2 \tag{67}$$

7. Compute

$$\boldsymbol{u}' = N[\boldsymbol{P}_{\mathbf{u}^\dagger}\hat{\boldsymbol{u}}]. \tag{68}$$

8. If $\boldsymbol{u}' \approx \boldsymbol{u}$, return $\boldsymbol{u}'$ and stop. Else, let $\boldsymbol{u} \leftarrow N[\boldsymbol{u} + \boldsymbol{u}']$ and go back to Step 2.

Justification. We first show that when the above iterations have converged, the eigenvectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ both have eigenvalue 0. From the definition of $\boldsymbol{Y}$ in Eq. (62) and $\boldsymbol{P}_{\mathbf{u}^\dagger}$ in Eq. (65), $\boldsymbol{u}^\dagger$ is always an eigenvector of $\boldsymbol{Y}$ with eigenvalue 0 and is equal to $\boldsymbol{u}^\dagger$ up to sign. 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 Eq. (67) belongs to the linear span of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ ($= \pm\boldsymbol{u}^\dagger$) and the vector $\boldsymbol{u}'$ in Eq. (68) is a projection of $\hat{\boldsymbol{u}}$ within that linear span onto the direction orthogonal to $\boldsymbol{u}^\dagger$. Hence, it coincides with $\pm\boldsymbol{v}_1$.

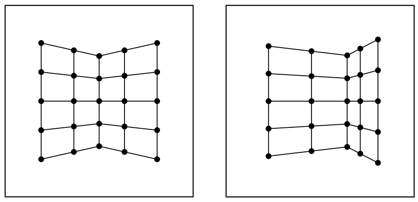


Figure 3. Simulated images of planar grid surfaces.

The iterations converge when $\boldsymbol{u} = \boldsymbol{u}'$ ($= \pm\boldsymbol{v}_1$). Thus, $\boldsymbol{v}_1$ is an eigenvector of $\boldsymbol{Y}$ with eigenvalue λ. Hence, $\boldsymbol{u}$ also satisfies Eq. (66). Computing the inner product with $\boldsymbol{u}$ on both sides, we have

$$(\boldsymbol{u}, \boldsymbol{Y}\boldsymbol{u}) = \lambda. \tag{69}$$

On the other hand, $\boldsymbol{u}$ ($= \pm\boldsymbol{v}_1$) is orthogonal to $\boldsymbol{u}^\dagger$ ($= \pm\boldsymbol{v}_2$), so

$$\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{u} = \boldsymbol{u}. \tag{70}$$

Hence,

$$(\boldsymbol{u}, \boldsymbol{Y}\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{X}\boldsymbol{P}_{\mathbf{u}^\dagger}\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) = 0, \tag{71}$$

since $(\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) = 0$ is an identity in $\boldsymbol{u}$ (see Eqs. (30)). Eqs. (69) and (71) contradict our assumption that $\lambda \neq 0$. So, $\boldsymbol{v}_1$ is also an eigenvector of $\boldsymbol{Y}$ with eigenvalue 0. □

It follows that both $\boldsymbol{X}\boldsymbol{u} = \boldsymbol{0}$ and Eq. (70) hold, and thus $\boldsymbol{u}$ is the desired solution. Of course, this conclusion relies on the premise that the iterations converge. According to our experience, if we let $\boldsymbol{u} \leftarrow \boldsymbol{u}'$ in Step 9, the next value of $\boldsymbol{u}'$ computed in Step 8 often reverts to the former value of $\boldsymbol{u}$, falling in infinite looping. So, we update $\boldsymbol{u}$ to the "midpoint" $(\boldsymbol{u}' + \boldsymbol{u})/2$ and normalized it to a unit vector $N[\boldsymbol{u}' + \boldsymbol{u}]$ in Step 9. By this, the iterations converged in all of our experiments.

CFNS vs. EFNS. Figure 3 shows simulated images of two planar grid surfaces viewed from different angles. 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 by CFNS and EFNS.

Figure 4 shows a typical instance ($\sigma = 1$) of the convergence of the determinant $\det \boldsymbol{F}$ and the residual J from different initial values. In the final step, $\det \boldsymbol{F}$ is forced to be 0 by SVD, as prescribed by Chojnacki et al. [4]. The dotted lines show the values to be converged.

The LS solution has a very low residual J, since the rank constraint $\det \boldsymbol{F} = 0$ is ignored. So, J needs to be increased to achieve $\det \boldsymbol{F} = 0$, but CFNS fails to do so. As a result, $\det \boldsymbol{F}$ remains nonzero and drops to 0 by the final SVD correction, causing a sudden jump in J. If we start from SVD-corrected LS, the residual J first increases, making $\det \boldsymbol{F}$ nonzero, but in the end both J and $\det \boldsymbol{F}$ converge in an

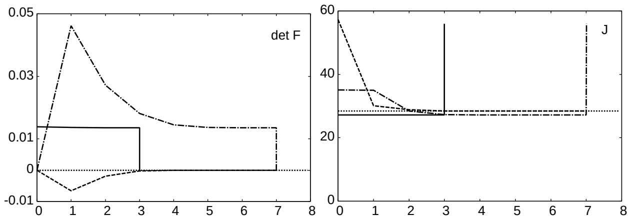


Figure 4. The convergence of $\det F$ and the residual J (σ = 1) for different initializations: LS (solid line), SVD-corrected LS (dashed line), and the true value (chained line). All solutions are SVD-corrected in the final step.

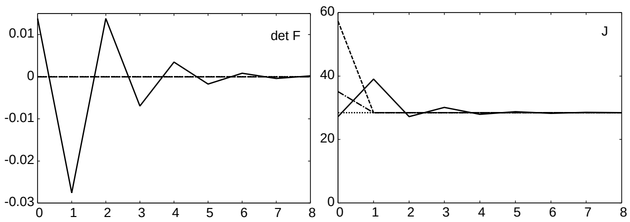


Figure 5. The results by EFNS corresponding to Fig. 5.

expected way. In contrast, the true value has a very large J, so CFNS tries to decrease it sharply at the cost of too much increase in $\det \boldsymbol{F}$, which never reverts to 0 until the final SVD. Figure 5 shows corresponding results by EFNS. Both J and $\det \boldsymbol{F}$ converge to their correct values with stably attenuating oscillations. Figures 6 and 7 show the results corresponding to Fig. 4 and 5 for another instance (σ = 3). We can observe similar behavior of CFNS and EFNS.

We mean by "convergence" the state of the same solution repeating itself in the course of iterations. In mathematical terms, the resulting solution is a *fixed point* of the iteration operator, i.e., the procedure to update the current solution. In [4], Chojnacki et al. [4] proved that the solution $\boldsymbol{u}$ satisfying Eqs. (61) is a fixed point of their CFNS. Apparently, they expected to arrive at that solution by their scheme. As demonstrated by Figs. 4, and 6, however, CFNS has many other fixed points, and which to arrive at depends on initialization. In contrast, we have proved that any fixed point of EFNS is *necessarily* the desired solution. This cannot be proved for CFNS.

6. Accuracy Comparison

Using the simulated images in Fig. 3, we compare the accuracy of the following methods:

1) SVD-corrected LS (Hartley's 8-point method),
2) SVD-corrected ML,

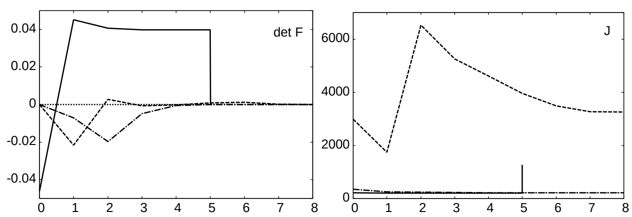


Figure 6. The convergence of $\det F$ and the residual J (σ = 3) for different initializations: LS (solid line), SVD-corrected LS (dashed line), and the true value (chained line). All solutions are SVD-corrected in the final step.

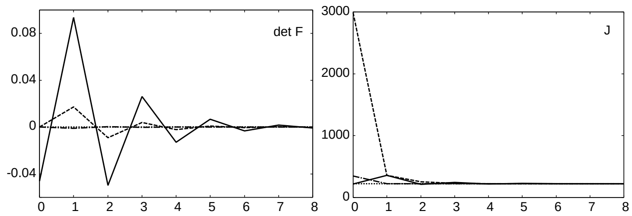


Figure 7. The results by EFNS corresponding to Fig. 6.

