10th Meeting on Image Recognition and Understanding (MIRU2007), Hiroshima, Japan, July 2007, pp. 219–224.

Extended FNS for Constrained Parameter Estimation

Kenichi KANATANI† and Yasuyuki SUGAYA‡

‡Department of Computer Science, Okayama University, Okayama 700-8530 Japan 5 Affiliation
†Department of Information and Computer Sciences,
Toyohashi University of Technology, Toyohashi, Aichi 441-8580 Japan
E-mail: † kanatani@suri.it.okayama-u.ac.jp, ‡ sugaya@iim.ics.tut.ac.jp

Abstract We present a new method, called "EFNS" ("extended FNS"), for linearizable constrained maximum likelihood estimation. This complements the CFNS of Chojnacki et al. and is a true extension of the FNS of Chojnacki et al. to an arbitrary number of intrinsic constraints. Computing the fundamental matrix as an illustration, we demonstrate that CFNS does not necessarily converge to a correct solution, while EFNS converges to an optimal value which nearly satisfies the theoretical accuracy bound (KCR lower bound).

Keywords parameter estimation, maximum likelihood, CFNS, KCR lower bound, fundamental matrix, rank constraint

1. Introduction

One of the fundamental principles of computer vision is to hypothesize a structure, expressed as parameterized equations, in the scene we are viewing and estimate the parameters by fitting the equations to the observed images. Such a structure is called a *model*, and the equations it implies *constraints*.

Most deeply studied in the past are the problems for which the constraints are linear in unknowns, for which many numerical techniques have been proposed for computing the maximum likelihood (ML) solution, including FNS (Fundamental Numerical Scheme) [2], HEIV (Heteroscedastic Errors-in-Variable) [9], renormalization [6], and projective Gauss-Newton iterations [8].

All these assume that the unknown parameters are unconstrained. In many problems, however, we introduce intrinsic constraints to avoid complications arising from minimal parameterization. For example, the "epipolar equation" can be succinctly described in terms of the fundamental matrix $\boldsymbol{F}$ constrained to be $\det \boldsymbol{F} = 0$.

Hereafter, we call the equations that link variables with "data" *extrinsic* constraints and those only "among" variables *intrinsic* constraints. Typical approaches to optimization with intrinsic constraints are:

A posteriori correction. We first compute the solution without considering the intrinsic constraints and then modify the solution so that they are satisfied (Fig. 1(a)).

Internal access. We minimally parameterize the problem and do optimization 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 intrinsic constraints automatically results (Fig. 1(c)).

Maximum likelihood (ML) estimation reduces to minimization of the negative logarithm of the likelihood, which is a relatively simple function if appropriate intrinsic constraints are introduced and can be easily minimized with a small number of iterations using a method like FNS [8, 7]. If the problem is minimally parameterized, however, we need to optimize a complicated nonlinear equation which has a lot of local minima [11], requiring close attention to avoid them.

The concept of external access 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. [3] that first presented a systematic scheme for constrained optimization of linearizable problem. They termed their method *CFNS* (*Constrained FNS*).

The purpose of this paper is i) to point out that CFNS does not necessarily converge to a correct solution and ii) to present a new scheme, called *EFNS* (*extended FNS*), which always converges to an optimal value. Our method is different from CFNS in the following respects:

- CFNS is for problems with a single intrinsic constraint, so multiple constraints must be combined into one, while EFNS incorporates any number of intrinsic constraints from the beginning.
- CFNS has no relation to FNS, while EFNS reduces to FNS if the number of intrinsic constraints is 0. In this sense, EFNS is a true extension of FNS.
- There is no guarantee that the CFNS solution is the desired value, while we can *prove* this for

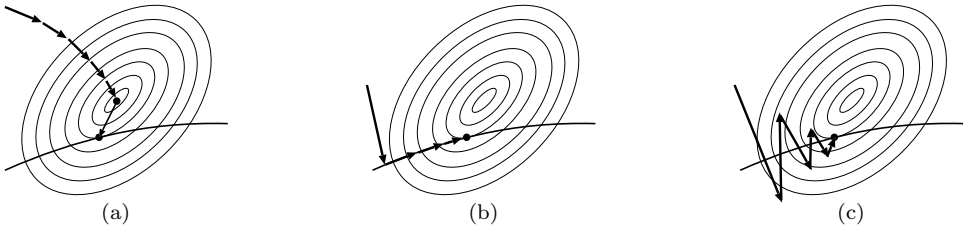

Figure 1: Contours of the function to minimize and the constraint surface. (a) A posteriori correction. (b) Internal access. (c) External access.

EFNS.

In Sec. 2, we summarize mathematical fundamentals. After briefly describing the CFNS of Chojnacki et al. [3] in Sec. 3, we present our EFNS in Sec. 4. In Sec. 5, we compare them by computing the fundamental matrix and demonstrate that CFNS does not necessarily converge to a correct value while EFNS converges to an optimal solution. We also show that our EFNS is more accurate than a posteriori correction and more robust than minimal parameterization. We conclude in Sec. 6.

