10th Meeting on Image Recognition and Understanding (MIRU2007), Hiroshima, Japan, July 2007, pp. 609–614.

High Accuracy Computation of Rank-constrained Fundamental Matrix by Efficient Search

Yasuyuki SUGAYA† and Kenichi KANATANI‡

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

‡Department of Computer Science, Okayama University, Okayama 700-8530 Japan

E-mail: † sugaya@iim.ics.tut.ac.jp, ‡ kanatani@suri.it.okayama-u.ac.jp

Abstract A new method is presented for computing the fundamental matrix from point correspondences: its singular value decomposition (SVD) is optimized by the Levenberg-Marquard (LM) method. There is no need for tentative 3-D reconstruction. The accuracy achieves the theoretical bound (the KCR lower bound).

Keywords fundamental matrix, rank constraint, statistical optimization, maximum likelihood, KCR lower bound

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].

A popular approach is to do maximum likelihood (ML) computation without imposing the constraint that the fundamental matrix has rank 2; the rank constraint was imposed a posteriori in a statistically optimal manner. The resulting solution has accuracy close to the theoretical bound (KCR lower bound [2, 8]).

Another possible approach is to do optimization subject to the rank constraint [1, 4, 13, 15]. In this paper, we propose a new method in this line. Following Bartoli and Sturm [1], we optimize the singular value decomposition (SVD) of the fundamental matrix, but there is no need to include 3-D coordinates or camera matrices as unknowns; we use the Levenberg-Marquard (LM) method in the reduced parameter space.

We summarize the mathematical background in Sec. 2. Then, we describe the optimal correction approach in Sec. 3 and the proposed method in Sec. 4. Sec. 5 shows numerical experiments. We conclude in Sec. 6.

2. Mathematical Fundamentals

Fundamental matrix

Given two images of the same scene, a point (x, y) in the first image and the corresponding point (x', y') in the second satisfy the *epipolar equation* [7]

$$\left(\begin{pmatrix} x \\ y \\ f_0 \end{pmatrix}, \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ f_0 \end{pmatrix}\right) = 0, \quad (1)$$

where f_0 is an arbitrary constant[1]. 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. (1) is of rank 2 and called the *fundamental matrix*. If we define

$$\boldsymbol{u} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top, \quad (2)$$

$$\boldsymbol{\xi} = (xx', xy', xf_0, yx', yy', yf_0, f_0x', f_0y', f_0^2)^\top, \quad (3)$$

Eq. (1) can be rewritten as

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

The magnitude of $\boldsymbol{u}$ is indeterminate, so we normalize it to $\|\boldsymbol{u}\| = 1$. If we write N observed noisy correspondence pairs as 9-D vectors $\{\boldsymbol{\xi}_\alpha\}$ in the form of Eq. (3), our task is to estimate the 9-D vector $\boldsymbol{u}$ from $\{\boldsymbol{\xi}_\alpha\}$ using Eq. (4).

Covariance matrices

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

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

where $E[\,\cdot\,]$ denotes expectation over the noise distribution. If 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

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

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

$$\begin{pmatrix} 0 & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & f_0\bar{x}_\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bar{x}'_\alpha\bar{y}'_\alpha & f_0\bar{x}'_\alpha & f_0\bar{y}_\alpha & 0 & 0 \\ \bar{y}_\alpha^2+\bar{y}_\alpha'^2 & f_0\bar{y}'_\alpha & 0 & f_0\bar{y}_\alpha & 0 \\ f_0\bar{y}'_\alpha & f_0^2 & 0 & 0 & 0 \\ 0 & 0 & f_0^2 & 0 & 0 \\ f_0\bar{y}_\alpha & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{6}$$

which we call the *normalized covariance matrix*. It depends only on the true positions $(\bar{x}_\alpha, \bar{y}_\alpha)$ and $(\bar{x}'_\alpha, \bar{y}'_\alpha)$ of the data points (x_α, y_α) and (x'_α, y'_α). In numerical computation, we replace the true values by the data values[2].