3) CFNS of Chojnacki et al.
4) Optimally corrected ML,
5) 7-parameter LM.
6) EFNS.

For brevity, we use the shorthand "ML" for unconstrained minimization of Eq. (14). For this, we used the FNS of Chojnacki et al. [3] initialized by LS. We confirmed that FNS, HEIV, and the projective Gauss-Newton iterations all converged to the same solution (up to rounding errors), although the speed of convergence varies (see [11] for the convergence comparison). We initialized the 7-parameter LM, CFNS, and EFNS by LS. All iterations are stopped when the update of $\boldsymbol{F}$ is less than 10^{-6} in norm.

Figure 8 plots for σ on the horizontal axis the following root-mean-square (RMS) error D corresponding to Eq. (8) over 10000 independent trials:

$$D = \sqrt{\frac{1}{10000} \sum_{a=1}^{10000} \| \boldsymbol{P}_{\mathcal{U}} \hat{\boldsymbol{u}}^{(a)} \|^2}. \tag{72}$$

Here, $\hat{\boldsymbol{u}}^{(a)}$ is the ath value, and $\boldsymbol{P}_{\mathcal{U}}$ is the projection matrix in Eq. (11); since the solution is always normalized into a unit vector, we measure the deviation of $\hat{\boldsymbol{u}}^{(a)}$ from $\boldsymbol{u}$ by orthogonally projecting $\hat{\boldsymbol{u}}^{(a)}$ onto the tangent space $T_{\mathbf{u}}(\mathcal{U})$ to $\mathcal{U}$ at $\boldsymbol{u}$ (see Eq. 8 and Fig. 2). The dotted line is the

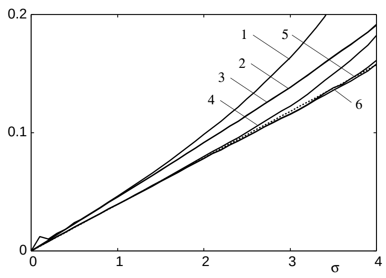


Figure 8. The RMS error D vs. noise level σ for Fig. 3. 1) SVD-corrected LS. 2) SVD-corrected ML. 3) CFNS. 4) Optimally corrected ML. 5) 7-parameter LM. 6) EFNS. The dotted line indicates the KCR lower bound.

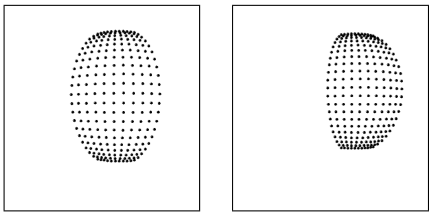


Figure 9. Simulated images of a spherical grid surface.

bound implied by the KCR lower bound (the trace of the right-hand side of Eq. (12)).

Note that the RMS error describes the "variation" from the true value; the computed solution is sometimes very close to the true value, other times very far from it, and D measures the "standard deviation".

Figure 9 shows simulated images (600×600 pixels) of a spherical grid surface viewed from different angles. We did similar experiments. Figure 10 shows the results corresponding to Fig. 8.

Preliminary observations. We can see that SVD-corrected LS (Hartley's 8-point algorithm) performs very poorly. We can also see that SVD-corrected ML is inferior to optimally corrected ML, whose accuracy is close to the KCR lower bound. The accuracy of the 7-parameter LM is nearly the same as optimally corrected ML when the noise is small but gradually outperforms it as the noise increases. Best performing is EFNS, exhibiting nearly the same accuracy as the KCR lower bound. The CFNS performs as poorly as SVD-corrected ML, because, as we observed in the preceding section, it is likely to stop at the unconstrained ML solution (we forced the determinant to be zero by SVD).

Doing many experiments (not all shown here), we have observed that:

i) The EFNS stably achieves the highest accuracy over a wide range of the noise level.

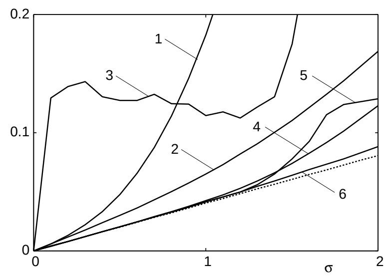


Figure 10. The RMS error D vs. noise level σ for Fig. 9. 1) SVD-corrected LS. 2) SVD-corrected ML. 3) CFNS. 4) Optimally corrected ML. 5) 7-parameter LM. 6) EFNS. The dotted line indicates the KCR lower bound.

ii) Optimally corrected ML is fairly accurate and very robust to noise but gradually deteriorates as noise grows.

iii) The 7-parameter LM achieves very high accuracy when started from a good initial value but is likely to fall into local minima if poorly initialized.

The robustness of EFNS and optimally corrected ML is due to the fact that the computation is done in the redundant ("external") $\boldsymbol{u}$-space, where J has a simple form of Eq. (14). In fact, we have never experienced local minima in our experiments. The deterioration of optimally corrected ML in the presence of large noise is because linear approximation is involved in Eq. (18).

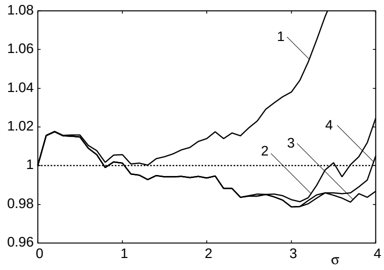
The fragility of the 7-parameter LM is attributed to the complexity of the function J when expressed in seven parameters, resulting in many local minima in the reduced ("internal") parameter space, as pointed out in [20].

Thus, the optimal correction of ML and the 7-parameter ML have complementary characteristics, which suggests that the 7-parameter ML started from optimally corrected ML may exhibit comparable accuracy to EFNS. We now confirm this.

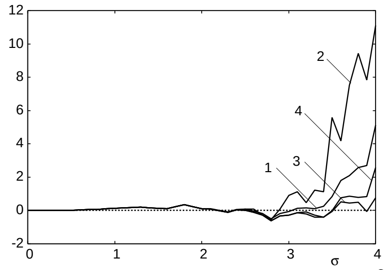
Detailed observations. Figure 11 compares for the images in Fig. 3:

1) optimally corrected ML.
2) 7-parameter LM started from LS.
3) 7-parameter LM started from optimally corrected ML.
4) EFNS.

For visual ease, we plot in Fig. 11(a) the ratio $D/D_{\rm KCR}$ of D in Eq. (72) to the corresponding KCR lower bound. Figure 11(b) plots the average residual $\hat{J}$ (minimum of Eq. (14)). Since direct plots of $\hat{J}$ nearly overlap, we plot here the difference $\hat{J} - (N-7)\sigma^2$, where N is the number of corresponding pairs. This is motivated by the fact that to a first approximation $\hat{J}/\sigma^2$ is subject to a χ^2 distribution with $N - 7$ degrees of freedom [9], so the expectation of

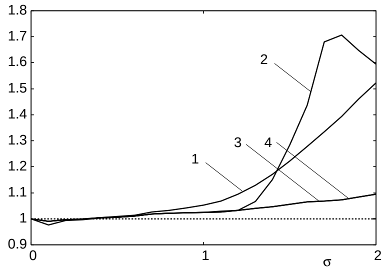


(a) The RMS error relative to the KCR lower bound.

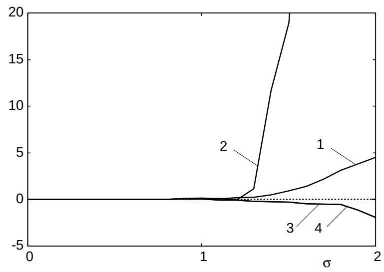


(b) Average residual minus minus $(N-7)\sigma^2$.

Figure 11. The RMS error and the average residual for Fig. 3. 1) Optimally corrected ML. 2) 7-parameter LM started from LS. 3) 7-parameter LM started from optimally corrected ML. 4) EFNS.



(a) The RMS error relative to the KCR lower bound.



(b) Average residual minus minus $(N-7)\sigma^2$.

Figure 12. The RMS error and the average residual for Fig. 9. 1) Optimally corrected ML. 2) 7-parameter LM started from LS. 3) 7-parameter LM started from optimally corrected ML. 4) EFNS.

