19th British Machine Vision Conference (BMVC 2008), 1–4 September 2008, Leeds, U.K., pp. 173–182.

Triangulation from Two Views Revisited: Hartley-Sturm vs. Optimal Correction

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

Abstract

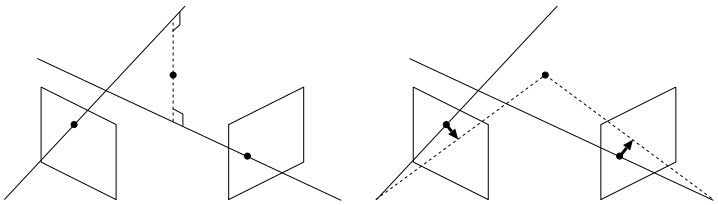
A higher order scheme is presented for the optimal correction method of Kanatani [5] for triangulation from two views and is compared with the method of Hartley and Sturm [3]. It is pointed out that the epipole is a singularity of the Hartley-Sturm method, while the proposed method has no singularity. Numerical simulation confirms that both compute identical solutions at other points. However, the proposed method is significantly faster.

1 Introduction

Stereo vision is a method for reconstructing the 3-D structure of a scene from corresponding points by triangulation: if the configuration of the cameras and their intrinsic parameters are known, one can compute the "line of sight" of each pixel, and the intersection of the corresponding lines of sight gives their 3-D position. If the camera configuration and the intrinsic parameters are not known, they can be estimated from point correspondences by computing the fundamental matrix; this procedure is known as *structure from motion* [4, 5].

In practice, correspondence detection entails uncertainty, so corresponding lines of sight may not meet in the scene. In the old days, this was handled by a practical compromise such as regarding the "midpoint" of the shortest line segment connecting the two lines of sight as the intersection (Fig. 1(a)). However, Kanatani [5] pointed out that an optimal method is to displace the corresponding pixels so that their lines of sight intersect in such a way that the amount of the displacement is minimum (Fig. 1(b)). Kanatani [5] called this strategy *optimal correction*, and Kanazawa and Kanatani [6] used this principle for reliability evaluation of stereo reconstruction.

Hartley and Sturm [3] independently introduced a similar idea and presented a numerical scheme of reducing the problem to solving a 6-degree polynomial. Some researchers compared the two methods and found that Kanatani's method was far more practical. For example, Torr and Zissermann [9] wrote "Hartley and Sturm provide ... This turns out to be equivalent to the optimally corrected correspondence of Kanatani. Comparisons of the computation of $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ by Hartley and Sturm's method and that of Kanatani have again



(a) (b)

Figure 1: Triangulation. (a) The mid-point method. (b) Optimal correction.

shown agreement to three or four significant figures, as mentioned, but Kanatani's method is several orders of magnitude faster, being a linear method and hence preferable for rapid evaluation purposes". Yet, the Hartley-Sturm method has won popularity and currently is widely regarded as a standard tool for triangulation.

In this paper, we modify Kanatani's method and compare it with the Hartley-Sturm method. First, we state the problem in mathematical terms (Sec. 2) and describe Kanatani's formula (Sec. 3). Then, we extend it to higher orders so that an exact optimal solution is computed (Sec. 4). Next, we recast it into a very compact form suitable for numerical computation (Sec. 5). We point out that the Hartley-Sturm method has a singularity at epipoles, while our method does not, and discuss the convergence issue (Sec. 6). From simulation, we observe that the Hartley-Strum method and our method compute identical solutions other than at epipoles, while our method is significantly faster (Sec. 7). We conclude that our method best suits practical use (Sec. 8).

2 Mathematical Background

Suppose point (x,y) in the first image corresponds to point (x',y') in the second. We represent them in 3-D vectors in the form

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

where f_0 is a scale constant of approximately the image size[1]. We refer to the point represented by vector $\mathbf{x}$ simply as "point $\mathbf{x}$". As is well known [4, 5], the necessary and sufficient condition for the lines of sight of points $\mathbf{x}$ and $\mathbf{x}'$ meet in the scene is the *epipolar equation*

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

where $\mathbf{F}$ is the fundamental matrix [4]. Throughout this paper, we denote the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ by $(\mathbf{a},\mathbf{b})$. We assume that the fundamental matrix $\mathbf{F}$ is already given.

The detected corresponding points $\mathbf{x}$ and $\mathbf{x}'$ may not exactly satisfy Eq. (2) due to uncertainty of image processing operations. Kanatani's principle of optimal correction [5] is to *minimally* correct points $\mathbf{x}$ and $\mathbf{x}'$ to points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ so that Eq. (2) is satisfied, where by "minimally" we mean that the sum of square distances, or the *reprojection error*

[1] This is for numerical stability [1]. In our experiment, we set $f_0 = 600$ pixels.