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

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

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

Geometry of the constraint

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

$$\boldsymbol{u}^\dagger \equiv N[\begin{pmatrix} u_5u_9 - u_8u_6 \\ u_6u_7 - u_9u_4 \\ u_4u_8 - u_7u_5 \\ u_8u_3 - u_2u_9 \\ u_9u_1 - u_3u_7 \\ u_7u_2 - u_1u_8 \\ u_2u_6 - u_5u_3 \\ u_3u_4 - u_6u_1 \\ u_1u_5 - u_4u_2 \end{pmatrix}], \tag{8}$$

where $N[\,\cdot\,]$ denotes normalization into unit norm. The inside of this operator is the 9-D representation (cf. Eq. (2)) of the transpose $\boldsymbol{F}^{\dagger\top}$ of the cofactor $\boldsymbol{F}^\dagger$ of $\boldsymbol{F}$. The constraint $\det \boldsymbol{F} = 0$ is equivalently written as

$$(\boldsymbol{u}^\dagger, \boldsymbol{u}) = 0. \tag{9}$$

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 $T_{\boldsymbol{u}}(\mathcal{U})$. It follows that the projection operator $\boldsymbol{P}_{\mathcal{U}}$ in Eq. (7) has the following matrix representation ($\boldsymbol{I}$ is the unit matrix):

$$\boldsymbol{P}_{\mathcal{U}} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top - \boldsymbol{u}^\dagger\boldsymbol{u}^{\dagger\top}. \tag{10}$$

KCR lower bound

If the noise in $\{\boldsymbol{\xi}_\alpha\}$ is independent and Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma^2 V_0[\boldsymbol{\xi}]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\boldsymbol{u}}$ of $\boldsymbol{u}$ [8]:

$$V[\hat{\boldsymbol{u}}] \succ \sigma^2\Big(\sum_{\alpha=1}^N \frac{(\boldsymbol{P}_{\mathcal{U}}\bar{\boldsymbol{\xi}}_\alpha)(\boldsymbol{P}_{\mathcal{U}}\bar{\boldsymbol{\xi}}_\alpha)^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}\Big)_8^-. \tag{11}$$

Here, $\succ$ means that the left-hand side minus the right is positive semidefinite, and $(\,\cdot\,)_r^-$ denotes the pseudoinverse of rank[3] r. Chernov and Lesort [2] called the right-hand side of Eq. (11) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that Eq. (11) 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 (ML)

If the noise in $\{\boldsymbol{\xi}_\alpha\}$ is independent and Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma^2 V_0[\boldsymbol{\xi}]$, *maximum likelihood* (*ML*) estimation of $\boldsymbol{u}$ is to minimize the sum of square *Mahalanobis distances*[4]

$$J = \sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha, V_0[\boldsymbol{\xi}_\alpha]_2^-(\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha)), \tag{12}$$

subject to $(\boldsymbol{u}, \bar{\boldsymbol{\xi}}_\alpha) = 0$, $\alpha = 1, ..., N$. Eliminating the constraint by using Lagrange multipliers, we obtain [8]

$$J = \sum_{\alpha=1}^N \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}. \tag{13}$$

The ML estimator $\hat{\boldsymbol{u}}$ minimizes this subject to $\|\boldsymbol{u}\| = 1$ and $(\boldsymbol{u}^\dagger, \boldsymbol{u}) = 0$. Its covariance matrix $V[\hat{\boldsymbol{u}}]$ agrees with the KCR lower bound (the right-side hand of Eq. (11)) up to $O(\sigma^4)$ [8].

3. A Posteriori Correction Approach

Optimal correction

A common approach to solve this problem is to minimizes Eq. (13) without considering the rank constraint, compute the SVD of the resulting fundamental matrix, and replace the smallest singular value by 0, producing a "closest" matrix of rank 2 in norm [6]. We call this *SVD correction*.

A more sophisticated method is the *optimal correction* [8, 12]. According to the statistical optimization theory [8], 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. The procedure goes as follows [8]:

[2]We have confirmed that this does not affect the results of our experiments.

[3]The matrix obtained by replacing the largest r eigenvalues by their reciprocals and the remaining ones by 0s in its spectral (or eigen) decomposition.

[4]Namely, we fit a hyperplane defined by Eq. (4) to N points $\{\boldsymbol{\xi}_\alpha\}$ in $\mathcal{R}^9$ so as to minimize the sum of the distances to the points inversely weighted by their covariance matrices $V_0[\boldsymbol{\xi}_\alpha]$.