$\hat{J}$ is approximately $(N-7)\sigma^2$. Figure 12 shows the corresponding results for Fig. 9. We observe:

i) The RMS error of optimally corrected ML increases as noise increases, *yet* the corresponding residual remains low.

ii) The 7-parameter LM started from LS appears to have high accuracy for noise levels for which the corresponding residual high.

iii) The accuracy of the 7-parameter LM improves if started from optimally corrected ML, resulting in the accuracy is comparable to EFNS.

The seemingly contradictory fact that solutions that are closer to the true value (measured by RMS) have higher residuals $\hat{J}$ means that the LM search failed to reach the true minimum of the function J, indicating existence of local minima located closer to the true value than to the true minimum of J. When started from optimally corrected ML, the LM search successfully reaches the true minimum of J, resulting in the smaller $\hat{J}$ but larger RMS errors.

RMS vs. KCR Lower Bound. One may wonder why the computed RMS errors are sometimes below the KCR lower bound. There are several reasons for this.

The KCR lower bound is shown here for a convenient reference, but it does not mean that errors of the values computed by any *algorithm* should be above it; it is a lower bound on *unbiased estimators*. By "estimator", we mean a *function* of the data, e.g., the minimizer of a given cost function. An iterative algorithm such as LM does not qualify as an estimator, since the final value depends not only the data but also on the starting value; the resulting value may not be the true minimizer of the cost function. Thus, it may happen, as we have observed above, that a solution closer to the true value has higher residual.

Next, the KCR lower bound is derived, without any approximation [9], from the starting identity that the expectation of the estimator (as a function of the data) should coincide with its true value. This is a very strong identity, from which we can derive the KCR lower bound in the same way as the Cramer-Rao lower bound is derived from the unbiasedness constraint in the framework of traditional statistical estimation. However, the ML estimator or the minimizer of the function J in Eq. (14) may not necessarily be unbiased when the noise is large. In fact, it has been reported that removing bias from the ML solution can result in better accuracy ("hyperaccuracy") for ellipse fitting in the presence of large noise [10].

Finally, the RMS error is computed from "finite" samples, not theoretical expectation. We did 10000 independent trials for each σ, but the result still has fluctuations. Theoretically, the plot should be a smooth function of σ, but zigzags remain to some extent how many samples we use.

Which is better?. We have seen the best performance by the 7-parameter ML started from optimally corrected ML and by EFNS. We tested which is really better by doing a hybrid method: we do both and choose the solution that

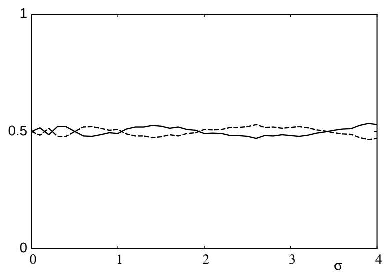


Figure 13. The ratio of the solution being chosen for Fig. 3. Solid line: 7-parameter LM started from optimally corrected ML. Dashed line: EFNS.

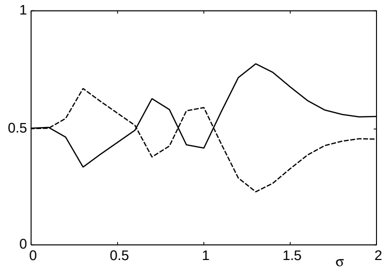


Figure 14. The ratio of the solution being chosen for Fig. 9. Solid line: 7-parameter LM started from optimally corrected ML. Dashed line: EFNS.

has a smaller value of J. Figure 13 plots the ratio of each solution being chosen for the images in Fig. 3; Figure 14 plots the corresponding result for Fig. 9. As we can see, the two are completely even; there is no distinction between them.

7. Bundle Adjustment (Gold Standard)

There is a subtle point to be clarified in the discussion of Sect. 2. The transition from Eq. (13) to Eq. (14) is exact; no approximation is involved. Although terms of $O(\sigma^4)$ are omitted and the true values are replaced by their data in Eq. (7), it is numerically confirmed that these do not affect the final results in any noticeable way.

However, although the "analysis" may be exact, the "interpretation" is not strict. Namely, despite the fact that Eq. (14) is the (squared) Mahalanobis distance in the $\boldsymbol{\xi}$-space, its minimization can be ML only when the noise in the $\boldsymbol{\xi}$-space is Gaussian, because then and only then is the likelihood proportional to $e^{-J/\text{constant}}$. Strictly speaking, if the noise in the image plane is Gaussian, the transformed noise in the $\boldsymbol{\xi}$-space is no longer Gaussian, so the proviso that "If the noise in $\{\boldsymbol{\xi}_\alpha\}$ is ..." above Eq. (13) (and for the KCR lower bound of Eq. (12), too) does not necessarily hold, and minimizing Eq. (14) is not strictly ML in the image plane.

In order to test how much difference is incurred by this, we minimize the Mahalanobis distance in the $\{\boldsymbol{x}, \boldsymbol{x}'\}$-space, called the *reprojection error*. This approach was endorsed by Hartley and Zisserman [7], who called it the *Gold Standard*.

This is usually done as search in a high-dimensional parameter space, as done by Bartoli and Sturm [1], computing tentative 3-D reconstruction and adjusting the reconstructed shape, 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*.

Here, we present a new numerical scheme for directly minimizing the reprojection error without reference to any tentative 3-D reconstruction (this result has not been presented anywhere else). Then, we compare its accuracy with those methods we described so far.

Problem. We minimize the reprojection error

$$E = \sum_{\alpha=1}^{N} \Big(\|\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha\|^2 + \|\boldsymbol{x}'_\alpha - \bar{\boldsymbol{x}}'_\alpha\|^2 \Big), \tag{73}$$

with respect to $\bar{\boldsymbol{x}}_\alpha$, $\bar{\boldsymbol{x}}_\alpha$, α = 1, ..., N, and $\boldsymbol{F}$ (constrained to be $\|\boldsymbol{F}\|$ = 1 and det $\boldsymbol{F}$ = 0) subject to the epipolar constraint

$$(\bar{\boldsymbol{x}}_\alpha, \boldsymbol{F}\bar{\boldsymbol{x}}'_\alpha) = 0, \qquad \alpha = 1, ..., N. \tag{74}$$

First approximation. Instead of estimating $\bar{\boldsymbol{x}}_\alpha$ and $\bar{\boldsymbol{x}}'_\alpha$ directly, we express them as

$$\bar{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - \Delta\boldsymbol{x}_\alpha, \qquad \bar{\boldsymbol{x}}'_\alpha = \boldsymbol{x}'_\alpha - \Delta\boldsymbol{x}'_\alpha, \tag{75}$$

and estimate the correction terms $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$. Substituting Eqs. (75) into Eq. (73), we have

$$E = \sum_{\alpha=1}^{N} \Big(\|\Delta\boldsymbol{x}_\alpha\|^2 + \|\Delta\boldsymbol{x}'_\alpha\|^2 \Big). \tag{76}$$

The epipolar equation of Eq. (74) becomes

$$(\boldsymbol{x}_\alpha - \Delta\boldsymbol{x}_\alpha, \boldsymbol{F}(\boldsymbol{x}'_\alpha - \Delta\boldsymbol{x}'_\alpha)) = 0. \tag{77}$$

Ignoring the second order terms in the correction terms, we obtain to a first approximation

$$(\boldsymbol{F}\boldsymbol{x}'_\alpha, \Delta\boldsymbol{x}_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \Delta\boldsymbol{x}'_\alpha) = (\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha). \tag{78}$$

Since the correction terms $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$ are constrained to be in the image plane, we have the constraints

$$(\boldsymbol{k}, \Delta\boldsymbol{x}_\alpha) = 0, \qquad (\boldsymbol{k}, \Delta\boldsymbol{x}'_\alpha) = 0, \tag{79}$$

where we define $\boldsymbol{k} \equiv (0,0,1)^\top$. Introducing Lagrange multipliers for Eqs. (75) and (79), we can easily determine $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$ that minimize Eq. (76) as follows (Appendix C):

$$\begin{aligned} \Delta\boldsymbol{x}_\alpha &= \frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)\boldsymbol{P}_\mathbf{k}\boldsymbol{F}\boldsymbol{x}'_\alpha}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_\mathbf{k}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_\mathbf{k}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}, \\ \Delta\boldsymbol{x}'_\alpha &= \frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)\boldsymbol{P}_\mathbf{k}\boldsymbol{F}^\top\boldsymbol{x}_\alpha}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_\mathbf{k}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_\mathbf{k}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}. \end{aligned} \tag{80}$$

