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

Kenichi Kanatani†, Yasuyuki Sugaya‡, and Hirotaka Niitsuma†
†Okayama University, ‡Toyohashi University of Technology
E-mail: kanatani@suri.cs.okayama-u.ac.jp

Abstract. A higher order scheme is presented for the optimal correction method previously proposed by the authors for computing the 3-D position of corresponding points in a stereo image pair, and this method is compared with the Hartley-Sturm method. It is pointed out that the epipole is a singularity of the Harley-Sturm method, while the our method has no singularity. It is confirmed by numerical simulation that both compute identical solutions at other points. It can be shown, however, that our method is significantly faster.

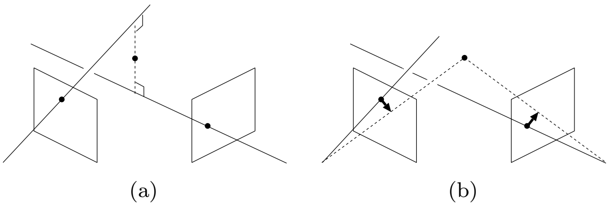


Figure 1 Triangulation. (a) The mid-point method. (b) Optimal correction.

1. Introduction

Stereo vision is a method for reconstructing the 3-D structure of a scene from corresponding points over two images 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 lines of sight of corresponding pixels 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].

However, correspondence detection entails uncertainty to some extent, and the lines of sight of detected corresponding pixels do not necessarily 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)).

In early 1990s, the authors pointed out that such a compromise is not optimal and that the 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)). This is the principle of *optimal correction* [5], for which the authors showed a simple numerical procedure [7].

Later, Hartley and Sturm [3] introduced a similar idea and presented a numerical scheme of reducing the problem to solving a 6-degree polynomial[1] (see Appendix). However, they only compared their method with the "mid-point method" and made no mention of the authors' optimal correction.

Yet, some researchers were aware that the authors' method was far more practical than the Hartley-Sturm method. For example, Torr and Zissermann [9] pointed out that the authors' method produced practically the same solution as the Hartley-Sturm method but was several orders of magnitude faster[2]. Nevertheless, the Hartley-Sturm method has won popularity and currently is widely regarded as a standard tool for triangulation.

The aim of this paper is to compare the Hartley-Sturm method with our optimal correction. First, we state the problem in mathematical terms (Sec. 2) and describe our previous formula of optimal correction (Sec. 3), which gives a first approximation. Then, we extend it to higher orders so that an exact optimal solution is computed (Sec. 4) and reformulate 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, where their computation fails, 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 far significantly faster than the Hartley-Strum method (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

[1] Their original paper [3] contained mathematical errors, which were corrected in a later published book [4].

[2] "... Hartley and Sturm [13] provide ... This turns out to be equivalent to the optimally corrected correspondence of Kanatani [16]. Comparisons of the computation of $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ by Hartley and Sturm's method and that of Kanatani have again 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. ..."

3-D vectors in the form

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

where f_0 is a scale constant of approximately the image size[3]. Hereafter, we refer to the point represented by vector $\boldsymbol{x}$ simply as "point $\boldsymbol{x}$".

As is well known [4, 5], the necessary and sufficient condition for the lines of sight of points $\boldsymbol{x}$ and $\boldsymbol{x}'$ meet in the scene is the following *epipolar equation*:

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

Throughout this paper, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$. In Eq. (2), $\boldsymbol{F}$ is a matrix of rank 2, called the *fundamental matrix*[4], determined by the relative configuration of the two cameras and their intrinsic parameters but independent of the 3-D structure of the scene or the choice of the corresponding points [4, 5]. The fundamental matrix $\boldsymbol{F}$ is determined by camera calibration. It can also be estimated from multiple pairs of point correspondences in the images [4, 5, 6]. Here, we assume that the fundamental matrix $\boldsymbol{F}$ is already given.