[4],

$$E = \|\mathbf{x} - \bar{\mathbf{x}}\|^2 + \|\mathbf{x}' - \bar{\mathbf{x}}'\|^2, \tag{3}$$

is kept minimum. In mathematical terms, we minimize Eq. (3) with respect to $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ subject to

$$(\bar{\mathbf{x}}, \mathbf{F}\bar{\mathbf{x}}') = 0. \tag{4}$$

In statistical terms, this is interpreted as follows. Suppose the points $\mathbf{x}$ and $\mathbf{x}'$ are observed after displaced from their true positions $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ by noise in the form

$$\mathbf{x} = \bar{\mathbf{x}} + \Delta\mathbf{x}, \qquad \mathbf{x}' = \bar{\mathbf{x}}' + \Delta\mathbf{x}'. \tag{5}$$

If the noise terms $\Delta\mathbf{x}$ and $\Delta\mathbf{x}'$ are independent and isotropic Gaussian random variables of mean $\mathbf{0}$ and a constant covariance, minimization of Eq. (3) is equivalent to maximum likelihood (ML) estimation of the true positions $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$.

3 Optimal Correction Procedure

Kanatani's formula for optimal correction [5, 6] is derived as follows. Instead of directly estimating $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$, we write

$$\bar{\mathbf{x}} = \mathbf{x} - \Delta\mathbf{x}, \qquad \bar{\mathbf{x}}' = \mathbf{x}' - \Delta\mathbf{x}', \tag{6}$$

and estimate the correction terms $\Delta\mathbf{x}$ and $\Delta\mathbf{x}'$. On substitution of Eqs. (6), Eq. (3) becomes

$$E = \|\Delta\mathbf{x}\|^2 + \|\Delta\mathbf{x}'\|^2, \tag{7}$$

and Eq. (4) is written as

$$(\mathbf{x} - \Delta\mathbf{x}, \mathbf{F}(\mathbf{x}' - \Delta\mathbf{x}')) = 0. \tag{8}$$

Expanding this and ignoring second order terms in $\Delta\mathbf{x}$ and $\Delta\mathbf{x}'$, we obtain

$$(\mathbf{F}\mathbf{x}', \Delta\mathbf{x}) + (\mathbf{F}^\top\mathbf{x}, \Delta\mathbf{x}') = (\mathbf{x}, \mathbf{F}\mathbf{x}'). \tag{9}$$

Since the correction is done on the image plane, the third components of $\Delta\mathbf{x}$ and $\Delta\mathbf{x}'$ should be zero, so we have

$$(\mathbf{k}, \Delta\mathbf{x}) = 0, \qquad (\mathbf{k}, \Delta\mathbf{x}') = 0, \tag{10}$$

where and throughout this paper, we define $\mathbf{k} \equiv (0,0,1)^\top$. Introducing Lagrange multipliers to Eqs. (9) and (10) in the form

$$\|\Delta\mathbf{x}\|^2 + \|\Delta\mathbf{x}'\|^2 - \lambda\Big((\mathbf{F}\mathbf{x}', \Delta\mathbf{x}) + (\mathbf{F}^\top\mathbf{x}, \Delta\mathbf{x}')\Big) - \mu(\mathbf{k}, \Delta\mathbf{x}) - \mu'(\mathbf{k}, \Delta\mathbf{x}'), \tag{11}$$

and letting the derivatives with respect to $\Delta\mathbf{x}$ and $\Delta\mathbf{x}'$ be zero, we obtain

$$2\Delta\mathbf{x} - \lambda\mathbf{F}\mathbf{x}' - \mu\mathbf{k} = \mathbf{0}, \qquad 2\Delta\mathbf{x}' - \lambda\mathbf{F}^\top\mathbf{x} - \mu'\mathbf{k} = \mathbf{0}. \tag{12}$$

Multiplying these by the projection matrix

$$\mathbf{P}_\mathbf{k} \equiv \mathrm{diag}(1,1,0), \tag{13}$$

which makes the third component 0, and noting that $\mathbf{P_k}\Delta\mathbf{x} = \Delta\mathbf{x}$, $\mathbf{P_k}\Delta\mathbf{x}' = \Delta\mathbf{x}'$, and $\mathbf{P_k}\mathbf{k} = \mathbf{0}$, we obtain

$$2\Delta\mathbf{x} - \lambda\mathbf{P_k}\mathbf{F}\mathbf{x}' = \mathbf{0}, \qquad 2\Delta\mathbf{x}' - \lambda\mathbf{P_k}\mathbf{F}^\top\mathbf{x} = \mathbf{0}. \tag{14}$$

Hence,

$$\Delta\mathbf{x} = \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}\mathbf{x}', \qquad \Delta\mathbf{x}' = \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}^\top\mathbf{x}. \tag{15}$$