Here, we define

$$\boldsymbol{P}_{\mathbf{k}} \equiv \mathrm{diag}(1,1,0). \tag{81}$$

Substituting Eq. (80) into Eq. (76), we obtain (Appendix C)

$$E = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)^2}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top \boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top \boldsymbol{x}_\alpha)}, \tag{82}$$

which is known as the *Sampson error* [7]. Suppose we have obtained the matrix $\boldsymbol{F}$ that minimizes this subject to $\|\boldsymbol{F}\|$ = 1 and $\det \boldsymbol{F}$ = 0. Writing it as $\hat{\boldsymbol{F}}$ and substituting it into Eq. (80), we obtain the solution

$$\hat{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - \frac{(\boldsymbol{x}_\alpha, \hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha}{(\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha) + (\hat{\boldsymbol{F}}^\top \boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top \boldsymbol{x}_\alpha)},$$
$$\hat{\boldsymbol{x}}'_\alpha = \boldsymbol{x}'_\alpha - \frac{(\boldsymbol{x}_\alpha, \hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top \boldsymbol{x}_\alpha}{(\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}\boldsymbol{x}'_\alpha) + (\hat{\boldsymbol{F}}^\top \boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\boldsymbol{F}}^\top \boldsymbol{x}_\alpha)}. \tag{83}$$

Second approximation. Eqs. (83) give only a first approximation solution. So, we estimate the true solution by writing, instead of Eqs. (75),

$$\bar{\boldsymbol{x}}_\alpha = \hat{\boldsymbol{x}}_\alpha - \Delta\hat{\boldsymbol{x}}_\alpha, \qquad \bar{\boldsymbol{x}}'_\alpha = \hat{\boldsymbol{x}}'_\alpha - \Delta\hat{\boldsymbol{x}}'_\alpha, \tag{84}$$

and by estimating the correction terms $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$, which are small quantities of higher order than the first order terms $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$ in Eqs. (75). Substitution of Eqs. (84) into Eq. (75) yields

$$E = \sum_{\alpha=1}^{N} \Big(\|\tilde{\boldsymbol{x}}_\alpha + \Delta\hat{\boldsymbol{x}}_\alpha\|^2 + \|\tilde{\boldsymbol{x}}'_\alpha + \Delta\hat{\boldsymbol{x}}'_\alpha\|^2 \Big), \tag{85}$$

where we define

$$\tilde{\boldsymbol{x}}_\alpha = \boldsymbol{x}_\alpha - \hat{\boldsymbol{x}}_\alpha, \qquad \tilde{\boldsymbol{x}}'_\alpha = \boldsymbol{x}'_\alpha - \hat{\boldsymbol{x}}'_\alpha. \tag{86}$$

The epipolar equation of Eq. (74) now becomes

$$(\hat{\boldsymbol{x}}_\alpha - \Delta\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}(\hat{\boldsymbol{x}}'_\alpha - \Delta\hat{\boldsymbol{x}}'_\alpha)) = 0. \tag{87}$$

Ignoring second order terms of $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$, which are themselves of higher order, we have

$$(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \Delta\hat{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \Delta\hat{\boldsymbol{x}}'_\alpha) = (\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha). \tag{88}$$

This is a higher order approximation of Eq. (74) than the first order approximation in Eq. (78). Introducing Lagrange multipliers to Eq. (88) and the constraints

$$(\boldsymbol{k}, \Delta\hat{\boldsymbol{x}}_\alpha) = 0, \qquad (\boldsymbol{k}, \Delta\hat{\boldsymbol{x}}'_\alpha) = 0 \tag{89}$$

we can obtain $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$ that minimize Eq. (82) in the following form (Appendix C):

$$\Delta\hat{\boldsymbol{x}}_\alpha = \frac{e_\alpha \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)} - \tilde{\boldsymbol{x}}_\alpha,$$
$$\Delta\hat{\boldsymbol{x}}'_\alpha = \frac{e_\alpha \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)} - \tilde{\boldsymbol{x}}'_\alpha. \tag{90}$$

Here, we define

$$e_\alpha = (\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha). \tag{91}$$

On substation of Eq. (90), Eq. (85) now has the following form (Appendix C):

$$E = \sum_{\alpha=1}^{N} \frac{e_\alpha^2}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)}. \tag{92}$$

Suppose we have obtained the matrix $\boldsymbol{F}$ that minimizes this subject to $\|\boldsymbol{F}\|$ = 1 and $\det \boldsymbol{F}$ = 0. Writing it as $\hat{\hat{\boldsymbol{F}}}$ and substituting it into Eq. (90), we obtain the solution

$$\hat{\hat{\boldsymbol{x}}}_\alpha = \boldsymbol{x}_\alpha - \frac{\hat{\hat{e}}_\alpha \boldsymbol{P}_{\mathbf{k}}\hat{\hat{\boldsymbol{F}}}\hat{\boldsymbol{x}}'_\alpha}{(\hat{\hat{\boldsymbol{F}}}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\hat{\boldsymbol{F}}}\hat{\boldsymbol{x}}'_\alpha) + (\hat{\hat{\boldsymbol{F}}}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\hat{\boldsymbol{F}}}^\top\hat{\boldsymbol{x}}_\alpha)},$$
$$\hat{\hat{\boldsymbol{x}}}'_\alpha = \boldsymbol{x}'_\alpha - \frac{\hat{\hat{e}}_\alpha \boldsymbol{P}_{\mathbf{k}}\hat{\hat{\boldsymbol{F}}}^\top\hat{\boldsymbol{x}}_\alpha}{(\hat{\hat{\boldsymbol{F}}}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\hat{\boldsymbol{F}}}\hat{\boldsymbol{x}}'_\alpha) + (\hat{\hat{\boldsymbol{F}}}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\hat{\hat{\boldsymbol{F}}}^\top\hat{\boldsymbol{x}}_\alpha)}, \tag{93}$$

where $\hat{\hat{e}}_\alpha$ is the value of Eq. (91) obtained by replacing $\boldsymbol{F}$ in it by $\hat{\hat{\boldsymbol{F}}}$. The resulting solution $\{\hat{\hat{\boldsymbol{x}}}_\alpha, \hat{\hat{\boldsymbol{x}}}'_\alpha\}$ is a better approximation than the solution $\{\hat{\boldsymbol{x}}_\alpha, \hat{\boldsymbol{x}}'_\alpha\}$ in Eqs. (83). We rewriting $\{\hat{\hat{\boldsymbol{x}}}_\alpha, \hat{\hat{\boldsymbol{x}}}'_\alpha\}$ as $\{\hat{\boldsymbol{x}}_\alpha, \hat{\boldsymbol{x}}'_\alpha\}$ and estimate yet better solution in the form of Eqs. (84). We repeat this until the iterations converge.

Fundamental matrix computation. The remaining problem is to compute the matrix $\boldsymbol{F}$ that minimizes Eqs. (82) and (92) subject to $\|\boldsymbol{F}\|$ = 1 and $\det \boldsymbol{F}$ = 0. If we use the representation in Eqs. (3) and (4), we can write

$$(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha) = \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)}{f_0^2}, \tag{94}$$

$$(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha) = \frac{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}{f_0^2}, \tag{95}$$

where $V_0[\boldsymbol{\xi}_\alpha]$ is the matrix in Eq. (7). From Eqs. (94) and (95), Eq. (82) is rewritten in the form

$$E = \frac{1}{f_0^2}\sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \tag{96}$$

which is Eq. (14) itself except the scale. Hence, the matrix $\boldsymbol{F}$ that minimizes this subject to $\|\boldsymbol{F}\|$ = 1 and $\det \boldsymbol{F}$ = 0 can be determined by the methods described in Sect. 3–5.