In practice, the detected corresponding points $\boldsymbol{x}$ and $\boldsymbol{x}'$ may not exactly satisfy Eq. (2) due to uncertainty of image processing operations. The principle of optimal correction [5] is to *optimally* correct points $\boldsymbol{x}$ and $\boldsymbol{x}'$ to points $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ so that Eq. (2) is satisfied, where by "optimally" we mean that the sum of square distances, or the *reprojection error* [4],

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

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

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

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

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

If the noise terms $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{x}'$ are independent and isotropic Gaussian random variables of mean $\boldsymbol{0}$ and a constant variance σ^2, minimization of Eq. (3) is equivalent to maximum likelihood (ML) estimation of the true positions $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{x}}'$.

3. Optimal Correction Procedure

The optimal correction formula in [5, 7] is derived as follows. Instead of directly estimating $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{x}}'$, we write

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

and estimate the correction terms $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{x}'$. If Eqs. (6) are substituted, Eq. (3) is written as

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

and Eq. (4) becomes

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

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

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

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

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

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

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

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

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

Multiplying these by the projection matrix

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

which makes the third component 0, and noting that $\boldsymbol{P}_{\mathrm{k}}\Delta\boldsymbol{x} = \Delta\boldsymbol{x}$, $\boldsymbol{P}_{\mathrm{k}}\Delta\boldsymbol{x}' = \Delta\boldsymbol{x}'$, and $\boldsymbol{P}_{\mathrm{k}}\boldsymbol{k} = \boldsymbol{0}$, we obtain

$$2\Delta\boldsymbol{x} - \lambda\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}' = \boldsymbol{0}, \quad 2\Delta\boldsymbol{x}' - \lambda\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x} = \boldsymbol{0}. \tag{14}$$

Hence, we have

$$\Delta\boldsymbol{x} = \frac{\lambda}{2}\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}', \qquad \Delta\boldsymbol{x}' = \frac{\lambda}{2}\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x}. \tag{15}$$

Substituting these into Eq. (9), we obtain

$$(\boldsymbol{F}\boldsymbol{x}', \frac{\lambda}{2}\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}') + (\boldsymbol{F}^\top\boldsymbol{x}, \frac{\lambda}{2}\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x}) = (\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}'), \tag{16}$$

from which λ is determined in the form

$$\frac{\lambda}{2} = \frac{(\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}')}{(\boldsymbol{F}\boldsymbol{x}', \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}') + (\boldsymbol{F}^\top\boldsymbol{x}, \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x})}. \tag{17}$$

Hence, Eqs. (15) becomes

$$\Delta\boldsymbol{x} = \frac{(\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}')\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}'}{(\boldsymbol{F}\boldsymbol{x}', \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}') + (\boldsymbol{F}^\top\boldsymbol{x}, \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x})}, \\ \Delta\boldsymbol{x}' = \frac{(\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}')\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x}}{(\boldsymbol{F}\boldsymbol{x}', \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}') + (\boldsymbol{F}^\top\boldsymbol{x}, \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x})}. \tag{18}$$

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

$$\hat{\boldsymbol{x}} = \boldsymbol{x} - \frac{(\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}')\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}'}{(\boldsymbol{F}\boldsymbol{x}', \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}') + (\boldsymbol{F}^\top\boldsymbol{x}, \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x})}, \\ \hat{\boldsymbol{x}}' = \boldsymbol{x}' - \frac{(\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}')\boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x}}{(\boldsymbol{F}\boldsymbol{x}', \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}\boldsymbol{x}') + (\boldsymbol{F}^\top\boldsymbol{x}, \boldsymbol{P}_{\mathrm{k}}\boldsymbol{F}^\top\boldsymbol{x})}. \tag{19}$$

This is the optimal correction formula given in [5, 7].

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

[4] In [1, 3], its transpose is defined to be the "fundamental matrix".