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

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

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

$$V_0[\tilde{\boldsymbol{u}}] = \tilde{\boldsymbol{M}}_8^-. \tag{15}$$

3. Update the solution $\tilde{\boldsymbol{u}}$ as follows[5]:

$$\tilde{\boldsymbol{u}} \leftarrow N[\tilde{\boldsymbol{u}} - \frac{1}{3}\frac{(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{u}}^\dagger)V_0[\tilde{\boldsymbol{u}}]\tilde{\boldsymbol{u}}^\dagger}{(\tilde{\boldsymbol{u}}^\dagger, V_0[\tilde{\boldsymbol{u}}]\tilde{\boldsymbol{u}}^\dagger)}]. \tag{16}$$

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

$$\boldsymbol{P}_{\tilde{\mathbf{u}}} = \boldsymbol{I} - \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}^\top, \quad V_0[\tilde{\boldsymbol{u}}] \leftarrow \boldsymbol{P}_{\tilde{\mathbf{u}}} V_0[\tilde{\boldsymbol{u}}] \boldsymbol{P}_{\tilde{\mathbf{u}}}, \tag{17}$$

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. (16) is essentially the Newton iteration formula for displacing $\tilde{\boldsymbol{u}}$ in the direction in the tangent space $T_{\tilde{\boldsymbol{u}}}(\mathcal{S}^8)$ 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 $T_{\tilde{\boldsymbol{u}}}(\mathcal{S}^8)$, so the operator $N[\,\cdot\,]$ pulls it back onto $\mathcal{S}^8$. From that point, the same procedure is repeated until $(\tilde{\boldsymbol{u}}^\dagger, \tilde{\boldsymbol{u}}) = 0$. Since the normalized covariance matrix $V_0[\tilde{\boldsymbol{u}}]$ is defined in the tangent space $T_{\tilde{\boldsymbol{u}}}(\mathcal{S}^8)$, it changes as $\tilde{\boldsymbol{u}}$ moves. Eq. (17) corrects it so that $V_0[\tilde{\boldsymbol{u}}]$ has the domain $T_{\tilde{\boldsymbol{u}}}(\mathcal{S}^8)$ at the displaced point $\tilde{\boldsymbol{u}}$. □

FNS

First, we need to solve unconstrained minimization of Eq. (13), for which several methods exist: the *FNS* (*Fundamental Numerical Scheme*) of Chojnacki et al. [3], the *HEIV* (*Heteroscedastic Errors-in-Variable*) of Leedan and Meer [11], and the *projective Gauss-Newton iterations* of Kanatani and Sugaya [9]. Their convergence properties were studied in [9].

In our experiment, we used the FNS of Chojnacki et al. [3]. This method is based on the fact that the derivative of Eq. (13) with respect to $\boldsymbol{u}$ has the form

$$\nabla_{\mathbf{u}} J = \boldsymbol{X}\boldsymbol{u}, \tag{18}$$

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

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

The procedure of FNS goes as follows [3, 9]:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrix $\boldsymbol{X}$ in Eq. (19).
3. Solve the eigenvalue problem

$$\boldsymbol{X}\boldsymbol{u}' = \lambda \boldsymbol{u}', \tag{20}$$

and compute the unit eigenvector $\boldsymbol{u}'$ for the smallest eigenvalue[6] 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.

The resulting solution is then optimally corrected to satisfy the rank constraint as described earlier.

Least squares (LS)

For computing the initial value for FNS, we use *least squares* (*LS*), minimizing

$$J_{\mathrm{LS}} = \sum_{\alpha=1}^{N} (\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 = (\boldsymbol{u}, \boldsymbol{M}_{\mathrm{LS}}\boldsymbol{u}), \tag{21}$$

where

$$\boldsymbol{M}_{\mathrm{LS}} = \sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top. \tag{22}$$

Equation (21) is minimized by the unit eigenvector of $\boldsymbol{M}_{\mathrm{LS}}$ for the smallest eigenvalue. We could obtain a more accurate solution using Taubin's method [9], but it requires more computation time.

4. Proposed Method

We now propose a new method for minimizing Eq. (13) subject to the rank constraint, using the parameterization of Bartoli and Sturm [1].

Bundle adjustment

The fundamental matrix $\boldsymbol{F}$ has nine elements, on which the normalization[7] $\|\boldsymbol{F}\| = 1$ and the rank constraint $\det \boldsymbol{F} = 0$ are imposed. So, it has seven degrees of freedom.

