PAPER

High Accuracy Fundamental Matrix Computation and Its Performance Evaluation

Kenichi KANATANI[†a)] ***and*** **Yasuyuki SUGAYA**[††], ***Members***

SUMMARY We compare the convergence performance of different numerical schemes for computing the fundamental matrix from point correspondences over two images. First, we state the problem and the associated KCR lower bound. Then, we describe the algorithms of three well-known methods: FNS, HEIV, and renormalization. We also introduce Gauss-Newton iterations as a new method for fundamental matrix computation. For initial values, we test random choice, least squares, and Taubin's method. Experiments using simulated and real images reveal different characteristics of each method. Overall, FNS exhibits the best convergence properties.

key words: *fundamental matrix, geometric fitting, KCR lower bound, maximum likelihood, convergence performance*

1. Introduction

Computing the fundamental matrix from point correspondences over two images is the first step of many vision applications including camera calibration, image rectification, structure from motion, and new view generation. Initially, least-squares-based algebraic methods were widely used, but Kanatani [9] pointed out that fundamental matrix computation should be regarded as statistical estimation and that maximum likelihood (ML) produces an optimal solution.

Since then, many numerical schemes have been proposed for computing ML: the best known are FNS [3] and HEIV [13], which attain a theoretical accuracy bound (KCR lower bound) expect for higher order terms in noise [2], [9]. Kanatani's renormalization [9] also computes a solution nearly equivalent to them [10]. In this paper, we newly introduce a fourth method: directly computing ML by Gauss-Newton iterations.

All these are iterative methods with different convergence properties, which also depend on the choice of initial values. Chernov [1] and Kanatani [11] investigated the convergence behavior of these schemes for ellipse fitting. The purpose of this paper is to examine their convergence performance for fundamental matrix computation.

Section 2 states the problem and the KCR lower bound. Section 3 describes FNS, HEIV, renormalization, and Gauss-Newton iterations. In Sect. 4, we introduce three types of initialization: random choice, least squares, and Taubin's method. Section 5 shows numerical examples using simulated and real images, together with discussions about the origins of the performance difference among them. In Sect. 6, we conclude that overall FNS has the best convergence properties.

Manuscript received April 6, 2006.

Manuscript revised September 14, 2006.

[†]The author is with the Department of Computer Science, Okayama University, Okayama-shi, 700–8530 Japan.

[††]The author is with the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.

a) E-mail: kanatani@suri.it.okayama-u.ac.jp

DOI: 10.1093/ietisy/e90-d.2.1

2. Fundamental Matrix Computation

If point (x, y) corresponds to point (x', y') over two images of the same scene taken from different positions, the following *epipolar equation* holds [7]:

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

Here, $\boldsymbol{F} = (F_{ij})$ is called the *fundamental matrix*, which 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. Throughout this paper, the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted by $(\boldsymbol{a}, \boldsymbol{b})$. In Eq. (1), f_0 is an appropriate scale constant for stabilizing numerical computation [6].

If we define

$$\begin{aligned} \boldsymbol{u} &= (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top, \\ \boldsymbol{\xi} &= (xx', xy', xf_0, yx', yy', yf_0, f_0x', f_0y', f_0^2)^\top, \end{aligned} \quad (2)$$

Eq. (1) is written as

$$(\boldsymbol{u}, \boldsymbol{\xi}) = 0. \quad (3)$$

Since the absolute scale of the vector $\boldsymbol{u}$ is indeterminate, we adopt normalization $\|\boldsymbol{u}\| = 1$.

Fundamental matrix computation thus reduces to fitting a hyperplane of the form of Eq. (3) to noisy vector data $\{\boldsymbol{\xi}_\alpha\}$ in $\mathcal{R}^9$. Let us write $\boldsymbol{\xi}_\alpha = \bar{\boldsymbol{\xi}}_\alpha + \Delta\boldsymbol{\xi}_\alpha$, where $\bar{\boldsymbol{\xi}}_\alpha$ is the noiseless value, and $\Delta\boldsymbol{\xi}_\alpha$ the noise term. We define the covariance matrix of $\boldsymbol{\xi}_\alpha$ by $V[\boldsymbol{\xi}_\alpha] = E[\Delta\boldsymbol{\xi}_\alpha\Delta\boldsymbol{\xi}_\alpha^\top]$, where $E[\,\cdot\,]$ denotes expectation for the noise distribution.

If each image coordinate of matching points is perturbed by independent random noise of mean 0 and standard deviation σ, the covariance matrix $V[\boldsymbol{\xi}_\alpha]$ has the form $\sigma^2 V_0[\boldsymbol{\xi}_\alpha]$ up to $O(\sigma^4)$, where

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

$$\begin{pmatrix} 0 & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & f_0\bar{x}_\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bar{x}'_\alpha\bar{y}'_\alpha & f_0\bar{x}'_\alpha & f_0\bar{y}_\alpha & 0 & 0 \\ \bar{y}_\alpha^2 + \bar{y}_\alpha'^2 & f_0\bar{y}'_\alpha & 0 & f_0\bar{y}_\alpha & 0 \\ f_0\bar{y}'_\alpha & f_0^2 & 0 & 0 & 0 \\ 0 & 0 & f_0^2 & 0 & 0 \\ f_0\bar{y}_\alpha & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (4)$$

Here, $(\bar{x}_\alpha, \bar{y}_\alpha)$ is the true position of (x_α, y_α). It is replaced by (x_α, y_α) in actual computation†.

