Solution Space Analysis of Essential Matrix based on Algebraic Error Minimization

Gaku Nakano[1]

NEC Corporation, Japan
g-nakano@nec.com

Abstract. This paper reports on a solution space analysis of the essential matrix based on algebraic error minimization. Although it has been known since 1988 that an essential matrix has at most 10 real solutions for five-point pairs, the number of solutions in the least-squares case has not been explored. We first derive that the Karush–Kuhn–Tucker conditions of algebraic errors satisfying the Demazure constraints can be represented by a system of polynomial equations without Lagrange multipliers. Then, using computer algebra software, we reveal that the simultaneous equation has at most 220 real solutions, which can be obtained by the Gauss-Newton method, Gröbner basis, and homotopy continuation. Through experiments on synthetic and real data, we quantitatively evaluate the convergence of the proposed and the existing methods to globally optimal solutions. Finally, we visualize a spatial distribution of the global and local minima in 3D space.

Keywords: Essential matrix, Two-view geometry, Structure-from-Motion, System of polynomial equations, Gröbner basis, Homotopy continuation

1 Introduction

Two-view geometry is a fundamental problem in computer vision for reconstructing the 3D shape of objects and camera motions using two images [17]. It has been widely used in various applications such as V-SLAM [5,34], novel view synthesis [2,33], visual localization [42,47], city-scale Structure-from-Motion [1,49], and human interaction understanding [20]. Particularly, the most basic form shown in Fig. 1, finding the relative motion between two calibrated cameras, is called the essential matrix estimation. An essential matrix is a rank-deficient 3×3 matrix of which two singular values are equal. Because of this constraint, the essential matrix estimation has been studied for decades.

It has been well known since 1913 that an essential matrix can be obtained from at least five point correspondences [24][1]. In 1988, Demazure [10] proved that there are at most 10 solutions in the five-point case. However, 6- [38,39], 7- [17], and 8-point [17] algorithms had been used until a practical 5-point algorithm was first developed by Nister [37] in 2004. The 5-point method was extended to various problems, *e.g.* 3-point with known gravity direction [13,21,36], 5-point with

[1] English translation: https://arxiv.org/abs/1801.01454

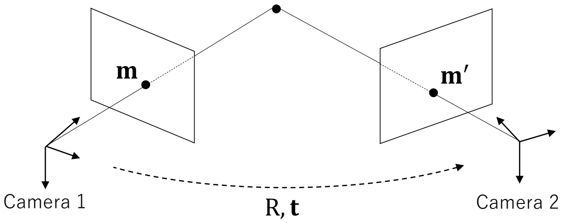


Fig. 1. Relative pose estimation problem.

a small motion [44], 6- and 7-point for uncalibrated cameras [19,25,28,40]. These methods, called minimal solvers, are usually incorporated with RANSAC [12] to remove outliers existing in the input point matches.

On the other hand, the essential matrix estimation for the least-squares case, where given more than five points, has been thought to be a more challenging problem than the minimal case due to the aforementioned constraints of the essential matrix. Many efforts have been extensively devoted to developing a method for solving the constrained least-squares, which can be classified into two categories: locally optimal methods [11,18,23,32] and globally optimal methods [4,7,16,48]. However, unlike the 5-point minimal case, the solution space of the least-square case has not been well investigated in the past. There are still open questions; *How many real solutions are there? Is a solution obtained by a local method the global optimum or a local minimum? Does a global method surely return the global optimum?*

To answer those questions, we analyze the solution space of the essential matrix estimation in the sense of the algebraic error minimization in this paper. We first derive the new KKT (Karush–Kuhn–Tucker) conditions as a system of polynomial equations without Lagrange multipliers. Utilizing an algebraic geometry software, we reveal the number of the solutions in the least-square case. Then, we propose three direct methods for solving the new KKT conditions. Finally, we conduct synthetic and real data evaluations to validate whether the proposed and existing methods provide globally optimal solutions or not.

2 Theoretical Background

The essential matrix $\mathtt{E}$ is a 3×3 matrix that gives a constraint on the relative motion and a pair of corresponding point between two images. Let $\mathtt{R} \in SO(3)$ and $\mathbf{t}$ be the relative rotation matrix and the translation vector, respectively, and $\{\mathbf{m} \leftrightarrow \mathbf{m}'\}$ an image point correspondence in 3×1 homogeneous coordinates between two calibrated cameras.The essential matrix satisfies

$$\mathbf{m}'^{\mathsf{T}}\mathtt{E}\mathbf{m} = 0, \tag{1}$$

where

$$\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}, \quad [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}. \tag{2}$$

Since Eq. (1) holds up to any scale of $\mathbf{t}$, the essential matrix $\mathtt{E}$ has 5 DoF (Degrees of Freedom). This property is equivalent to that two of its singular values are equal, and the third is zero:

$$\mathtt{E} = \mathtt{U}\,\mathrm{diag}(1,1,0)\;\mathtt{V}^\mathsf{T}. \tag{3}$$

Demazure [10] expressed the above constraint in a matrix form

$$2\mathtt{E}\mathtt{E}^\mathsf{T}\mathtt{E} - \mathrm{tr}\big(\mathtt{E}\mathtt{E}^\mathsf{T}\big)\mathtt{E} = \mathbf{0}_{3\times 3}. \tag{4}$$

Given more than five point pairs, the essential matrix estimation can be formulated as a non-linear constrained problem

$$\begin{array}{ll} \min\limits_{\mathtt{E}} & \mathbf{e}^\mathsf{T}\mathtt{M}\mathbf{e} \\ \text{s.t.} & \|\mathbf{e}\|^2 = 1,\; 2\mathtt{E}\mathtt{E}^\mathsf{T}\mathtt{E} - \mathrm{tr}\big(\mathtt{E}\mathtt{E}^\mathsf{T}\big)\mathtt{E} = \mathbf{0}_{3\times 3}, \end{array} \tag{5}$$

where $\mathtt{M} = \sum_i (\mathbf{m}'_i \otimes \mathbf{m}_i)(\mathbf{m}'_i \otimes \mathbf{m}_i)^\mathsf{T}$ and $\mathbf{e}$ is a 9-dimensional vector representation of $\mathtt{E}$. The L2 norm constraint is required to avoid the trivial solution $\mathbf{e} = \mathbf{0}$.

The objective function of Eq. (5), $\mathbf{e}^\mathsf{T}\mathtt{M}\mathbf{e}$, is called the algebraic error, which is not a geometrically meaningful metric but widely used in computer vision due to its simple form. In this paper, we focus on discussing approaches for finding the global optimum of the algebraic error minimization.

3 Previous Work

In this section, we briefly introduce the existing methods for estimating an essential matrix in the sense of least-squares.