Substituting these into Eq. (9), we obtain

$$(\mathbf{F}\mathbf{x}', \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}\mathbf{x}') + (\mathbf{F}^\top\mathbf{x}, \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}^\top\mathbf{x}) = (\mathbf{x}, \mathbf{F}\mathbf{x}'), \tag{16}$$

from which λ is determined in the form

$$\frac{\lambda}{2} = \frac{(\mathbf{x}, \mathbf{F}\mathbf{x}')}{(\mathbf{F}\mathbf{x}', \mathbf{P_k}\mathbf{F}\mathbf{x}') + (\mathbf{F}^\top\mathbf{x}, \mathbf{P_k}\mathbf{F}^\top\mathbf{x})}. \tag{17}$$

Hence, Eqs. (15) becomes

$$\Delta\mathbf{x} = \frac{(\mathbf{x}, \mathbf{F}\mathbf{x}')\mathbf{P_k}\mathbf{F}\mathbf{x}'}{(\mathbf{F}\mathbf{x}', \mathbf{P_k}\mathbf{F}\mathbf{x}') + (\mathbf{F}^\top\mathbf{x}, \mathbf{P_k}\mathbf{F}^\top\mathbf{x})}, \qquad \Delta\mathbf{x}' = \frac{(\mathbf{x}, \mathbf{F}\mathbf{x}')\mathbf{P_k}\mathbf{F}^\top\mathbf{x}}{(\mathbf{F}\mathbf{x}', \mathbf{P_k}\mathbf{F}\mathbf{x}') + (\mathbf{F}^\top\mathbf{x}, \mathbf{P_k}\mathbf{F}^\top\mathbf{x})}. \tag{18}$$

Thus, the true positions $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ are estimated from Eqs. (6) in the form

$$\hat{\mathbf{x}} = \mathbf{x} - \frac{(\mathbf{x}, \mathbf{F}\mathbf{x}')\mathbf{P_k}\mathbf{F}\mathbf{x}'}{(\mathbf{F}\mathbf{x}', \mathbf{P_k}\mathbf{F}\mathbf{x}') + (\mathbf{F}^\top\mathbf{x}, \mathbf{P_k}\mathbf{F}^\top\mathbf{x})}, \qquad \hat{\mathbf{x}}' = \mathbf{x}' - \frac{(\mathbf{x}, \mathbf{F}\mathbf{x}')\mathbf{P_k}\mathbf{F}^\top\mathbf{x}}{(\mathbf{F}\mathbf{x}', \mathbf{P_k}\mathbf{F}\mathbf{x}') + (\mathbf{F}^\top\mathbf{x}, \mathbf{P_k}\mathbf{F}^\top\mathbf{x})}. \tag{19}$$

These are Kanatani's formula given in [5, 6, 9].

4 Higher Order Correction

Since Eq. (9) is a first approximation, the points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ computed by Eqs. (19) may not strictly satisfy $(\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') = 0$. We now present an iterative scheme for computing $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ that exactly minimize Eq. (3). Instead of directly estimating $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$, we write

$$\bar{\mathbf{x}} = \hat{\mathbf{x}} - \Delta\hat{\mathbf{x}}, \qquad \bar{\mathbf{x}}' = \hat{\mathbf{x}}' - \Delta\hat{\mathbf{x}}', \tag{20}$$

and compute the higher order correction terms $\Delta\hat{\mathbf{x}}$ and $\Delta\hat{\mathbf{x}}'$, which are small quantities of a higher order. Substituting Eqs. (20) into Eq. (3), we have

$$E = \|\tilde{\mathbf{x}} + \Delta\hat{\mathbf{x}}\|^2 + \|\tilde{\mathbf{x}}' + \Delta\hat{\mathbf{x}}'\|^2, \tag{21}$$

where we define

$$\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}, \qquad \tilde{\mathbf{x}}' = \mathbf{x}' - \hat{\mathbf{x}}'. \tag{22}$$

Equation (4) now becomes

$$(\hat{\mathbf{x}} - \Delta\hat{\mathbf{x}}, \mathbf{F}(\hat{\mathbf{x}}' - \Delta\hat{\mathbf{x}}')) = 0. \tag{23}$$

Expanding this and ignoring second order terms in $\Delta\hat{\mathbf{x}}$ and $\Delta\hat{\mathbf{x}}'$, we obtain

$$(\mathbf{F}\hat{\mathbf{x}}', \Delta\hat{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \Delta\hat{\mathbf{x}}') = (\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}'). \tag{24}$$

Since $\Delta\hat{\mathbf{x}}$ and $\Delta\hat{\mathbf{x}}'$ are small quantities of a higher order, Eq. (24) is a higher order approximation of Eq. (4). The correction is done on the image plane, so we have the constraint

$$(\mathbf{k}, \Delta\hat{\mathbf{x}}) = 0, \qquad (\mathbf{k}, \Delta\hat{\mathbf{x}}') = 0. \tag{25}$$

