SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Minimal solution for estimating fundamental matrix under planar motion

Qiang FU[1,2*], Xinxing Mu[1] & Yu Wang[3]

[1]*School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China;*
[2]*Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China;*
[3]*Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

Appendix A Related works

The 8-point method [1] can be used to estimate the fundamental matrix when the number of points $N \geqslant 8$. It first involves solving a set of linear equations without considering the singularity constraint of the fundamental matrix. Then the singularity constraint is enforced by singular value decomposition (SVD) correction. The 7-point method [1] is a minimal solver and is only applicable when $N = 7$, which enforces the singularity constraint directly. It is often used with the random sample consensus (RANSAC) algorithm to deal with outliers. Both the 8-point method and the 7-point method are designed for general motion. The 6-point method [2] can be used to estimate the fundamental matrix under planar motion when $N \geqslant 6$. Similar to the 8-point method, the 6-point method enforces the singularity constraint by SVD correction. The proposed 4-point method is a minimal solver designed for planar motion and is only applicable when $N = 4$. Similar to the 7-point method, it enforces the singularity constraint directly. Note that the importance of the singularity constraint is pointed out in [3]. A detailed comparison of the four different methods is shown in Table A1.

Table A1 Comparison of the four different methods.

Method	Singularity constraint	Planar-motion constraint	Motion type	Minimal solver
8-point	posteriorly enforced	Not exploiting	general motion	No
7-point	directly enforced	Not exploiting	general motion	Yes
6-point	posteriorly enforced	Exploiting	planar motion	No
4-point	directly enforced	Exploiting	planar motion	Yes

Appendix B Definition of coefficient matrix $\mathbf{C} \in \mathbb{R}^{2\times 10}$

Suppose that we have got

$$\mathbf{F}_1 = \begin{bmatrix} 0 & a_1 & a_2 \\ a_3 & 0 & a_4 \\ a_5 & a_6 & a_7 \end{bmatrix}, \mathbf{F}_2 = \begin{bmatrix} 0 & b_1 & b_2 \\ b_3 & 0 & b_4 \\ b_5 & b_6 & b_7 \end{bmatrix}, \mathbf{F}_3 = \begin{bmatrix} 0 & c_1 & c_2 \\ c_3 & 0 & c_4 \\ c_5 & c_6 & c_7 \end{bmatrix}. \tag{B1}$$

Then

$$\mathbf{F} = x\mathbf{F}_1 + y\mathbf{F}_2 + \mathbf{F}_3 = \begin{bmatrix} 0 & a_1x + b_1y + c_1 & a_2x + b_2y + c_2 \\ a_3x + b_3y + c_3 & 0 & a_4x + b_4y + c_4 \\ a_5x + b_5y + c_5 & a_6x + b_6y + c_6 & a_7x + b_7y + c_7 \end{bmatrix}. \tag{B2}$$

According to the first constraint $\det(\mathbf{F}) = 0$, we can rewrite this constraint equation as

$$[\mathbf{C}(1,1), \mathbf{C}(1,2), \ldots, \mathbf{C}(1,10)][x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1]^T = 0. \tag{B3}$$

* Corresponding author (email: fuqiang@ustb.edu.cn)

Therefore, we can get

$$
\begin{aligned}
\mathbf{C}(1,1) &= a_1a_4a_5 - a_1a_3a_7 + a_2a_3a_6\\
\mathbf{C}(1,2) &= a_1a_4b_5 + a_1a_5b_4 + a_4a_5b_1 - a_1a_3b_7 - a_1a_7b_3 + a_2a_3b_6 + a_2a_6b_3 + a_3a_6b_2 - a_3a_7b_1\\
\mathbf{C}(1,3) &= a_1b_4b_5 + a_4b_1b_5 + a_5b_1b_4 - a_1b_3b_7 + a_2b_3b_6 - a_3b_1b_7 + a_3b_2b_6 + a_6b_2b_3 - a_7b_1b_3\\
\mathbf{C}(1,4) &= b_1b_4b_5 - b_1b_3b_7 + b_2b_3b_6\\
\mathbf{C}(1,5) &= a_1a_4c_5 + a_1a_5c_4 + a_4a_5c_1 - a_1a_3c_7 - a_1a_7c_3 + a_2a_3c_6 + a_2a_6c_3 + a_3a_6c_2 - a_3a_7c_1\\
\mathbf{C}(1,6) &= a_1b_4c_5 + a_1b_5c_4 + a_4b_1c_5 + a_4b_5c_1 + a_5b_1c_4 + a_5b_4c_1 - a_1b_3c_7 - a_1b_7c_3 + a_2b_3c_6\\
&\quad + a_2b_6c_3 - a_3b_1c_7 + a_3b_2c_6 + a_3b_6c_2 - a_3b_7c_1 + a_6b_2c_3 + a_6b_3c_2 - a_7b_1c_3 - a_7b_3c_1\\
\mathbf{C}(1,7) &= b_1b_4c_5 + b_1b_5c_4 + b_4b_5c_1 - b_1b_3c_7 - b_1b_7c_3 + b_2b_3c_6 + b_2b_6c_3 + b_3b_6c_2 - b_3b_7c_1\\
\mathbf{C}(1,8) &= a_1c_4c_5 + a_4c_1c_5 + a_5c_1c_4 - a_1c_3c_7 + a_2c_3c_6 - a_3c_1c_7 + a_3c_2c_6 + a_6c_2c_3 - a_7c_1c_3\\
\mathbf{C}(1,9) &= b_1c_4c_5 + b_4c_1c_5 + b_5c_1c_4 - b_1c_3c_7 + b_2c_3c_6 - b_3c_1c_7 + b_3c_2c_6 + b_6c_2c_3 - b_7c_1c_3\\
\mathbf{C}(1,10) &= c_1c_4c_5 - c_1c_3c_7 + c_2c_3c_6.
\end{aligned}
$$