Eight-point DLT solver [16]. Ignoring the Demazure constraint, we can reformulate Eq. (5) by a single constrained problem, *i.e.*

$$\min_{\mathtt{E}} \mathbf{e}^\mathsf{T}\mathtt{M}\mathbf{e}, \quad \text{s.t. } \|\mathbf{e}\|^2 = 1. \tag{6}$$

Using the method of Lagrange multiplier, the solution of the above equation can be written by a linear equation:

$$\mathtt{M}\mathbf{e} = \lambda\mathbf{e}, \tag{7}$$

where λ is a Lagrange multiplier and also an eigenvalue of $\mathtt{M}$. Since $\mathbf{e}^\mathsf{T}\mathtt{M}\mathbf{e} = \lambda\mathbf{e}^\mathsf{T}\mathbf{e} = \lambda$, the optimal solution $\mathbf{e}$ can be obtained by an eigenvector corresponding to the smallest eigenvalue $\lambda_{\min}$. The optimum of Eq. (6) may not be an essential matrix due to the lack of the Demazure constraint. To ensure the constraint, the singular value correction

$$\mathtt{E} \leftarrow \mathtt{U}\,\mathrm{diag}(1,1,0)\;\mathtt{V}^\mathsf{T} \tag{8}$$

is typically applied as a post-processing. The above procedure is called the DLT (Direct Linear Transform) method and has been regarded as the standard approach for decades because of its simpleness.

Local parameterization solver [18]. A straightforward way of satisfying the Demazure constraint is to parameterize an essential matrix by 5 DoF. Two orthogonal matrices $\mathtt{U}$ and $\mathtt{V}$ can be represented by the exponential map, *i.e.*

$$\mathtt{U} = \exp([\mathbf{x}_1]_\times), \quad \mathtt{V} = \exp([\mathbf{x}_2]_\times), \tag{9}$$

where $\mathbf{x}_1 = \frac{1}{\sqrt{2}}[x_1, x_2, \frac{x_3}{\sqrt{2}}]^\mathsf{T}$ and $\mathbf{x}_2 = \frac{1}{\sqrt{2}}[x_4, x_5, -\frac{x_3}{\sqrt{2}}]^\mathsf{T}$. Note that $\mathtt{U}$ and $\mathtt{V}$ share a common variable x_3. An essential matrix is parameterized by

$$\mathtt{E}(\mathbf{x}_1, \mathbf{x}_2) = \mathtt{U}(\mathbf{x}_1)\,\mathrm{diag}(1,1,0)\;\mathtt{V}(\mathbf{x}_2)^\mathsf{T}. \tag{10}$$

Thus, Eq. (5) can be rewritten in an unconstrained problem

$$\min_{x_1,\ldots,x_5} \; \mathbf{e}(\mathbf{x}_1, \mathbf{x}_2)^\mathsf{T}\mathtt{M}\,\mathbf{e}(\mathbf{x}_1, \mathbf{x}_2). \tag{11}$$

Eigenvalue minimization solver [23]. Another way to satisfy the Demazure constraint is to find a rotation that minimizes the smallest eigenvalue $\lambda_{\min}$. Using Eq. (2), the vector expression $\mathbf{e}$ can be written by

$$\mathbf{e} = \mathtt{A}\mathbf{t}, \quad \mathtt{A} = \begin{bmatrix} \mathbf{0} & -\mathbf{r}_3 & \mathbf{r}_2 \\ \mathbf{r}_3 & \mathbf{0} & -\mathbf{r}_1 \\ -\mathbf{r}_2 & \mathbf{r}_1 & \mathbf{0} \end{bmatrix}, \tag{12}$$

where $\mathbf{r}_k$ denotes the k-th row of $\mathtt{R}$. Equation (5) can be reformulated by

$$\begin{aligned} \min_{\mathtt{R},\mathbf{t}} \quad & \mathbf{t}^\mathsf{T}\mathtt{A}^\mathsf{T}\mathtt{M}\mathtt{A}\mathbf{t} \\ \text{s.t.} \quad & \|\mathbf{t}\|^2 = 1, \quad \mathtt{R} \in SO(3). \end{aligned} \tag{13}$$

The optimal translation $\mathbf{t}$ can be given in the same way as the DLT method by the eigenvector associated with the smallest eigenvalue:

$$\mathtt{A}^\mathsf{T}\mathtt{M}\mathtt{A}\mathbf{t} = \lambda_{\min}\mathbf{t}. \tag{14}$$

The 3×3 matrix $\mathtt{A}^\mathsf{T}\mathtt{M}\mathtt{A}$ consists of unknown rotation of 3 DoF. Using the Cayley transform, $\mathtt{R}(\mathbf{x}) = (\mathtt{I} - [\mathbf{x}]_\times)(\mathtt{I} + [\mathbf{x}]_\times)^{-1}$ with $\mathbf{x} = [x_1, x_2, x_3]^\mathsf{T}$, Eq. (13) can be further rewritten by an unconstrained problem

$$\min_{x_1,x_2,x_3} \; \lambda_{\min}(\mathtt{R}(\mathbf{x})). \tag{15}$$

SDP-relaxation solver [48]. The above two solvers are locally optimal methods that require a good initial guess to converge to the global optimum. To avoid the convergence to a local minimum, a globally optimal solution was proposed using SDP (Semi-Definite Programming) relaxation. Using the orthogonality condition $\mathtt{R}^\mathsf{T}\mathtt{R} = \mathtt{I}$, Eq. (5) can be rewritten by

$$\begin{aligned} \min_{\mathbf{e},\mathbf{t}} \quad & \mathbf{e}^\mathsf{T}\mathtt{M}\mathbf{e} \\ \text{s.t.} \quad & \mathtt{E}\mathtt{E}^\mathsf{T} = [\mathbf{t}]_\times[\mathbf{t}]_\times^\mathsf{T}, \quad \|\mathbf{t}\|^2 = 1. \end{aligned} \tag{16}$$

The objective and constraints are quadratic with respect to $\mathtt{E}$ and $\mathbf{t}$, therefore, the above equation is a non-convex QCQP (Quadratically Constrained Quadratic Program). Letting $\mathtt{X} = \begin{bmatrix} \mathbf{e} \\ \mathbf{t} \end{bmatrix} [\mathbf{e}^\mathsf{T} \mathbf{t}^\mathsf{T}]$, we can write the non-convex QCQP by

$$\begin{aligned} \min_{\mathtt{X}} \quad & \operatorname{tr}\left(\begin{bmatrix} \mathtt{M} & \\ & \mathbf{0}_{3\times 3} \end{bmatrix} \mathtt{X} \right) \\ \text{s.t.} \quad & \mathtt{X} \succeq 0, \ \operatorname{tr}(\mathtt{B}_i \mathtt{X}) = c_i, \ 1 \le i \le 7, \end{aligned} \tag{17}$$

where $\mathtt{X} \succeq 0$ represents that $\mathtt{X}$ is positive semi-definite, and $\mathtt{B}_i$ and c_i are coefficients corresponding to the quadratic constraints of Eq. (16). The rank constraint $\operatorname{rank}(\mathtt{X}) = 1$ is dropped due to NP-hard in Eq. (17), which is called SDP-relaxation. Owing to the relaxation, Eq. (17) becomes a convex optimization that the global optimum can always be obtainable. It should be noted that $\mathtt{E}$ and $\mathbf{t}$ recovered from $\mathtt{X}$ do not strictly satisfy the original constraints in Eq. (16) because generally $\operatorname{rank}(\mathtt{X}) > 1$ even for the global optimum of Eq. (17).