Let $\hat{\boldsymbol{u}}$ be an estimate of $\boldsymbol{u}$. We define its covariance matrix $V[\hat{\boldsymbol{u}}]$ by

$$V[\hat{\boldsymbol{u}}] = E[(\boldsymbol{P}_{\boldsymbol{u}}\hat{\boldsymbol{u}})(\boldsymbol{P}_{\boldsymbol{u}}\hat{\boldsymbol{u}})^\top], \quad (5)$$

where $\boldsymbol{P}_{\boldsymbol{u}}$ is the following projection matrix ($\boldsymbol{I}$ denotes the unit matrix):

$$\boldsymbol{P}_{\boldsymbol{u}} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top. \quad (6)$$

Since $\boldsymbol{u}$ is normalized to unit norm, its domain is the unit sphere $\mathcal{S}^8$ in $\mathcal{R}^9$. Equation (5) means that the error is evaluated after projected onto the tangent space to $\mathcal{S}^8$ at $\boldsymbol{u}$, assuming that the noise is sufficiently small.

It has been shown by Kanatani [9] that if $\boldsymbol{\xi}_\alpha$ is identified with an independent Gaussian random variable of mean $\bar{\boldsymbol{\xi}}_\alpha$ and covariance matrix $V[\boldsymbol{\xi}_\alpha]$, the covariance matrix $V[\hat{\boldsymbol{u}}]$ of any unbiased estimator satisfies

$$V[\hat{\boldsymbol{u}}] \succ \sigma^2\Big(\sum_{\alpha=1}^N \frac{\bar{\boldsymbol{\xi}}_\alpha\bar{\boldsymbol{\xi}}_\alpha^\top}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}\Big)_8^-, \quad (7)$$

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

Chernov and Lesort [2] called the right-hand side of Eq. (7) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that this holds except for $O(\sigma^4)$ even if $\hat{\boldsymbol{u}}$ is not unbiased; it is sufficient that $\hat{\boldsymbol{u}}$ is "consistent" in the sense that $\hat{\boldsymbol{u}} \to \boldsymbol{u}$ as $\sigma \to 0$.

The fundamental matrix $\boldsymbol{F}$ should also satisfy the constraint that $\det \boldsymbol{F} = 0$ [7]. If this constraint is taken into account, the KCR lower bound involves the corresponding projection operation [9], [12].

3. Maximum Likelihood (ML)

If $\boldsymbol{\xi}_\alpha$ is regarded as an independent Gaussian random variable of mean $\bar{\boldsymbol{\xi}}_\alpha$ and covariance matrix $V[\boldsymbol{\xi}_\alpha]$, *maximum likelihood* (*ML*) is to minimize the sum of the square Mahalanobis distances of the data points $\boldsymbol{\xi}_\alpha$ from the hyperplane to be fitted in $\mathcal{R}^9$, i.e.,

$$J = \frac{1}{2}\sum_{\alpha=1}^N (\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha, V_0[\boldsymbol{\xi}_\alpha]_4^-(\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha)), \quad (8)$$

subject to the constraint $(\boldsymbol{u}, \bar{\boldsymbol{\xi}}_\alpha) = 0$, $\alpha = 1, ..., N$, where we can use $V_0[\boldsymbol{\xi}_\alpha]$ instead of $V[\boldsymbol{\xi}_\alpha]$ because the solution is unchanged if $V_0[\boldsymbol{\xi}_\alpha]$ is multiplied by a positive constant††. Introducing Lagrange multipliers for the constraint $(\boldsymbol{u}, \bar{\boldsymbol{\xi}}_\alpha) = 0$, we can reduce the problem to unconstrained minimization of the following function [3], [9], [13]:

$$J = \frac{1}{2}\sum_{\alpha=1}^N \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}. \quad (9)$$

The solution is obtained by solving

$$\begin{aligned} \nabla_{\boldsymbol{u}} J &= \sum_{\alpha=1}^N \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)\boldsymbol{\xi}_\alpha}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} - \sum_{\alpha=1}^N \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u}}{(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2} \\ &= (\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u} = \boldsymbol{0}, \end{aligned} \quad (10)$$

where we define

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

We need not consider the normalization constraint $\|\boldsymbol{u}\| = 1$, because Eq. (9) is a homogeneous expression of degree 0 in $\boldsymbol{u}$. In fact, multiplication of $\boldsymbol{u}$ by a nonzero constant does not affect the value of J, and hence the gradient $\nabla_{\boldsymbol{u}} J$ is always orthogonal to $\boldsymbol{u}$. It can be shown that the covariance matrix of the solution $\hat{\boldsymbol{u}}$ coincides with the right-hand side of Eq. (7) (the KCR lower bound) except for $O(\sigma^4)$ [2], [9], [10].