Similarly, according to the second constraint $\det(\mathbf{F}+\mathbf{F}^T)=0$, we can get

$$
\begin{aligned}
\mathbf{C}(2,1) &= 2a_1a_2a_4 - 2a_3^2a_7 - 2a_1^2a_7 + 2a_1a_2a_6 + 2a_2a_3a_4 + 2a_1a_4a_5 - 4a_1a_3a_7 + 2a_2a_3a_6 + 2a_1a_5a_6\\
&\quad + 2a_3a_4a_5 + 2a_3a_5a_6\\
\mathbf{C}(2,2) &= 2a_1a_2b_4 - 2a_3^2b_7 - 2a_1^2b_7 + 2a_1a_4b_2 + 2a_2a_4b_1 + 2a_1a_2b_6 + 2a_1a_6b_2 - 4a_1a_7b_1 + 2a_2a_3b_4\\
&\quad + 2a_2a_4b_3 + 2a_2a_6b_1 + 2a_3a_4b_2 + 2a_1a_4b_5 + 2a_1a_5b_4 + 2a_4a_5b_1 - 4a_1a_3b_7 - 4a_1a_7b_3 + 2a_2a_3b_6\\
&\quad + 2a_2a_6b_3 + 2a_3a_6b_2 - 4a_3a_7b_1 + 2a_1a_5b_6 + 2a_1a_6b_5 + 2a_3a_4b_5 + 2a_3a_5b_4 + 2a_4a_5b_3 + 2a_5a_6b_1\\
&\quad - 4a_3a_7b_3 + 2a_3a_5b_6 + 2a_3a_6b_5 + 2a_5a_6b_3\\
\mathbf{C}(2,3) &= 2a_1b_2b_4 - 2a_7b_3^2 - 2a_7b_1^2 + 2a_2b_1b_4 + 2a_4b_1b_2 - 4a_1b_1b_7 + 2a_1b_2b_6 + 2a_2b_1b_6 + 2a_2b_3b_4\\
&\quad + 2a_3b_2b4 + 2a_4b_2b_3 + 2a_6b_1b_2 + 2a_1b_4b_5 + 2a_4b_1b_5 + 2a_5b_1b_4 - 4a_1b_3b_7 + 2a_2b_3b_6 - 4a_3b_1b_7\\
&\quad + 2a_3b_2b_6 + 2a_6b_2b_3 - 4a_7b_1b_3 + 2a_1b_5b_6 + 2a_3b_4b_5 + 2a_4b_3b_5 + 2a_5b_1b_6 + 2a_5b_3b_4 + 2a_6b_1b_5\\
&\quad - 4a_3b_3b_7 + 2a_3b_5b_6 + 2a_5b_3b_6 + 2a_6b_3b_5\\
\mathbf{C}(2,4) &= 2b_1b_2b_4 - 2b_3^2b_7 - 2b_1^2b_7 + 2b_1b_2b_6 + 2b_2b_3b_4 + 2b_1b_4b_5 - 4b_1b_3b_7 + 2b_2b_3b_6 + 2b_1b_5b_6 + 2b_3b_4b_5 + 2b_3b_5b_6\\
\mathbf{C}(2,5) &= 2a_1a_2c_4 - 2a_3^2c_7 - 2a_1^2c_7 + 2a_1a_4c_2 + 2a_2a_4c_1 + 2a_1a_2c_6 + 2a_1a_6c_2 - 4a_1a_7c_1 + 2a_2a_3c_4\\
&\quad + 2a_2a_4c_3 + 2a_2a_6c_1 + 2a_3a_4c_2 + 2a_1a_4c_5 + 2a_1a_5c_4 + 2a_4a_5c_1 - 4a_1a_3c_7 - 4a_1a_7c_3 + 2a_2a_3c_6\\
&\quad + 2a_2a_6c_3 + 2a_3a_6c_2 - 4a_3a_7c_1 + 2a_1a_5c_6 + 2a_1a_6c_5 + 2a_3a_4c_5 + 2a_3a_5c_4 + 2a_4a_5c_3 + 2a_5a_6c_1\\
&\quad - 4a_3a_7c_3 + 2a_3a_5c_6 + 2a_3a_6c_5 + 2a_5a_6c_3\\
\mathbf{C}(2,6) &= 2a_1b_2c_4 + 2a_1b_4c_2 + 2a_2b_1c_4 + 2a_2b_4c_1 + 2a_4b_1c_2 + 2a_4b_2c_1 - 4a_1b_1c_7 + 2a_1b_2c_6 + 2a_1b_6c_2\\
&\quad - 4a_1b_7c_1 + 2a_2b_1c_6 + 2a_2b_3c_4 + 2a_2b_4c_3 + 2a_2b_6c_1 + 2a_3b_2c_4 + 2a_3b_4c_2 + 2a_4b_2c_3 + 2a_4b_3c_2\\