4 Analysis of Solution Space

In this section, we first formulate the new KKT conditions, or the first-order optimality conditions, without Lagrange multipliers. Our derivation is inspired by the optimal PnP method [35]. The new KKT conditions are represented as a system of polynomials in an essential matrix. Then, we reveal the number of the solutions of the polynomial system by using Gröbner basis. Finally, we introduce three direct methods for finding the solutions of the new KKT conditions.

4.1 New KKT conditions

The Lagrangian function of Eq. (5) can be written by

$$L = \frac{1}{2}\mathbf{e}^\mathsf{T}\mathtt{M}\mathbf{e} + \frac{\mu}{2}\left(1 - \operatorname{tr}\left(\mathtt{E}\mathtt{E}^\mathsf{T}\right)\right) + \operatorname{tr}\left(\mathtt{S}\left(2\mathtt{E}\mathtt{E}^\mathsf{T}\mathtt{E} - \operatorname{tr}\left(\mathtt{E}\mathtt{E}^\mathsf{T}\right)\mathtt{E}\right)\right), \tag{18}$$

where μ and $\mathtt{S}$ are a Lagrange multiplier and a 3×3 matrix consisting of nine Lagrange multipliers, respectively. Note that $\operatorname{tr}\left(\mathtt{E}\mathtt{E}^\mathsf{T}\right) = \|\mathbf{e}\|^2$ and the multiplier $1/2$ is merely for convenience. The gradient of L with respect to $\mathtt{E}$ is given by

$$\begin{aligned} \frac{\partial L}{\partial \mathtt{E}} = & \operatorname{mat}(\mathtt{M}\mathbf{e}) - \mu\mathtt{E} + 2\mathtt{S}^\mathsf{T}\mathtt{E}^\mathsf{T}\mathtt{E} + 2\mathtt{E}\mathtt{S}\mathtt{E} \\ & + 2\mathtt{E}\mathtt{E}^\mathsf{T}\mathtt{S}^\mathsf{T} - \operatorname{tr}\left(\mathtt{E}\mathtt{E}^\mathsf{T}\right)\mathtt{S}^\mathsf{T} - 2\operatorname{tr}(\mathtt{S}\mathtt{E})\mathtt{E} = \mathbf{0}_{3\times 3}, \end{aligned} \tag{19}$$

where $\operatorname{mat}(\,\cdot\,)$ is a linear operator converting a 9-dimensional vector to a 3×3 matrix: $\mathbb{R}^9 \to \mathbb{R}^{3\times 3}$. Multiplying $\mathtt{E}^\mathsf{T}$ from left and right to Eq. (19), we obtain the following two equations:

$$\begin{aligned}\mathtt{E}^{\mathsf{T}}\frac{\partial L}{\partial \mathtt{E}} =& \mathtt{E}^{\mathsf{T}}\operatorname{mat}(\mathtt{M}\mathbf{e}) - s\mathtt{E}^{\mathsf{T}}\mathtt{E} + 2\mathtt{E}^{\mathsf{T}}(\mathtt{S}^{\mathsf{T}}\mathtt{E}^{\mathsf{T}} + \mathtt{E}\mathtt{S})\mathtt{E} \\ &+ (2\mathtt{E}\mathtt{E}^{\mathsf{T}}\mathtt{E} - \operatorname{tr}(\mathtt{E}\mathtt{E}^{\mathsf{T}})\mathtt{E})^{\mathsf{T}}\mathtt{S}^{\mathsf{T}} = \mathbf{0}_{3\times 3},\end{aligned} \tag{20a}$$

$$\begin{aligned}\frac{\partial L}{\partial \mathtt{E}}\mathtt{E}^{\mathsf{T}} =& \operatorname{mat}(\mathtt{M}\mathbf{e})\mathtt{E}^{\mathsf{T}} - s\mathtt{E}\mathtt{E}^{\mathsf{T}} + 2\mathtt{E}(\mathtt{S}\mathtt{E} + \mathtt{E}^{\mathsf{T}}\mathtt{S}^{\mathsf{T}})\mathtt{E}^{\mathsf{T}} \\ &+ \mathtt{S}^{\mathsf{T}}(2\mathtt{E}\mathtt{E}^{\mathsf{T}}\mathtt{E} - \operatorname{tr}(\mathtt{E}\mathtt{E}^{\mathsf{T}})\mathtt{E})^{\mathsf{T}} = \mathbf{0}_{3\times 3},\end{aligned} \tag{20b}$$

where $s = \mu + 2\operatorname{tr}(\mathtt{S}\mathtt{E})$. Using $2\mathtt{E}\mathtt{E}^{\mathsf{T}}\mathtt{E} - \operatorname{tr}(\mathtt{E}\mathtt{E}^{\mathsf{T}})\mathtt{E} = \mathbf{0}$, we can rewrite Eqs. (20a) and (20b) by

$$\begin{aligned}\mathtt{E}^{\mathsf{T}}\operatorname{mat}(\mathtt{M}\mathbf{e}) &= s\mathtt{E}^{\mathsf{T}}\mathtt{E} - 2\mathtt{E}^{\mathsf{T}}\left(\mathtt{S}^{\mathsf{T}}\mathtt{E}^{\mathsf{T}} + \mathtt{E}\mathtt{S}\right)\mathtt{E} \\ &= \mathtt{E}^{\mathsf{T}}\left(s\mathtt{I} - 2(\mathtt{E}\mathtt{S})^{\mathsf{T}} - 2\mathtt{E}\mathtt{S}\right)\mathtt{E},\end{aligned} \tag{21a}$$

$$\begin{aligned}\operatorname{mat}(\mathtt{M}\mathbf{e})\mathtt{E}^{\mathsf{T}} &= s\mathtt{E}\mathtt{E}^{\mathsf{T}} - 2\mathtt{E}\left(\mathtt{S}\mathtt{E} + \mathtt{E}^{\mathsf{T}}\mathtt{S}^{\mathsf{T}}\right)\mathtt{E}^{\mathsf{T}} \\ &= \mathtt{E}\left(s\mathtt{I} - 2\mathtt{S}\mathtt{E} - 2(\mathtt{S}\mathtt{E})^{\mathsf{T}}\right)\mathtt{E}^{\mathsf{T}}.\end{aligned} \tag{21b}$$