Introducing Lagrange multipliers to Eqs. (24) and (25) in the form

$$\|\tilde{\mathbf{x}} + \Delta\hat{\mathbf{x}}\|^2 + \|\tilde{\mathbf{x}}' + \Delta\hat{\mathbf{x}}'\|^2 - \lambda\Big((\mathbf{F}\hat{\mathbf{x}}', \Delta\hat{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \Delta\hat{\mathbf{x}}')\Big) - \mu(\mathbf{k}, \Delta\hat{\mathbf{x}}) - \mu'(\mathbf{k}, \Delta\hat{\mathbf{x}}'), \tag{26}$$

and letting the derivatives with respect to $\Delta\hat{\mathbf{x}}$ and $\Delta\hat{\mathbf{x}}'$ be zero, we obtain

$$2(\tilde{\mathbf{x}} + \Delta\hat{\mathbf{x}}) - \lambda\mathbf{F}\hat{\mathbf{x}}' - \mu\mathbf{k} = \mathbf{0}, \qquad 2(\tilde{\mathbf{x}}' + \Delta\hat{\mathbf{x}}') - \lambda\mathbf{F}^\top\hat{\mathbf{x}} - \mu'\mathbf{k} = \mathbf{0}. \tag{27}$$

Multiplying these by the projection matrix $\mathbf{P_k}$ in Eq. (13) and noting that $\mathbf{P_k}\tilde{\mathbf{x}} = \tilde{\mathbf{x}}$, $\mathbf{P_k}\tilde{\mathbf{x}}' = \tilde{\mathbf{x}}'$ from the definition of $\tilde{\mathbf{x}}'$ in Eq. (22), we obtain

$$2\tilde{\mathbf{x}} + 2\Delta\hat{\mathbf{x}} - \lambda\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}' = \mathbf{0}, \quad 2\tilde{\mathbf{x}} + 2\Delta\hat{\mathbf{x}}' - \lambda\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}} = \mathbf{0}. \tag{28}$$

Hence, we have

$$\Delta\hat{\mathbf{x}} = \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}' - \tilde{\mathbf{x}}, \qquad \Delta\hat{\mathbf{x}}' = \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}} - \tilde{\mathbf{x}}'. \tag{29}$$

Substituting these into Eq. (24), we obtain

$$(\mathbf{F}\hat{\mathbf{x}}', \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}' - \tilde{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \frac{\lambda}{2}\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}} - \tilde{\mathbf{x}}') = (\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}'), \tag{30}$$

from which λ is determined in the form

$$\frac{\lambda}{2} = \frac{(\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}\hat{\mathbf{x}}', \tilde{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \tilde{\mathbf{x}}')}{(\mathbf{F}\hat{\mathbf{x}}', \mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}^\top\hat{\mathbf{x}}, \mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})}. \tag{31}$$

Hence, Eq. (29) becomes

$$\begin{aligned}
\Delta\hat{\mathbf{x}} &= \frac{\Big((\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}\hat{\mathbf{x}}', \tilde{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \tilde{\mathbf{x}}')\Big)\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}'}{(\mathbf{F}\hat{\mathbf{x}}', \mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}^\top\hat{\mathbf{x}}, \mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})} - \tilde{\mathbf{x}}, \\
\Delta\hat{\mathbf{x}}' &= \frac{\Big((\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}\hat{\mathbf{x}}', \tilde{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \tilde{\mathbf{x}}')\Big)\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}}}{(\mathbf{F}\hat{\mathbf{x}}', \mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}^\top\hat{\mathbf{x}}, \mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})} - \tilde{\mathbf{x}}'.
\end{aligned} \tag{32}$$

Thus, the true positions $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ are estimated from Eqs. (22) and (20) in the form

$$\begin{aligned}
\hat{\hat{\mathbf{x}}} &= \mathbf{x} - \frac{\Big((\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}\hat{\mathbf{x}}', \tilde{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \tilde{\mathbf{x}}')\Big)\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}'}{(\mathbf{F}\hat{\mathbf{x}}', \mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}^\top\hat{\mathbf{x}}, \mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})}, \\
\hat{\hat{\mathbf{x}}}' &= \mathbf{x}' - \frac{\Big((\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}\hat{\mathbf{x}}', \tilde{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \tilde{\mathbf{x}}')\Big)\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}}}{(\mathbf{F}\hat{\mathbf{x}}', \mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}^\top\hat{\mathbf{x}}, \mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}})}.
\end{aligned} \tag{33}$$

This is a higher order approximation. Still, they may not strictly satisfy $(\hat{\mathbf{x}}, \mathbf{F}\hat{\hat{\mathbf{x}}}') = 0$, so we let $\hat{\mathbf{x}} \leftarrow \hat{\hat{\mathbf{x}}}$ and $\hat{\mathbf{x}}' \leftarrow \hat{\hat{\mathbf{x}}}'$ and repeat the same computation until the iterations converge. In the end, $\Delta\hat{\mathbf{x}}_\alpha$ and $\Delta\hat{\mathbf{x}}'_\alpha$ in Eq. (23) become $\mathbf{0}$.