2. Mathematical Fundamentals

Intrinsic constraints

We assume that the data $\{\boldsymbol{\xi}_\alpha\}$, $\alpha = 1, ..., N$, are n-D vectors whose noiseless values $\{\bar{\boldsymbol{\xi}}_\alpha\}$ should satisfy

$$(\boldsymbol{u}, \bar{\boldsymbol{\xi}}_\alpha) = 0, \qquad \alpha = 1, ..., N, \tag{1}$$

where $\boldsymbol{u}$ is an unknown n-D vector. In this paper, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$. Our task is to estimate $\boldsymbol{u}$ from noisy data $\{\boldsymbol{\xi}_\alpha\}$.

Since Eq. (1) does not constrain the scale of $\boldsymbol{u}$, we impose normalization $\|\boldsymbol{u}\| = 1$, which is an intrinsic constraint. In addition, we allow r constraints on $\boldsymbol{u}$:

$$\phi_k(\boldsymbol{u}) = 0, \qquad k = 1, ..., r. \tag{2}$$

We assume that these r equations and the normalization $\|\boldsymbol{u}\| = 1$ are algebraically independent and mutually transversal[1] [12]. We also assume that all these constrain the scale indeterminate $\boldsymbol{u}$. So, like Chojnacki et al. [3], we assume that each $\phi_k(\boldsymbol{u})$ is a homogeneous equation of degree κ_k in $\boldsymbol{u}$. Hence, the following identity holds for arbitrary nonzero t:

$$\phi_k(t\boldsymbol{u}) = t^{\kappa_k}\phi_k(\boldsymbol{u}). \tag{3}$$

Covariance matrices

We define the covariance matrix $V[\hat{\boldsymbol{u}}]$ of an estimate $\hat{\boldsymbol{u}}$ of $\boldsymbol{u}$ computed from noisy data $\{\boldsymbol{\xi}_\alpha\}$ by

$$V[\hat{\boldsymbol{u}}] = E[(\boldsymbol{P}_{\mathcal{U}}\hat{\boldsymbol{u}})(\boldsymbol{P}_{\mathcal{U}}\hat{\boldsymbol{u}})^\top], \tag{4}$$

where $E[\,\cdot\,]$ denotes expectation with respect to the noise distribution. The operator $\boldsymbol{P}_{\mathcal{U}}$ projects $\mathcal{R}^n$ onto the domain $\mathcal{U}$ of $\boldsymbol{u}$ defined by $\|\boldsymbol{u}\| = 1$ and $\phi_k(\boldsymbol{u}) = 0$, $k = 1, ..., r$. By the transversality assumption, it is an $(n - r - 1)$-D submanifold of $\mathcal{R}^n$. Eq. (4) means that the error of $\hat{\boldsymbol{u}}$ is evaluated after projected onto the tangent space $T_{\bar{\boldsymbol{u}}}(\mathcal{U})$ to $\mathcal{U}$ at $\bar{\boldsymbol{u}}$.

Complement space

Each constraint $\phi_k(\boldsymbol{u}) = 0$ defines a hypersurface in $\mathcal{R}^n$, whose unit surface normal is $\nabla_{\mathbf{u}}\phi_k(\boldsymbol{u})$. By the transversality assumption, vectors $\boldsymbol{u}$, $\nabla_{\mathbf{u}}\phi_1(\boldsymbol{u})$, ..., $\nabla_{\mathbf{u}}\phi_r(\boldsymbol{u})$ are linearly independent. Let

$$\mathcal{N} = \{\boldsymbol{u}, \nabla_{\mathbf{u}}\phi_1(\boldsymbol{u}), ..., \nabla_{\mathbf{u}}\phi_r(\boldsymbol{u})\}_{\mathcal{L}} \tag{5}$$

be the $(r + 1)$-D linear subspace they spanned[2]. This is the orthogonal complement of the tangent space $T_{\mathbf{u}}(\mathcal{U})$ of the domain $\mathcal{U}$ of $\boldsymbol{u}$:

$$\mathcal{N} = T_{\mathbf{u}}(\mathcal{U})^\perp. \tag{6}$$

We call $\mathcal{N}$ the *complement space* of $\boldsymbol{u}$.

KCR lower bound

If the noise in $\{\boldsymbol{\xi}_\alpha\}$ is independent and Gaussian with mean $\mathbf{0}$ and covariance matrix $V[\boldsymbol{\xi}_\alpha]$, the following inequality holds for an arbitrary unbiased estimator $\hat{\boldsymbol{u}}$ of $\boldsymbol{u}$ [6]:

$$V[\hat{\boldsymbol{u}}] \succ \Big(\sum_{\alpha=1}^{N} \frac{(\boldsymbol{P}_{\mathcal{U}}\bar{\boldsymbol{\xi}}_\alpha)(\boldsymbol{P}_{\mathcal{U}}\bar{\boldsymbol{\xi}}_\alpha)^\top}{(\boldsymbol{u}, V[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}\Big)^-_{n-r-1}. \tag{7}$$

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 [1] called the right-hand side of Eq. (9) the *KCR* (*Kanatani-Cramer-Rao*) *lower bound* and showed that Eq. (7) holds up to higher order noise terms even if $\hat{\boldsymbol{u}}$ is not unbiased; it is sufficient that $\hat{\boldsymbol{u}} \to \boldsymbol{u}$ as the noise decreases.

[1] This means that the hypersurfaces defined by individual constrains $\phi_k(\boldsymbol{u}) = 0$ do not share their tangent spaces at their intersections [12].

[2] Namely, this is the set of all directions along which the intrinsic constraint is violated.

[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.