We further need to impose the constraint $\det \boldsymbol{F} = 0$. However, once the solution $\hat{\boldsymbol{u}}$ of Eq. (10) is obtained, it can be easily corrected so as to satisfy $\det \boldsymbol{F} = 0$ in such a way that the accuracy is equivalent to the constrained minimization of Eq. (9) subject to $\det \boldsymbol{F} = 0$ except for higher order terms in σ [12] (see Appendix A for the procedure). Since this is common to all methods for solving Eq. (10), we compare in the following the performance difference among different methods before imposing the constraint $\det \boldsymbol{F} = 0$.

3.1 Fundamental Numerical Scheme (FNS)

The procedure called *FNS* (*fundamental numerical scheme*) of Chojnacki et al. [3] for solving Eq. (10) is described as follows:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ in Eqs. (11).

†It has been confirmed by simulation that this replacement or omission of terms $O(\sigma^4)$ does not produce any significant changes.

††Note that $V_0[\boldsymbol{\xi}_\alpha]$ has rank 4: it has only four degrees of freedom for $\bar{x}_\alpha$, $\bar{y}_\alpha$, $\bar{x}'_\alpha$, and $\bar{y}'_\alpha$.

3. Solve the eigenvalue problem
$$(\boldsymbol{M}-\boldsymbol{L})\boldsymbol{u}'=\lambda\boldsymbol{u}', \tag{12}$$
and compute the unit eigenvector $\boldsymbol{u}'$ for the eigenvalue λ closest to 0.
4. If $\boldsymbol{u}'\approx\boldsymbol{u}$ except for sign, return $\boldsymbol{u}'$ and stop. Else, let $\boldsymbol{u}\leftarrow\boldsymbol{u}'$ and go back to Step 2.

Chojnacki et al. [4] also showed how to incorporate the constraint $\det\boldsymbol{F}=0$ in the above iterations. Later, they pointed out that convergence performance improves if we choose in Step 3 not the eigenvalue closest to 0 but the smallest one [5]. We call the above procedure the *original FNS* and the one using the smallest eigenvalue the *modified FNS*.

Whichever eigenvalue is chosen for λ, we have $\lambda=0$ after convergence. In fact, convergence means
$$(\boldsymbol{M}-\boldsymbol{L})\boldsymbol{u}=\lambda\boldsymbol{u} \tag{13}$$
for some $\boldsymbol{u}$. Computing the inner product of both sides with $\boldsymbol{u}$, we have
$$(\boldsymbol{u},\boldsymbol{M}\boldsymbol{u})-(\boldsymbol{u},\boldsymbol{L}\boldsymbol{u})=\lambda. \tag{14}$$
On the other hand, Eqs. (11) imply that $(\boldsymbol{u},\boldsymbol{M}\boldsymbol{u})=(\boldsymbol{u},\boldsymbol{L}\boldsymbol{u})$ identically, meaning $\lambda=0$.

3.2 Heteroscedastic Errors in Variables (HEIV)

Equation (10) can be rewritten as
$$\boldsymbol{M}\boldsymbol{u}=\boldsymbol{L}\boldsymbol{u}. \tag{15}$$
The *HEIV* (*heteroscedastic errors-in-variables*) method of Leedan and Meer [13] is to iteratively solve the generalized eigenvalue problem $\boldsymbol{M}\boldsymbol{u}=\lambda\boldsymbol{L}\boldsymbol{u}$. However, we cannot directly solve this, because $\boldsymbol{L}$ is not positive definite. So, we write
$$\boldsymbol{\xi}_\alpha=\begin{pmatrix}\boldsymbol{z}_\alpha\\ f_0^2\end{pmatrix},\quad \boldsymbol{u}=\begin{pmatrix}\boldsymbol{v}\\ F_{33}\end{pmatrix},$$
$$V_0[\boldsymbol{\xi}_\alpha]=\begin{pmatrix}V_0[\boldsymbol{z}_\alpha] & \boldsymbol{0}\\ \boldsymbol{0}^\top & 0\end{pmatrix}, \tag{16}$$
and define 8×8 matrices $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{L}}$ by
$$\tilde{\boldsymbol{M}}=\sum_{\alpha=1}^{N}\frac{\tilde{\boldsymbol{z}}_\alpha\tilde{\boldsymbol{z}}_\alpha^\top}{(\boldsymbol{v},V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})},\quad \tilde{\boldsymbol{L}}=\sum_{\alpha=1}^{N}\frac{(\boldsymbol{v},\tilde{\boldsymbol{z}}_\alpha)^2V_0[\boldsymbol{z}_\alpha]}{(\boldsymbol{v},V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})^2}, \tag{17}$$
where we put
$$\tilde{\boldsymbol{z}}_\alpha=\boldsymbol{z}_\alpha-\bar{\boldsymbol{z}},$$
$$\bar{\boldsymbol{z}}=\sum_{\alpha=1}^{N}\frac{\boldsymbol{z}_\alpha}{(\boldsymbol{v},V_0[\boldsymbol{z}_\alpha]\boldsymbol{v})}\Bigg/\sum_{\beta=1}^{N}\frac{1}{(\boldsymbol{v},V_0[\boldsymbol{z}_\beta]\boldsymbol{v})}. \tag{18}$$
Then, Eq. (15) splits into the following two equations [5], [13]:
$$\tilde{\boldsymbol{M}}\boldsymbol{v}=\tilde{\boldsymbol{L}}\boldsymbol{v},\qquad (\boldsymbol{v},\bar{\boldsymbol{z}})+f_0^2F_{33}=0. \tag{19}$$
Thus, if an 8-dimensional unit vector $\boldsymbol{v}$ that satisfies the first equation is computed, the second equation gives F_{33}, and we obtain
$$\boldsymbol{u}=N\Big[\begin{pmatrix}\boldsymbol{v}\\ F_{33}\end{pmatrix}\Big], \tag{20}$$
where $N[\,\cdot\,]$ denotes normalization to unit norm. The vector $\boldsymbol{u}$ that satisfies the first of Eqs. (19) is computed by the following iterations [5], [13]:

1. Initialize $\boldsymbol{v}$.
2. Compute the matrices $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{L}}$ in Eqs. (17).
3. Solve the generalized eigenvalue problem (cf. Appendix B)
$$\tilde{\boldsymbol{M}}\boldsymbol{v}'=\lambda\tilde{\boldsymbol{L}}\boldsymbol{v}', \tag{21}$$
and compute the unit generalized eigenvector $\boldsymbol{v}'$ for the generalized eigenvalue λ closest to 1.
4. If $\boldsymbol{v}'\approx\boldsymbol{v}$ except for sign, return $\boldsymbol{v}'$ and stop. Else, let $\boldsymbol{v}\leftarrow\boldsymbol{v}'$ and go back to Step 2.

However, Leedan and Meer [13] observed that choosing in Step 3 not the generalized eigenvalue closest to 1 but the smallest one improves the convergence performance. Here, we call the above procedure the *original HEIV* and the one using the smallest generalized eigenvalue the *modified HEIV*.

Whichever generalized eigenvalue is chosen for λ, we have $\lambda=1$ after convergence. In fact, convergence means
$$\tilde{\boldsymbol{M}}\boldsymbol{v}=\lambda\tilde{\boldsymbol{L}}\boldsymbol{v} \tag{22}$$
for some $\boldsymbol{v}$. Computing the inner product of both sides with $\boldsymbol{v}$, we have
$$(\boldsymbol{v},\tilde{\boldsymbol{M}}\boldsymbol{v})=\lambda(\boldsymbol{v},\tilde{\boldsymbol{L}}\boldsymbol{v}). \tag{23}$$
On the other hand, Eqs. (17) imply that $(\boldsymbol{v},\tilde{\boldsymbol{M}}\boldsymbol{v})=(\boldsymbol{v},\tilde{\boldsymbol{L}}\boldsymbol{v})$ identically, meaning $\lambda=1$.

3.3 Renormalization

The *renormalization* of Kanatani [9] is to approximate the matrix $\boldsymbol{L}$ in Eqs. (11) in the form
$$\boldsymbol{L}\approx c\boldsymbol{N},\qquad \boldsymbol{N}=\sum_{\alpha=1}^{N}\frac{V_0[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u},V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}. \tag{24}$$
The constant c is determined so that $\boldsymbol{M}-c\boldsymbol{N}$ has eigenvalue 0. This is done by the following iterations [9]:

1. Initialize $\boldsymbol{u}$ and let $c=0$.
2. Compute the matrix $\boldsymbol{M}$ in Eqs. (11) and the matrix $\boldsymbol{N}$ in Eqs. (24).
3. Solve the eigenvalue problem
$$(\boldsymbol{M}-c\boldsymbol{N})\boldsymbol{u}'=\lambda\boldsymbol{u}', \tag{25}$$
and compute the unit eigenvector $\boldsymbol{u}'$ for the eigenvalue λ closest to 0.
4. If $\lambda\approx0$, return $\boldsymbol{u}'$ and stop. Else, let
$$c\leftarrow c+\frac{\lambda}{(\boldsymbol{u}',\boldsymbol{N}\boldsymbol{u}')},\qquad \boldsymbol{u}\leftarrow\boldsymbol{u}' \tag{26}$$
and go back to Step 2.

3.4 Gauss-Newton Iterations

Since the gradient $\nabla_{\boldsymbol{u}} J$ is given by Eq. (10), we can minimize the function J in Eq. (9) by Newton iterations. If we evaluate the Hessian $\nabla^2_{\boldsymbol{u}} J$, the increment $\Delta\boldsymbol{u}$ in $\boldsymbol{u}$ is determined by solving

$$(\nabla^2_{\boldsymbol{u}} J)\Delta\boldsymbol{u} = -\nabla_{\boldsymbol{u}} J. \tag{27}$$

Since $\nabla^2_{\boldsymbol{u}} J$ is singular (recall that J is constant in the direction of $\boldsymbol{u}$), the solution is indeterminate. However, if we use pseudoinverse and compute

$$\Delta\boldsymbol{u} = -(\nabla^2_{\boldsymbol{u}} J)^-_8 \nabla_{\boldsymbol{u}} J, \tag{28}$$

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

Differentiating Eq. (10) 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 $\boldsymbol{M}$ in Eqs. (11). We enforce $\boldsymbol{M}$ to have eigenvalue 0 for $\boldsymbol{u}$, using the projection matrix $\boldsymbol{P}_{\boldsymbol{u}}$ of Eq. (6). The iteration procedure goes as follows:

1. Initialize $\boldsymbol{u}$.
2. Compute
$$\boldsymbol{u}' = N[\boldsymbol{u} - (\boldsymbol{P}_{\boldsymbol{u}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{u}})^-_8 (\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u}]. \tag{29}$$
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.