The right-hand side of Eqs. (21a) and (21b) is a 3×3 symmetric matrix. Thus, the left-hand side, $\mathtt{E}^{\mathsf{T}}\operatorname{mat}(\mathtt{M}\mathbf{e})$ and $\operatorname{mat}(\mathtt{M}\mathbf{e})\mathtt{E}^{\mathsf{T}}$, also must be a symmetric matrix. Defining $\mathtt{P} = \mathtt{E}^{\mathsf{T}}\operatorname{mat}(\mathtt{M}\mathbf{e})$ and $\mathtt{Q} = \operatorname{mat}(\mathtt{M}\mathbf{e})\mathtt{E}^{\mathsf{T}}$, we can formulate the symmetry constraint as the following six polynomial equations:

$$\begin{aligned}P_{12} - P_{21} = 0, \quad P_{13} - P_{31} = 0, \quad P_{32} - P_{23} = 0, \\ Q_{12} - Q_{21} = 0, \quad Q_{13} - Q_{31} = 0, \quad Q_{32} - Q_{23} = 0,\end{aligned} \tag{22}$$

where P_{ij} and Q_{ij} denote the (i, j) elements of $\mathtt{P}$ and $\mathtt{Q}$, respectively. Finally, the new KKT conditions can be given by

$$\begin{gathered}P_{ij} - P_{ji} = 0,\ Q_{ij} - Q_{ji} = 0,\ \forall i, j \in \{1, 2, 3\}, \\ \operatorname{tr}(\mathtt{E}\mathtt{E}^{\mathsf{T}}) = 1,\ 2\mathtt{E}\mathtt{E}^{\mathsf{T}}\mathtt{E} - \operatorname{tr}(\mathtt{E}\mathtt{E}^{\mathsf{T}})\mathtt{E} = \mathbf{0}_{3\times 3}.\end{gathered} \tag{23}$$

Equation (23) is a polynomial system in 16 equations with nine variables of $\mathtt{E}$.

4.2 Solution space

We can use an algebraic geometry software Macaulay2 [15][2] to analyze the solution space of Eq. (23). Macaulay2 computes the Gröbner basis of a polynomial system in a finite prime field, and we have found that Eq. (23) have 440 solutions. Due to the L2 norm constraint, $\operatorname{tr}(\mathtt{E}\mathtt{E}^{\mathsf{T}}) = 1$, there is a 2-fold ambiguity in the 440 solutions. In other words, $\mathtt{E}$ and $-\mathtt{E}$ are exactly the same solutions. To remove the ambiguity, we can set one of $\mathtt{E}$ as a scalar instead of $\operatorname{tr}(\mathtt{E}\mathtt{E}^{\mathsf{T}}) = 1$. For example, $E_{13} = 1$ leads to 220 solutions. Readers may refer to [26] for more details of how to use Macaulay2 for geometric problems in computer vision.

[2] http://www.math.uiuc.edu/Macaulay2/

```
% m1, m2: 3xN set of homogeneous 2D points
% E_ini : Initial guess of essential matrix
% E_opt : Optimized essential matrix
function E_opt = Emat_KKTGN(m1, m2, E_ini)
    M     = [m2(1,:)’.*m1’, m2(2,:)’.*m1’, m2(3,:)’.*m1’];
    E_opt = fsolve(@(x)KKTeqs(x,M’*M), E_ini);
end
function eqs = KKTeqs(E, M)
    e     = reshape(E’,9,1);
    matMe = reshape(M*e, 3, 3)’;
    P     = E’*matMe;
    Q     = matMe*E’;
    ceq1  = trace(E*E’) - 1;
    ceq2  = 2*(E*E’)*E - E;
    eqs   = [P(1,2)-P(2,1); P(1,3)-P(3,1); P(2,3)-P(3,2);
             Q(1,2)-Q(2,1); Q(1,3)-Q(3,1); Q(2,3)-Q(3,2);
             ceq1; ceq2(:)];
end
```

Fig. 2. 15-line MATLAB code for solving Eq. (23) by the Gauss-Newton method.

4.3 Direct least-squares solvers

We propose three solvers to directly find an essential matrix that satisfies Eq. (23).

Gauss-Newton solver. A typical approach to finding a single solution to a polynomial system is to apply the Gauss-Newton method with an initial guess. The Gauss-Newton method is easy to implement and is often already built-in as a subroutine in optimization libraries. Figure 2 shows a MATLAB example with `fsolve` function written in 15-lines of code. Note that a single solution satisfying the KKT conditions is a local minimum, which is not guaranteed to be the global optimum.

Gröbner basis solver. As a method for finding all solutions of multivariate polynomials, Gröbner basis method has been widely used in computer vision for the last decade [27,29]. We applied an automatic generator [29] on Eq. (23) with replacing $\operatorname{tr}\left(\mathtt{EE}^\mathsf{T}\right) = 1$ by $E_{13} = 1$ and obtained a Gröbner basis solver that returns 220 solutions. Unfortunately, the generated solver is not numerically stable due to a large elimination template of size 5149 × 5369. We have found that the solver does not return meaningful solutions that satisfy Eq. (23).

Homotopy continuation solver. Another approach for obtaining all solutions is to use a homotopy continuation method, a classical but useful tool widely known in numerical algebraic geometry [30]. Several software packages are publicly available, such as Bertini [3][3], PHCPack [45][4], Hom4PS [6][5]. We will use Bertini for experiments in Sec. 5 because it is the fastest among the three software for solving Eq. (23).

[3] https://bertini.nd.edu/

[4] https://github.com/janverschelde/PHCpack

[5] http://www.hom4ps3.org/

5 Experiment

We report experimental results on both synthetic and real data in this section. All experiments were conducted on a PC with Core i7-7920X.

5.1 Implementation

We have implemented the existing methods (Sec. 3) and the new direct methods (Sec. 4.3) in MATLAB. The details are as follows:

DLT The standard eight-point DLT algorithm [17] shown in Eqs. (6) to (8).

Expm-GN Local parameterization method, Eq. (11), using the exponential map with a Gauss-Newton optimization. Readers may refer to Sec. 4 in [18] for the implementation details.

minEig Eigenvalue minimization method [23] implemented in OpenGV [22][6], a C++ library for camera pose estimation problems. The Jacobian of Eq. (15) is supplied using numerical differentiation to avoid complex arithmetic (See Sec. 3.1 in [23]). We selected a rotation matrix from the initial $\mathtt{E}$ by checking the orientation of the projective depth to be positive [9].

QCQP SDP-relaxation method for a non-convex QCQP [48] shown in Eq. (17). We have rewritten the original C++ code[7] in MATLAB using SDPA-M 7.3.9 [46], a MATLAB wrapper for an SDP solver, SDPA[8].

Bertini Direct method for obtaining all real roots of Eq. (23) using Bertini 1.6 [3], a software for solving polynomial systems based on homotopy continuation. Since $P_{ij} - P_{ji} = 0$ and $Q_{ij} - Q_{ji} = 0$ in Eq. (23) hold up to scale, we set $E_{13} = 1$ instead of $\mathrm{tr}\left(\mathtt{E}\mathtt{E}^\top\right) = 1$. Moreover, a random 3×3 rotation was multiplied to the point correspondences to avoid a numerical degeneracy, $E_{13} = 0$. We selected the global optimum from all real solutions that gives the smallest algebraic error, $\mathbf{e}^\top \mathtt{M} \mathbf{e}$.