Many types of 7-degree parameterization have been proposed in the past. Typical ones are based on epipoles (e.g., [13, 15]), but the resulting expressions are often 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 3-D positions of the observed feature points, the relative positions of the two cameras, and their intrinsic parameters as unknowns. Using assumed values, they

[5] $\tilde{\boldsymbol{u}}^\dagger$ means the operation of Eq. (8) applied to $\tilde{\boldsymbol{u}}$.

[6] Originally, the eigenvalue closest to 0 was chosen [3]. Later, Chojnacki, et al. [5] pointed out that the choice of the smallest eigenvalue improves the convergence. This was confirmed by the experiments of Kanatani and Sugaya [9].

[7] The matrix norm is defined by $\|\boldsymbol{F}\| = \sqrt{\sum_{i,j=1}^{3} F_{ij}^2}$. This gives the same value as the vector norm $\|\boldsymbol{u}\|$ of the transformed vector $\boldsymbol{u}$ in Eq. (2).

computed tentative 3-D reconstruction and evaluated the reprojection error. Then, they searched the high-dimensional parameter space for the value that minimizes the reprojection error. 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").

From the underlying geometry, however, we can see that the necessary and sufficient condition for the corresponding points $\{(x_\alpha, y_\alpha)\}$ and $\{(x'_\alpha, y'_\alpha)\}$ to be a projection of "some" 3-D structure is the epipolar equation of Eq. (1) for "some" $\boldsymbol{F}$, and Eq. (12) describes the corresponding reprojection error. Hence, bundle adjustment for minimizing the reprojection error by assuming "some" 3-D reconstruction is practically equivalent to minimize Eq. (12), so we need not consider indeterminate 3-D reconstruction.

Here, using the parameterization of Bartoli and Sturm [1], we directly minimize Eq. (13) by the Levenberg-Marquard (LM) method.

SVD-based Lie algebraic method

The fundamental matrix $\boldsymbol{F}$ has rank 2, so its SVD has the form[8]

$$\boldsymbol{F} = \boldsymbol{U}\mathrm{diag}(\sigma_1, \sigma_2, 0)\boldsymbol{V}^\top, \tag{23}$$

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

$$\sigma_1 = \cos\theta, \qquad \sigma_2 = \sin\theta. \tag{24}$$

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 (also see [14]). 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. (13) with respect to them are as follows (we omit the details):

$$\nabla_\omega J = \boldsymbol{F}_U^\top \boldsymbol{X}\boldsymbol{u}, \qquad \nabla_{\omega'} J = \boldsymbol{F}_V^\top \boldsymbol{X}\boldsymbol{u}. \tag{25}$$

[8]The symbol $\mathrm{diag}(\cdots)$ denotes the diagonal matrix with $\cdots$ as its diagonal elements in that order.

[9]Bartoli and Sturm [1] took the ratio $\gamma = \sigma_2/\sigma_1$ as a variable. Here, we adopt the angle θ by retaining the symmetry. 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, 8]. Since we take the frame center to be the image origin, there is little danger of σ_1 or σ_2 becoming 0.

Here, $\boldsymbol{X}$ is the matrix defined in Eq. (19), and the matrices $\boldsymbol{F}_U$ and $\boldsymbol{F}_V$ are defined as follows:

$$\boldsymbol{F}_U \equiv \begin{pmatrix} 0 & F_{31} & -F_{21} \\ 0 & F_{32} & -F_{22} \\ 0 & F_{33} & -F_{23} \\ -F_{31} & 0 & F_{11} \\ -F_{32} & 0 & F_{12} \\ -F_{33} & 0 & F_{13} \\ F_{21} & -F_{11} & 0 \\ F_{22} & -F_{12} & 0 \\ F_{23} & -F_{13} & 0 \end{pmatrix}, \tag{26}$$

$$\boldsymbol{F}_V \equiv \begin{pmatrix} 0 & F_{13} & -F_{12} \\ -F_{13} & 0 & F_{11} \\ F_{12} & -F_{11} & 0 \\ 0 & F_{23} & -F_{22} \\ -F_{23} & 0 & F_{21} \\ F_{22} & -F_{21} & 0 \\ 0 & F_{33} & -F_{32} \\ -F_{33} & 0 & F_{31} \\ F_{32} & -F_{31} & 0 \end{pmatrix}. \tag{27}$$