If we define

$$\hat{\boldsymbol{\xi}}_\alpha = \begin{pmatrix} \hat{x}_\alpha \hat{x}'_\alpha + \hat{x}'_\alpha \tilde{x}_\alpha + \hat{x}_\alpha \tilde{x}'_\alpha \\ \hat{x}_\alpha \hat{y}'_\alpha + \hat{y}'_\alpha \tilde{x}_\alpha + \hat{x}_\alpha \tilde{y}'_\alpha \\ f_0(\hat{x}_\alpha + \tilde{x}_\alpha) \\ \hat{y}_\alpha \hat{x}'_\alpha + \hat{x}'_\alpha \tilde{y}_\alpha + \hat{y}_\alpha \tilde{x}'_\alpha \\ \hat{y}_\alpha \hat{y}'_\alpha + \hat{y}'_\alpha \tilde{y}_\alpha + \hat{y}_\alpha \tilde{y}'_\alpha \\ f_0(\hat{y}_\alpha + \tilde{y}_\alpha) \\ f_0(\hat{x}'_\alpha + \tilde{x}'_\alpha) \\ f_0(\hat{y}'_\alpha + \tilde{y}'_\alpha) \\ f_0^2 \end{pmatrix}, \tag{97}$$

the expression e_α in Eq. (91) is written as

$$e_\alpha = \frac{(\boldsymbol{u}, \hat{\boldsymbol{\xi}}_\alpha)}{f_0^2}. \tag{98}$$

Hence, Eq. (92) is rewritten as

$$E = \frac{1}{f_0^2} \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \hat{\boldsymbol{\xi}}_\alpha)^2}{(\boldsymbol{u}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{u})}, \tag{99}$$

where $V_0[\hat{\boldsymbol{\xi}}_\alpha]$ is the matrix in Eq. (7) obtained by replacing x_α, y_α, x'_α, and y'_α by $\hat{x}_\alpha$, $\hat{y}'_\alpha$, $\hat{x}'_\alpha$, and $\hat{y}'_\alpha$, respectively. Since Eq. (99) again has the same for as Eq. (14) except the scale, we can obtain the matrix $\boldsymbol{F}$ that minimizes this subject to $\|\boldsymbol{F}\| = 1$ and $\det \boldsymbol{F} = 0$ can be determined by the methods in Sect. 3–5.

Procedure. Our bundle adjustment computation is summarized as follows.

1. Let $\boldsymbol{u}_0 = \mathbf{0}$.
2. For $\alpha = 1, ..., N$, let

$$\hat{x}_\alpha = x_\alpha, \quad \hat{y}_\alpha = y_\alpha, \quad \hat{x}'_\alpha = x'_\alpha, \quad \hat{y}'_\alpha = y'_\alpha,$$
$$\tilde{x}_\alpha = \tilde{y}_\alpha = \tilde{x}'_\alpha = \tilde{y}'_\alpha = 0. \tag{100}$$

3. Compute the vector $\hat{\boldsymbol{\xi}}_\alpha$, $\alpha = 1, ..., N$, in Eq. (97).
4. Compute the matrix $V_0[\hat{\boldsymbol{\xi}}_\alpha]$, $\alpha = 1, ..., N$, by replacing $x_\alpha, y_\alpha, x'_\alpha$, and y'_α by $\hat{x}_\alpha, \hat{y}'_\alpha, \hat{x}'_\alpha$, and $\hat{y}'_\alpha$, respectively in Eq. (7).
5. Compute the vector $\boldsymbol{u}$ that minimizes the following function E subject to $\|\boldsymbol{u}\| = 1$ and $(\boldsymbol{u}^\dagger, \boldsymbol{u}) = 0$:

$$E = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \hat{\boldsymbol{\xi}}_\alpha)^2}{(\boldsymbol{u}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{u})}. \tag{101}$$

6. 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 $\tilde{y}'_\alpha$ as follows:

$$\tilde{\boldsymbol{x}}_\alpha \leftarrow \frac{(\boldsymbol{u}, \hat{\boldsymbol{\xi}}_\alpha)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha}{(\boldsymbol{u}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{u})}, \quad \tilde{\boldsymbol{x}}'_\alpha \leftarrow \frac{(\boldsymbol{u}, \hat{\boldsymbol{\xi}}_\alpha)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha}{(\boldsymbol{u}, V_0[\hat{\boldsymbol{\xi}}_\alpha]\boldsymbol{u})}. \tag{102}$$

7. Go back to Step 3 after the following update:

$$\boldsymbol{u}_0 \leftarrow \boldsymbol{u}, \quad \hat{x}_\alpha \leftarrow x_\alpha - \tilde{x}_\alpha, \quad \hat{y}_\alpha \leftarrow y_\alpha - \tilde{y}_\alpha,$$
$$\hat{x}'_\alpha \leftarrow x'_\alpha - \tilde{x}'_\alpha, \quad \hat{y}'_\alpha \leftarrow y'_\alpha - \tilde{y}'_\alpha. \tag{103}$$

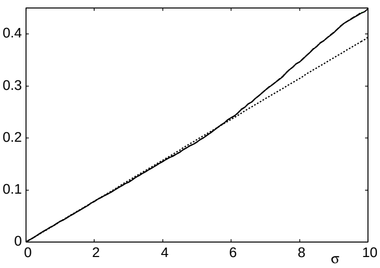


Figure 15. The RMS errors for Fig. 3. Dashed line: Sampson solution. Solid line: Gold Standard solution. Dotted line: KCR lower bound.

8. Effect of Bundle Adjustment

The above computation reduces to the Mahalanobis distance minimization in the $\boldsymbol{\xi}$-space if we stop at Step 5. So, the issue is how the accuracy improves by the subsequent iterations. Borrowing the terminology of Hartley and Zisserman [7], let us call the solution obtained at Step 5 the *Sampson solution* and the solution obtained after the iterations the *Gold Standard solution*.

Simulations. Using the simulated images in Fig. 3, we computed the RMS error D in Eq. (72) for 10000 trials. Figure 15 corresponds to Fig. 8 except that the horizontal axis is now extended to an extremely large noise level.

For minimizing Eq. (101), we used EFNS initialized by the Taubin method. If the iterations did not converge after 100 iterations, we switched to the projective Gauss-Newton iterations followed by optimal correction followed by the 7-parameter LM search. We did preliminary experiments for testing the convergence properties of FNS, HEIV, projective Gauss-Newton iterations, and EFNS and found that projective Gauss-Newton iterations and EFNS can tolerate larger noise than others.

As we can see from Fig. 15, the RMS errors of the Sampson and the Gold Standard solutions coincide in the plot; the two solutions did differ, but the difference is a few order smaller than the magnitude of the KCR lower bound.

Figure 16 compares the reprojection error for the two solutions. The dashed line is the value of E in Eq. (101) when the computation is stopped there, which equals the minimum of the function J in Eq. (14); we call it the *Sampson error*. For each Sampson solution $\boldsymbol{F}$ (or $\boldsymbol{u}$), we computed the reprojection error by minimizing E in Eq. (73) with respect to $\bar{\boldsymbol{x}}_\alpha$ and $\bar{\boldsymbol{x}}'_\alpha$, $\alpha = 1, ..., N$, subject to Eq. (74) with the computed $\boldsymbol{F}$ fixed. The computation goes the same as described in the preceding section except that $\boldsymbol{F}$ is fixed.

From Fig. 16, we observe that the Sampson error is very close to the first order estimate $(N - 7)\sigma^2/f_0^2$, reflecting the fact that the Sampson error is a first approximation to the reprojection error.

We also observe that the reprojection error of the Gold Standard solution is certainly smaller than the Sampson er-

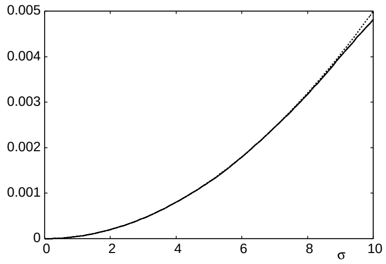


Figure 16. The reprojection error for Fig. 3. Chained line: Sampson residual. Dashed line: Sampson solution. Solid line: Gold Standard solution. Dotted line: $(N-7)\sigma^2/f_0^2$.

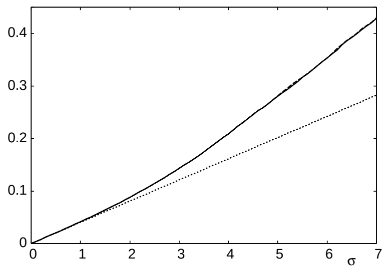


Figure 17. The RMS errors for Fig. 9. Dashed line: Sampson solution. Solid line: Gold Standard solution. Dotted line: KCR lower bound.

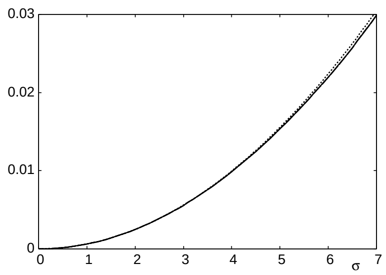


Figure 18. The reprojection error for Fig. 9. Chained line: Sampson error. Dashed line: Sampson solution. Solid line: Gold Standard solution. Dotted line: $(N-7)\sigma^2/f_0^2$.

ror if the noise level is above a certain level (about 6 pixels in this case), *yet* the reprojection error is virtually identical for the Sampson and the Gold Standard solutions.