4. Initialization

We test the following three types of initialization to examine the dependence of convergence properties on initial values.

4.1 Random Choice

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

4.2 Least Squares (LS)

Approximating the denominators in Eq. (9) by a constant, we minimize

$$J_{\mathrm{LS}} = \frac{1}{2}\sum_{\alpha=1}^{N} (\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 = \frac{1}{2}(\boldsymbol{u}, \boldsymbol{M}_{\mathrm{LS}}\boldsymbol{u}), \tag{30}$$

where we define

$$\boldsymbol{M}_{\mathrm{LS}} = \sum_{\alpha=1}^{N} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top. \tag{31}$$

Equation (30) is minimized by the unit eigenvalue $\boldsymbol{u}$ of $\boldsymbol{M}_{\mathrm{LS}}$ for the smallest eigenvalue.

4.3 Taubin's Method

Replacing the denominators in Eq. (9) by their average, we minimize the following function[†] [14]:

$$J_{\mathrm{TB}} = \frac{1}{2}\frac{\sum_{\alpha=1}^{N}(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2}{\sum_{\alpha=1}^{N}(\boldsymbol{u}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{u})} = \frac{1}{2}\frac{(\boldsymbol{u}, \boldsymbol{M}_{\mathrm{LS}}\boldsymbol{u})}{(\boldsymbol{u}, \boldsymbol{N}_{\mathrm{TB}}\boldsymbol{u})}. \tag{32}$$

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

$$\boldsymbol{N}_{\mathrm{TB}} = \sum_{\alpha=1}^{N} V_0[\boldsymbol{\xi}_\alpha]. \tag{33}$$

Equation (32) is minimized by solving the generalized eigenvalue problem

$$\boldsymbol{M}_{\mathrm{LS}}\boldsymbol{u} = \lambda \boldsymbol{N}_{\mathrm{TB}}\boldsymbol{u} \tag{34}$$

for the smallest generalized eigenvalue. However, we cannot directly solve this, because $\boldsymbol{N}_{\mathrm{TB}}$ is not positive definite. So, we decompose $\boldsymbol{\xi}_\alpha$, $\boldsymbol{u}$, and $V_0[\boldsymbol{\xi}_\alpha]$ in the form of Eqs. (16) and define 8×8 matrices $\tilde{\boldsymbol{M}}_{\mathrm{LS}}$ and $\tilde{\boldsymbol{N}}_{\mathrm{TB}}$ by

$$\tilde{\boldsymbol{M}}_{\mathrm{LS}} = \sum_{\alpha=1}^{N} \tilde{\boldsymbol{z}}_\alpha \tilde{\boldsymbol{z}}_\alpha^\top, \qquad \tilde{\boldsymbol{N}}_{\mathrm{TB}} = \sum_{\alpha=1}^{N} V_0[\boldsymbol{z}_\alpha], \tag{35}$$

where

$$\tilde{\boldsymbol{z}}_\alpha = \boldsymbol{z}_\alpha - \bar{\boldsymbol{z}}, \qquad \bar{\boldsymbol{z}} = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{z}_\alpha. \tag{36}$$

Then, Eq. (34) splits into two equations

$$\tilde{\boldsymbol{M}}_{\mathrm{LS}}\boldsymbol{v} = \lambda \tilde{\boldsymbol{N}}_{\mathrm{TB}}\boldsymbol{v}, \qquad (\boldsymbol{v}, \bar{\boldsymbol{z}}) + f_0^2 F_{33} = 0. \tag{37}$$

We compute the unit generalized eigenvector $\boldsymbol{v}$ of the first equation for the smallest generalized eigenvalue λ (see Appendix B). The second equation gives F_{33}, and $\boldsymbol{u}$ is given in the form of Eq. (20).

5. Numerical Examples

5.1 Simulated Images

Figure 1 shows two simulated images of two planar grid planes joined at angle 60°. The image size is

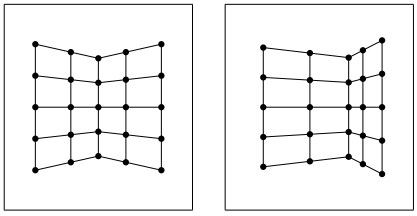


Fig. 1 Simulated images of planar grid surfaces.

[†]Taubin [14] did not take the covariance matrix into account. This is a modification of his method.

600×600 (pixels), and the focal length is 1200 (pixels). We added random Gaussian noise of mean 0 and standard deviation σ (pixels) to the image coordinates of each grid point independently and computed the fundamental matrix by FNS, HEIV, renormalization, and Gauss-Newton iterations.

Figure 2 plots for each σ the root-mean-squares of $\|\boldsymbol{P}_{\boldsymbol{u}}\hat{\boldsymbol{u}}\|$ over 1000 independent trials. We compared LS, Taubin's method, and the four iterative methods starting from the Taubin solution and confirmed that for each method the final solution does not depend on the initial value as long as the iterations converge. The dotted line indicates the KCR lower bound implied by Eq. (7).

