Machine Vision and Applications
https://doi.org/10.1007/s00138-019-01012-0

ORIGINAL PAPER

Camera pose estimation from lines: a fast, robust and general method

Ping Wang[1,2] · Guili Xu[2] · Yuehua Cheng[3] · Qida Yu[2]

Received: 23 March 2018 / Revised: 5 December 2018 / Accepted: 10 February 2019

Abstract
In this paper, we revisit the perspective-n-line problem and propose a closed-form solution that is fast, robust and generally applicable. Our main idea is to formulate the pose estimation problem into an optimal problem. Our method only needs to solve a fifteenth-order and a fourth-order univariate polynomial, respectively, which makes the processes more easily understood and significantly improves the performance. Experiment results show that our method offers accuracy and precision comparable or better than existing state-of-the-art methods, but with significantly lower computational cost. This superior computational efficiency is particularly suitable for real applications.

Keywords Perspective-n-line problem (PnL) · Camera pose estimation · Absolute position and orientation · Computer vision

1 Introduction

Determining the position and orientation of a calibrated camera from n correspondences between 3D reference features and their 2D projections, where the features are either points or lines, is an important step in many vision-based tasks such as robot location [3], augmented reality [4], structure-from-motion (SfM) [25], and spacecraft pose estimation during descent and landing [21]. For point features, the problem becomes the well-known Perspective-n-point (PnP) problem, which has been well studied in recent years [12,14,17–19,22,27,31]. For line features, it corresponds to the Perspective-n-Line (PnL) problem, which remains a challenging topic.

In this paper, we propose a fast, robust, and general method for the PnL problem. Our proposed method has several advantages compared to existing methods:

- *Optimality* Some recent works (i.e., [2,20]) formulate the PnL problem into a unconstrained optimization problem and compute all stationary points by solving the first-order optimal conditions of the object function. The first-order optimality conditions are usually a complex polynomial equation system, and are solved by using matrix resultant technique (i.e., the Gröbner basis technique [15]). However, this processing will take significant time and is difficult to assure reliability. Similarly, to these recent works, our method also transfers the PnL problem into an optimal one. However, our method only needs to solve a fifteenth-order and a fourth-order univariate polynomial without solving the complex polynomial equation systems. The number of the solutions for our method is substantially smaller than existing globally optimal methods. All of these make our method more easily applicable and significantly improve the performance.
- *Universal applicability* Our method achieves accurate results for both non-redundant line sets ($n = 4$ or 5) and redundant line sets ($n \geq 6$). In contrast, some existing methods have poor results when $n = 4$ or 5 (e.g., [2,20,30]), and other methods can only work for $n \geq 6$ (e.g., [1,23,26]).
- *High accuracy and efficiency* Experiment results show that our method offers accuracy and precision comparable or better than existing state-of-the-art methods.

✉ Guili Xu
guilixu2002@163.com

Ping Wang
pingwangsky@gmail.com

Yuehua Cheng
chengyuehua@nuaa.edu.cn

Qida Yu
dannyhyer@163.com

[1] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

[2] College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

[3] College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Published online: 20 February 2019

Additionally, our method, in terms of computational efficiency, is significantly better than existing globally optimal methods. The computational efficiency of our method is even better than some linear-formulation-based PnL methods [1,26], with increasing number of lines. This superior computational efficiency is particularly suitable for real applications.

The rest of the paper is organized as follows: Sect. 2 provides an overview of the related work on PnL. We describe our proposed method in Sect. 3, while we present a thorough analysis of the proposed method by simulated and real experiments in Sect. 4. Lastly, we conclude the work in Sect. 5.