KKT-GN Gauss-Newton-based method for solving Eq. (23). A code example is shown in Fig. 2, but we used a simple Gauss-Newton implementation with the analytic Jacobian for efficiency instead of `fsolve` function.

BA Bundle adjustment minimizing reprojection error [43]. In addition to a rotation matrix and a translation vector, 3D points are jointly optimized. This is the only one method minimizing a geometrically meaningful error in this experiment.

For quantitative evaluations, we measured three criteria: rotation error $E_{\mathtt{R}}$, translation error $E_{\mathbf{t}}$, and the first-order optimality E_{KKT}. The three metrics were computed by

$$E_{\mathtt{R}} = \frac{180^\circ}{\pi} \cos^{-1}\left(\frac{\mathrm{tr}\left(\mathtt{R}_{\mathrm{gt}}^\top \mathtt{R}_{\mathrm{est}}\right) - 1}{2}\right), \quad E_{\mathbf{t}} = \frac{180^\circ}{\pi} \cos^{-1}\left(\frac{\mathbf{t}_{\mathrm{gt}}^\top \mathbf{t}_{\mathrm{est}}}{\|\mathbf{t}_{\mathrm{gt}}\| \, \|\mathbf{t}_{\mathrm{est}}\|}\right),$$
$$E_{\mathrm{KKT}} = \frac{\max(|P_{12} - P_{21}|, \ldots, |Q_{32} - Q_{23}|)}{\|\mathtt{M}\|_F \, \|\mathtt{E}\|_F}, \qquad (24)$$

[6] http://laurentkneip.github.io/opengv

[7] https://github.com/jizhaox/npt-pose

[8] http://sdpa.sourceforge.net/

where $\|\cdot\|_F$ represents the Frobenius norm of a matrix. E_{KKT} represents a convergence to a local minimum under 5 DoF; an estimated $\mathtt{E}$ having a sufficiently low E_{KKT} is a valid local minimum that can be decomposed into $\mathtt{R}$ and $\mathbf{t}$.

5.2 Synthetic data

We generated synthetic scenes to quantitatively compare the methods. We set the first camera as the origin of the world coordinates, *i.e.* the rotation matrix as an 3×3 identity matrix and the translation as a zero vector. The second camera was randomly positioned within $[\pm1, \pm1, \pm2.5]$. The rotation matrix of the second camera was given by random Euler angles in the range of $[10°, \pm60°, \pm90°]$. The image resolution is 640×480, and both cameras have a focal length of 800 pixels and the optical center at $[320, 240]$. To generate 3D points, we first uniformly determined 2D points in the normalized image plane of the first camera and then randomly set their depth from 5 to 10. We projected the 3D points onto the two cameras and obtained 2D point correspondences. DLT was used as the initial guess for the iterative methods, *i.e.* Expm-GN, minEig, KKT-GN, and BA.

Accuracy w.r.t. image noise We evaluated the robustness against image noise. The number of the 2D points was fixed to 100, and zero-mean Gaussian noise was added to the points with the standard deviation $0.5 \le \sigma \le 5$ pixels. Figure 3 summarizes the average estimation error of 1000 independent trials for each noise level. Bertini gives the best performance among the algebraic error minimization methods. This result shows that we can obtain the global optimum by solving Eq. (23). Expm-GN and minEig are comparable each other. QCQP gives almost the same accuracy with DLT for rotation estimation. KKT-GN performs well for small noise levels $\sigma \le 3$ but becomes worse than DLT as the noise increases whereas KKT-GN is initialized by DLT. According to the result of E_{KKT}, KKT-GN and QCQP get trapped by a bad local minimum specially for high noise levels. Feature points are generally detected on subpixel accuracy, *e.g.* $\sigma \le 2$, therefore, Expm-GN, minEig, and KKT-GN are expected to have a similar performance in practice.

Accuracy w.r.t. number of points We compared the methods by varying the number of the points from 10 to 100. The noise level was fixed to $\sigma = 2$. The average error over 1000 independent trials is shown in Fig. 4. Expm-GN and minEig have similar performance with Bertini, which provides the global optimum close to BA, while E_{KKT} of minEig is 10^{-8} that does not reach to the machine epsilon like Expm-GN and Bertini. This observation suggests that $E_{\mathrm{KKT}} < 10^{-8}$ is sufficiently low to obtain a local minimum with preserving the decomposability of an essential matrix. For ≤ 25 points, QCQP and KKT-GN are less accurate than the other methods even though their E_{KKT} is less than 10^{-8}. It is inferred that KKT-GN and QCQP converge to a bad local minimum for few point correspondences. QCQP is slightly inferior to KKT-GN in terms of $E_{\mathtt{R}}$ even for > 25 points, therefore, KKT-GN can be comparable to Expm-GN and minEig if many points are available.

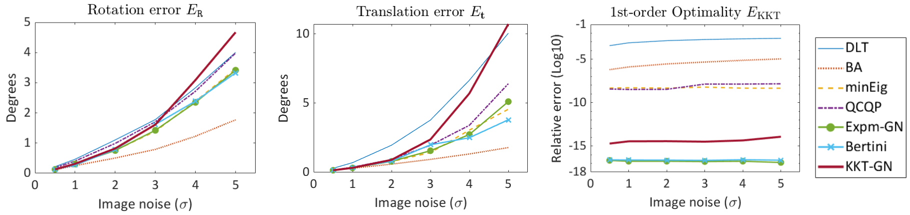


Fig. 3. Mean estimation error w.r.t. the image noise level over 1000 independent trials for each noise level.

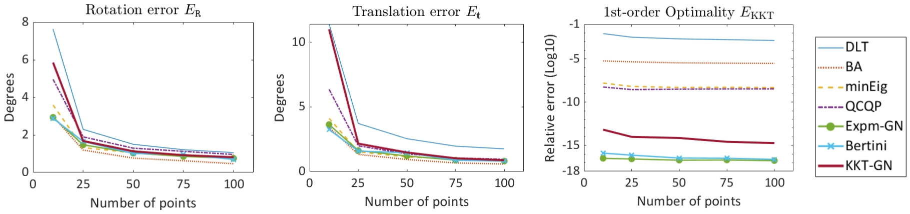


Fig. 4. Mean estimation error w.r.t. the number of point correspondences over 1000 independent trials for each point.