The derivative of Eq. (13) with respect to θ is

$$\frac{\partial J}{\partial \theta} = (\boldsymbol{u}_\theta, \boldsymbol{X}\boldsymbol{u}), \tag{28}$$

where we define

$$\boldsymbol{u}_\theta \equiv \begin{pmatrix} \sigma_1 u_{12} v_{12} - \sigma_2 u_{11} v_{11} \\ \sigma_1 u_{12} v_{22} - \sigma_2 u_{11} v_{21} \\ \sigma_1 u_{12} v_{32} - \sigma_2 u_{11} v_{31} \\ \sigma_1 u_{22} v_{12} - \sigma_2 u_{21} v_{11} \\ \sigma_1 u_{22} v_{22} - \sigma_2 u_{21} v_{21} \\ \sigma_1 u_{22} v_{32} - \sigma_2 u_{21} v_{31} \\ \sigma_1 u_{32} v_{12} - \sigma_2 u_{31} v_{11} \\ \sigma_1 u_{32} v_{22} - \sigma_2 u_{31} v_{21} \\ \sigma_1 u_{32} v_{32} - \sigma_2 u_{31} v_{31} \end{pmatrix}. \tag{29}$$

Second derivatives are similarly computed. Adopting Gauss-Newton approximation[10], we obtain the following (we omit the details):

$$\nabla_\omega^2 J = \boldsymbol{F}_U^\top \boldsymbol{X}\boldsymbol{F}_U, \qquad \nabla_{\omega'}^2 J = \boldsymbol{F}_V^\top \boldsymbol{X}\boldsymbol{F}_V, \tag{30}$$

$$\nabla_{\omega\omega'} J = \boldsymbol{F}_U^\top \boldsymbol{X}\boldsymbol{F}_V, \qquad \frac{\partial J^2}{\partial \theta^2} = (\boldsymbol{u}_\theta, \boldsymbol{X}\boldsymbol{u}_\theta), \tag{31}$$

$$\frac{\partial \nabla_\omega J}{\partial \theta} = \boldsymbol{F}_U^\top \boldsymbol{X}\boldsymbol{u}_\theta, \qquad \frac{\partial \nabla_{\omega'} J}{\partial \theta} = \boldsymbol{F}_V^\top \boldsymbol{X}\boldsymbol{u}_\theta. \tag{32}$$

Computational procedure

The LM procedure goes as follows:

1. Initialize $\boldsymbol{F}$ in such a way that $\det \boldsymbol{F} = 0$ and $\|\boldsymbol{F}\| = 1$, and express it as $\boldsymbol{F} = \boldsymbol{U}\mathrm{diag}(\cos\theta, \sin\theta, 0)\boldsymbol{V}^\top$.
2. Compute J in Eq. (13), and let $c = 0.0001$.
3. Compute $\boldsymbol{F}_U$, $\boldsymbol{F}_V$, and $\boldsymbol{u}_\theta$ in Eqs. (26), (27), and (29).

[10]This amounts to ignoring terms involving $(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)$.

4. Compute $\boldsymbol{X}$ in Eq. (19), the first derivatives in Eqs. (25) an (28), and the second derivatives in Eqs. (30), (31), and (32).
5. Compute the following matrix $\boldsymbol{H}$:

$$\boldsymbol{H} = \begin{pmatrix} \nabla^2_{\omega} J & \nabla_{\omega\omega'} J & \partial \nabla_{\omega} J / \partial \theta \\ (\nabla_{\omega\omega'} J)^\top & \nabla^2_{\omega'} J & \partial \nabla_{\omega'} J / \partial \theta \\ (\partial \nabla_{\omega} J / \partial \theta)^\top & (\partial \nabla_{\omega'} J / \partial \theta)^\top & \partial J^2 / \partial \theta^2 \end{pmatrix}. \tag{33}$$

6. Solve the 7-D simultaneous linear equations

$$(\boldsymbol{H} + cD[\boldsymbol{H}]) \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\omega}' \\ \Delta\theta \end{pmatrix} = - \begin{pmatrix} \nabla_{\omega} J \\ \nabla_{\omega'} J \\ \partial J / \partial \theta \end{pmatrix}, \tag{34}$$

