UDC 512.643.2

A COMPACT FORM OF A CUBIC EQUATION FROM THE 7-POINT ALGORITHM

O. Monastyrskyi[1, a], V. Krygin[1,2, b]

[1] National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Institute of Physics and Technology

[2] International Research and Training Center for Information Technologies and Systems of the National Academy of Sciences of Ukraine and Ministry of Education and Science of Ukraine

Abstract

One of the popular approaches to estimate the fundamental matrix of a stereo pair is a 7-point algorithm. One of the key elements of the algorithm is a solution of a cubic equation. This paper provides a compact representation of the equation, helping researchers to implement less error-prone computer vision software faster.

Keywords: computer vision, fundamental matrix, cubic equation, determinant

Introduction

The fundamental matrix plays a significant role in computer stereo vision, helping to build 3D models from sets of images [1, 2]. The cubic equation, needed by the seven-point algorithm, can be derived using symbolic computing tools of Macaulay2 [3] or SymPy [4]. There is a work dedicated to the estimation of the essential matrix [5], containing an explicit form of equations, which can be used for the fundamental matrix estimation in partially calibrated case [6]. Our paper is about the uncalibrated case.

The first section of the paper provides the 7-point algorithm. If you are already familiar with it, you can skip to the second section, where we derive a compact form of the cubic equation appearing in the algorithm.

1. 7-point algorithm

1.1. Overview

Given a set of points' correspondences $X \subset \mathbb{R}^{3\times 2}$ from two images, we want to estimate the rank 2 fundamental matrix F, satisfying

$$\boldsymbol{x}'^T F \boldsymbol{x} = 0, \quad \forall \langle \boldsymbol{x}', \boldsymbol{x} \rangle \in X.$$

Noisy data and wrong correspondences lead to

$$\left|\boldsymbol{x}'^T F \boldsymbol{x}\right| < \varepsilon, \quad \forall \langle \boldsymbol{x}', \boldsymbol{x} \rangle \in X' \subset X, \tag{1}$$

where $\varepsilon \sim 10^{-3}$, and X' is a set, for which the inequality holds. The bigger $|X'|$ means the better F.

First, you should find points' correspondences on two images. They may be imprecise and may not cover entire images – you can use [7] for this purpose. One iteration of the F estimation procedure:

1) choose seven pairs of corresponding points;
2) find a solution to the equations described below;
3) check how many correspondences satisfy (1).

As a result (after thousands of iterations), you should choose the best matrix – the one, for which you've got the biggest X'.

1.2. Building the system of linear equations

The expression $\boldsymbol{x}'^T F \boldsymbol{x}$ is linear with respect to components of F

$$\boldsymbol{x}'^T F \boldsymbol{x} = \begin{bmatrix} x'_1 & x'_2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = $$

$$= \operatorname{vec}\left(\boldsymbol{x}'\boldsymbol{x}^T\right)^T \operatorname{vec}(F),$$

where vec is a row-major matrix flattening. Now we can write the system of equations

$$\boldsymbol{x}_i'^T F \boldsymbol{x}_i = 0, \quad i = \overline{1,7}$$

as a system of linear equations

$$\begin{bmatrix} \operatorname{vec}\left(\boldsymbol{x}'_1\boldsymbol{x}_1^T\right)^T \\ \vdots \\ \operatorname{vec}\left(\boldsymbol{x}'_7\boldsymbol{x}_7^T\right)^T \end{bmatrix} \operatorname{vec}(F) = 0.$$

The system contains seven equations and nine unknowns, so the solution is a point in two-dimensional nullspace of the matrix of coefficients

$$F = \alpha_1 F_1 + \alpha_2 F_2, \quad \alpha_1^2 + \alpha_2^2 > 0, \tag{2}$$

where F_1 and F_2 are orthogonal elements of the nullspace gathered back into matrix view.

1.3. Building the cubic equation

We need the fundamental matrix to have rank 2. One of the methods of enforcing this constraint is to find such α_1 and α_2 that the final matrix has zero determinant. If $\det(F_1) = 0$ (or $\det(F_2) = 0$), it's the result of the current iteration. Otherwise, we have an equation

$$\det(F_1 + \alpha F_2) = 0. \tag{3}$$

[a] stryitg@gmail.com
[b] valeriy.krygin@gmail.com

2. Expressing the cubic equation

Theorem 1. *Let* A *and* B *be* 3×3 *nonsingular matrices, then*

$$\det(A+B) = \det(A) + \det(A)\operatorname{tr}\left(BA^{-1}\right) + \det(B)\operatorname{tr}\left(AB^{-1}\right) + \det(B)$$

Proof. Let ε_{ijk} be the Levi-Civita symbol. Then

$$\det(A) = \sum_{i,j,k} \varepsilon_{ijk} a_{i1} a_{j2} a_{k3}$$

and

$$\begin{aligned}
&\det(A+B) = \\
&= \sum_{i,j,k} \varepsilon_{ijk} (a_{i1}+b_{i1})(a_{j2}+b_{j2})(a_{k3}+b_{k3}) = \\
&= \sum_{i,j,k} \varepsilon_{ijk} a_{i1} a_{j2} a_{k3} + \sum_{i,j,k} \varepsilon_{ijk} a_{i1} a_{j2} b_{k3} + \\
&+ \sum_{i,j,k} \varepsilon_{ijk} a_{i1} b_{j2} a_{k3} + \sum_{i,j,k} \varepsilon_{ijk} a_{i1} b_{j2} b_{k3} + \\
&+ \sum_{i,j,k} \varepsilon_{ijk} b_{i1} a_{j2} a_{k3} + \sum_{i,j,k} \varepsilon_{ijk} b_{i1} a_{j2} b_{k3} + \\
&+ \sum_{i,j,k} \varepsilon_{ijk} b_{i1} b_{j2} a_{k3} + \sum_{i,j,k} \varepsilon_{ijk} b_{i1} b_{j2} b_{k3}.
\end{aligned}$$