2 Related work

As the minimal case, P3L ($n = 3$) has been thoroughly investigated in the literature. In one of the earliest methods, Dhome et al. [8] proposed an analytical method to solve the P3L problem. They transformed 3D lines and 2D lines into a model coordinate system and a virtual viewer coordinate system, respectively, and then derived a eighth-order polynomial to determine the closed-form solution of the P3L problem. However, its mathematical expression is too complex to solve. In Chen [7] investigated the necessary conditions under which the P3L problem has finite number of solutions, and proposed an algebraic method to the P3L problem. Unfortunately, this method is highly unstable in the presences of noise. Caglioti [6] addressed a special case of the P3L problem where three lines lie in a common plane and intersect at a common point, which is called the planar 3-line junction perspective problem. Qin and Zhu [24] proposed a solution of another special situation of the P3L problem, where three lines form a Z-shape in space, i.e., two lines are parallel and the 3rd line intersects with both of them. Recently, Zhang et al. [30] decomposed the overall rotation from the world frame to the camera frame into a sequence of simple rotations, and proposed a geometric method to the P3L method. In practice, a P3L solution is usually used in combination with RANSAC to remove outliers, because the solutions of P3L problem are not uniquely determined [7].

Considering that data redundancy generally contributes to improving accuracy, most of existing works on PnL focus on overconstrained cases with more than three lines. To properly account for the recent works, we would like to roughly categorize them into two groups—*the iterative methods* and *the non-iterative methods*.

Typically, *the iterative methods* [9,13,16] formulate the PnL problem into a nonlinear least-squares problem, and then solve it using iterative optimization methods, i.e., Gauss–Newton and Levenberg–Marquardt method [5]. However, the iterative methods are sensitive to the initialization and are easily trapped into a local minimum, which will lead to poor accuracy, especially when no redundant lines ($n = 4$ or 5) are available.

For *the non-iterative methods*, the most straight-forward method is the Direct Linear Transformation (DLT) method [1,26], which transforms the measured line correspondences into a homogeneous system of linear equations and solves the linear equations using singular value decomposition (SVD) method [11]. As the improved method, DLT-Plücker-Lines was presented in [23]. DLT-Plücker-Lines parameterizes 3D reference lines using Plücker coordinates and uses linear least squares to estimate the camera projection matrix. Recently, Xu et al. [28] explored the similarity between PnL and PnP (Perspective n point problem) in terms of linear formulation, and proposed a series of linear-formulation-based PnL methods.

To sum up, all the aforementioned linear methods have an advantage of less computing costs, but are sensitive to noise, and cannot deal with small line sets ($n = 4$ or 5). To overcome these problems, Ansar and Daniilidis [2] developed a method that is able to handle $n = 4$ or more lines, with computational complexity $O(n^2)$. However, this method is still inaccurate for small line sets, due to its underlying linearization scheme. To improve accuracy, Mirzaei and Roumeliotis [20] proposed the first globally optimal and non-iterative method (called AlgLS) with complexity $O(n)$, which formulates the PnL problem into a multivariate polynomial system using the camera measurement equations and employs the matrix resultant technique to determine all roots of the system. Unfortunately, they parameterized rotation by using the Cayley Gibbs Rodriguez (CGR) representation, which leads to unstable results because the CGR representation has a singularity for any 180° rotations. To resolve these drawbacks, Zhang et al. [30] proposed another non-iterative $O(n)$ solution, named RPnL, which transfers the PnL problem into a suboptimal problem by solving a fifteenth-order polynomial. RPnL is very efficient and works well for both non-redundant ($n = 4$ or 5) and redundant ($n \geq 6$) lines cases. Even more recently, the RPnL method was modified by Xu et al. [28] into the Accurate Subset-based PnL (ASPnL) method, which acts more accurately on small line sets. To our knowledge, ASPnL is one of the most accurate non-iterative methods until now, which represents the state-of-the-art solution. However, ASPnL has a primary disadvantage that its computational time increases strongly for higher number of lines, which will limit its applications for scenarios with many lines.