5 Compact Numerical Scheme

We now reduce our algorithm to a compact form suitable for numerical computation. First, we encode the fundamental matrix $\mathbf{F} = (F_{ij})$ and the corresponding point pair $\{(x,y), (x',y')\}$ in the following 9-D vectors:

$$\begin{aligned} \mathbf{u} &= (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top, \\ \xi &= (xx', xy', f_0 x, yx', yy', f_0 y, f_0 x', f_0 y', f_0^2)^\top. \end{aligned} \tag{34}$$

The epipolar equation in Eq. (2) now becomes $(\mathbf{u}, \xi) = 0$. We can also rewrite part of the numerators and denominators in Eqs. (19) as follows:

$$(\mathbf{x}, \mathbf{F}\mathbf{x}') = \frac{1}{f_0^2}(\mathbf{u}, \xi), \qquad (\mathbf{F}\mathbf{x}', \mathbf{P_k}\mathbf{F}\mathbf{x}') + (\mathbf{F}^\top\mathbf{x}, \mathbf{P_k}\mathbf{F}^\top\mathbf{x}) = \frac{1}{f_0^2}(\mathbf{u}, V_0[\xi]\mathbf{u}). \tag{35}$$

Here, we define the matrix $V_0[\xi]$ as follows:

$$V_0[\xi] = \begin{pmatrix} x^2+x'^2 & x'y' & f_0x' & xy & 0 & 0 & f_0x & 0 & 0 \\ x'y' & x^2+y'^2 & f_0y' & 0 & xy & 0 & 0 & f_0x & 0 \\ f_0x' & f_0y' & f_0^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ xy & 0 & 0 & y^2+x'^2 & x'y' & f_0x' & f_0y & 0 & 0 \\ 0 & xy & 0 & x'y' & y^2+y'^2 & f_0y' & 0 & f_0y & 0 \\ 0 & 0 & 0 & f_0x' & f_0y' & f_0^2 & 0 & 0 & 0 \\ f_0x & 0 & 0 & f_0y & 0 & 0 & f_0^2 & 0 & 0 \\ 0 & f_0x & 0 & 0 & f_0y & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{36}$$

From Eqs. (35), Eqs. (19) can be written as

$$\hat{\mathbf{x}} = \mathbf{x} - \frac{(\mathbf{u}, \xi)\mathbf{P_k}\mathbf{F}\mathbf{x}'}{(\mathbf{u}, V_0[\xi]\mathbf{u})}, \qquad \hat{\mathbf{x}}' = \mathbf{x}' - \frac{(\mathbf{u}, \xi)\mathbf{P_k}\mathbf{F}^\top\mathbf{x}}{(\mathbf{u}, V_0[\xi]\mathbf{u})}. \tag{37}$$

Now, if we define the vector

$$\hat{\xi} = \begin{pmatrix} \hat{x}\hat{x}' + \hat{x}'\tilde{x} + \hat{x}\tilde{x}' \\ \hat{x}\hat{y}' + \hat{y}'\tilde{x} + \hat{x}\tilde{y}' \\ f_0(\hat{x} + \tilde{x}) \\ \hat{y}\hat{x}' + \hat{x}'\tilde{y} + \hat{y}\tilde{x}' \\ \hat{y}\hat{y}' + \hat{y}'\tilde{y} + \hat{y}\tilde{y}' \\ f_0(\hat{y} + \tilde{y}) \\ f_0(\hat{x}' + \tilde{x}') \\ f_0(\hat{y}' + \tilde{y}') \\ f_0^2 \end{pmatrix}, \tag{38}$$

part of the numerators and denominators in Eqs. (33) can be rewritten as

$$(\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}\hat{\mathbf{x}}', \tilde{\mathbf{x}}) + (\mathbf{F}^\top\hat{\mathbf{x}}, \tilde{\mathbf{x}}') = \frac{1}{f_0^2}(\mathbf{u}, \hat{\xi}), \tag{39}$$

$$(\mathbf{F}\hat{\mathbf{x}}', \mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}') + (\mathbf{F}^\top\hat{\mathbf{x}}, \mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}}) = \frac{1}{f_0^2}(\mathbf{u}, V_0[\hat{\xi}]\mathbf{u}), \tag{40}$$