Maximum likelihood (ML)

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

$$J = \sum_{\alpha=1}^{N} (\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha, V[\boldsymbol{\xi}_\alpha]_{n-s}^{-}(\boldsymbol{\xi}_\alpha - \bar{\boldsymbol{\xi}}_\alpha)), \tag{8}$$

subject to Eq. (1). We are assuming that the data $\{\boldsymbol{\xi}_\alpha\}$ are also subject to s algebraically independent intrinsic constraints, such as being unit vectors, but all the necessary information is contained in the form of $V[\boldsymbol{\xi}_\alpha]$, so we need not explicitly indicate them.

Eliminating the constraint of Eq. (1) by using Lagrange multipliers, we obtain [6]

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

The ML estimator $\hat{\boldsymbol{u}}$ minimizes J subject to $\|\boldsymbol{u}\| = 1$, $\phi_k(\boldsymbol{u}) = 0$, $k = 1, ..., r$. It is known that its covariance matrix $V[\hat{\boldsymbol{u}}]$ agrees with the KCR lower bound (the right-side hand of Eq. (9)) up to higher order noise terms [6].

3. Constrained FNS

The *CFNS* (*Constrained FNS*) of Chojnacki et al. [3] minimizes Eq. (9) in the same form as their FNS [2]. Namely, we find a symmetric matrix $\boldsymbol{Q}$ such that the stationarity condition of Eq. (9) subject to the intrinsic constraints is written in the form

$$\boldsymbol{Q}\boldsymbol{u} = \boldsymbol{0}, \tag{10}$$

and do the following iterations:

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

$$\boldsymbol{Q}\boldsymbol{v} = \lambda\boldsymbol{v}, \tag{11}$$

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

Chojnacki et al. [3] recommend to correct the resulting solution appropriately so that the intrinsic constraints are strictly imposed.

Infinitely many candidates exist for the matrix $\boldsymbol{Q}$ with which the problem is written as Eq. (10), but not all of them allow the above iterations to converge. Chojnacki et al. [3] gave one, but the derivation is not written in their paper. In Sec. 5, we show that CFNS does not necessarily converge to a correct solution.

[4] This means we fit a hyperplane $(\boldsymbol{u}, \boldsymbol{\xi}) = 0$ to N points $\{\boldsymbol{\xi}_\alpha\}$ in $\mathcal{R}^9$ so as to minimize the sum of their distances inversely weighted by their covariance matrices $V[\boldsymbol{\xi}_\alpha]$.

4. Extended FNS

Here, we present a new scheme, called *EFNS* (*Extended FNS*), very similar to CFNS. Later, we show that EFNS is superior to CFNS (Sec. 5).

Stationarity condition

According to the variational principle, the necessary and sufficient condition for the function J to be stationary at a point in the submanifold $\mathcal{U} \subset \mathcal{R}^n$ is that its gradient $\nabla_{\mathbf{u}} J$ is orthogonal to $\mathcal{U}$ at that point. In terms of the complement space $\mathcal{N}$ defined by Eq. (6), this can be written as $\nabla_{\mathbf{u}} J \in \mathcal{N}$.

Eq. (9) is homogeneous with degree 0 in $\boldsymbol{u}$, so its value is unchanged by multiplication of $\boldsymbol{u}$ by a nonzero constant. This means that $\nabla_{\mathbf{u}} J$ is always orthogonal to $\boldsymbol{u}$. Hence, if we define

$$\mathcal{M} = \{\nabla_{\mathbf{u}}\phi_1(\boldsymbol{u}), ..., \nabla_{\mathbf{u}}\phi_r(\boldsymbol{u})\}_{\mathcal{L}}, \tag{12}$$

the condition $\nabla_{\mathbf{u}} J \in \mathcal{N}$ is equivalent to $\nabla_{\mathbf{u}} J \in \mathcal{M}$. Thus, if we let $\boldsymbol{P}_{\mathcal{V}}$ be the projection operator onto the orthogonal complement $\mathcal{M}^\perp$ of $\mathcal{M}$, the stationarity condition is written as

$$\boldsymbol{P}_{\mathcal{V}} \nabla_{\mathbf{u}} J = \boldsymbol{0}. \tag{13}$$

Let $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_r\}$ be the orthonormal system obtained by the Gram-Schmidt orthogonalization of $\nabla_{\mathbf{u}}\phi_1(\boldsymbol{u})$, ..., $\nabla_{\mathbf{u}}\phi_r(\boldsymbol{u})$. The projection operator $\boldsymbol{P}_{\mathcal{V}}$ has the following matrix representation:

$$\boldsymbol{P}_{\mathcal{V}} = \boldsymbol{I} - \sum_{k=1}^{r} \boldsymbol{u}_k \boldsymbol{u}_k^\top. \tag{14}$$

Intrinsic constraints

In addition to the stationarity condition of Eq. (13), the solution $\boldsymbol{u}$ must satisfy the intrinsic constraints $\phi_k(\boldsymbol{u}) = 0$, $k = 1, ..., r$. Since Eq. (3) is an identity in t, its derivation with respect to t on both sides is also an identity in t:

$$(\nabla_{\mathbf{u}}\phi_k(t\boldsymbol{u}), \boldsymbol{u}) = \kappa_k t^{\kappa_k - 1}\phi_k(\boldsymbol{u}). \tag{15}$$