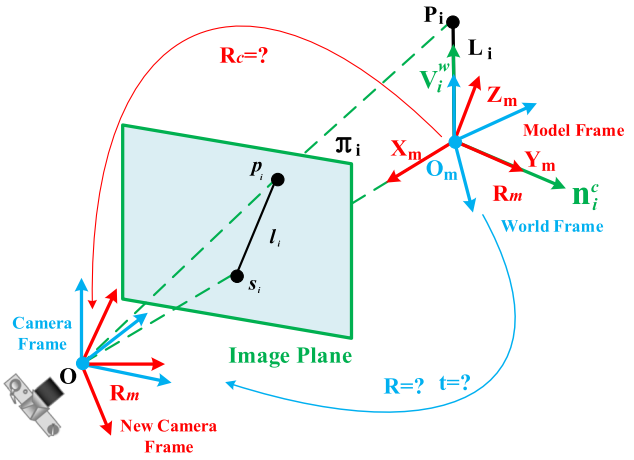


Fig. 1 Illustration of the PnL problem

3 Proposed method

3.1 Problem statement

The problem considered in this paper is illustrated in Fig. 1. Given a calibrated camera and n known 3D reference lines $L_i (i = 1, 2, \ldots, n)$ with their corresponding 2D projections on the image plane as l_i. The goal is to recover the rotation R and translation t of the camera. Let $L_i = (\mathbf{V}_i^w, P_i)$ be the known 3D line in the world frame, where $\mathbf{V}_i^w \in \mathbb{R}^3$ is the normalized vector giving the direction of the line and $P_i \in \mathbb{R}^3$ is any point on the line. Let $l_i = (s_i, p_i)$ be the corresponding 2D projection of L_i on the image plane, where s_i and p_i are the endpoints of l_i. For a given l_i, a projection plane Π_i is determined which passes through the projection center O, l_i and L_i. The normal of Π_i is denoted as $\mathbf{n}_i^c \in \mathbb{R}^3$ in the camera frame, which can be easily calculated using the cross product of s_i and p_i.

3.2 Defining an intermediate rotation from correspondences

The first step involves the definition of a new, intermediate rotation matrix R_m from the reference lines L_i and their corresponding normals $\mathbf{n}_i^c$. To obtain the intermediate rotation R_m, we select a line $L_i = (\mathbf{V}_i^w, P_i)$ with the longest projection length $|s_i p_i|$, and create an intermediate model frame $[O_m - \mathbf{X}_m, \mathbf{Y}_m, \mathbf{Z}_m]$ (see Fig. 1), where

$$\begin{aligned}
\mathbf{Y}_m &= \frac{\mathbf{n}_i^c}{\|\mathbf{n}_i^c\|} \\
\mathbf{X}_m &= \frac{\mathbf{n}_i^c \times \mathbf{V}_i^w}{\|\mathbf{n}_i^c \times \mathbf{V}_i^w\|} \\
\mathbf{Z}_m &= \frac{\mathbf{X}_m \times \mathbf{Y}_m}{\|\mathbf{X}_m \times \mathbf{Y}_m\|}.
\end{aligned} \tag{1}$$

The Y-axis of the model frame aligns with $\mathbf{n}_i^c$, and the origin of the model frame is located at the origin of the world frame. Hence, the intermediate rotation matrix R_m can be determined as $R_m = [\mathbf{X}_m, \mathbf{Y}_m, \mathbf{Z}_m]^T$.

Via the rotation matrix R_m, feature vectors $\mathbf{V}_i^w$ and their corresponding normals $\mathbf{n}_i^c$ can be easily rotated into a new world frame and a new camera frame using

$$\mathbf{V}_i^{w_n} = R_m \mathbf{V}_i^w \quad i = 1, 2, \ldots, n, \tag{2}$$

$$\mathbf{n}_i^{c_n} = R_m \mathbf{n}_i^c \quad i = 1, 2, \ldots, n. \tag{3}$$

Let's define the rotation matrix between the new world frame and the new camera frame as R_c. Now if we are able to obtain the R_c, and the complete rotation R can be easily given by R_c and R_m. We use the Euler Angle to express the rotation R_c as

$$\begin{aligned}
R_c &= \mathrm{Rot}(Y, \beta)\mathrm{Rot}(Z, \gamma)\mathrm{Rot}(X, \alpha) \\
&= \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix},
\end{aligned} \tag{4}$$

in which $\mathrm{Rot}(X, \alpha)$, $\mathrm{Rot}(Y, \beta)$ and $\mathrm{Rot}(Z, \gamma)$ denote rotation around the X-axis, Y-axis and Z-axis, respectively. $\mathrm{Rot}(X, \alpha)$ can be easily calculated because α is the angle between Z-axis and $\mathbf{V}_i^{w_n}$. Hence, the R_c can be determined by two unknown variables β and γ.