where $V_0[\hat{\boldsymbol{\xi}}]$ is the matrix obtained by replacing x, y, x', and y' by $\hat{x}$, $\hat{y}$, $\hat{x}'$, and $\hat{y}'$, respectively, in Eq. (36). From Eqs. (39) and (40), we can write Eqs. (33) as

$$\hat{\hat{\mathbf{x}}} = \mathbf{x} - \frac{(\mathbf{u}, \hat{\boldsymbol{\xi}})\mathbf{P_k}\mathbf{F}\hat{\mathbf{x}}'}{(\mathbf{u}, V_0[\hat{\boldsymbol{\xi}}]\mathbf{u})}, \qquad \hat{\hat{\mathbf{x}}}' = \mathbf{x}' - \frac{(\mathbf{u}, \hat{\boldsymbol{\xi}})\mathbf{P_k}\mathbf{F}^\top\hat{\mathbf{x}}}{(\mathbf{u}, V_0[\hat{\boldsymbol{\xi}}]\mathbf{u})}. \tag{41}$$

Thus, we can obtain the following compact procedure:

1. Let $E_0 = \infty$ (a sufficiently large number), $\hat{x} = x$, $\hat{y} = y$, $\hat{x}' = x'$, $\hat{y}' = y'$, and $\tilde{x} = \tilde{y} = \tilde{x}' = \tilde{y}' = 0$.
2. Compute the 9-D vector $\hat{\boldsymbol{\xi}}$ in Eq. (38) and the corresponding 9×9 matrix $V_0[\hat{\boldsymbol{\xi}}]$.
3. Update $\tilde{x}$, $\tilde{y}$, $\tilde{x}'$, and $\tilde{y}'$ as follows:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \leftarrow \frac{(\mathbf{u}, \hat{\boldsymbol{\xi}})}{(\mathbf{u}, V_0[\hat{\boldsymbol{\xi}}]\mathbf{u})} \begin{pmatrix} u_1 & u_2 & u_3 \\ u_4 & u_5 & u_6 \end{pmatrix} \begin{pmatrix} \hat{x}' \\ \hat{y}' \\ f_0 \end{pmatrix}, \quad \begin{pmatrix} \tilde{x}' \\ \tilde{y}' \end{pmatrix} \leftarrow \frac{(\mathbf{u}, \hat{\boldsymbol{\xi}})}{(\mathbf{u}, V_0[\hat{\boldsymbol{\xi}}]\mathbf{u})} \begin{pmatrix} u_1 & u_4 & u_7 \\ u_2 & u_5 & u_8 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \\ f_0 \end{pmatrix} \tag{42}$$

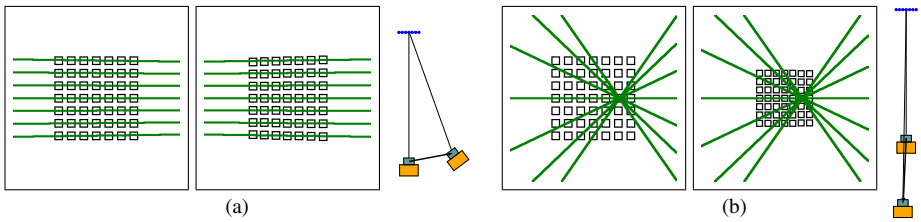
4. Compute the reprojection error $E = (\tilde{x}^2 + \tilde{y}^2 + \tilde{x}'^2 + \tilde{y}'^2)/f_0^2$.
5. If $E \approx E_0$, return $(\hat{x}, \hat{y})$ and $(\hat{x}', \hat{y}')$ and stop. Else, let $\hat{x} \leftarrow x - \tilde{x}$, $\hat{y} \leftarrow y - \tilde{y}$, $\hat{x}' \leftarrow x' - \tilde{x}'$, and $\hat{y}' \leftarrow y' - \tilde{y}'$, and go back to Step 2.

6 Theoretical Issues

Effect of epipoles. Point $\mathbf{x}$ in the first image is called its *epipole* if $\mathbf{F}^\top\mathbf{x} = \mathbf{0}$; point $\mathbf{x}'$ in the second is called its epipole if $\mathbf{F}\mathbf{x}' = \mathbf{0}$. The Hartley-Sturm method [3] first translates the images so that the corresponding points are at the coordinate origins and next rotates the images so that their epipoles are on the x-axes. Then, the observed points at the origins are orthogonally projected onto the parameterized epipolar lines passing through the respective epipoles. The parameter of the epipolar lines is determined so that the sum of the square distances from the origins is minimized, which reduces to solving a 6-degree polynomial.