4. Higher Order Extension

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

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

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

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

where we define

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

Equation (4) now becomes

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

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

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

Since $\Delta\hat{\boldsymbol{x}}$ and $\Delta\hat{\boldsymbol{x}}'$ 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

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

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

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

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

$$\begin{aligned} 2(\tilde{\boldsymbol{x}} + \Delta\hat{\boldsymbol{x}}) - \lambda\boldsymbol{F}\hat{\boldsymbol{x}}' - \mu\boldsymbol{k} &= \boldsymbol{0}, \\ 2(\tilde{\boldsymbol{x}}' + \Delta\hat{\boldsymbol{x}}') - \lambda\boldsymbol{F}^\top\hat{\boldsymbol{x}} - \mu'\boldsymbol{k} &= \boldsymbol{0}. \end{aligned} \tag{27}$$

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

$$\begin{aligned} 2\tilde{\boldsymbol{x}} + 2\Delta\hat{\boldsymbol{x}} - \lambda\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}' &= \boldsymbol{0}, \\ 2\tilde{\boldsymbol{x}} + 2\Delta\hat{\boldsymbol{x}}' - \lambda\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}. &= \boldsymbol{0} \end{aligned} \tag{28}$$

Hence, we have

$$\Delta\hat{\boldsymbol{x}} = \frac{\lambda}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}' - \tilde{\boldsymbol{x}}, \quad \Delta\hat{\boldsymbol{x}}' = \frac{\lambda}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}'. \tag{29}$$

Substituting these into Eq. (24), we obtain

$$\begin{aligned} &(\boldsymbol{F}\hat{\boldsymbol{x}}', \frac{\lambda}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}' - \tilde{\boldsymbol{x}}) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}}, \frac{\lambda}{2}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}') \\ &= (\hat{\boldsymbol{x}}, \boldsymbol{F}\hat{\boldsymbol{x}}'), \end{aligned} \tag{30}$$

from which λ is determined in the form

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

Hence, Eq. (29) becomes

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

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

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

This is a second order approximation. Still, they may not strictly satisfy the epipolar equation $(\hat{\hat{\boldsymbol{x}}}, \boldsymbol{F}\hat{\hat{\boldsymbol{x}}}') = 0$, so we let

$$\hat{\boldsymbol{x}} \leftarrow \hat{\hat{\boldsymbol{x}}}, \qquad \hat{\boldsymbol{x}}' \leftarrow \hat{\hat{\boldsymbol{x}}}' \tag{34}$$

and repeat the same computation until the iterations converge. In the end, $\Delta\hat{\boldsymbol{x}}_\alpha$ and $\Delta\hat{\boldsymbol{x}}'_\alpha$ in Eq. (23) become $\boldsymbol{0}$.

5. Compact Numerical Scheme

The computation described in the preceding sections looks very complicated. We now show that it reduces to a very compact form suitable for numerical computation. This is the core of our algorithm.

First, we encode the fundamental matrix $\boldsymbol{F} = (F_{ij})$ and the corresponding point pair $\{(x, y),\ (x', y')\}$ in the following 9-D vectors:

$$\begin{aligned} \boldsymbol{u} &= (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top, \\ \boldsymbol{\xi} &= (xx', xy', f_0x, yx', yy', f_0y, f_0x', f_0y', f_0^2)^\top. \end{aligned} \tag{35}$$

The epipolar equation in Eq. (2) now becomes simply $(\boldsymbol{u}, \boldsymbol{\xi}) = 0$ [6]. We can also rewrite part of the numerators and denominators in Eqs. (19) as follows:

$$(\boldsymbol{x}, \boldsymbol{F}\boldsymbol{x}') = \frac{1}{f_0^2}(\boldsymbol{u}, \boldsymbol{\xi}), \tag{36}$$

$$(\boldsymbol{F}\boldsymbol{x}', \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}') + (\boldsymbol{F}^\top\boldsymbol{x}, \boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}) = \frac{1}{f_0^2}(\boldsymbol{u}, V_0[\boldsymbol{\xi}]\boldsymbol{u}). \tag{37}$$