3.3 Determining Rot(Z, γ) by using least-square residual

In the following, we select another line L_j, whose projection $|s_j p_j|$ is the second longest, as an auxiliary line. Every remaining line L_k together with the L_i and L_j forms a 3-line subsets $\{L_i L_j L_k | k = 1, \ldots, n; k \neq i \& k \neq j\}$, and the line set $\{L_i, i = 1, 2, \ldots, n\}$ is finally divided into $n - 2$ subsets. By using the P3L (perspective-three-line) constraint [30], and letting $x = \cos\gamma$, each subset can build an eighth-order polynomial as follows:

$$\begin{cases} f_1(x) = \sum_{k=0}^{8} \delta_{1k} x^k = 0 \\ f_2(x) = \sum_{k=0}^{8} \delta_{2k} x^k = 0 \\ \cdots \\ f_{n-2}(x) = \sum_{k=0}^{8} \delta_{(n-2)k} x^k = 0 \end{cases}, \tag{5}$$

where δ_k is known coefficients.

Instead of directly solving a series of eighth-order polynomials, a cost function $F = \sum_{i=1}^{n-2} f_i^2(x)$ is defined as the square sum of these polynomials. The minima of F can then be determined by finding the roots of its derivative $F' = \sum_{i=1}^{n-2} f_i(x) f_i'(x) = 0$. F' is a fifteenth-order polynomial, which has at most 8 minima, and can be easily solved

by the eigenvalue method [10]. Once the minimal of F is determined, the $\mathrm{Rot}(Z,\gamma)$ in Eq. (4) can be easily calculated by the sine and cosine of the rotation angle γ.

3.4 Retrieving Rot(Y, β) by solving an optimal problem

When the $\mathrm{Rot}(Z,\gamma)$ is determined, from Eq. (4), the R_c can be expressed as:

$$R_c = \mathrm{Rot}(Y,\beta)R' = \begin{bmatrix} s_1 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & s_1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \tag{6}$$

in which $R' = \mathrm{Rot}(Z,\gamma)\mathrm{Rot}(X,\alpha)$, $s_1 = \cos\beta$ and $s_2 = \sin\beta$.

As L_i lies on the plane Π_i, the line direction vector $\mathbf{V}_i^{w_n}$ is perpendicular to the plane normal $\mathbf{n}_i^{c_n}$. Hence, the rotation matrix R_c must satisfy the constraint that

$$\left(\mathbf{n}_i^{c_n}\right)^T R_c \mathbf{V}_i^{w_n} = 0 \quad i = 1, 2, \ldots, n. \tag{7}$$

By substituting Eq. (6) into Eq. (7), and letting $\mathbf{n}_i^{c_n} = [x_i, y_i, z_i]^T$ and $(R'\mathbf{V}_i^{w_n}) = [X_i, Y_i, Z_i]^T$, we have

$$\begin{bmatrix} x_i & y_i & z_i \end{bmatrix} \begin{bmatrix} s_1 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & s_1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = 0 \tag{8}$$

By denoting a new unknown $s = [s_1, s_2, 1]^T$, Eq. (8) can be represented as

$$A_i s = 0, \tag{9}$$

where

$$A_i = \begin{bmatrix} x_i X_i + z_i Z_i, & x_i Z_i - z_i X_i, & y_i Y_i \end{bmatrix}.$$

Equation (9) is satisfied for every reference lines, hence

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} s = 0 \Longleftrightarrow As = 0 \tag{10}$$