Letting $t = 1$, we have

$$(\nabla_{\mathbf{u}}\phi_k(\boldsymbol{u}), \boldsymbol{u}) = \kappa_k \phi_k(\boldsymbol{u}). \tag{16}$$

Hence, $\phi_k(\boldsymbol{u}) = 0$, $k = 1, ..., r$, is equivalent to $(\nabla_{\mathbf{u}}\phi_k(\boldsymbol{u}), \boldsymbol{u}) = 0$, $k = 1, ..., r$, which means $\boldsymbol{u} \in \mathcal{M}^\perp$ (see Eq. (12)). It follows that the r intrinsic constraints are expressed in a single form as

$$\boldsymbol{P}_{\mathcal{V}}\boldsymbol{u} = \boldsymbol{u}. \tag{17}$$

The necessary and sufficient condition for J to be stationary subject to the intrinsic constraints is Eqs. (13) and (17).

Formulation

Differentiating Eq. (9) with respect to $\boldsymbol{u}$, we obtain

$$\nabla_{\mathbf{u}} J = \boldsymbol{M}\boldsymbol{u} - \boldsymbol{L}\boldsymbol{u}, \tag{18}$$

where we define

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} \frac{\boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top}{(\boldsymbol{u}, V[\boldsymbol{\xi}_\alpha]\boldsymbol{u})}, \quad \boldsymbol{L} = \sum_{\alpha=1}^{N} \frac{(\boldsymbol{u}, \boldsymbol{\xi}_\alpha)^2 V[\boldsymbol{\xi}_\alpha]}{(\boldsymbol{u}, V[\boldsymbol{\xi}_\alpha]\boldsymbol{u})^2}. \tag{19}$$

Hence, Eq. (13) is written as

$$\boldsymbol{P}_{\mathcal{V}}(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{u} = \boldsymbol{0}. \tag{20}$$

Using Eq. (17), we can write the left-hand side as $\boldsymbol{P}_{\mathcal{V}}(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{P}_{\mathcal{V}}\boldsymbol{u}$. Hence, if we define the symmetric matrix

$$\boldsymbol{X} = \boldsymbol{P}_{\mathcal{V}}(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{P}_{\mathcal{V}}, \tag{21}$$

our task is to find $\boldsymbol{u}$ that satisfies

$$\boldsymbol{X}\boldsymbol{u} = \boldsymbol{0}, \qquad \boldsymbol{P}_{\mathcal{V}}\boldsymbol{u} = \boldsymbol{u}. \tag{22}$$

The first equation implies that $\boldsymbol{u}$ belongs to the null space of $\boldsymbol{X}$. Since $\boldsymbol{P}_{\mathcal{V}}$ is the projection operator onto $\mathcal{M}^\perp$, Eq. (21) implies $\boldsymbol{X}\mathcal{M} = \boldsymbol{0}$. Hence, $\boldsymbol{u}$ and $\mathcal{M}$ both belong to the null space of $\boldsymbol{X}$, and they are mutually orthogonal by the second of Eqs. (22). Thus, the complement space $\mathcal{N}$ of Eq. (5) coincides with the null space of $\boldsymbol{X}$ and is the direct sum of $\boldsymbol{u}$ and $\mathcal{M}$:

$$\mathcal{N} = \{\boldsymbol{u}\}_{\mathcal{L}} \oplus \mathcal{M}. \tag{23}$$

As a result, the projection operator $\boldsymbol{P}_{\mathcal{U}}$ used in Eq. (4) has the following matrix form:

$$\boldsymbol{P}_{\mathcal{U}} = \boldsymbol{P}_{\mathcal{V}} - \boldsymbol{u}\boldsymbol{u}^\top. \tag{24}$$

Procedure

The above observation implies that $\boldsymbol{u}$ can be obtained by finding the complement space $\mathcal{N}$, because $\boldsymbol{u}$ is an element orthogonal to $\nabla_{\mathbf{u}}\phi_1(\boldsymbol{u})$, ..., $\nabla_{\mathbf{u}}\phi_r(\boldsymbol{u})$ in $\mathcal{N}$. The actual procedure is as follows:

1. Initialize $\boldsymbol{u}$.
2. Compute the matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ in Eqs. (19).
3. Compute the orthonormal system $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_r\}$ by the Gram-Schmidt orthogonalization of $\nabla_{\mathbf{u}}\phi_1(\boldsymbol{u})$, ..., $\nabla_{\mathbf{u}}\phi_r(\boldsymbol{u})$.
4. Compute the projection matrix $\boldsymbol{P}_{\mathcal{V}}$ in Eq. (14).
5. Compute the matrix $\boldsymbol{X}$ in Eq. (21).
6. Solve the eigenvalue problem
$$\boldsymbol{X}\boldsymbol{v} = \lambda\boldsymbol{v}, \tag{25}$$
and compute $r+1$ unit eigenvectors $\boldsymbol{v}_0$, ..., $\boldsymbol{v}_r$ for the smallest eigenvalues in absolute terms.
7. Project the current value $\boldsymbol{u}$ onto $\hat{\mathcal{N}} = \{\boldsymbol{v}_0, ..., \boldsymbol{v}_r\}_{\mathcal{L}}$ to compute
$$\hat{\boldsymbol{u}} = \sum_{k=0}^{r} (\boldsymbol{u}, \boldsymbol{v}_k)\boldsymbol{v}_k. \tag{26}$$
8. Compute
$$\boldsymbol{u}' = N[\boldsymbol{P}_{\mathcal{V}}\hat{\boldsymbol{u}}], \tag{27}$$
where $N[\,\cdot\,]$ denotes normalization to unit norm.
9. 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

When the above iterations have converged, the subspace $\hat{\mathcal{N}}$ coincides with the null space of $\boldsymbol{X}$.