Here, we define the matrix $V_0[\boldsymbol{\xi}]$ as follows[5]:

$$V_0[\boldsymbol{\xi}] = \left(\begin{array}{cccc} x^2+x'^2 & x'y' & f_0x' & xy \\ x'y' & x^2+y'^2 & f_0y' & 0 \\ f_0x' & f_0y' & f_0^2 & 0 \\ xy & 0 & 0 & y^2+x'^2 \\ 0 & xy & 0 & x'y' \\ 0 & 0 & 0 & f_0x' \\ f_0x & 0 & 0 & f_0y \\ 0 & f_0x & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right.$$

$$\left.\begin{array}{ccccc} 0 & 0 & f_0x & 0 & 0 \\ xy & 0 & 0 & f_0x & 0 \\ 0 & 0 & 0 & 0 & 0 \\ x'y' & f_0x' & f_0y & 0 & 0 \\ y^2+y'^2 & f_0y' & 0 & f_0y & 0 \\ f_0y' & f_0^2 & 0 & 0 & 0 \\ 0 & 0 & f_0^2 & 0 & 0 \\ f_0y & 0 & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right). \qquad (38)$$

From Eqs. (36) and (37), Eqs. (19) can be written as

$$\hat{\boldsymbol{x}} = \boldsymbol{x} - \frac{(\boldsymbol{u},\boldsymbol{\xi})\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\boldsymbol{x}'}{(\boldsymbol{u},V_0[\boldsymbol{\xi}]\boldsymbol{u})}, \quad \hat{\boldsymbol{x}}' = \boldsymbol{x}' - \frac{(\boldsymbol{u},\boldsymbol{\xi})\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\boldsymbol{x}}{(\boldsymbol{u},V_0[\boldsymbol{\xi}]\boldsymbol{u})}. \qquad (39)$$

Now, if we define the vector

$$\hat{\boldsymbol{\xi}} = \begin{pmatrix} \hat{x}\hat{x}' + \hat{x}'\tilde{x} + \hat{x}\tilde{x}' \\ \hat{x}\hat{y}' + \hat{y}'\tilde{x} + \hat{x}\tilde{y}' \\ f_0(\hat{x}+\tilde{x}) \\ \hat{y}\hat{x}' + \hat{x}'\tilde{y} + \hat{y}\tilde{x}' \\ \hat{y}\hat{y}' + \hat{y}'\tilde{y} + \hat{y}\tilde{y}' \\ f_0(\hat{y}+\tilde{y}) \\ f_0(\hat{x}'+\tilde{x}') \\ f_0(\hat{y}'+\tilde{y}') \\ f_0^2 \end{pmatrix}, \qquad (40)$$

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

$$(\hat{\boldsymbol{x}},\boldsymbol{F}\hat{\boldsymbol{x}}') + (\boldsymbol{F}\hat{\boldsymbol{x}}',\tilde{\boldsymbol{x}}) + (\boldsymbol{F}^\top\hat{\boldsymbol{x}},\tilde{\boldsymbol{x}}') = \frac{1}{f_0^2}(\boldsymbol{u},\hat{\boldsymbol{\xi}}), \qquad (41)$$

$$(\boldsymbol{F}\hat{\boldsymbol{x}}',\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}') + (\boldsymbol{F}^\top\hat{\boldsymbol{x}},\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}) = \frac{1}{f_0^2}(\boldsymbol{u},V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{u}), \qquad (42)$$

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. (38). From Eqs. (41) and (42), we can write Eqs. (33) as

$$\hat{\hat{\boldsymbol{x}}} = \boldsymbol{x} - \frac{(\boldsymbol{u},\hat{\boldsymbol{\xi}})\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'}{(\boldsymbol{u},V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{u})}, \quad \hat{\hat{\boldsymbol{x}}}' = \boldsymbol{x}' - \frac{(\boldsymbol{u},\hat{\boldsymbol{\xi}})\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}}{(\boldsymbol{u},V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{u})}. \qquad (43)$$

Thus, we can obtain the following compact procedure:

1. Let $E_0 = \infty$ (a sufficiently large number), and let

$$\hat{\boldsymbol{x}} = \boldsymbol{x}, \quad \hat{\boldsymbol{x}}' = \boldsymbol{x}', \quad \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}' = \mathbf{0} \qquad (44)$$