If *either* of the corresponding points is at the epipole, no epipolar line is defined, so an ad hoc procedure is necessary. In our procedure, $(\mathbf{u}, V_0[\hat{\boldsymbol{\xi}}]\mathbf{u})$ is the only quantity that appears in denominators. From Eqs. (35), (39), and (40), we see that it becomes zero only when $\mathbf{x}$ and $\mathbf{x}'$ (or $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ in the course of iterations) are *both* at the epipoles, in which case the numerators in Eqs. (42) are also zero, so the computation ends without any correction. If *one* of the corresponding points is at the epipole, we see from Eqs. (42) that our method displaces it *away from the epipole*, while the other point (not at the epipole) is unchanged. In contrast, the ad hoc rule of the Hartley-Sturm method [3] moves the point not at the epipole *to the epipole*. Thus, our method is theoretically more consistent, although this difference has little effect in practical situations.

Convergence. Our method consists of iterative updates, so one may wonder if the correct solution is reached. In fact, Hartley and Sturm [3] emphasized this as the *raison d'être* of their method. They parameterized the pair of corresponding epipolar lines and argued, showing numerical examples, that Newton-type search could be trapped into local minima



(a) (b)

Figure 2: Stereo image pairs of a planar grid (the epipolar lines are overlaid), and the camera configuration. (a) Stable camera configuration. (b) Unstable camera configuration.

if arbitrarily started in the parameter space. This does not apply to our method, since our problem does *not* have any local minima (or even a global minimum).

The Hartley-Sturm method searches the space in which *the epipolar equation is satisfied* for minimizing the reprojection error. We search the joint image plane for the positions $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ that satisfy $(\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') = 0$ *in the end*: intermediate positions do *not*. If $(\hat{\mathbf{x}}, \mathbf{F}\hat{\mathbf{x}}') = 0$ is first satisfied, we are done.

Evidently, whatever local search should start near the true solution to converge to it, since we may be trapped into a false solution if it is located closer to us than the true solution. In our problem, however, we are searching for the positions $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ that are the *closest* to the data $\mathbf{x}$ and $\mathbf{x}'$, *starting from them*. If there happens to be a "false" solution that satisfies the epipolar equation and is closer to us than the true solution, it should be the true solution by the very definition of the reprojection error[2].

Our method can be imagined to move points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ that are connected to $\mathbf{x}$ and $\mathbf{x}'$ with "springs" whose energy is proportional to the square distance of extension. The points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are gradually "pulled" toward the hypersurface $(\mathbf{x}, \mathbf{F}\mathbf{x}') = 0$ in the joint space until they reach it. This is similar to the well-known global optimization technique of gradually raising the reprojection error threshold from 0 and testing if a feasible solution exists [2, 7]; the one first found is the globally optimal solution.

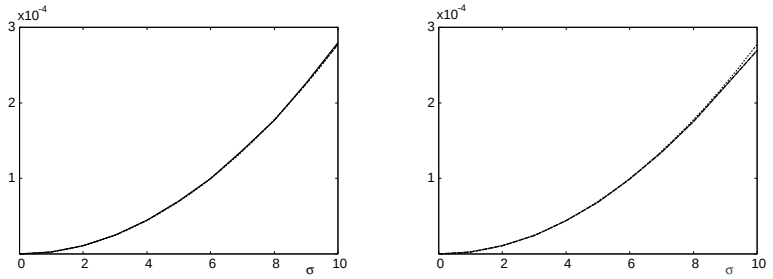
7 Experiments

Setting Figure 2(a), (b) shows simulated images (supposedly 400×400 pixels) of a grid pattern viewed by two cameras with focal length 1200 pixels. In Fig. 2(a), the baseline is nearly perpendicular to the camera optical axes (call this the "stable camera configuration"); in Fig. 2(b), it is nearly parallel to them (call this the "unstable camera configuration"). Some of the corresponding epipolar lines are overlaid; the epipoles are located at their intersections.

Adding independent Gaussian noise of mean 0 and standard deviation σ pixels to the x- and y-coordinates of the grid points, we computed their 3-D positions. We varied σ from 0 to an extremely large value 10 to see if our method still works. The iterations of our method are terminated when the update of the reprojection error E is less than 10^{-6}.

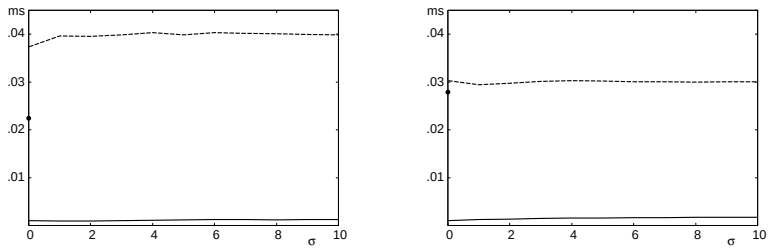
Reprojection error. Figure 3 plots, for each σ, the reprojection error E averaged for all the grid points over 1000 independent trials. The solid line shows the result of our optimal correction; the dashed line the Hartley-Sturm method; the dotted line the theoretical

[2]There may exist a pathological situation in which this argument does not hold in the presence of extremely large noise, but we doubt if there is any.