Computational time Figure 5 shows the computational time up to 1000 points with $\sigma = 2$. Expm-GN, minEig, and KTT-GN spend only 1 milliseconds for 1000 points, which is fast enough for real-time applications such as V-SLAM. Moreover, Expm-GN and KTT-GN are written in MATLAB, therefore, they can be further accelerated if implemented in C++. QCQP is much slower by orders of magnitude than the runtime reported in [48]. In fact, we had confirmed that the original QCQP written in C++ runs in 4 milliseconds. The main cause is that the Windows binary of SDPA-M is not well optimized due to a cross-compilation on Debian. This observation indicates that QCQP requires a highly-optimized SDP library whereas Expm-GN/KKT-GN does not. However, the conclusion still remains that QCQP suffers from high noise levels even run on C++. Bertini requires more than 1 minute, which is difficult to use for real-time applications.

Number of real solutions We counted the number of real solutions obtained by Bertini in two scenarios: varying the number of the points from 6 to 1000 with $\sigma = 2$, and varying the noise level up to $\sigma = 5$ with 100 points. We ran 100 independent trials for each point and noise level. As shown in Fig. 6, the number of real solutions ranges between 25 to 40 regardless of the number of the points and an image noise level. One point pair leads to a rank-1 update to the coefficient matrix $\mathtt{M}$ up to $\mathrm{rank}(\mathtt{M}) = 9$; however, no statistical significance was found from 6 to 1000 points. This is an interesting result that neither many points nor low noise level does not reduce the number of real solutions.

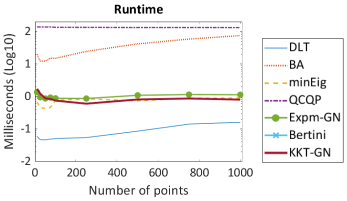


Fig. 5. Computational time w.r.t. the number of point correspondences. The result of Bertini, > 1 minute, is not displayed.

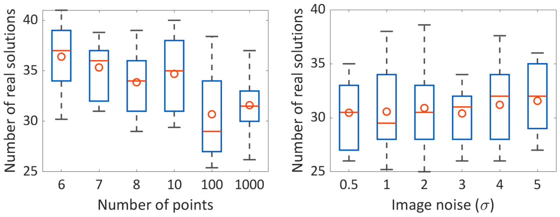
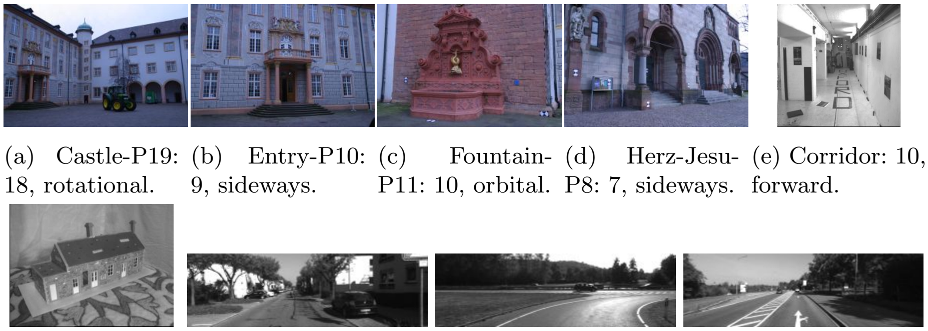


Fig. 6. The number of real solutions obtained by Bertini. Circles and bars in orange show the mean and the median values, respectively.



(a) Castle-P19: 18, rotational. (b) Entry-P10: 9, sideways. (c) Fountain-P11: 10, orbital. (d) Herz-Jesu-P8: 7, sideways. (e) Corridor: 10, forward.

(f) ModelHouse: 9, orbital. (g) 00: 454, forward. (h) 01: 110, forward. (i) 02: 466, forward.

Fig. 7. Images used in the real data evaluation. Each subcaption describes its sequence name followed by the number of image pairs and the camera motion. (a)–(d): Strecha, (c)–(d): VGG, (g)–(i): KITTI.

5.3 Real data

We evaluated the methods for various conditions of scenes and camera motions in publicly available dataset: four sequences from Strecha *et al.*'s dense MVS dataset [41][9], twos from VGG multi-view dataset[10], threes from KITTI visual odometry dataset [14][11]. The details of the sequences are shown in Figure 7.

Quantitative evaluation We conducted quantitative evaluations as follows. First we obtained initial point correspondences between two consecutive frames by using SIFT [31] in OpenCV 4.5[12]. We picked every 10th and 11th frames for KITTI because a single sequence of KITTI consists of more than 1000 frames.

[9] https://www.epfl.ch/labs/cvlab/data/
[10] https://www.robots.ox.ac.uk/~vgg/data/mview/
[11] http://www.cvlibs.net/datasets/kitti/eval_odometry.php
[12] https://github.com/opencv/opencv/

	Method	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$E_{\mathtt{R}}$ [degrees]	DLT	0.23	0.32	0.07	0.14	0.20	0.81	0.14	0.37	0.12
	QCQP	0.37	0.62	0.04	0.14	0.17	0.64	0.09	0.17	0.08
	minEig	0.25	0.24	0.05	0.09	0.15	0.60	0.08	0.17	0.07
	Expm-GN	0.25	0.24	0.05	0.09	0.16	0.58	0.08	0.18	0.07
	KKT-GN	0.25	0.22	0.05	0.10	0.17	0.59	0.09	0.18	0.07
	Bertini	0.25	0.22	0.05	0.09	0.16	0.58	0.08	0.17	0.07
	BA	0.24	0.18	0.05	0.10	0.14	0.58	0.08	0.15	0.06
$E_{\mathbf{t}}$ [degrees]	DLT	1.40	1.89	0.54	0.82	3.10	3.95	2.45	2.25	1.48
	QCQP	1.69	4.78	0.19	0.76	2.84	2.84	2.28	2.19	1.34
	minEig	1.07	1.27	0.25	0.42	2.71	1.85	2.09	2.21	1.29
	Expm-GN	1.05	1.21	0.23	0.39	2.84	1.85	2.08	2.23	1.29
	KKT-GN	1.05	1.15	0.23	0.43	2.93	1.88	2.15	2.27	1.27
	Bertini	1.05	1.15	0.23	0.39	2.84	1.85	2.08	2.23	1.27
	BA	1.00	1.00	0.25	0.43	2.40	1.81	2.32	2.31	1.26
E_{KKT} ($\log_{10}$)	DLT	−3.17	−3.52	−3.79	−3.51	−4.73	−2.82	−4.32	−3.47	−4.49
	QCQP	−8.19	−7.84	−8.14	−7.91	−8.78	−7.82	−8.56	−8.53	−8.60
	minEig	−8.20	−8.74	−8.74	−8.78	−9.28	−7.25	−8.46	−8.39	−9.67
	Expm-GN	−16.08	−16.42	−16.41	−16.29	−15.85	−15.82	−16.18	−15.91	−16.30
	KKT-GN	−12.24	−12.35	−12.19	−12.16	−9.33	−12.23	−9.95	−9.27	−12.34
	Bertini	−16.32	−16.21	−15.91	−15.70	−16.41	−16.28	−15.85	−15.59	−15.56
	BA	−6.23	−6.47	−6.79	−6.62	−5.10	−6.10	−5.25	−5.09	−5.33

Table 1. Estimation errors of the rotation, the translation, and the first-order optimality on real image dataset. (a)–(i) correspond to the sequences shown in Fig. 7. The best and the second-best results for each sequence are colored in red and blue, respectively.