Proof. From the definition of $\boldsymbol{X}$ in Eq. (21) and $\boldsymbol{P}_{\mathcal{V}}$ in Eq. (14), $\boldsymbol{u}_1$, ..., $\boldsymbol{u}_r$ are eigenvectors of $\boldsymbol{X}$ with eigenvalue 0. Hence, r of the eigenvectors $\boldsymbol{v}_0$, ..., $\boldsymbol{v}_r$ computed in Step 5 always have eigenvalue 0. If $\hat{\mathcal{N}}$ is not the null space of $\boldsymbol{X}$, one of them, say $\boldsymbol{v}_*$, has nonzero eigenvalue λ ($\neq 0$) and is orthogonal to the remaining eigenvectors, which span[5] $\mathcal{M} = \{\boldsymbol{u}_1, ..., \boldsymbol{u}_r\}_{\mathcal{L}}$. It follows that $\boldsymbol{v}_*$ is orthogonal to $\mathcal{M}$.

By construction, the vector $\hat{\boldsymbol{u}}$ in Eq. (26) belongs to $\hat{\mathcal{N}} = \{\boldsymbol{v}_*\}_{\mathcal{L}} \oplus \mathcal{M}$. Since the vector $\boldsymbol{u}'$ in Eq. (27) is a projection of $\hat{\boldsymbol{u}}$ within $\hat{\mathcal{N}}$ onto the direction orthogonal to $\mathcal{M}$, it coincides with $\pm\boldsymbol{v}_*$. The iterations converge when $\boldsymbol{u} = \boldsymbol{u}'$ ($= \pm\boldsymbol{v}_*$), and $\boldsymbol{v}_*$ is an eigenvector of $\boldsymbol{X}$ with eigenvalue λ. Hence, $\boldsymbol{u}$ also satisfies Eq. (25). Computing the inner product with $\boldsymbol{u}$ on both sides, we have

$$(\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) = \lambda. \tag{28}$$

On the other hand, $\boldsymbol{u}$ ($= \pm\boldsymbol{v}_*$) is orthogonal to all the eigenvectors of $\boldsymbol{X}$ with eigenvalue 0, so it is orthogonal to the subspace $\mathcal{M}$ they span. Hence, we have

$$\boldsymbol{P}_{\mathcal{V}}\boldsymbol{u} = \boldsymbol{u}. \tag{29}$$

Then,

$$\begin{aligned}(\boldsymbol{u}, \boldsymbol{X}\boldsymbol{u}) &= (\boldsymbol{u}, \boldsymbol{P}_{\mathcal{V}}(\boldsymbol{M} - \boldsymbol{L})\boldsymbol{P}_{\mathcal{V}}\boldsymbol{u}) \\ &= (\boldsymbol{u}, \boldsymbol{M}\boldsymbol{u}) - (\boldsymbol{u}, \boldsymbol{L}\boldsymbol{u}) = 0.\end{aligned} \tag{30}$$

The identity $(\boldsymbol{u}, \boldsymbol{M}\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{L}\boldsymbol{u})$ in $\boldsymbol{u}$ is easily confirmed by the definition of $\boldsymbol{M}$ and $\boldsymbol{L}$ in Eqs. (19).

[5] Eigenvectors with a multiple eigenvalue are indeterminate, so the computed eigenvectors of $\boldsymbol{X}$ with eigenvalue 0 need not coincide with any of $\boldsymbol{u}_1$, ..., $\boldsymbol{u}_r$. Uniquely determined is the subspace $\mathcal{M}$ that they span.

Eqs. (28) and (30) contradict our assumption that $\lambda \neq 0$. Hence, all elements of $\hat{\mathcal{N}}$ are eigenvectors of $\boldsymbol{X}$ with eigenvalue 0, i.e., $\hat{\mathcal{N}}$ is the null space of $\boldsymbol{X}$ when the iterations have converged. □

It follows that both $\boldsymbol{X}\boldsymbol{u} = \boldsymbol{0}$ and Eq. (29) hold, so $\boldsymbol{u}$ 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.

Reduction to FNS

If the number r of the intrinsic constraints is 0, we have $\mathcal{M} = \emptyset$ and $\boldsymbol{P}_{\mathcal{V}} = \boldsymbol{I}$. Hence, $\boldsymbol{u}$ is the eigenvector of $\boldsymbol{X}$ with the eigenvalue closest to 0. This is nothing but the FNS of Chojnacki et. [2]. In this sense, EFNS is a true extension of FNS to multiple intrinsic constraints. The FNS iterations converge simply by updating $\boldsymbol{u}$ by $\boldsymbol{u}'$ unless the initial value is very different from the solution [8]. So, we need not use the "midpoint" $N[\boldsymbol{u}'+\boldsymbol{u}]$, but of course we can.

On the other hand, the matrix $\boldsymbol{Q}$ used in CFNS has nothing to do with the matrix $\boldsymbol{X}$ used in FNS; for one thing, $\boldsymbol{Q}$ is positive semidefinite, while $\boldsymbol{X}$ is not. The basic difference of EFNS from CFNS is that while CFNS iteratively computes an eigenvector (1-D subspace), EFNS iteratively computes an $(r+1)$-D subspace.