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

$$\boldsymbol{U}' = \mathcal{R}(\boldsymbol{\omega})\boldsymbol{U}, \quad \boldsymbol{V}' = \mathcal{R}(\boldsymbol{\omega}')\boldsymbol{V}, \quad \theta' = \theta + \Delta\theta, \tag{35}$$

where $\mathcal{R}(\boldsymbol{\omega})$ denotes rotation around $N[\boldsymbol{\omega}]$ by angle $\|\boldsymbol{\omega}\|$.
8. Update $\boldsymbol{F}$ as follows:

$$\boldsymbol{F}' = \boldsymbol{U}' \mathrm{diag}(\cos\theta', \sin\theta', 0) \boldsymbol{V}'^\top. \tag{36}$$

9. Let J' be the value of Eq. (13) 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. Experiments

Simulation setting

Figure 1 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 1) SVD-corrected LS, 2) SVD-corrected ML, 3) CFNS, 4) optimally corrected ML, and 5) the proposed method.

The CFNS is a method proposed by Chojnacki et al. [4]. It modifies the matrix $\boldsymbol{X}$ in Eq. (19) in such a way that the FNS iterations result in a minimizer of Eq. (13) subject to the rank constraint.

All iterations are initialized by LS and stopped when the update of $\boldsymbol{F}$ is less than 10^{-6} in norm.

Accuracy comparison

Figure 2 plots, for σ on the horizontal axis, the following root-mean-square (RMS) error D corresponding

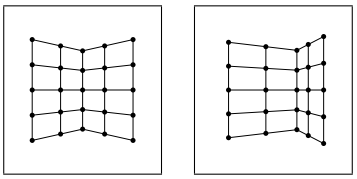

Figure 1: Simulated images of planar grid surfaces.

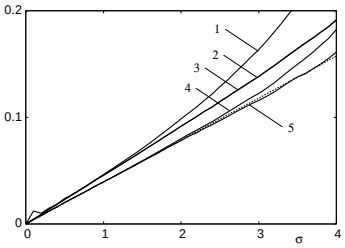

Figure 2: RMS error for Fig. 1 vs. noise level. 1) SVD-corrected LS. 2) SVD-corrected ML. 3) CFNS. 4) Optimally corrected ML. 5) Proposed method. The dotted line indicates the KCR lower bound.

to Eq. (7) over 10000 independent trials:

$$D = \sqrt{\frac{1}{10000} \sum_{a=1}^{10000} \|\boldsymbol{P}_{\mathcal{U}} \hat{\boldsymbol{u}}^{(a)}\|^2}. \tag{37}$$

Here, $\hat{\boldsymbol{u}}^{(a)}$ is the ath value, and $\boldsymbol{P}_{\mathcal{U}}$ is the projection matrix defined by Eq. (10). The dotted line is the bound implied by the KCR lower bound (the trace of the right-hand side of Eq. (11)).

As we can see, SVD-corrected LS is inferior to SVD-corrected ML. However, optimally corrected ML has accuracy close to the KCR lower bound. The accuracy of the proposed method is nearly the same as optimally corrected ML when the noise is small but gradually outperforms it as the noise increases.

The CFNS of Chojnacki et al. [4] performs only as much as SVD-corrected ML over the entire range of σ. They asserted superiority of CFNS to optimally corrected ML by numerical examples [4], but Fig. 2 contradicts their assertion. The reason for this is studied in [10], so we do not go into the details here.

Execution time comparison

Figure 3 plots the execution time of optimally corrected ML (dashed line) and the proposed method (solid line); for each σ, we averaged over 10000 trials. We used Core2Duo E6700 2.66GHz for the CPU with 4GB main memory and Linux for the OS. We conclude that the proposed method is more efficient than optimally corrected ML when the noise not so large. As the noise increases, the search takes more time, making optimally corrected ML more efficient in the end.

This is explained as follows. For optimally corrected ML, we first need to compute unconstrained ML, which requires eigenvalue computation of a 9×9 matrix in

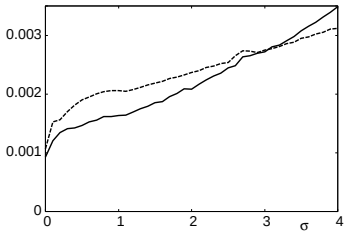

Figure 3: Execution time (including initialization) vs. noise level. The dashed line is for the optimally corrected ML; the solid line is for the proposed method.

each iteration step. We used FNS [3] for this, but the situation is the same if we use the HEIV [11], the projective Gauss-Newton iterations [9], or renormalization [8]. In contrast, the proposed method solves 7-D simultaneous linear equations in each iteration step, so there is no need to compute eigenvalues or SVD except in the initial step, resulting in high efficiency per step. As the noise increases, however, the search requires more steps, consuming longer time.