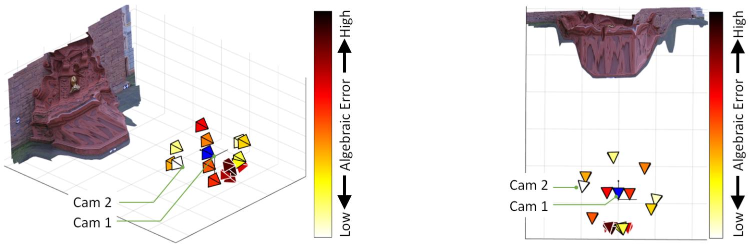
Then we perform LO-RANSAC [8] with the five-point method [37] as a hypothesis generator and each least-squares method as a local optimizer. Since Bertini is not efficient for RANSAC scenarios, we ran Bertini for the final refinement over inliers obtained by the Expm-GN local optimizer. LO-RANSAC was configured to have a threshold by 3-pixels, a confidence by 0.995, the maximum number of iterations by 10000, and the inner loops of an LO step by five. We conducted 100 independent trials for each sequence and fixed a random seed in each trial so that LO-RANSAC draws exactly the same samples for all methods.

Table 1 shows the average values of the three criteria, $E_{\mathtt{R}}$, $E_{\mathbf{t}}$, E_{KKT}, for each sequence. First of all, we can observe that three locally optimal methods, *i.e.* minEig, Expm-GN, and KKT-GN, have similar performance comparable to the globally optimal method, Bertini. Therefore, the three methods can be almost considered as globally optimal solvers in practice. On the other hand, QCQP is slightly worse than the three local methods in half of the sequences. Considering the two motion errors and E_{KKT}, we can say that the numerical accuracy of E_{KKT} is practically enough at 10^{-8} to 10^{-10}. These results are consistent with the synthetic data evaluation in Sec. 5.2.

Table 2 summarizes the RANSAC iterations and computational time. There are no significant differences on the number of RANSAC iterations. Since the same random seed was set for a single trial of all methods, this result suggests that each method obtains almost the same inliers at an LO step. DLT is the fastest due to its simpleness, followed by mingEig, Expm-GN, and KKT-GN.

	Method	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
# of iters.	DLT	1049	63	30	95	11	77	52	36	53
	QCQP	951	60	29	89	11	70	52	35	53
	minEig	1002	60	29	88	11	70	52	35	53
	Expm-GN	1002	60	29	88	11	68	52	34	53
	KKT-GN	1003	60	29	88	11	68	52	34	53
	BA	1001	60	29	88	11	68	52	34	53
Time [msec]	DLT	880	112	40	95	9	55	45	28	50
	QCQP	3067	1864	1581	1975	1229	1767	1340	1417	1379
	minEig	766	96	38	83	16	63	46	39	53
	Expm-GN	853	114	45	99	17	64	54	35	58
	KKT-GN	858	115	45	99	21	63	56	38	60
	BA	1348	510	367	441	222	373	352	299	383
# of real sols.		29 ± 7	30 ± 5	32 ± 9	33 ± 6	29 ± 6	30 ± 8	28 ± 7	33 ± 7	32 ± 5

Table 2. The number of RANSAC iterations and computational time on the sequences (a)–(i) shown in Fig. 7. The runtime of Bertini is omitted here because it takes several minutes for a single trial. The best and the second-best results for each sequence are colored in red and blue, respectively. The last row denotes the average of the number of real solutions with one standard deviation, which were obtained by Bertini.



(a) Bird's eye view.

(b) Top view.

Fig. 8. 3D visualization of real solutions obtained by Bertini. The first camera (the world origin) is depicted in blue. The second camera (the global optimum) and the others (local minima) are colored based on their algebraic error.

However, the difference between the three methods is slight. Expm-GN and KKT-GN can be potentially faster than minEig if implemented in C++.

Finally, we report the number of real solutions in each sequence. We constructed the coefficient matrix $\mathtt{M}$ from the inliers detected by LO-RANSAC with Expm-GN and solved Eq. (23) by Bertini. The last row of Table 2 indicates an interesting result, as in Sec. 5.2, that the number of real solutions is in the range of 20 to 40 in almost all cases irrespective of scenes and camera motions.

Qualitative evaluation We visualized the distribution of local minima, which are real solutions computed by Bertini. Figure 6 shows a 3D scene of Fountain-P11, which was reconstructed using the 5th and 6th frames of the sequence. We colored the first camera (the 5th frame, the world origin) in blue and the second cameras (the 6th frame, local minima) based on their algebraic error. The global optimum of the second camera is colored in white. For ease of viewing, we display only 17 second-cameras facing the direction of the fountain.

There is a local minimum in orange near the global optimum in white. Iterative methods may fall into the local minimum, not the global one, if an initial guess is closer to the local minimum than the global optimum. Moreover, there are many local minima having various error values behind the first camera. If an optimization scheme starts around them, it is difficult to escape the valley of the local minima.

As shown in the example above, we can visually validate the difficulty of a sequence by calculating all real solutions. The solution space visualization tells us why and how an estimation method successes or fails.

6 Discussion

As the results of the experiments in Sec. 5, we obtained the following findings.

- The global optimum of the algebraic error can be obtained by solving the new KKT conditions, Eq. (23). Homotopy continuation methods can be used to obtain the global optimum but are not fast enough for real-time applications.
- The number of real solutions is in the range from 20 to 40 irrespective of the camera motion and the number of the points.
- Three local methods, Expm-GN, minEig, and KKT-GN, provide almost globally optimal solutions, which are comparable with BA. Expm-GN gives the lowest first-order optimality, followed by KKT-GN and minEig.
- QCQP is worse than DLT in some cases due to the drop of $\mathrm{rank}(\mathtt{X}) = 1$ constraint in Eq. (17).