2. Compute the 9-D vector $\hat{\boldsymbol{\xi}}$ in Eq. (40) and the 9×9 matrix $V_0[\hat{\boldsymbol{\xi}}]$ in Eq. (38).
3. Update $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$ as follows:

$$\tilde{\boldsymbol{x}} \leftarrow \frac{(\boldsymbol{u},\hat{\boldsymbol{\xi}})\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}\hat{\boldsymbol{x}}'}{(\boldsymbol{u},V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{u})}, \quad \tilde{\boldsymbol{x}}' \leftarrow \frac{(\boldsymbol{u},\hat{\boldsymbol{\xi}})\boldsymbol{P}_{\mathbf{k}}\boldsymbol{F}^\top\hat{\boldsymbol{x}}}{(\boldsymbol{u},V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{u})}. \qquad (45)$$

4. Compute the reprojection error E by

$$E = \|\tilde{\boldsymbol{x}}\|^2 + \|\tilde{\boldsymbol{x}}'\|^2. \qquad (46)$$

5. If $E \approx E_0$, return $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ and stop. Else, let

$$E_0 \leftarrow E, \quad \hat{\boldsymbol{x}} \leftarrow \boldsymbol{x} - \tilde{\boldsymbol{x}}, \quad \hat{\boldsymbol{x}}' \leftarrow \boldsymbol{x}' - \tilde{\boldsymbol{x}}', \qquad (47)$$

and go back to Step 3.

If we stop at Step 3, we compute the first approximation solution given in [5, 7].

6. Theoretical Issues

6.1 Effect of Epipoles

Point $\boldsymbol{x}$ in the first image is called its *epipole* if $\boldsymbol{F}^\top\boldsymbol{x} = \mathbf{0}$; point $\boldsymbol{x}'$ in the second image is called its epipole if $\boldsymbol{F}\boldsymbol{x}' = \mathbf{0}$. They represent the projections of the viewpoints of the other cameras [4].

The Hartley-Sturm method [3] first translates the images so that the corresponding points are at the coordinate origins and then rotates the images so that their epipoles are on the x-axes (see Appendix). Then, the observed points at the origins are displaced to the foot of the perpendicular lines to 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.

If *either* of the corresponding points is at the epipole, no epipolar line is defined and the computation breaks down, so Hartley and Sturm [3] state that one needs an ad hoc procedure. In our procedure, $(\boldsymbol{u},V_0[\hat{\boldsymbol{\xi}}]\boldsymbol{u})$ is the only quantity that appears in denominators. From Eqs. (36), (37), (41), and (42), we see that it becomes zero only when $\boldsymbol{x}$ and $\boldsymbol{x}'$ (or $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ in the course of iterations) are *both* at the epipoles, in which case the numerators in Eqs. (45) are also zero, so no correction is done and the computation ends[6].

If one of the corresponding points are at the epipole, we see from Eqs. (45) 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 than the Hartley-Sturm method [3], although this this difference has little effect in practical situations.

[5] This coincides, up to scale, with the covariance matrix of $\boldsymbol{\xi}$ in Eqs. (35) obtained by assuming that independent and identical Gaussian noise of mean 0 and a constant variance is added to each coordinate of the point pair [6].

[6] We regard a fraction as zero if the numerator is zero, whatever the denominator.

6.2 Convergence

Our method consists of iterative updates, so one may wonder if the search is trapped into local minima. In fact, Hartley and Sturm [3] emphasized this aspect 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 if arbitrarily started in the parameter space. This is true in general. However, our method does not parameterize anything. We directly search the joint image plane for the positions $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ that satisfy the epipolar equation.