&\quad + 2a_6b_1c_2 + 2a_6b_2c_1 - 4a_7b_1c_1 + 2a_1b_4c_5 + 2a_1b_5c_4 + 2a_4b_1c_5 + 2a_4b_5c_1 + 2a_5b_1c_4 + 2a_5b_4c_1\\
&\quad - 4a_1b_3c_7 - 4a_1b_7c_3 + 2a_2b_3c_6 + 2a_2b_6c_3 - 4a_3b_1c_7 + 2a_3b_2c_6 + 2a_3b_6c_2 - 4a_3b_7c_1 + 2a_6b_2c_3\\
&\quad + 2a_6b_3c_2 - 4a_7b_1c_3 - 4a_7b_3c_1 + 2a_1b_5c_6 + 2a_1b_6c_5 + 2a_3b_4c_5 + 2a_3b_5c_4 + 2a_4b_3c_5 + 2a_4b_5c_3\\
&\quad + 2a_5b_1c_6 + 2a_5b_3c_4 + 2a_5b_4c_3 + 2a_5b_6c_1 + 2a_6b_1c_5 + 2a_6b_5c_1 - 4a_3b_3c_7 - 4a_3b_7c_3 - 4a_7b_3c_3\\
&\quad + 2a_3b_5c_6 + 2a_3b_6c_5 + 2a_5b_3c_6 + 2a_5b_6c_3 + 2a_6b_3c_5 + 2a_6b_5c_3\\
\mathbf{C}(2,7) &= 2b_1b_2c_4 - 2b_3^2c_7 - 2b_1^2c_7 + 2b_1b_4c_2 + 2b_2b_4c_1 + 2b_1b_2c_6 + 2b_1b_6c_2 - 4b_1b_7c_1 + 2b_2b_3c_4\\
&\quad + 2b_2b_4c_3 + 2b_2b_6c_1 + 2b_3b_4c_2 + 2b_1b_4c_5 + 2b_1b_5c_4 + 2b_4b_5c_1 - 4b_1b_3c_7 - 4b_1b_7c_3 + 2b_2b_3c_6\\
&\quad + 2b_2b_6c_3 + 2b_3b_6c_2 - 4b_3b_7c_1 + 2b_1b_5c_6 + 2b_1b_6c_5 + 2b_3b_4c_5 + 2b_3b_5c_4 + 2b_4b_5c_3 + 2b_5b_6c_1\\
&\quad - 4b_3b_7c_3 + 2b_3b_5c_6 + 2b_3b_6c_5 + 2b_5b_6c_3\\
\mathbf{C}(2,8) &= 2a_1c_2c_4 - 2a_7c_3^2 - 2a_7c_1^2 + 2a_2c_1c_4 + 2a_4c_1c_2 - 4a_1c_1c_7 + 2a_1c_2c_6 + 2a_2c_1c_6 + 2a_2c_3c_4\\
&\quad + 2a_3c_2c_4 + 2a_4c_2c_3 + 2a_6c_1c_2 + 2a_1c_4c_5 + 2a_4c_1c_5 + 2a_5c_1c_4 - 4a_1c_3c_7 + 2a_2c_3c_6 - 4a_3c_1c_7\\
&\quad + 2a_3c_2c_6 + 2a_6c_2c_3 - 4a_7c_1c_3 + 2a_1c_5c_6 + 2a_3c_4c_5 + 2a_4c_3c_5 + 2a_5c_1c_6 + 2a_5c_3c_4 + 2a_6c_1c_5\\
&\quad - 4a_3c_3c_7 + 2a_3c_5c_6 + 2a_5c_3c_6 + 2a_6c_3c_5\\
\mathbf{C}(2,9) &= 2b_1c_2c_4 - 2b_7c_3^2 - 2b_7c_1^2 + 2b_2c_1c_4 + 2b_4c_1c_2 - 4b_1c_1c_7 + 2b_1c_2c_6 + 2b_2c_1c_6 + 2b_2c_3c_4\\
&\quad + 2b_3c_2c_4 + 2b_4c_2c_3 + 2b_6c_1c_2 + 2b_1c_4c_5 + 2b_4c_1c_5 + 2b_5c_1c_4 - 4b_1c_3c_7 + 2b_2c_3c_6 - 4b_3c_1c_7\\
&\quad + 2b_3c_2c_6 + 2b_6c_2c_3 - 4b_7c_1c_3 + 2b_1c_5c_6 + 2b_3c_4c_5 + 2b_4c_3c_5 + 2b_5c_1c_6 + 2b_5c_3c_4 + 2b_6c_1c_5\\
&\quad - 4b_3c_3c_7 + 2b_3c_5c_6 + 2b_5c_3c_6 + 2b_6c_3c_5\\
\mathbf{C}(2,10) &= 2c_1c_2c_4 - 2c_3^2c_7 - 2c_1^2c_7 + 2c_1c_2c_6 + 2c_2c_3c_4 + 2c_1c_4c_5 - 4c_1c_3c_7 + 2c_2c_3c_6 + 2c_1c_5c_6 + 2c_3c_4c_5 + 2c_3c_5c_6.
\end{aligned}
$$