Let C^A_{ij}, C^B_{ij} be the elements of a cofactor matrix of A and B respectivly. Then the last sum could be written as

$$\begin{aligned}
&\det(A) + \sum_k C^A_{k3} b_{k3} + \sum_j C^A_{j2} b_{j2} + \sum_i C^B_{i1} a_{i1} + \\
&+ \sum_i C^A_{i1} b_{i1} + \sum_j C^B_{j2} a_{j2} + \sum_k C^B_{k3} a_{k3} + \det(B) = \\
&= \det(A) + \sum_{ij} C^A_{ij} b_{ij} + \sum_{ij} C^B_{ij} a_{ij} + \det(B).
\end{aligned}$$

We have

$$\begin{aligned}
&\det(A+B) = \\
&= \det(A) + \sum_{ij} C^A_{ij} b_{ij} + \sum_{ij} C^B_{ij} a_{ij} + \det(B).
\end{aligned} \tag{4}$$

Considering that $A^{-T} = \frac{1}{\det(A)} C^A$ and denoting elements of the A^{-1} as A^{-1}_{ij}, the last sum simplifies to

$$\begin{aligned}
&\det(A) + \det(A) \sum_{ij} A^{-1}_{ji} b_{ij} + \\
&+ \det(B) \sum_{ij} B^{-1}_{ji} a_{ij} + \det(B) = \\
&= \det(A) + \det(A)\operatorname{tr}\left(BA^{-1}\right) + \\
&+ \det(B)\operatorname{tr}\left(AB^{-1}\right) + \det(B).
\end{aligned}$$

□

Since F_1 and F_2 are rank 3 matrices, we can express the (3) as

$$\begin{aligned}
&\det(F_1+\alpha F_2) = \\
&= \det(F_1) + \det(F_1)\operatorname{tr}\left(\alpha F_2 F_1^{-1}\right) + \\
&+ \det(\alpha F_2)\operatorname{tr}\left(F_1 (\alpha F_2)^{-1}\right) + \det(\alpha F_2).
\end{aligned}$$

Since F_2 is a 3×3 matrix,

$$\det(\alpha F_2) = \alpha^3 \det(F_2),$$

the *compact form* for the polynomial of α is

$$\begin{aligned}
&\det(F_1+\alpha F_2) = \\
&= \det(F_1) + \alpha\det(F_1)\operatorname{tr}\left(F_2 F_1^{-1}\right) + \\
&+ \alpha^2 \det(F_2)\operatorname{tr}\left(F_1 F_2^{-1}\right) + \alpha^3 \det(F_2).
\end{aligned} \tag{5}$$

This equation is compact but its straightforward implementation may have performance issues. During the proof of the theorem 1, we've got a valuable intermediate result (4), which is still compact enough but is less computationally consumptive

$$\begin{aligned}
&\det(F_1+\alpha F_2) = \\
&= \det(F_1) + \alpha \sum_{ij} C^{F_1}_{ij} f_{2ij} + \\
&+ \alpha^2 \sum_{ij} C^{F_2}_{ij} f_{1ij} + \alpha^3 \det(F_1).
\end{aligned} \tag{6}$$

Conclusion

The current work presents a compact form (5) and a semi-compact form (6) (which is more desirable for software implementation) of the cubic equation, appearing in the 7-point algorithm. The compact form has multiple advantages: it's not hard to remember, its derivation refreshes linear algebra knowledge, it's simple in implementation and debug.

References

1. Hartley R. I., Zisserman A. Multiple View Geometry in Computer Vision. — Second edition. — Cambridge University Press, ISBN: 0521540518, 2004.
2. Szeliski Richard. Computer vision algorithms and applications. — London; New York : Springer, 2011. — Access mode: `http://dx.doi.org/10.1007/978-1-84882-935-0`.
3. Grayson Daniel R., Stillman Michael E. Macaulay2, a software system for research in algebraic geometry. — Available at `http://www.math.uiuc.edu/Macaulay2/`.
4. SymPy: symbolic computing in Python / Aaron Meurer, Christopher P. Smith, Mateusz Paprocki et al. // PeerJ Computer Science. — 2017. — Jan. — Vol. 3. — P. e103. — Access mode: `https://doi.org/10.7717/peerj-cs.103`.
5. Nister David. An efficient solution to the five-point relative pose problem // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2004. — Vol. 26, no. 6. — P. 756–770.
6. Kukelova Z., Bujnak M., Pajdla T. Polynomial Eigenvalue Solutions to the 5-pt and 6-pt Relative Pose Problems // Proceedings of the British Machine Vision Conference. — BMVA Press, 2008. — P. 56.1–56.10. — doi:10.5244/C.22.56.
7. Lowe David G. Object Recognition from Local Scale-Invariant Features // Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2. — ICCV '99. — USA : IEEE Computer Society, 1999. — P. 1150.