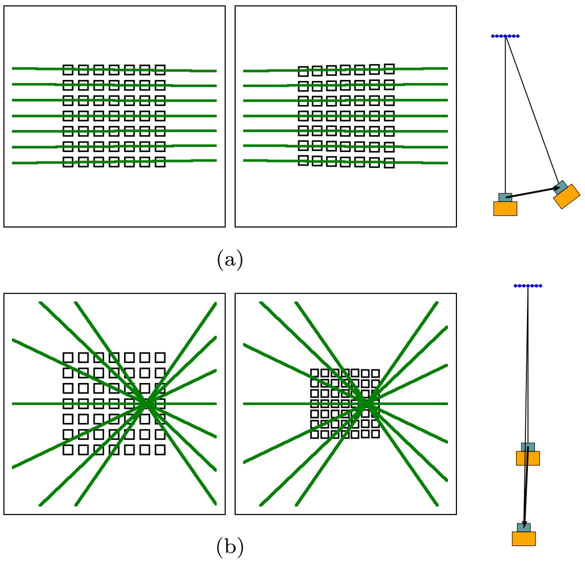
Evidently, local search in a parameter space should start near the true solution to converge to it, since we may be trapped into a false solution if it is located closer to us than the true solution. In our problem, however, we are searching for the positions $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ that are the *closest* to the data $\boldsymbol{x}$ and $\boldsymbol{x}'$, *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. The underlying principle is the same as the well-known global optimization technique of gradually raising the reprojection error threshold from 0 and testing if a feasible solution exists [2]; the one first found is the globally optimal solution.

7. Experiments

7.1 Setting

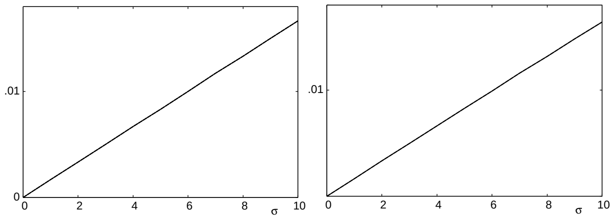
Figure 2(a), (b) shows simulated images (supposedly 400 × 400 pixels) of a grid pattern viewed by two cameras with focal length 1200 pixels. In Fig. 2(a), the baseline is nearly perpendicular to the camera optical axes (call this the "stable camera configuration"); in Fig. 2(b), it is nearly parallel 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. To see if our method works even in the presence of extremely large noise, we varied σ from 0 to 10 pixels. The iterations of our method are terminated when the update of the reprojection error E is less than 10^{-6}.


(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.



(a) (b)

Figure 3 Root mean reprojection error. Solid line: Our optimal correction. Dashed line: The Hartley-Sturm method. (a) Stable camera configuration. (b) Unstable camera configuration.

7.2 Reprojection Error

Figure 3 plots, for each σ, the "root mean reprojection error", i.e., the square root of the average of the reprojection error

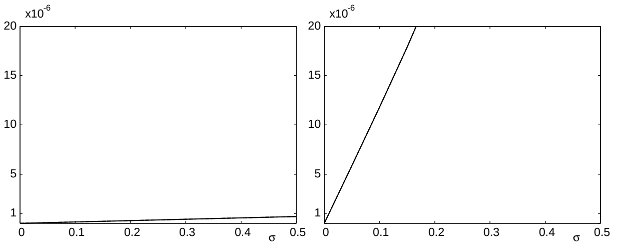
$$E = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 + \|\boldsymbol{x}' - \hat{\boldsymbol{x}}'\|^2 \qquad (48)$$

over all the grid points and over 1000 independent trials. The solid line shows the result of our optimal correction; the dashed line the Hartley-Sturm method. The two plots completely coincide in whichever camera configuration. In fact, the solutions of the two methods completely agree to significant digits, confirming that the two algorithms are mathematically equivalent.

7.3 Depth Error

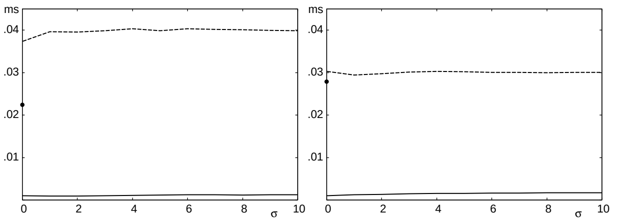
As Fig. 3 shows, the reprojection error is basically the same in the stable and unstable camera configurations. This is because optimal correction depends only on the statistical properties of noise. Namely, if the noise distribution is the same, say Gaussian, minimization of the reprojection error does *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*. To see this, Fig. 4 plots



(a) (b)

Figure 4 Root mean square of the inverse depth of 3-D reconstruction. Solid line: Our optimal correction. Dashed line: The Hartley-Sturm method. (a) Stable camera configuration. (b) Unstable camera configuration.