Note that Eq. (10) is not perfectly satisfied due to the noise. The residual of Eq. (10) can be expressed as

$$\eta = As \tag{11}$$

In addition, there is a constraint that $s_1^2 + s_2^2 = 1$. Hence, we directly minimize the sum of the squared residuals to build a cost function with a constraint. The simplified cost function is

$$\varepsilon = s^T G s + \lambda(1 - s_1^2 - s_2^2) \quad i = 1, 2, \ldots, n, \tag{12}$$

in which

$$G = A^T A = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{12} & G_{22} & G_{23} \\ G_{13} & G_{23} & G_{33} \end{bmatrix}$$

is a know 3×3 symmetric matrix, and λ is a Lagrange multiplier. The minima of Eq. (12) can be determined by solving the polynomial system of its first-order optimality condition.

$$\begin{aligned} \frac{\partial \varepsilon}{\partial s_1} &= G_{11}s_1 + G_{12}s_2 - \lambda s_1 + G_{13} = 0 \\ \frac{\partial \varepsilon}{\partial s_2} &= G_{12}s_1 + G_{22}s_2 - \lambda s_2 + G_{23} = 0 \\ \frac{\partial \varepsilon}{\partial \lambda} &= 1 - s_1^2 - s_2^2 = 0 \end{aligned} \tag{13}$$

By eliminating λ, we can express s_2 via

$$s_2 = \frac{2G_{12}s_1^2 + G_{23}s_1 - G_{12}}{(G_{11} - G_{22})s_1 + G_{13}}. \tag{14}$$

By squaring both sides of Eq. (14), and substituting $s_2^2 = 1 - s_1^2$ into it, we finally obtain a fourth-order polynomial of the form

$$s_1^4 F_4 + s_1^3 F_3 + s_1^2 F_2 + s_1 F_1 + F_0 = 0, \tag{15}$$

where

$$\begin{aligned} F_4 &= 4G_{12}^2 + G_{22}^2 + G_{11}^2 - 2G_{11}G_{22} \\ F_3 &= 4G_{12}G_{23} + 2G_{11}G_{13} - 2G_{13}G_{22} \\ F_2 &= G_{23}^2 + 2G_{11}G_{22} + G_{13}^2 - 4G_{12}^2 - G_{11}^2 - G_{22}^2 \\ F_1 &= 2G_{13}G_{22} - 2G_{11}G_{13} - 2G_{12}G_{23} \\ F_0 &= G_{12}^2 - G_{13}^2. \end{aligned}$$

s_1 can be easily solved from Eq. (15) by using the eigenvalue method [10]. After plugging s_1 back into Eq. (14), s_2 can also be determined. Eq. (15) has at most 2 minima, and up to 16 minima can be obtained in our method. For each minimum, we first evaluate the rotation R_c via Eq. (6) and then calculate the orthogonal error E_{er} as the following equation

$$E_{\mathrm{er}} = \sum_{i=1}^{n} \left(\left(\mathbf{n}_i^{c_n}\right)^T R_c \mathbf{V}_i^{w_n} \right)^2. \tag{16}$$

When $n \geq 6$, the PnL problem has a unique solution. Therefore, we choose the R_c with smallest objective value in

Eq. (16) as the final solution, and retrieve the complete R between the camera frame and the world frame by using $R = R_m^T R_c R_m$. When $4 \leq n < 6$, the PnL problem has multiple solutions in general, and we return all R_c to calculate the complete R.

3.5 Rotation refinement and translation estimation

To further polish the estimated R, we reformulate the pose estimation problem into a least-squares problem with three variables, then solve the least-squares problem via a single Gauss–Newton step. According to the projection from 3D lines to 2D lines in the normalized image plane [29], we obtain

$$\begin{aligned} &\left(\mathbf{n}_i^c\right)^T R\mathbf{V}_i^w = 0 \quad i = 1, 2, \ldots, n, \\ &\left(\mathbf{n}_i^c\right)^T (RP_i + t) = 0 \quad i = 1, 2, \ldots, n, \end{aligned} \tag{17}$$

in which $\mathbf{n}_i^c = [\hat{x}_i, \hat{y}_i, \hat{z}_i]^T$, $\mathbf{V}_i^w = [\hat{X}_i, \hat{Y}_i, \hat{Z}_i]^T$ and $P_i = [Px_i, Py_i, Pz_i]^T$. We adopt the Cayley parameterization to express the rotation R, which is given by

$$R = \frac{1}{H}\begin{bmatrix} 1+b^2-c^2-d^2 & 2bc-2d & 2bd+2c \\ 2bc+2d & 1-b^2+c^2-d^2 & 2cd-2b \\ 2bd-2c & 2cd+2b & 1-b^2-c^2+d^2 \end{bmatrix}, \tag{18}$$

where $H = 1 + b^2 + c^2 + d^2$.