7 Conclusions

In this paper, we analyzed the solution space of the essential matrix estimation and revealed that there are at most 220 solutions in the algebraic error minimization. This is the first theoretical contribution in the three decades since Demazure gave the number of the solutions for the five-point minimal case in 1988. Exhaustive experiments showed that three locally optimal methods compute almost globally optimal solutions in effect. The proposed theory can be used to stabilize essential matrix estimation in real applications, *e.g.* reliable scene graph construction in Structure-from-Motion, motion analysis in near-degenerate situations, and so on. For future work in this research field, we pointed out the limitations of the current direct solvers based on Gröbner basis or homotopy continuation, namely the need to improve stability and efficiency for solving large polynomial systems such as the proposed KKT conditions. The mathematical condition also needs to be analyzed to clarify what determines the number of real solutions and their spatial distribution. Furthermore, our mathematical derivation contributes to explore the future development of globally optimal methods for multi-view geometry, especially for two-view geometry with both or either unknown focal length and unknown lens distortions.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10), 105–112 (2011)
2. Avidan, S., Shashua, A.: Novel view synthesis in tensor space. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1034–1040. IEEE (1997)
3. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Bertini: Software for numerical algebraic geometry, https://bertini.nd.edu/
4. Briales, J., Kneip, L., Gonzalez-Jimenez, J.: A certifiably globally optimal solution to the non-minimal relative pose problem. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 145–154 (2018)
5. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics **32**(6), 1309–1332 (2016)
6. Chen, T., Lee, T.L., Li, T.Y.: Hom4ps-3: a parallel numerical solver for systems of polynomial equations based on polyhedral homotopy continuation methods. In: International Congress on Mathematical Software. pp. 183–190. Springer (2014)
7. Chesi, G.: Camera displacement via constrained minimization of the algebraic error. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 370–375 (2008)
8. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Joint Pattern Recognition Symposium. pp. 236–243. Springer (2003)
9. Chum, O., Werner, T., Matas, J.: Epipolar geometry estimation via ransac benefits from the oriented epipolar constraint. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. vol. 1, pp. 112–115. IEEE (2004)
10. Demazure, M.: Sur deux problemes de reconstruction. Tech. Rep. RR-0882, INRIA (Jul 1988), https://hal.inria.fr/inria-00075672
11. Fathy, M.E., Rotkowitz, M.C.: Essential matrix estimation using adaptive penalty formulations. J. Comput. Vision **74**(2), 117–136 (2007)
12. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
13. Fraundorfer, F., Tanskanen, P., Pollefeys, M.: A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In: European Conference on Computer Vision. pp. 269–282. Springer (2010)
14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
15. Grayson, D.R., Stillman, M.E.: Macaulay2, a software system for research in algebraic geometry. http://www.math.uiuc.edu/Macaulay2/
16. Hartley, R.I., Kahl, F.: Global optimization through rotation space search. International Journal of Computer Vision **82**(1), 64–79 (2009)
17. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edn. (2004)
18. Helmke, U., Hüper, K., Lee, P.Y., Moore, J.: Essential matrix estimation using gauss-newton iterations on a manifold. International Journal of Computer Vision **74**(2), 117–136 (2007)

19. Jiang, F., Kuang, Y., Solem, J.E., Åström, K.: A minimal solution to relative pose with unknown focal length and radial distortion. In: Asian Conference on Computer Vision. pp. 443–456. Springer (2014)
20. Joo, H., Simon, T., Li, X., Liu, H., Tan, L., Gui, L., Banerjee, S., Godisart, T., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social interaction capture. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(1), 190–204 (2019). https://doi.org/10.1109/TPAMI.2017.2782743
21. Kalantari, M., Hashemi, A., Jung, F., Guédon, J.P.: A new solution to the relative orientation problem using only 3 points and the vertical direction. Journal of Mathematical Imaging and Vision **39**(3), 259–268 (2011)
22. Kneip, L., Furgale, P.: Opengv: A unified and generalized approach to real-time calibrated geometric vision. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 1–8. IEEE (2014)
23. Kneip, L., Lynen, S.: Direct optimization of frame-to-frame rotation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2352–2359 (2013)
24. Kruppa, E.: Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung. Sitzungsberichte der Mathematisch-Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften pp. 1939–1948 (1913)
25. Kuang, Y., Solem, J.E., Kahl, F., Astrom, K.: Minimal solvers for relative pose with a single unknown radial distortion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 33–40 (2014)
26. Kukelova, Z.: Algebraic methods in computer vision. Ph.D. thesis, Czech Technical University in Prage (2013)
27. Kukelova, Z., Bujnak, M., Pajdla, T.: Automatic generator of minimal problem solvers. In: European Conference on Computer Vision. pp. 302–315. Springer (2008)
28. Kukelova, Z., Bujnak, M., Pajdla, T.: Polynomial eigenvalue solutions to minimal problems in computer vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(7), 1381–1393 (2011)
29. Larsson, V., Astrom, K., Oskarsson, M.: Efficient solvers for minimal problems by syzygy-based reduction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 820–829 (2017)
30. Leykin, A.: Numerical algebraic geometry. Journal of Software for Algebra and Geometry **3**(1), 5–10 (2011)
31. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
32. Ma, Y., Košecká, J., Sastry, S.: Optimization criteria and geometric algorithms for motion and structure estimation. International Journal of Computer Vision **44**(3), 219–249 (2001)
33. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
34. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
35. Nakano, G.: Globally optimal dls method for pnp problem with cayley parameterization. In: Proceedings of the British Machine Vision Conference (BMVC). pp. 78.1–78.11 (2015)

36. Nakano, G., Takada, J.: A robust least squares solution to the calibrated two-view geometry with two known orientation angles. In: International Conference on Computer Vision, Imaging and Computer Graphics. pp. 132–145. Springer (2013)
37. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE transactions on pattern analysis and machine intelligence **26**(6), 756–770 (2004)
38. Philip, J.: Critical point configurations of the 5-, 6-, 7-, and 8-point algorithms for relative orientation. Tech. Rep. TRITA-MAT-1998-MA-13, Department of Mathematics, Royal Institute of Technology (1998)
39. Pizarro, O., Eustice, R.M., Singh, H.: Relative pose estimation for instrumented, calibrated imaging platforms. In: DICTA. pp. 601–612. Sydney, Australia (2003)
40. Stewénius, H., Nistér, D., Kahl, F., Schaffalitzky, F.: A minimal solution for relative pose with unknown focal length. Image and Vision Computing **26**(7), 871–877 (2008)
41. Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: 2008 IEEE conference on computer vision and pattern recognition. pp. 1–8. Ieee (2008)
42. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: Inloc: Indoor visual localization with dense matching and view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7199–7209 (2018)
43. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International workshop on vision algorithms. pp. 298–372. Springer (1999)
44. Ventura, J., Arth, C., Lepetit, V.: Approximated relative pose solvers for efficient camera motion estimation. In: European Conference on Computer Vision. pp. 180–193. Springer (2014)
45. Verschelde, J.: Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. ACM Trans. Math. Softw. **25**(2), 251–276 (Jun 1999). https://doi.org/10.1145/317275.317286
46. Yamashita, M., Fujisawa, K., Fukuda, M., Kobayashi, K., Nakata, K., Nakata, M.: Latest developments in the sdpa family for solving large-scale sdps. In: Handbook on semidefinite, conic and polynomial optimization, pp. 687–713. Springer (2012)
47. Zeisl, B., Sattler, T., Pollefeys, M.: Camera pose voting for large-scale image-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2704–2712 (2015)
48. Zhao, J.: An efficient solution to non-minimal case essential matrix estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
49. Zhu, S., Zhang, R., Zhou, L., Shen, T., Fang, T., Tan, P., Quan, L.: Very large-scale global sfm by distributed motion averaging. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4568–4577 (2018)