(a) (b)

Figure 5 Computation time (in ms). Solid line: Our optimal correction. Dashed line: The Hartley-Sturm method. (a) Stable camera configuration. (b) Unstable camera configuration.

the root mean square of the "inverse depth"[7]. For the computed depth Z, we let $\bar{\rho} = 1/\bar{Z}$ and $\rho = 1/Z$ and averaged $(\rho - \bar{\rho})^2$ over all the grid points and over 1000 independent trials. As had been predicted, we observed very irregular and meaningless results when σ is extremely large, so we restrict σ here to a realistic range of $[0, 0.5]$.

We can confirm from Fig. 4(a),(b) that if the camera configuration is unstable, a small amount of noise, even if the points are optimally corrected, results in a large variation of the reconstructed depth, and our method and the Hartley-Sturm methods compute the same solution.

7.4 Computation Time

We have seen that our method and the Hartley-Strum method compute identical solutions (except theoretically at epipoles). The biggest difference between them is the computation time. Figure 5 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.

We can see from Fig. 5 that 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, doing the same computation for whatever data. The exception is at $\sigma = 0$, at which the computation is faster (see the black dot on the vertical axis). If the data are exact, 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, in contrast, iterative. From Fig. 5, 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. 5, our method is significantly faster than the Hartley-Sturm method.

8. Conclusions

We have extended the first order optimal correction of the authors [5, 7] for computing the 3-D position of corresponding points in stereo images by triangulation 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.

[7] The inverse depth is widely used for evaluating the reliability of depth computation to handle the cases where the 3-D position is computed to be very far a way in front of the cameras or very far a way *behind* the cameras in the presence of large noise.

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

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

[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.

Appendix: Hartley-Sturm Method

The Hartley-Sturm method [3] consists of the following computation (symbols are somewhat altered to conform to our usage):

Input: The fundamental matrix $\boldsymbol{F}$ and corresponding points (x, y) and (x', y').

Output: The corrected positions $(\hat{x}, \hat{y})$ and $(\hat{x}', \hat{y}')$.

Procedure:

1. Let $\boldsymbol{e} = (e_1, e_2, e_3)^\top$ and $\boldsymbol{e}' = (e'_1, e'_2, e'_3)^\top$ be the unit eigenvectors of $\boldsymbol{F}\boldsymbol{F}^\top$ and $\boldsymbol{F}^\top\boldsymbol{F}$, respectively, with eigenvalue 0.
2. Compute the angles
$$\theta = \arg(f_0 e_1 - x e_3, f_0 e_2 - y e_3),$$
$$\theta' = \arg(f_0 e'_1 - x' e'_3, f_0 e'_2 - y' e'_3), \qquad (49)$$
where $\arg(x, y)$ denotes the argument of point (x, y) measured from the x-axis.
3. Compute the following matrices:
$$\boldsymbol{L} = \begin{pmatrix} 1 & & -x/f_0 \\ & 1 & -y/f_0 \\ & & 1 \end{pmatrix}, \quad \boldsymbol{L}' = \begin{pmatrix} 1 & & -x'/f_0 \\ & 1 & -y'/f_0 \\ & & 1 \end{pmatrix},$$
$$\boldsymbol{R} = \begin{pmatrix} \cos\theta & \sin\theta & \\ -\sin\theta & \cos\theta & \\ & & 1 \end{pmatrix},$$
$$\boldsymbol{R}' = \begin{pmatrix} \cos\theta' & \sin\theta' & \\ -\sin\theta' & \cos\theta' & \\ & & 1 \end{pmatrix}. \qquad (50)$$
4. Transform the fundamental matrix $\boldsymbol{F}$ as follows:
$$\tilde{\boldsymbol{F}} = \boldsymbol{R}(\boldsymbol{L}^{-1})^\top \boldsymbol{F} \boldsymbol{L}'^{-1} \boldsymbol{R}'^{-1}. \qquad (51)$$
5. Compute the following a, b, c, d, f, and f'.
$$a = \tilde{F}_{22}, \quad b = \tilde{F}_{32}, \quad c = \tilde{F}_{23}, \quad d = \tilde{F}_{33},$$
$$f = \begin{cases} -\dfrac{\tilde{F}_{12}}{b} & |b| \geq |d| \\ -\dfrac{\tilde{F}_{13}}{d} & |b| < |d| \end{cases},$$
$$f' = \begin{cases} -\dfrac{\tilde{F}_{21}}{c} & |c| \geq |d| \\ -\dfrac{\tilde{F}_{31}}{d} & |c| < |d| \end{cases} \qquad (52)$$
6. Solve the 6-degree polynomial in t
$$a_6 t^6 + a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 = 0, \quad (53)$$
where
$$\begin{aligned} a_6 &= -f^4 ac(ad - bc) \\ a_5 &= (a^2 + f'^2 c^2)^2 - f^4 (ad - bc)(ad + bc) \\ a_4 &= 2(a^2 + f'^2 c^2)(ab + f'^2 cd) \\ &\quad -f^2 (ad - bc)(2ac + f^2 bd) \\ a_3 &= 2(a^2 + f'^2 c^2)(b^2 + f'^2 d^2) + 4(ab + f'^2 cd)^2 \\ &\quad -2f^2 (ad - bc)(ad + bc) \\ a_2 &= 4(ab + f'^2 cd)(b^2 + f'^2 d^2) \\ &\quad -(ad - bc)(ac + 2f^2 bd) \\ a_1 &= (b^2 + f'^2 d^2)^2 - (ad - bc)(ad + bc) \\ a_0 &= -(ad - bc)bd. \end{aligned} \qquad (54)$$
7. From the (at most 6) real roots, let t_0 the one that minimizes.
$$E(t) = \frac{t^2}{1 + f^2 t^2} + \frac{(ct + d)^2}{(at + b)^2 + f'^2 (ct + d)^2}. \quad (55)$$
8. If $E(t_0) \leq E(\infty)$, the compute
$$\tilde{x} = \frac{f_0 f t_0^2}{1 + f^2 t_0^2}, \qquad \tilde{y} = \frac{f_0 t_0}{1 + f^2 t_0^2},$$
$$\tilde{x}' = \frac{f_0 f' (ct_0 + d)^2}{f'^2 (ct_0 + d)^2 + (at_0 + b)^2},$$
$$\tilde{y}' = -\frac{f_0 (at_0 + b)(ct_0 + d)}{f'^2 (ct_0 + d)^2 + (at_0 + b)^2}. \qquad (56)$$
Else, let
$$\tilde{x} = \frac{f_0}{f}, \qquad \tilde{y} = 0,$$
$$\tilde{x}' = \frac{f_0 f' c^2}{f'^2 c^2 + a^2}, \qquad \tilde{y}' = -\frac{f_0 ac}{f'^2 c^2 + a^2}. \qquad (57)$$
9. Compute $(\hat{x}, \hat{y})$ and $(\hat{x}', \hat{y}')$ by
$$\begin{pmatrix} \hat{x}/f_0 \\ \hat{y}/f_0 \\ 1 \end{pmatrix} = \boldsymbol{L}^{-1} \boldsymbol{R}^{-1} \begin{pmatrix} \tilde{x}/f_0 \\ \tilde{y}/f_0 \\ 1 \end{pmatrix},$$
$$\begin{pmatrix} \hat{x}'/f_0 \\ \hat{y}'/f_0 \\ 1 \end{pmatrix} = \boldsymbol{L}'^{-1} \boldsymbol{R}'^{-1} \begin{pmatrix} \tilde{x}'/f_0 \\ \tilde{y}'/f_0 \\ 1 \end{pmatrix}. \qquad (58)$$