Now letting $\hat{s} = [1, b, c, d, b^2, bc, bd, c^2, cd, d^2]^T$, and Eq. (17) can be transformed into the following matrix form

$$Q_i^T \hat{s} = N_i t \quad i = 1, 2, \ldots, n, \tag{19}$$

in which

$$Q_i = \begin{bmatrix} \hat{x}_i\hat{X}_i + \hat{y}_i\hat{Y}_i + \hat{z}_i\hat{Z}_i, & \hat{x}_iPx_i + \hat{y}_iPy_i + \hat{z}_iPz_i \\ 2\hat{z}_i\hat{Y}_i - 2\hat{y}_i\hat{Z}_i, & 2\hat{z}_iPy_i - 2\hat{y}_iPz_i \\ 2\hat{x}_i\hat{Z}_i - 2\hat{z}_i\hat{X}_i, & 2\hat{x}_iPz_i - 2\hat{z}_iPx_i \\ 2\hat{y}_i\hat{X}_i - 2\hat{x}_i\hat{Y}_i, & 2\hat{y}_iPx_i - 2\hat{x}_iPy_i \\ \hat{x}_i\hat{X}_i - \hat{y}_i\hat{Y}_i - \hat{z}_i\hat{Z}_i, & \hat{x}_iPx_i - \hat{y}_iPy_i - \hat{z}_iPz_i \\ 2\hat{y}_i\hat{X}_i + 2\hat{x}_i\hat{Y}_i, & 2\hat{y}_iPx_i + 2\hat{x}_iPy_i \\ 2\hat{z}_i\hat{X}_i + 2\hat{x}_i\hat{Z}_i, & 2\hat{z}_iPx_i + 2\hat{x}_iPz_i \\ \hat{y}_i\hat{Y}_i - \hat{x}_i\hat{X}_i - \hat{z}_i\hat{Z}_i, & \hat{y}_iPy_i - \hat{x}_iPx_i - \hat{z}_iPz_i \\ 2\hat{z}_i\hat{Y}_i + 2\hat{y}_i\hat{Z}_i, & 2\hat{z}_iPy_i + 2\hat{y}_iPz_i \\ \hat{z}_i\hat{Z}_i - \hat{x}_i\hat{X}_i - \hat{y}_i\hat{Y}_i, & \hat{z}_iPz_i - \hat{x}_iPx_i - \hat{y}_iPy_i \end{bmatrix}$$

and

$$N_i = \begin{bmatrix} 0 \\ -\left(\mathbf{n}_i^c\right)^T \end{bmatrix}.$$

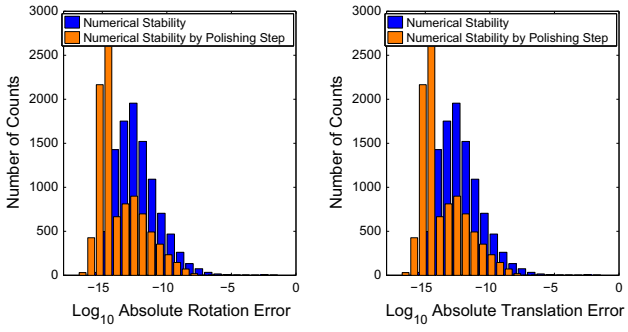


Fig. 2 Numerical stability of the proposed solver. The horizontal axis shows the $\log_{10}$ value of the absolute rotation error (left) and the absolute translation error (right)

Equation (19) is also satisfied for every reference lines, hence

$$\begin{bmatrix} Q_1^T \\ Q_2^T \\ \vdots \\ Q_n^T \end{bmatrix} \hat{s} = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{bmatrix} t \Longleftrightarrow Q\hat{s} = Nt \Longleftrightarrow t = C\hat{s}, \tag{20}$$

where $\hat{C} = (N^T N)^{-1} N^T \hat{Q}$.