5. Fundamental Matrix Computation

We now compare EFNS and CFNS by computing the fundamental matrix.

Epipolar equation

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* [5]

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

where f_0 is an arbitrary constant[6]. The matrix $\boldsymbol{F} = (F_{ij})$ is of rank 2 and called the *fundamental matrix*; it is determined by the relative positions and orientations of the two cameras and their intrinsic parameters independently of the scene.

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

If we define

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

Eq. (31) is written as $(\boldsymbol{u}, \boldsymbol{\xi}) = 0$. Letting $\boldsymbol{\xi}_\alpha$ represent the αth corresponding point pair $\{(x_\alpha, y_\alpha), (x'_\alpha, y'_\alpha)\}$ and $\bar{\boldsymbol{\xi}}_\alpha$ their true values $\{(\bar{x}_\alpha, \bar{y}_\alpha), (\bar{x}'_\alpha, \bar{y}'_\alpha)\}$, we have Eq. (1).

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

$$V_0[\boldsymbol{\xi}_\alpha] = \begin{pmatrix} \bar{x}_\alpha^2 + \bar{x}_\alpha'^2 & \bar{x}'_\alpha\bar{y}'_\alpha & f_0\bar{x}'_\alpha & \bar{x}_\alpha\bar{y}_\alpha \\ \bar{x}'_\alpha\bar{y}'_\alpha & \bar{x}_\alpha^2 + \bar{y}_\alpha'^2 & f_0\bar{y}'_\alpha & 0 \\ f_0\bar{x}'_\alpha & f_0\bar{y}'_\alpha & f_0^2 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & \bar{y}_\alpha^2 + \bar{x}_\alpha'^2 \\ 0 & \bar{x}_\alpha\bar{y}_\alpha & 0 & \bar{x}'_\alpha\bar{y}'_\alpha \\ 0 & 0 & 0 & f_0\bar{x}'_\alpha \\ f_0\bar{x}_\alpha & 0 & 0 & f_0\bar{y}_\alpha \\ 0 & f_0\bar{x}_\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\begin{pmatrix} 0 & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & 0 & 0 & f_0\bar{x}_\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bar{x}'_\alpha\bar{y}'_\alpha & f_0\bar{x}'_\alpha & f_0\bar{y}_\alpha & 0 & 0 \\ \bar{y}_\alpha^2 + \bar{y}_\alpha'^2 & f_0\bar{y}'_\alpha & 0 & f_0\bar{y}_\alpha & 0 \\ f_0\bar{y}'_\alpha & f_0^2 & 0 & 0 & 0 \\ 0 & 0 & f_0^2 & 0 & 0 \\ f_0\bar{y}_\alpha & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad (33)$$

Since the minimizer of Eq. (9) is not affected by multiplication of $V[\boldsymbol{\xi}_\alpha]$ by a positive constant, we can replace $V[\boldsymbol{\xi}_\alpha]$ by $V_0[\boldsymbol{\xi}_\alpha]$ in the minimization computation. The true positions $(\bar{x}_\alpha, \bar{y}_\alpha)$ and $(\bar{x}'_\alpha, \bar{y}'_\alpha)$ are replaced by their data values (x_α, y_α) and (x'_α, y'_α) in the computation[7].

The intrinsic constraint $\det \boldsymbol{F} = 0$ is written as $\phi(\boldsymbol{u}) = 0$, where $\phi(\boldsymbol{u})$ is the following homogeneous polynomial of degree 3:

$$\phi(\boldsymbol{u}) = u_1u_5u_9 + u_2u_6u_7 + u_3u_8u_4 - u_3u_5u_7 - u_2u_4u_9 - u_1u_8u_6. \qquad (34)$$

Its gradient is

$$\nabla_{\mathbf{u}}\phi = \begin{pmatrix} u_5u_9 - u_8u_6 \\ u_6u_7 - u_9u_4 \\ u_4u_8 - u_7u_5 \\ u_8u_3 - u_2u_9 \\ u_9u_1 - u_3u_7 \\ u_7u_2 - u_1u_8 \\ u_2u_6 - u_5u_3 \\ u_3u_4 - u_6u_1 \\ u_1u_5 - u_4u_2 \end{pmatrix}. \qquad (35)$$

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

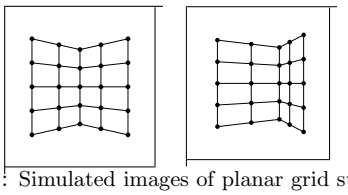

Figure 2: Simulated images of planar grid surfaces.

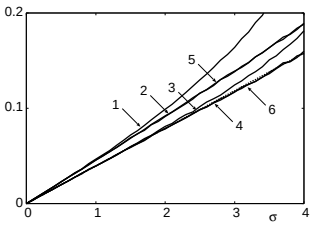

Figure 3: RMS error for Fig. 2 vs. noise level. 1) SVD-corrected LS. 2) SVD-corrected ML. 3) Optimally corrected ML. 4) Direct search. 5) CFNS. 6) EFNS. The dotted line indicates the KCR lower bound.