We see that Taubin's method is considerably better[†] than LS. The four iterative methods indeed improve the Taubin solution, but the improvement is rather small. All the solutions nearly agree with the KCR lower bound when noise is small and gradually deviate from it as noise increases. Since FNS, HEIV, and Gauss-Newton minimize the same function, the resulting solution is virtually the same. The renormalization solution is nearly equivalent to them.

Figure 3 shows the average number of iterations of each method for 1000 trials. We stopped when the increment in $\boldsymbol{u}$ was less than 10^{-6} in norm (the sign of the eigenvector was chosen so that the orientation aligns with the previous solution). Figure 3(a) is for random initialization. The original FNS did not converge for about 99% of the trials after 100 iterations; the original HEIV about 40%. We stopped after 100 iterations and set the iteration count to 100.

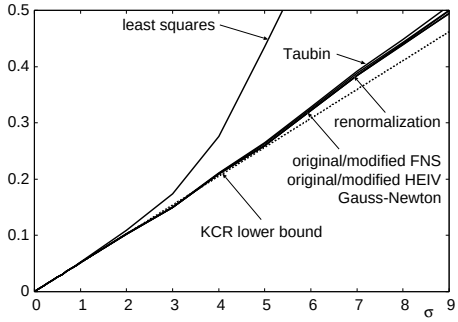


Fig. 2 Root-mean-squares error vs. noise level.

Figure 3(a) shows that the modified FNS/HEIV converge much more quickly than the original FNS/HEIV. This can be explained as follows. If the computed $\boldsymbol{u}'$ is close to the true value $\boldsymbol{u}$, the matrix $\boldsymbol{L}$ in Eqs. (11) and the matrix $\tilde{\boldsymbol{L}}$ in Eqs. (17) are both close to $\boldsymbol{O}$. Initially, however, they may be very different from $\boldsymbol{O}$ when the initial value is randomly chosen. Equations (12) and (21) are written, respectively, as

$$(\boldsymbol{M} - \boldsymbol{L} - \lambda\boldsymbol{I})\boldsymbol{u}' = \boldsymbol{0}, \quad (\tilde{\boldsymbol{M}} - \lambda\tilde{\boldsymbol{L}})\boldsymbol{v}' = \boldsymbol{0}. \quad (38)$$

Note that $\boldsymbol{L}$ and $\tilde{\boldsymbol{L}}$ are both positive definite. In order to cancel their effects, we need to choose λ to be negative in the first equation and smaller than 1 in the second.

As predicted from this explanation, the difference between the original FNS/HEIV and the modified FNS/HEIV shrinks as we use better initial values, as seen from Fig. 3(b), (c). We also see that the (original or modified) FNS is more efficient than (original or modified) HEIV.

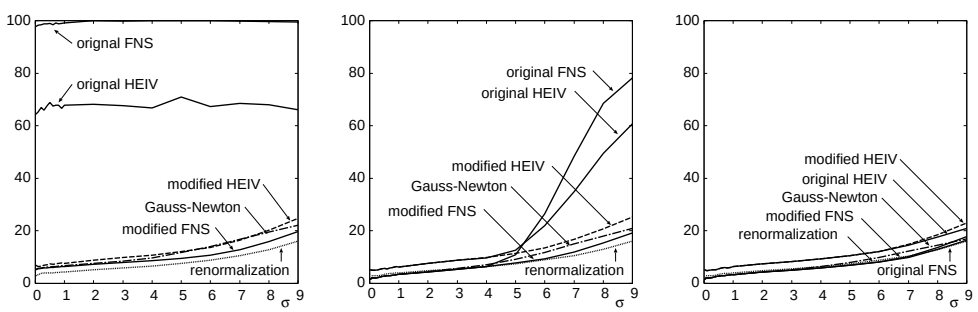
Another finding is that, for random initialization, renormalization is the most efficient. This is because we start solving Eq. (25) with $c = 0$, canceling the effect of $\boldsymbol{N}$ whatever it is, and the resulting $\boldsymbol{u}'$ is close to the LS solution. In contrast, FNS and HEIV may produce a solution very different from the true value when initially the matrices $\boldsymbol{L}$ and $\tilde{\boldsymbol{L}}$ are very different from $\boldsymbol{O}$.

As Fig. 3(b), (c) shows, however, the convergence performance of FNS and HEIV improves as we use better initial values. Naturally, Gauss-Newton iterations converge faster when started from better initial values. In contrast, renormalization behaves almost independently of initialization, confirming the above explanation. Overall, Taubin-initialized (original or modified) FNS shows the best convergence performance.

5.2 Real Images

Figure 4 shows two images of the same scene. We manually chose corresponding 100 points as marked there and computed the fundamental matrix by six different

[†]The mechanism of the superiority of Taubin's method over LS is analyzed in detail in [10].



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

Fig.3 Average number of iterations vs. noise level.

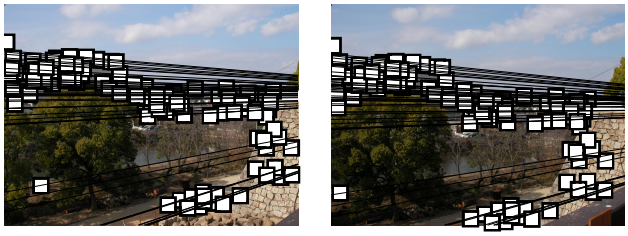


Fig. 4 Corresponding points in real images and computed epipolar lines.