After plugging $t = C\hat{s}$ back into Eq. (19), we finally obtain the least-squares problem as follows

$$\hat{\varepsilon} = \sum_{i=1}^{n} \|(Q_i^T - N_i C)\hat{s}\|^2 = \sum_{i=1}^{n} \|E_i\hat{s}\|^2, \tag{21}$$

where $\hat{E}_i$ is a 3×10 matrix that can be computed ahead. We then use the typical Gauss–Newton method to solve the least-squares problem. Since the initialization is accurate enough, only one-step iteration is used. Specifically, assuming that $\hat{s}$ is a stationary point of Eq. (21), we refine $\hat{s}$ through the updating rule $\hat{s} = \hat{s} + \Delta\hat{s}$. The increment $\Delta\hat{s}$ is determined by $\Delta\hat{s} = -[J^T J]^{-1} J^T F(\hat{s})$, where $F = [E_1, E_2, \ldots, E_n]^T$ and J is the Jacobian matrix of F.

Once the refined $\hat{s}$ is obtained, and the complete R and t are finally given by substituting $\hat{s}$ into Eqs. (18) and (20). The refined results are shown in Fig. 2 and show that the polishing step can drastically improve the numerical precision, although only one-step iteration is used. Additionally, to improve the efficiency of our method, we construct all rows of Q and N in Eq. (19) simultaneously by using vectorization method, instead of every two rows in an iterative way. Similarly, the matrix A in Eq. (10) can also be constructed by using this strategy.

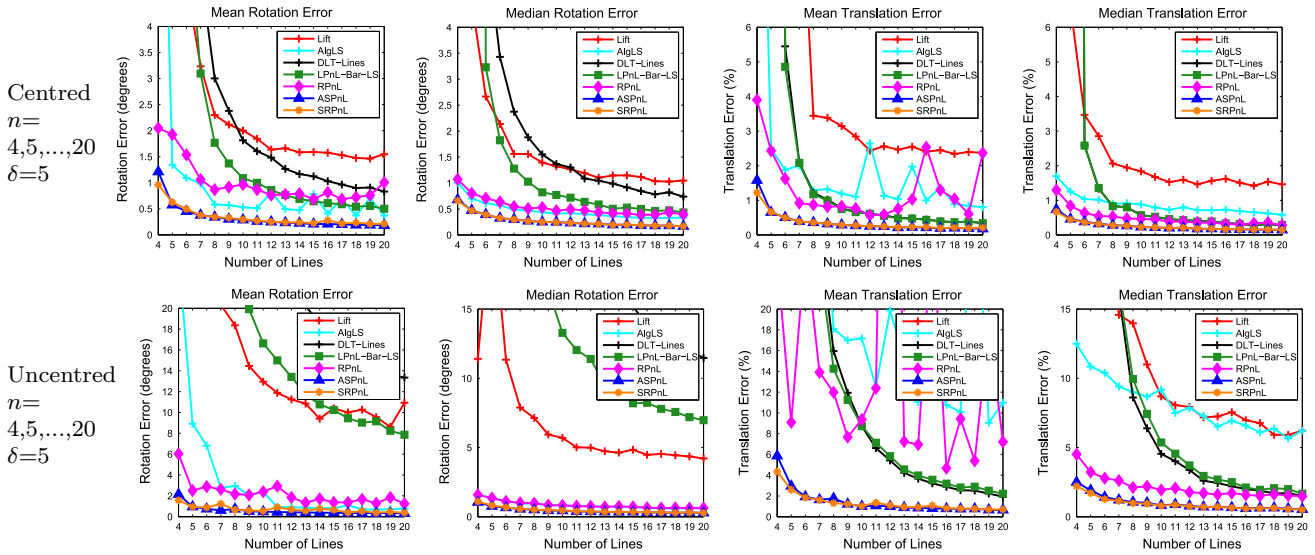


Fig. 3 Synthetic experimental results. n denotes the number of lines, and δ denotes the standard deviation of image noise. The accuracies of the compared methods were tested in centered and uncentered cases by varing n from 4 to 20