The projection matrix $\boldsymbol{P}_{\mathcal{V}}$ in Eq. (14) is now

$$\boldsymbol{P}_{\mathcal{V}} = \boldsymbol{I} - N[\nabla_{\mathbf{u}}\phi]N[\nabla_{\mathbf{u}}\phi]^{\top}. \tag{36}$$

Hence, the projection matrix $\boldsymbol{P}_{\mathcal{U}}$ in Eq. (24) is

$$\boldsymbol{P}_{\mathcal{U}} = \boldsymbol{I} - N[\nabla_{\mathbf{u}}\phi]N[\nabla_{\mathbf{u}}\phi]^{\top} - \boldsymbol{u}\boldsymbol{u}^{\top}. \tag{37}$$

Accuracy comparison

Fig. 2 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) optimally corrected ML, 4) direct search, 5) CFNS, and 6) EFNS.

The "SVD correction" means doing singular value decomposition (SVD) and replacing the smallest singular value by 0 [4]. "LS" means least-squares (also called "eight-point algorithm" [4]) minimizing $\sum_{\alpha=1}^{N}(\boldsymbol{u},\boldsymbol{\xi}_{\alpha})^2$: the solution is the unit eigenvector with the smallest eigenvalue of the matrix obtained by replacing $V[\boldsymbol{\xi}_{\alpha}]$ by $\boldsymbol{I}$ in the definition of $\boldsymbol{M}$ in Eqs. (19). For brevity, we use the shorthand "ML" for unconstrained ML, for which we used the FNS of Chojnacki et al. [2]. "Optimal correction" refers to the a posteriori correction approach used in [6, 10]. By "direct search", we mean the Levenberg-Marquard (LM) optimization using minimal parameterization [13]. All iterations are initialized by LS and stopped when the update of $\boldsymbol{F}$ is less than 10^{-6} in norm.

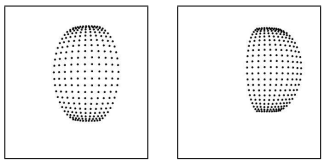

Figure 4: Simulated images of a spherical surface.

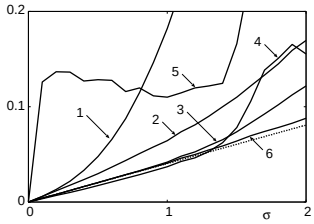

Figure 5: RMS error for Fig. 4 vs. noise level. 1) SVD-corrected LS. 2) SVD-corrected ML. 3) Optimally corrected ML. 4) Direct search. 5) CFNS. 6) EFNS. The dotted line indicates the KCR lower bound.

For each method, Figure 3 plots the root-mean-square (RMS) error

$$D = \sqrt{\frac{1}{10000}\sum_{a=1}^{10000}\|\boldsymbol{P}_{\mathcal{U}}\hat{\boldsymbol{u}}^{(a)}\|^2}, \tag{38}$$

corresponding to Eq. (6) over 10000 trials using independent noise for each σ, where $\hat{\boldsymbol{u}}^{(a)}$ is the ath value. The dotted line is the corresponding KCR lower bound (the trace of the right-hand side of Eq. (7)).

Figure 4 shows simulated images of a spherical surface (600×600 pixels with 1200 pixels focal length). Figure 5 shows the result corresponding to Fig. 3.

Observations

From Fig. 2, we see that the error behavior of EFNS is very similar to optimally corrected ML and direct search; all these nearly achieve the KCR lower bound. The SVD correction of LS or ML fails to produce comparable accuracy. Notable is the fact that CFNS performs as poorly as SVD-corrected ML.

In Fig. 5, CFNS performs even worse, while EFNS has higher accuracy than optimally corrected ML. Direct search performs very well for small noise but quickly deteriorates for large noise. This indicates growth of local minima as the noise increases.

Convergence of CFNS

Showing experimental evidences, Chojnacki et al. [3] asserted that CFNS is superior to optimally corrected ML. Figure 3 contradicts their assertion. Close examination has revealed that the solution of CFNS depends on the initial value of the iterations.

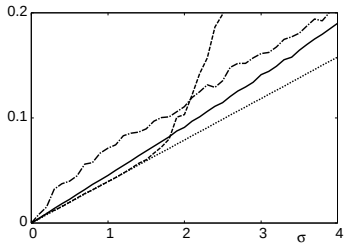

Figure 6: Dependence of CFNS on initializations: LS (solid line), SVD-corrected LS (dashed line), and the true value (chained line). The dotted line indicates the KCR lower bound.

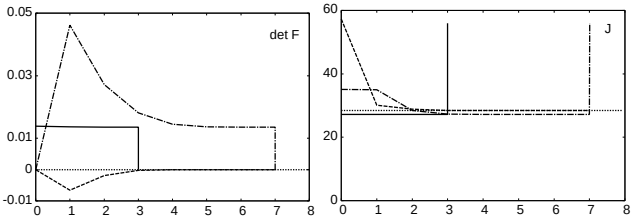

Figure 7: The convergence of $\det \boldsymbol{F}$ and the residual J ($\sigma = 1$) for different initializations: LS (solid line), SVD-corrected LS (dashed line), and the true value (chained line). All solutions are SVD-corrected in the final step.