Table 1 Number of iterations and the computation time (seconds) of the original/modified FNS/HEIV, renormalization and Gauss-Newton iterations for random initialization, LS initialization and Taubin initialization.

	random		LS		Taubin	
original FNS	94.3	.168s	5	.009s	5	.012s
modified FNS	12.0	.030s	5	.010s	5	.013s
original HEIV	74.6	.264s	7	.019s	7	.025s
modified HEIV	9.1	.037s	7	.020s	7	.026s
renormalization	7.0	.022s	7	.013s	7	.017s
Gauss-Newton	10.3	.038s	5	.009s	6	.017s

methods. The solution is the same whichever is used, and the computed epipolar lines are drawn in the images.

Table 1 lists the number of iterations and the computation time (seconds) of each method. For random initialization, we computed the average over 100 independent trials. We used Pentium 4 3.4GHz for the CPU with 2GB main memory and Linux for the OS.

We observe that for whichever initialization, FNS is always better than HEIV. For both, the choice of the eigenvalue is irrelevant if the iterations are initialized by LS or Taubin' method; for random initialization, the original FNS/HEIV do not converge in most of the trials (recall that 100 means nonconvergence). As predicted, the number of iterations of renormalization does not depend on initialization.

The difference in computation time between LS and Taubin initializations is due to the initialization computation: 0.0009s for LS vs. 0.0015s for Taubin. Overall, LS initialized (original or modified) FNS shows the best convergence performance.

6. Conclusions

We have compared the convergence performance of different numerical schemes for computing the fundamental matrix from point correspondences over two images. First, we stated the problem and the associated KCR lower bound. Then, we described FNS, HEIV, and renormalization. We also introduced Gauss-Newton iterations as a new method for fundamental matrix computation. For initial values, we tested random choice, LS, and Taubin's method. Experiments using simulated and real images revealed different characteristics of each method. Overall, FNS exhibited the best convergence performance.

Acknowledgments

The author thanks Nikolai Chernov of the University of Alabama, U.S.A., and Wojciech Chojnacki of the University of Adelaide, Australia, for helpful discussions. This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C (No. 17500112).

References

[1] N. Chernov, "On the convergence of numerical schemes in computer vision," J. Math. Imaging Vis., vol.25, 2007, to appear.
[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.
[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 Vis., vol.23, no.2, pp.175–183, Sept. 2005.
[6] R.I. Hartley, "In defense of the eight-point algorithm," IEEE Trans. Pattern 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, Geometric Computation for Machine Vision, Oxford University Press, Oxford, U.K., 1993.
[9] K. Kanatani, Statistical Optimization for Geometric Computation: Theory and Practice, Elsevier Science, Amsterdam, The Netherlands, 1996; Dover, New York, 2005.
[10] K. Kanatani, "Hyperaccuracy for geometric fitting," 4th Int. Workshop on Total Least Squares and Errors-in-Variables Modelling, Leuven, Belgium, Aug. 2006.
[11] K. Kanatani, "Performance evaluation of accurate ellipse fitting," Proc. 21th Int. Conf. Image Vis. Comput. New Zealand, Great Barrier Island, New Zealand, pp.7–12, Nov. 2006.
[12] K. Kanatani and N. Ohta, "Comparing optimal three-dimensional reconstruction for finite motion and optical flow," J. Elec. Imaging, vol.12, no.3, pp.478–488, July 2003.
[13] Y. Leedan and P. Meer, "Heteroscedastic regression in computer vision: Problems with bilinear constraint," Int. J. Comput. Vis., vol.37, no.2, pp.127–150, June 2000.
[14] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol.13, no.11, pp.1115–1138, Nov. 1991.

Appendix A. Rank Constraint Optimization

The computed fundamental matrix $\boldsymbol{F}$ can be optimally corrected so as to satisfy $\det \boldsymbol{F} = 0$ as follows [9], [12].

Let $\hat{\boldsymbol{u}}$ be the 9-dimensional vector representation of the ML estimate $\hat{\boldsymbol{F}}$ of the fundamental matrix $\boldsymbol{F}$ computed without the constraint $\det \boldsymbol{F} = 0$. Compute

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

where $\boldsymbol{P}_{\hat{\boldsymbol{u}}}$ is the projection matrix in Eq. (6) defined for $\hat{\boldsymbol{u}}$. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_9 \ (= 0)$ be the eigenvalues of $\tilde{\boldsymbol{M}}$, and $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, ..., $\boldsymbol{u}_9$ $(= \hat{\boldsymbol{u}})$ the corresponding orthonormal system of eigenvectors. The covariance matrix $V[\hat{\boldsymbol{u}}]$ of $\hat{\boldsymbol{u}}$ is estimated to be $\hat{\sigma}^2 V_0[\hat{\boldsymbol{u}}]$, where $\hat{\sigma}$ is an estimate of the noise standard deviation and $V_0[\hat{\boldsymbol{u}}]$ has the following form† [9]:

$$V_0[\hat{\boldsymbol{u}}] = \frac{\boldsymbol{u}_1\boldsymbol{u}_1^\top}{\lambda_1} + \cdots + \frac{\boldsymbol{u}_8\boldsymbol{u}_8^\top}{\lambda_8}. \tag{A·2}$$