Figures 17 and 18 show the results corresponding to Figs. 15 and 16 for Fig. 9. Again, we observe similar behavior.

Observations. From our experiments, we conclude that 1) the reprojection is smaller than the Sampson error if the noise is very large, but that 2) *the solution that minimizes the Sampson error also minimizes the reprojection error, and vice versa.*

Let us call the computed fundamental matrix *meaningful* if its relative error is less than 50%. Certainly, we cannot expect meaningful applications of camera calibration or 3-D reconstruction if the computed fundamental matrix has 50% or larger error. We can see that Figs. 15 and 17 nearly covers the noise level range for which meaningful estimation is possible (recall that the solution is normalized to unit norm, so the RMS error roughly corresponds to the relative error).

If the noise is very large, the objective function becomes very flat around its minimum, so large deviations are inevitable whatever computational method is used; the KCR lower bound exactly describes this situation. From such a wide distribution, we may sometimes observe a solution very close to the true value and other times a very wrong one. So, the accuracy evaluation must be done with a large number of trials. In fact, we observed that the RMS error plots of the Sampson and the Gold Standard solutions were visibly different with 1000 trials for each σ. However, they coincided after 10000 trials. In the past, a hasty conclusion was often drawn after a few experiments. Doing many experiments, we failed to observe that the Gold Standard solution is any better than the Sampson solution, quite contrary to the assertion by Hartley and Zisserman [7].

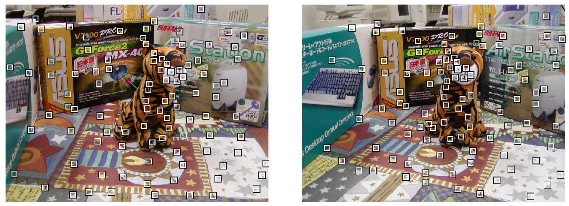


Figure 19. Real images and 100 corresponding points.

Table 1. Residuals and execution times (sec).

method	residual	time
SVD-corrected LS	45.550	.00052
SVD-corrected ML	45.556	.00652
CFNS	45.556	.01300
opt. corrected ML	45.378	.00764
7-LM from LS	45.378	.01136
7-LM from opt. corrected ML	45.378	.01748
EFNS	45.379	.01916
bundle adjustment	45.379	.02580

Real image example. We manually selected 100 pairs of corresponding points in the two images in Fig. 19 and computed the fundamental matrix from them. The final residual J and the execution time (sec) are listed in Table 1. We used Core2Duo E6700 2.66GHz for the CPU with 4GB main memory and Linux for the OS.

We can see that for this example optimally corrected

ML, 7-parameter LM (abbr. 7-LM) started from either LS or optimally corrected ML, EFNS, and bundle adjustment all converged to the same solution, indicating that all are optimal. For this solution, the reprojection error E numerically coincides with the residual J. We can also see that SVD-corrected LS (Hartley's 8-point method) and SVD-corrected ML have higher residual than the optimal solution and that CFNS has as high a residual as SVD-corrected ML.

9. Conclusions

We categorized algorithms for computing the fundamental matrix from point correspondences into "a posteriori correction", "internal access", and "external access" and reviewed existing methods in this framework. Then, we proposed new effective schemes[10]:

1. a new internal access method: 7-parameter LM search.
2. a new external access method: EFNS.
3. a new bundle adjustment algorithm using EFNS.

We conducted experimental comparison and observed that the popular SVD-corrected LS (Hartley's 8-point algorithm) has poor performance. We also observed that the CFNS of Chojnacki et al. [4], a pioneering external access method, does not necessarily converge to a correct solution, while our EFNS always yields an optimal value; we gave a mathematical justification to this.

After many experiments (not all shown here), we concluded that EFNS and the 7-parameter LM search started from optimally corrected ML exhibited the best performance. We also observed that additional bundle adjustment (Gold Standard) does not increase the accuracy to any noticeable degree.

Acknowledgments: This work was done in part in collaboration with Mitsubishi Precision, Co. Ldt., Japan. The authors thank Mike Brooks, Wojciech Chojnacki, and Anton van den Hengel of the University Adelaide, Australia, for providing software and helpful discussions. They also thank Nikolai Chernov of the University of Alabama at Birmingham, U.S.A. for helpful discussions.

References

[1] A. Bartoli and P. Sturm, Nonlinear estimation of fundamental matrix with minimal parameters, *IEEE Trans. Patt. Anal. Mach. Intell.*, **26**-3 (2004-3), 426–432.

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

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

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

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

[6] R. I. Hartley, In defense of the eight-point algorithm, *IEEE Trans. Patt. Anal. Mach. Intell.*, **19**-6 (1997-6), 580–593.

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

[8] C. Harris and M. Stephens, A combined corner and edge detector, *Proc. 4th Alvey Vision Conf.*, August 1988, Manchester, U.K., pp.147–151.

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

[10] K. Kanatani, Ellipse fitting with hyperaccuracy, *IEICE Trans. Inf. & Sys.*,, **E89-D**-10 (2006-10), 2653–2660.

[11] K. Kanatani and Y. Sugaya, High accuracy fundamental matrix computation and its performance evaluation, *IEICE Trans. Inf. & Syst.*, **E90-D**-2 (2007-2), 579–585.

[12] K. Kanatani and Y. Sugaya, Extended FNS for constrained parameter estimation, *Proc. 10th Meeting Image Recog. Understand.*, July 2007, Hiroshima, Japan, pp. 219–226.

[13] Y. Kanazawa and K. Kanatani Robust image matching preserving global consistency, Proc. 6th Asian Conf. Comput. Vision, January 2004, Jeju Island, Korea, Vol. 2, pp. 1128–1133.

[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] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision*, **60**-2 (2004-11), 91–110.

[16] J. Matei and P. Meer, Estimation of nonlinear errors-in-variables models for computer vision applications, *IEEE Trans. Patt. Anal. Mach. Intell.*, **28**-10 (2006-10), 1537–1552.

[17] J. Matasu, O. Chum, M. Urban, and T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, *Image Vision Comput.*, **22**-10 (2004-9), 761–767.

[18] K. Mikolajczyk and C. Schmidt, Scale and affine invariant interest point detectors, *Int. J. Comput. Vision*, **60**-1 (2004-10), 63–86.

[19] P. Meer, Robust techniques for computer vision, in G. Medioni and S. B. Kang (Eds.), *Emerging Topics in Computer Vision*, Prentice Hall, Upper Saddle River, NJ, U.S.A., 2004, pp. 107–190.

[20] T. Migita and T. Shakunaga, One-dimensional search for reliable epipole estimation, *Proc. IEEE Pacific Rim Symp. Image and Video Technology*, December 2006, Hsinchu, Taiwan, pp. 1215–1224.

[21] S. M. Smith and J. M. Brady, SUSAN—A new approach to low level image processing, *Int. J. Comput. Vision*, **23**-1 (1997-1), pp. 45–78.

[22] Y. Sugaya and K. Kanatani, High accuracy computation of rank-constrained fundamental matrix, *Proc. 18th British*

[10] Source codes are available at http://www.iim.ics.tut.ac.jp/~sugaya/public-e.html

Machine Vision Conf., September 2007, Coventry, U.K.

[23] G. Taubin, 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 (1991-11), 1115–1138.

[24] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Artif. Intell.*, **78** (1995), 87–119.

[25] Z. Zhang and C. Loop, Estimating the fundamental matrix by transforming image points in projective space, *Comput. Vision Image Understand.*, **82**-2 (2001-5), 174–180.