(a) (b)

Figure 3: Mean reprojection error. Solid line: optimal correction. Dashed line: Hartley-Sturm. Dotted line: theoretical expectation. (a) Stable camera configuration. (b) Unstable camera configuration.



(a) (b)

Figure 4: Computation time (in ms). Solid line: optimal correction. Dashed line: Hartley-Sturm. (a) Stable camera configuration. (b) Unstable camera configuration.

expectation[3] $(\sigma/f_0)^2$. The plots of the two methods completely coincide in whichever camera configuration. Numerically, they agree up to significant digits, confirming that the two algorithms are mathematically equivalent. Also, the reprojection error nearly coincides with the theoretical expectation, indicating that the true ML solution is computed indeed.

As Fig. 3 shows, the reprojection error is basically the same in the stable and unstable camera configurations. This is because optimal correction depends only on the statistical properties of noise: if the noise distribution is the same, say Gaussian, minimization of the reprojection error does *not* depend on the camera configuration, the image content, or the 3-D structure of the scene. What the camera configuration affects is the *reliability of 3-D reconstruction*, irrespective of the method for triangulation, for which our method and the Hartley-Sturm methods produce the same result.

Computation time. Figure 4 shows the average computation time over 1000 trials. We implemented the two method using the C language and used the eigenvalue method [8] for solving a 6-degree polynomial. We used Intel Core2Duo E6700 2.66GHz for the CPU with main memory 4GB and Linux for the OS.

As we see from Fig. 4, the computation time of the Hartley-Sturm method almost does not depend on the noise level σ. As we confirmed, most of the time is spent on solving a 6-degree polynomial. At $\sigma = 0$, the computation is slightly faster (see the black dots

[3]For an ML solution, $f_0^2 E/\sigma^2$ is to a first approximation subject to a χ^2 distribution with one degree of freedom, hence has expectation 1 [5].

on the vertical axes). For exact data, the 6-degree polynomial degenerates into degree 5, which is probably easier to solve. We also observe that the computation is faster for the unstable camera configuration, in which case the 6-degree polynomial is probably easier to solve, but we did not analyzed the details.

Our method is, on the other hand, iterative. From Fig. 4, we observe that the number of iterations slightly increases as noise increases and also when the camera configuration is unstable. However, the increase is very small, converging after at most three to four iterations. We have also found that because our method consists of simple vector and matrix operations only, the computation time depends to a large extent on the vector-matrix calculus library and the complier that we use, while the Hartley-Sturm method takes almost the same time in whatever implementation. Overall, however, as demonstrated by Fig. 4, our method is significantly faster than the Hartley-Sturm method.

8 Conclusions

We have extended the first order optimal correction of Kanatani [5] for triangulation from two views to higher orders and shown that it can be written in a very compact form suitable for numerical computation. We compared it with the Hartley-Sturm method [3], widely regarded as a standard tool for triangulation.

We have pointed out that the epipole is a singularity of the Hartley-Sturm method, at which the computation breaks down, while our method does not. We have also argued that we need not worry about local minima for our iterations. By simulation, we have demonstrated that our method and the Hartley-Sturm method compute identical solutions, yet our method is significantly faster, confirming the observation of Torr and Zissermann [9]. We conclude that our method best suits practical use.

References

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

[2] R. Hartley and F. Kahl, Optimal algorithms in multiview geometry, *Proc. 8th Asian Conf. Comput. Vision (ACCV2007)*, November 2007, Tokyo Japan, Vol. 1, pp. 13–34.

[3] R. I. Hartley and P. Sturm, Triangulation, *Comput. Vision Image Understand.*, **68**-2 (1997-11), 146–157.

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

[5] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice* Elsevier, Amsterdam, the Netherlands, 1996; reprinted, Dover, York, NY, U.S.A., 2005.

[6] Y. Kanazawa and K. Kanatani, Reliability of 3-D reconstruction by stereo vision, *IEICE Trans. Inf. & Syst.*, **E78-D**-10 (1995-10), 1301–1306.

[7] Q. Ke and T. Kanade, Quasiconvex optimization for robust geometric reconstruction, *IEEE Trans. Patt. Anal. Mach. Intell.*, **29**-10 (2007-10), 1834–1847.

[8] W. H. Press, S. A. Teukolsky, W. T. Vettering, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition, Cambridge University Press, Cambridge, U.K., 1992.

[9] P. H. S. Torr and A. Zissermann, Performance characterization of fundamental matrix estimation under image degradation, *Mach. Vis. Appl.*, **9**-5/6 (1997-3), 321–333.