Appendix C Real experiments

Three images acquired from a camera mounted on a vehicle moving on a plane (available at https://urlc.cn/Eb55BNH) are used in the real experiments. Six fundamental matrices can be obtained from these images. For each image pair, after building tentative matches by matching SIFT points [4], potential outliers are removed through RANSAC combined with the proposed 4-point method or the 7-point method. Then the fundamental matrix is estimated from all correspondences classified as inliers using the normalized 8-point method [1]. As for the evaluation criterion, the root-mean-squared (RMS) error is used, i.e.,

$$E_{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{\left(\mathbf{x}_i'^T\mathbf{F}\mathbf{x}_i\right)^2}{(\xi)_1^2+(\xi)_2^2+(\xi')_1^2+(\xi')_2^2}}, \tag{C1}$$

where $\xi = \mathbf{F}\mathbf{x}_i$, $\xi' = \mathbf{F}^T\mathbf{x}_i'$. Besides, $(\xi)_j^2$ and $(\xi')_j^2$ denote the square of the j-th entry of the vector ξ and ξ' respectively, and n is the number of correspondences classified as inliers.

The computation efficiency of the 8-point method, the 7-point method, the 6-point method and the proposed 4-point method combining with RANSAC is also investigated for the above real images. Experiments were performed by Matlab 2011b running on a desktop computer (Intel Core Processor at 3.2GHz). From Table C1, it can be concluded that when combined with RANSAC, the proposed 4-point method is more accurate than the other methods for the real images. The RANSAC running time of the proposed 4-point method is a little larger than that of the 8-point method and the 6-point method, but is much smaller than that of the 7-point method.

Table C1 Performance of different methods for the real image pairs.

Method	RANSAC running time (ms)	E_{RMS}
8-point	25.8	1.6996
7-point	649.4	1.6708
6-point	18.1	1.7482
4-point	59.8	1.5648

Note that RANSAC is usually used with a minimal solver. The 8-point method and the 6-point method are not minimal solvers and their results are listed here for the purpose of comparison. The 4-point method combined with RANSAC gives better accuracy than the other methods because it is the minimal solver for planar motion and is more effective to deal with outliers.

Note that the calculation complexity of the methods combined with RANSAC can be roughly analyzed as follows. Suppose that the number of input image points is N and the number of RANSAC iterations is N_{iter}. Then the overall calculation complexity of the four methods combined with RANSAC is [5]

$$Complexity = O\left(N_{iter}\left(C_{fitting} + NC_{dis}\right)\right), \tag{C2}$$

where $C_{fitting}$ is the cost associated to compute the fitting of the fundamental matrix from s matched points ($s = 8, 7, 6, 4$ in this paper), and C_{dis} is the cost associated to compute the geometric error of the fundamental matrix with respect to each pair of image points.

References

1 Hartley R, Zisserman A. Multiple View Geometry in Computer Vision (2nd Edition). Cambridge: Cambridge University Press, 2004.

2 Ortín D, Montiel J M M. Indoor robot motion based on monocular images. Robotica, 2001, 19: 331-342.

3 Migita T, Shakunaga T. Evaluation of epipole estimation methods with/without rank-2 constraint across algebraic/geometric error functions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, 2007, 1-7.

4 Lowe D G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60: 91-110.

5 Zuliani M. RANSAC for Dummies. Vision Research Lab, University of California, Santa Barbara, 2009.