Appendix

A. Derivation of the 6-parameter LM

First, note that if $\boldsymbol{F}$ has the form of Eq. (48), we have

$$\begin{aligned}\|\boldsymbol{F}\|^2 &= \sum_{i,j=1}^{3} F_{ij}^2 = \mathrm{tr}[\boldsymbol{F}\boldsymbol{F}^\top] \\ &= \mathrm{tr}[\boldsymbol{U}\mathrm{diag}(\sigma_1,\sigma_2,0)\boldsymbol{V}^\top\boldsymbol{V}\mathrm{diag}(\sigma_1,\sigma_2,0)\boldsymbol{U}^\top] \\ &= \mathrm{tr}[\mathrm{diag}(\sigma_1^2,\sigma_2^2,0)] \\ &= \sigma_1^2+\sigma_2^2, \end{aligned} \tag{104}$$

where we have used the identity $\mathrm{tr}[\boldsymbol{A}\boldsymbol{B}] = \mathrm{tr}[\boldsymbol{B}\boldsymbol{A}]$ for the matrix trace. Thus, the parameterization of Eqs. (49) ensures the normalization $\|\boldsymbol{F}\| = 1$.

Suppose the orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ undergo a small change into $\boldsymbol{U}+\Delta\boldsymbol{U}$ and $\boldsymbol{V}+\Delta\boldsymbol{V}$, respectively. According to the Lie group theory, there exist small vectors $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$ such that the increments $\Delta\boldsymbol{U}$ and $\Delta\boldsymbol{V}$ are written as

$$\Delta\boldsymbol{U} = \boldsymbol{\omega}\times\boldsymbol{U}, \qquad \Delta\boldsymbol{V} = \boldsymbol{\omega}'\times\boldsymbol{V} \tag{105}$$

to a first approximation, where the operator $\times$ means column-wise vector product. Hence, the increment $\Delta\boldsymbol{F}$ in $\boldsymbol{F}$ is to a first approximation

$$\begin{aligned}\Delta\boldsymbol{F} &= \boldsymbol{\omega}\times\boldsymbol{U}\mathrm{diag}(\cos\theta,\sin\theta,0)\boldsymbol{V}^\top \\ &\quad +\boldsymbol{U}\mathrm{diag}(-\sin\theta\Delta\theta,\cos\theta\Delta\theta,0)\boldsymbol{V}^\top \\ &\quad +\boldsymbol{U}\mathrm{diag}(\cos\theta,\sin\theta,0)(\boldsymbol{\omega}'\times\boldsymbol{V})^\top. \end{aligned} \tag{106}$$

Taking out the elements, we can rearrange this in the vector form

$$\Delta\boldsymbol{u} = \boldsymbol{F}_U\boldsymbol{\omega} + \boldsymbol{u}_\theta\Delta\theta + \boldsymbol{F}_V\boldsymbol{\omega}', \tag{107}$$

where $\boldsymbol{F}_U$ and $\boldsymbol{F}_V$ are the matrices in Eqs. (51) and $\boldsymbol{u}_\theta$ is defined by Eq. (53). The resulting increment ΔJ in J is written to a first approximation

$$\begin{aligned}\Delta J &= (\nabla_{\mathbf{u}}J,\Delta\boldsymbol{u}) = (2\boldsymbol{X}\boldsymbol{u},\boldsymbol{F}_U\boldsymbol{\omega}+\boldsymbol{u}_\theta\Delta\theta+\boldsymbol{F}_V\boldsymbol{\omega}') \\ &= 2(\boldsymbol{F}^\top\boldsymbol{X}\boldsymbol{u},\boldsymbol{\omega}) + 2(\boldsymbol{u}_\theta,\boldsymbol{X}\boldsymbol{u})\Delta\theta + 2(\boldsymbol{F}_V\boldsymbol{X}_{\mathbf{u}}J,\boldsymbol{\omega}'), \end{aligned} \tag{108}$$

which shows that the first derivatives of J are given by Eqs. (50) and (52). If we further change $\boldsymbol{u}$ into $\boldsymbol{u}+\Delta\boldsymbol{u}$ in the above expression, we have to a first approximation (i.e., up to the second order in $\Delta\boldsymbol{u}$),

$$\begin{aligned}\Delta^2 J &= (\Delta\boldsymbol{u},\nabla_{\mathbf{u}}^2 J\Delta\boldsymbol{u}) \\ &= (\boldsymbol{F}_U\boldsymbol{\omega}+\boldsymbol{u}_\theta\Delta\theta+\boldsymbol{F}_V\boldsymbol{\omega}', 2\boldsymbol{M}(\boldsymbol{F}_U\boldsymbol{\omega}+\boldsymbol{u}_\theta\Delta\theta+\boldsymbol{F}_V\boldsymbol{\omega}')) \\ &= 2(\boldsymbol{\omega},\boldsymbol{F}_U^\top\boldsymbol{M}\boldsymbol{F}_U\boldsymbol{\omega}) + 2(\boldsymbol{\omega}',\boldsymbol{F}_V^\top\boldsymbol{M}\boldsymbol{F}_V\boldsymbol{\omega}') \\ &\quad +2(\boldsymbol{u}_\theta,\boldsymbol{M}\boldsymbol{u}_\theta)\Delta\theta^2 + 4(\boldsymbol{\omega},\boldsymbol{F}_U^\top\boldsymbol{M}\boldsymbol{\omega}') \\ &\quad +4(\boldsymbol{\omega},\boldsymbol{F}_U^\top\boldsymbol{M}\boldsymbol{u}_\theta)\Delta\theta + 4(\boldsymbol{\omega}',\boldsymbol{F}_V^\top\boldsymbol{M}\boldsymbol{u}_\theta)\Delta\theta, \end{aligned} \tag{109}$$

where we have used the Gauss-Newton approximation $\nabla_{\mathbf{u}}^2 J \approx 2\boldsymbol{M}$. From this, we obtain the second derivatives in Eqs. (54).

B. Details of CFNS

According to Chojnacki et al. [4], the matrix $\boldsymbol{Q}$ used in Eq. (63) is given, without any background reasoning, as follows (their original symbols are somewhat altered to conform to the use in this paper).

The gradient $\nabla_{\mathbf{u}}J = (\partial J/\partial u_i)$ and the Hessian $\nabla_{\mathbf{u}}^2 J = (\partial^2 J/\partial u_i\partial u_j)$ of the function J in Eq. (14) are

$$\nabla_{\mathbf{u}}J = 2(\boldsymbol{M}-\boldsymbol{L})\boldsymbol{u}, \quad \nabla_{\mathbf{u}}^2 J = 2(\boldsymbol{M}-\boldsymbol{L}) - 8(\boldsymbol{S}-\boldsymbol{T}), \tag{110}$$

where $\boldsymbol{M}$ and $\boldsymbol{L}$ are the matrices in Eqs. (22), and we define

$$\begin{aligned}\boldsymbol{S} &= \sum_{\alpha=1}^{N}\frac{(\boldsymbol{u},\boldsymbol{\xi}_\alpha)\mathcal{S}[\boldsymbol{\xi}_\alpha(V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^\top]}{(\boldsymbol{u},V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}, \\ \boldsymbol{T} &= \sum_{\alpha=1}^{N}\frac{(\boldsymbol{u},\boldsymbol{\xi}_\alpha)^2(V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})(V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u}^\top)^\top}{(\boldsymbol{u},V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^3}. \end{aligned} \tag{111}$$

Here, $\mathcal{S}[\,\cdot\,]$ is the symmetrization operator ($\mathcal{S}[\boldsymbol{A}] = (\boldsymbol{A}+\boldsymbol{A}^\top)/2$). Let

$$\begin{aligned}\boldsymbol{A} &= \boldsymbol{P}_{\mathbf{u}^\dagger}(\nabla_{\mathbf{u}}^2 J)(2\boldsymbol{u}\boldsymbol{u}^\top - \boldsymbol{I}), \\ \boldsymbol{B} &= \frac{2}{\|\det\boldsymbol{u}\|}\Big(\sum_{i=1}^{9}\mathcal{S}[((\nabla_{\mathbf{u}}^2\det\boldsymbol{u})\boldsymbol{e}_i)\boldsymbol{u}^{\dagger\top}](\nabla_{\mathbf{u}}J)\boldsymbol{e}_i^\top \\ &\qquad -(\boldsymbol{u}^\dagger,\nabla_{\mathbf{u}}J)\boldsymbol{u}^\dagger\boldsymbol{u}^{\dagger\top}\nabla_{\mathbf{u}}^2\det\boldsymbol{u}\Big), \\ \boldsymbol{C} &= 3\Big(\frac{\det\boldsymbol{u}}{\|\nabla_{\mathbf{u}}\det\boldsymbol{u}\|^2}\nabla_{\mathbf{u}}^2\det\boldsymbol{u} \\ &\qquad +\boldsymbol{u}^\dagger\boldsymbol{u}^{\dagger\top}\Big(\boldsymbol{I} - \frac{2\det\boldsymbol{u}}{\|\nabla_{\mathbf{u}}\det\boldsymbol{u}\|^2}\nabla_{\mathbf{u}}^2\det\boldsymbol{u}\Big)\Big), \end{aligned} \tag{112}$$

where $\boldsymbol{u}^\dagger$ is the vector in Eq. (9), $\boldsymbol{P}_{\mathbf{u}^\dagger}$ is the projection matrix in Eq. (59), and $\boldsymbol{e}_i$ is the ith coordinate basis vector (with 0 components except 1 in the ith position). The matrix $\boldsymbol{Q}$ is given by

$$\boldsymbol{Q} = (\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C})(\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C})^\top. \tag{113}$$