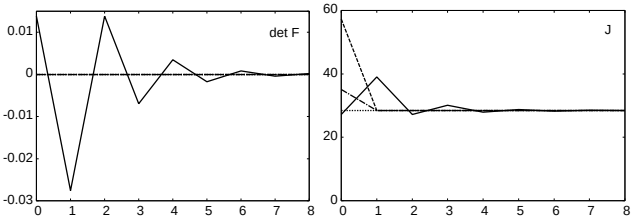

Figure 8: The results by EFNS corresponding to Fig. 7.

The results in Fig. 3 was obtained by LS initialization for all methods. Figure 6 compares the effects of different initializations on CFNS: we used LS, SVD-corrected LS, and the true value. We can see that the SVD correction of the initial value greatly increases the accuracy for small noise, but it quickly deteriorates for larger noise. If started from the true value, CFNS cannot obtain a correct value even in the presence of very small noise. In contrast, EFNS did not exhibit such initialization dependence.

Figure 7 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. [3]. The dotted lines show the values to be converged.

The LS solution has a very low residual J, because 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

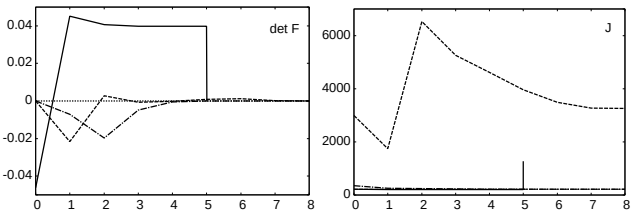

Figure 9: The convergence of $\det \boldsymbol{F}$ and the residual J ($\sigma = 3$) for different initializations: LS (solid line), SVD-corrected LS (dashed line), and the true value (chained line). All solutions are SVD-corrected in the final step.

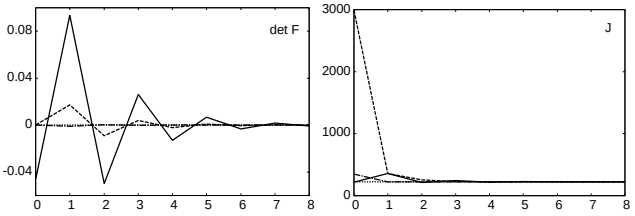

Figure 10: The results by EFNS corresponding to Fig. 9.

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 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 8 shows corresponding results by EFNS. Both J and $\det \boldsymbol{F}$ converge to their correct values with stably attenuating oscillations.

Figures 9 and 10 show the results corresponding to Fig. 7 and 8 for another instance ($\sigma = 3$). We can observe similar behavior of CFNS and EFNS.

CFNS vs. EFNS

From many experiments (not all shown here), we observe as follows. "Convergence" means 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 [3], Chojnacki et al. [3] proved that the solution $\boldsymbol{u}$ satisfying Eqs. (13) and (17) is a fixed point of their CFNS. Apparently, they expected to arrive at that solution by their scheme. However, as demonstrated by Fig. 7, CFNS has many other fixed points, and which to arrive at depends on initialization. In contrast, we proved in Sec. 3 that any fixed point of EFNS is *necessarily* the desired solution. This cannot be proved for CFNS.

Real image examples

We manually selected 100 pairs of corresponding

Figure 11: 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. 11.

method	residual	time
SVD-corrected LS	45.550	.00052
SVD-corrected ML	45.556	.00652
CFNS	45.378	.01300
Optimally corrected ML	45.378	.00764
direct search	45.378	.01136
EFNS	45.379	.01916

points in the two images in Fig. 11 and computed the fundamental matrix from them. Since its true value is unknown, we evaluated the final residual J together with the execution time (sec). We used Core2Duo E6700 2.66GHz for the CPU with 4GB main memory and Linux for the OS.

Evidently, SVD-corrected LS is poor, and again CFNS performs just as much as SVD-corrected ML, while optimally corrected ML, LM (from LS or optimally corrected ML), and EFNS arrive at the same optimal solution.

6. Conclusions

We presented a new method, called EFNS[8], for linearizable constrained ML. This complements the CFNS of Chojnacki et al. [3] and is a true extension of the FNS of Chojnacki et al. [2]. Computing the fundamental matrix as an illustration, we demonstrated that CFNS does not necessarily converge to a correct solution depending on initialization, while EFNS is insensitive to initialization, always arriving at an optimal value.

The advantage of EFNS is that the computation is done in the "external" $\boldsymbol{u}$-space, where expressions are relatively simple, while direct search is done in the "internal" space of minimal parameterization, where a lot of local minima exist due to the complication of expressions, as pointed out in [11]. As a result, EFNS performs better than minimal search in the presence of large noise as demonstrated in Fig. 4. The only disadvantage of EFNS is its slightly longer computation time, but this is a very small cost for its higher accuracy and robustness. More about this subject is discussed in [13].

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

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

References

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

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

[3] 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.

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

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

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

[7] K. Kanatani, "Performance evaluation of accurate ellipse fitting," Proc. 21th Int. Conf. Image Vision Computing New Zealand (IVCNZ 2006), pp.7–12, Great Barrier Island, New Zealand, Nov. 2006.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] T. Poston and L. Stewart, "Catastrophe Theory and Its Applications," Pitman, London, U.K., 1978; Reprinted Dover, New York, N.Y., U.S.A., 1996.

[13] Y. Sugaya and K. Kanatani, "High accuracy computation of rank-constrained fundamental matrix by efficient search," Proc. 10th Meeting Image Recog. Understand. (MIRU2007), Hiroshima, Japan, July 2007, this volume.