Real image experiments

We manually selected 100 pairs of corresponding points in the two images in Fig. 4 and computed the fundamental matrix from them. The final residual J and the execution time (sec) are listed in Table 1.

We can see that optimally correction ML and the proposed method converged to the same solution, while SVD corrections of LS and ML resulted in different values with higher residuals. For this data set, the proposed method took longer time than optimally corrected ML.

6. Conclusions

We presented a new method for computing the fundamental matrix from point correspondences over two images[11]: we adopted the SVD representation of Bartoli and Sturm [1] and optimized it by LM. There is no need for tentative 3-D reconstruction. We compared its accuracy with the KCR lower bound by numerical experiments (not all are shown here) and concluded that our method is generally superior to optimally corrected ML.

Acknowledgments: The authors thank Wojciech Chojnacki of the University of Adelaide, Australia for providing software and having helpful discussions.

References

[1] A. Bartoli and P. Sturm, "Nonlinear estimation of fundamental matrix with minimal parameters," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.3, pp.426–432, March 2004.

[2] N. Chernov and C. Lesort, "Statistical efficiency of curve fitting algorithms," Comput. Stat. Data Anal., vol.47, no.4, pp.713–728, Nov. 2004.

[11] The source code is available from the authors' Web page `http://www.iim.ics.tut.ac.jp/~sugaya/public-e.html`

Figure 4: Real images and 100 corresponding points.

Table 1: The residual and execution time (sec) for computing the fundamental matrix from the corresponding points in Fig. 4.

method	residual	time
SVD-corrected LS	45.550	.00052
SVD-corrected ML	45.556	.00652
Optimally corrected ML	45.378	.00764
Proposed method	45.378	.01136

[3] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, "On the fitting of surfaces to data with covariances," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.11, pp.1294–1303, Nov. 2000.

[4] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, "A new constrained parameter estimator for computer vision applications" Image Vis. Comput., vol.22, no.2, pp.85–91, Feb. 2004.

[5] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, "FNS, CFNS and HEIV: A unifying approach," J. Math. Imaging Vision, vol.23, no.2, pp.175–183, Sept. 2005.

[6] R. I. Hartley, "In defense of the eight-point algorithm," IEEE Trans. Patt. Anal. Mach. Intell., vol.19, no.6, pp.580–593, June 1997.

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

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

[9] K. Kanatani and Y. Sugaya, "High accuracy fundamental matrix computation and its performance evaluation," Proc. 17th British Machine Vision Conf (BMVC 2006), vol.1, pp. 217–226, Edinburgh, U.K., Sept. 2006.

[10] K. Kanatani and Y. Sugaya, "Extended FNS for constrained parameter estimation," Proc. 10th Meeting Image Recog. Understand. (MIRU2007), Hiroshima, Japan, July 2007, this volume.

[11] Y. Leedan and P. Meer, "Heteroscedastic regression in computer vision: Problems with bilinear constraint," Int. J. Comput. Vision., vol.37, no.2, pp.127–150, June 2000.

[12] J. Matei and P. Meer, "Estimation of nonlinear errors-in-variables models for computer vision applications," IEEE Trans. Patt. Anal. Mach. Intell., vol.28, no.10, pp.1537–1552, Oct. 2006.

[13] T. Migita and T. Shakunaga, "One-dimensional search for reliable epipole estimation," Proc. IEEE Pacific Rim Sypm. Image and Video Technology, pp. 1215–1224, Hsinchu, Taiwan, Dec. 2006.

[14] M. Sakamoto, Y. Sugaya, and K. Kanatani, "Homography optimization for consistent circular panorama generation," Proc. IEEE Pacific Rim Symp. Image and Video Technology, pp. 1195–1205, Hsinchu, Taiwan, Dec. 2006.

[15] Z. Zhang and C. Loop, "Estimating the fundamental matrix by transforming image points in projective space," Comput. Vis. Image Understand., vol.82, no.2, pp.174–180, May 2001.