C. Details of bundle adjustment

Introducing Lagrange multipliers λ_α, μ_α, and μ'_α for the constraints of Eqs. (78), and (79) to Eq. (76), we let

$$L = \sum_{\alpha=1}^{N}\Big(\|\Delta\boldsymbol{x}_\alpha\|^2 + \|\Delta\boldsymbol{x}'_\alpha\|^2\Big) - \sum_{\alpha=1}^{N}\lambda_\alpha\Big((\boldsymbol{F}\boldsymbol{x}'_\alpha, \Delta\boldsymbol{x}_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \Delta\boldsymbol{x}'_\alpha)\Big) - \sum_{\alpha=1}^{N}\mu_\alpha(\boldsymbol{k}, \Delta\boldsymbol{x}_\alpha) - \sum_{\alpha=1}^{N}\mu'_\alpha(\boldsymbol{k}, \Delta\boldsymbol{x}'_\alpha). \quad (114)$$

Putting the derivatives of L with respect to $\Delta\boldsymbol{x}_\alpha$ and $\Delta\boldsymbol{x}'_\alpha$ to $\boldsymbol{0}$, we have

$$2\Delta\boldsymbol{x}_\alpha - \lambda_\alpha\boldsymbol{F}\boldsymbol{x}'_\alpha - \mu_\alpha\boldsymbol{k} = \boldsymbol{0},\quad 2\Delta\boldsymbol{x}'_\alpha - \lambda_\alpha\boldsymbol{F}^\top\boldsymbol{x}_\alpha - \mu'_\alpha\boldsymbol{k} = \boldsymbol{0}. \quad (115)$$

Multiplying the projection matrix $\boldsymbol{P}_{\mathbf{k}}$ in Eq. (81) on both sides from left and noting that $\boldsymbol{P}_{\mathbf{k}}\Delta\boldsymbol{x}_\alpha = \Delta\boldsymbol{x}_\alpha$, $\boldsymbol{P}_{\mathbf{k}}\Delta\boldsymbol{x}'_\alpha = \Delta\boldsymbol{x}'_\alpha$, and $\boldsymbol{P}_{\mathbf{k}}\boldsymbol{k} = \boldsymbol{0}$, we have

$$2\Delta\boldsymbol{x}_\alpha - \lambda_\alpha\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha = \boldsymbol{0},\quad 2\Delta\boldsymbol{x}'_\alpha - \lambda_\alpha\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha = \boldsymbol{0}. \quad (116)$$

Hence, we obtain

$$\Delta\boldsymbol{x}_\alpha = \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha, \qquad \Delta\boldsymbol{x}'_\alpha = \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha. \quad (117)$$

Subsitituting these into Eq. (78), we have

$$(\boldsymbol{F}\boldsymbol{x}'_\alpha, \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha) = (\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha), \quad (118)$$

and hence

$$\frac{\lambda_\alpha}{2} = \frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}. \quad (119)$$

Substituting this into Eq. (117), we obtain Eqs. (80). If we substitute Eqs. (80) into Eq. (76), we have

$$E = \sum_{\alpha=1}^{N}\Big(\Big\|\frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha}{(\boldsymbol{x}'_\alpha, \boldsymbol{F}^\top\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}\Big\|^2 + \Big\|\frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha}{(\boldsymbol{x}'_\alpha, \boldsymbol{F}^\top\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}\Big\|^2\Big)$$
$$= \sum_{\alpha=1}^{N}\frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)^2(\|\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha\|^2 + \|\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha\|^2)}{\Big((\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)\Big)^2}$$
$$= \sum_{\alpha=1}^{N}\frac{(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)^2}{(\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)}, \quad (120)$$

where we have noted due to the identity $\boldsymbol{P}_{\mathbf{k}}^2 = \boldsymbol{P}_{\mathbf{k}}$ that $\|\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha\|^2 = (\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha) = (\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}^2\boldsymbol{F}\boldsymbol{x}'_\alpha) = (\boldsymbol{F}\boldsymbol{x}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'_\alpha)$. Similarly, we have $\|\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha\|^2 = (\boldsymbol{F}^\top\boldsymbol{x}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}_\alpha)$.

Introducing Lagrange multipliers λ_α, μ_α, and μ'_α for the constraints of Eqs. (87), and (89) to Eq. (76), we let

$$L = \sum_{\alpha=1}^{N}\Big(\|\tilde{\boldsymbol{x}}_\alpha + \Delta\hat{\boldsymbol{x}}_\alpha\|^2 + \|\tilde{\boldsymbol{x}}'_\alpha + \Delta\hat{\boldsymbol{x}}'_\alpha\|^2\Big) - \sum_{\alpha=1}^{N}\lambda_\alpha\Big((\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \Delta\hat{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \Delta\hat{\boldsymbol{x}}'_\alpha)\Big) - \sum_{\alpha=1}^{N}\mu_\alpha(\boldsymbol{k}, \Delta\hat{\boldsymbol{x}}_\alpha) - \sum_{\alpha=1}^{N}\mu'_\alpha(\boldsymbol{k}, \Delta\hat{\boldsymbol{x}}'_\alpha). \quad (121)$$

Putting the derivatives of L with respect to $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$ to $\boldsymbol{0}$, we have

$$2(\tilde{\boldsymbol{x}}_\alpha + \Delta\hat{\boldsymbol{x}}_\alpha) - \lambda_\alpha\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha - \mu_\alpha\boldsymbol{k} = \boldsymbol{0},\quad 2(\tilde{\boldsymbol{x}}'_\alpha + \Delta\hat{\boldsymbol{x}}'_\alpha) - \lambda_\alpha\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha - \mu'_\alpha\boldsymbol{k} = \boldsymbol{0}. \quad (122)$$

Multiplying $\boldsymbol{P}_{\mathbf{k}}$ on both sides from left, we have

$$2\tilde{\boldsymbol{x}}_\alpha + 2\Delta\hat{\boldsymbol{x}}_\alpha - \lambda_\alpha\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha = \boldsymbol{0},\quad 2\tilde{\boldsymbol{x}}_\alpha + 2\Delta\hat{\boldsymbol{x}}'_\alpha - \lambda_\alpha\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha = \boldsymbol{0}. \quad (123)$$

Substituting these into Eq. (88), we have

$$\Delta\hat{\boldsymbol{x}}_\alpha = \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha - \tilde{\boldsymbol{x}}_\alpha,\quad \Delta\hat{\boldsymbol{x}}'_\alpha = \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}'_\alpha. \quad (124)$$

Subsitituting these into Eq. (88), we obtain

$$(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha - \tilde{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \frac{\lambda_\alpha}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}'_\alpha) = (\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha), \quad (125)$$

and hence

$$\frac{\lambda_\alpha}{2} = \frac{(\hat{\boldsymbol{x}}_\alpha, \boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \tilde{\boldsymbol{x}}_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \tilde{\boldsymbol{x}}'_\alpha)}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)}. \quad (126)$$

Subsitituting this into Eq. (124), we obtain Eq. (92). If we substitute Eq. (92) into Eq. (85), we have

$$E = \sum_{\alpha=1}^{N}\Big(\Big\|\frac{e_\alpha\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)}\Big\|^2 + \Big\|\frac{e_\alpha\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)}\Big\|^2\Big)$$
$$= \sum_{\alpha=1}^{N}\frac{e_\alpha^2(\|\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha\|^2 + \|\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha\|^2)}{\Big((\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)\Big)^2}$$
$$= \sum_{\alpha=1}^{N}\frac{e_\alpha^2}{(\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'_\alpha) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}_\alpha)}. \quad (127)$$