We update $\hat{\boldsymbol{u}}$ and $V_0[\hat{\boldsymbol{u}}]$ iteratively until they converge as follows:

$$\hat{\boldsymbol{u}} \leftarrow N\Big[\hat{\boldsymbol{u}} - \frac{(\det \boldsymbol{F})V_0[\hat{\boldsymbol{u}}]\hat{\boldsymbol{u}}^\dagger}{(\hat{\boldsymbol{u}}^\dagger, V_0[\hat{\boldsymbol{u}}]\hat{\boldsymbol{u}}^\dagger)}\Big], \tag{A·3}$$

$$V_0[\hat{\boldsymbol{u}}] \leftarrow \boldsymbol{P}_{\hat{\boldsymbol{u}}} V_0[\hat{\boldsymbol{u}}] \boldsymbol{P}_{\hat{\boldsymbol{u}}}. \tag{A·4}$$

Here, $\hat{\boldsymbol{u}}^\dagger$ is the vector representation of the transposed cofactor matrix $\hat{\boldsymbol{F}}^{\dagger\top}$:

$$\hat{\boldsymbol{u}}^\dagger = \begin{pmatrix} \hat{u}_5\hat{u}_9 - \hat{u}_8\hat{u}_6 \\ \hat{u}_6\hat{u}_7 - \hat{u}_9\hat{u}_4 \\ \hat{u}_4\hat{u}_8 - \hat{u}_7\hat{u}_5 \\ \hat{u}_8\hat{u}_3 - \hat{u}_2\hat{u}_6 \\ \hat{u}_9\hat{u}_1 - \hat{u}_3\hat{u}_7 \\ \hat{u}_7\hat{u}_2 - \hat{u}_1\hat{u}_8 \\ \hat{u}_2\hat{u}_6 - \hat{u}_5\hat{u}_3 \\ \hat{u}_3\hat{u}_4 - \hat{u}_6\hat{u}_1 \\ \hat{u}_1\hat{u}_5 - \hat{u}_4\hat{u}_2 \end{pmatrix}. \tag{A·5}$$

Appendix B. Generalized Eigenvalue Problem

Given a positive definite symmetric matrix $\boldsymbol{G}$ and a symmetric matrix $\boldsymbol{A}$, we want to compute a real number λ and a vector $\boldsymbol{w}$ such that

$$\boldsymbol{A}\boldsymbol{w} = \lambda \boldsymbol{G}\boldsymbol{w}. \tag{A·6}$$

In n dimensions, there exist n such real numbers (allowing multiplicity) λ_1, ..., λ_n, called *generalized eigenvalues*, and corresponding vectors $\boldsymbol{w}_1$, ..., $\boldsymbol{w}_n$, called the *generalized eigenvectors*. If $\boldsymbol{G}$ is the unit matrix $\boldsymbol{I}$, Eq. (A·6) reduces to the usual eigenvalue problem.

Equation (A·6) is solved as follows [9]. Let μ_1, ..., μ_n be the eigenvalues of $\boldsymbol{G}$, and $\boldsymbol{g}_1$, ..., $\boldsymbol{g}_n$ the corresponding orthonormal system of eigenvectors. Define

$$\boldsymbol{T} = \frac{\boldsymbol{g}_1\boldsymbol{g}_1^\top}{\sqrt{\mu_1}} + \frac{\boldsymbol{g}_2\boldsymbol{g}_2^\top}{\sqrt{\mu_2}} + \cdots + \frac{\boldsymbol{g}_n\boldsymbol{g}_n^\top}{\sqrt{\mu_n}}. \tag{A·7}$$

Let λ_1, ..., λ_n be the eigenvalues of $\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}$, and $\boldsymbol{u}_1$, ..., $\boldsymbol{u}_n$ the corresponding orthonormal system of eigenvectors. Then, the generalized eigenvalues of $\boldsymbol{A}$ with respect to $\boldsymbol{W}$ are λ_1, ..., λ_n, and the corresponding generalized eigenvectors are $\boldsymbol{w}_1 = \boldsymbol{T}\boldsymbol{u}_1$, ..., $\boldsymbol{w}_n = \boldsymbol{T}\boldsymbol{u}_n$.

†For numerical computation, we multiply this expression by λ_8 to make it $O(1)$ to prevent numerical instability.

Kenichi Kanatani received his B.E., M.S., and Ph.D. in applied mathematics from the University of Tokyo in 1972, 1974 and 1979, respectively. After serving as Professor of computer science at Gunma University, Gunma, Japan, he is currently Professor of computer science at Okayama University, Okayama, Japan. He is the author of many books on computer vision, and he has received many awards including best paper awards from IPSJ (1987) and IEICE (2005). He is an IEEE Fellow.

Yasuyuki Sugaya received his B.E., M.S., and Ph.D. in computer science from the University of Tsukuba, Ibaraki, Japan, in 1996, 1998, and 2001, respectively. From 2001 to 2006, he was Assistant Professor of computer science at Okayama University, Okayama, Japan. Currently, he is Lecturer of information and computer sciences at Toyohashi University of Technology, Toyohashi, Aichi, Japan. His research interests include image processing and computer vision. He received the IEICE best paper award in 2005.