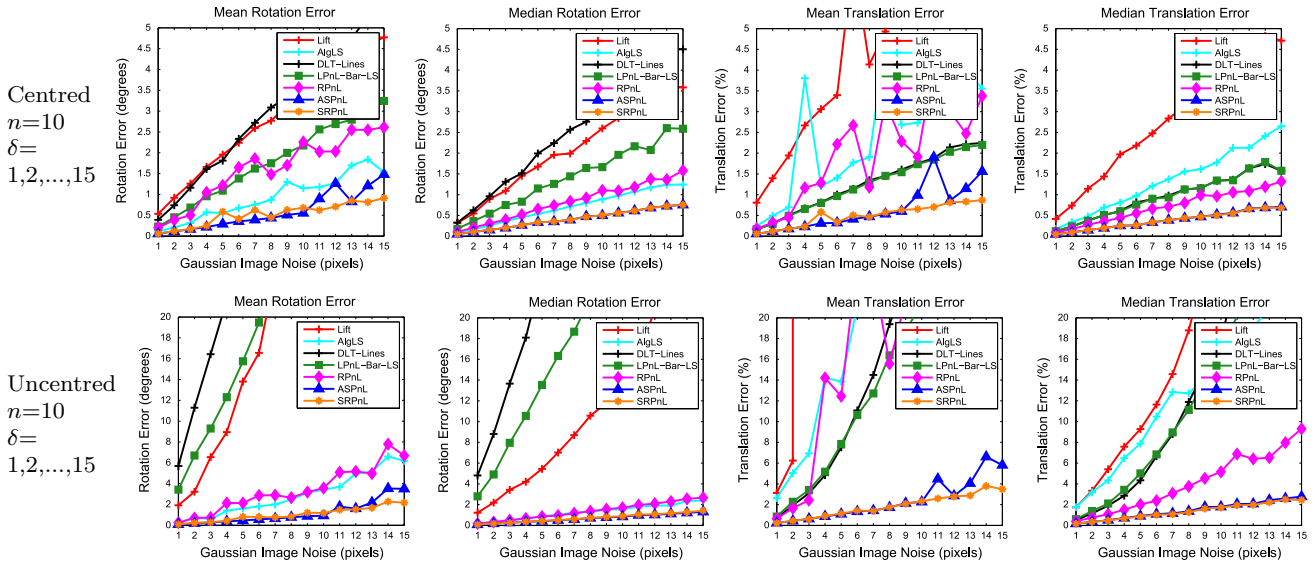


Fig. 4 Synthetic experimental results. n denotes the number of lines, and δ denotes the standard deviation of image noise. The robustness against noisy was tested in centered and uncentered cases by varing δ from 1 to 15 pixels

4 Experiment results

In this section, we investigated the performance of the proposed method, referred to as **SRPnL**,[1] by means of synthetic data and real images, and compared the accuracy and efficiency with the leading PnL methods:

– *DLT-Lines* An efficient linear method, which uses linear least squares to estimate the camera pose parameters. This method requires at least 6 line correspondences [1,26].
– *Lift* An efficient linear method, which converts the polynomial system to a set of linear systems by re-parameterizing the nonlinear terms with new variables. **Lift** method is sensitive to the measurement-noise [2].
– *AlgLS* One of the most accurate non-iterative method, which estimates the camera's pose by directly solving the corresponding least-squares problem algebraically. This method is very time-consuming when n is large [20].
– *RPnL* A robust and efficient non-iterative method, which works well for both non-redundant ($n \leq 6$) and redun-

[1] The source code of the proposed method (SRPnL) can be downloaded from https://sites.google.com/view/ping-wang-homepage.

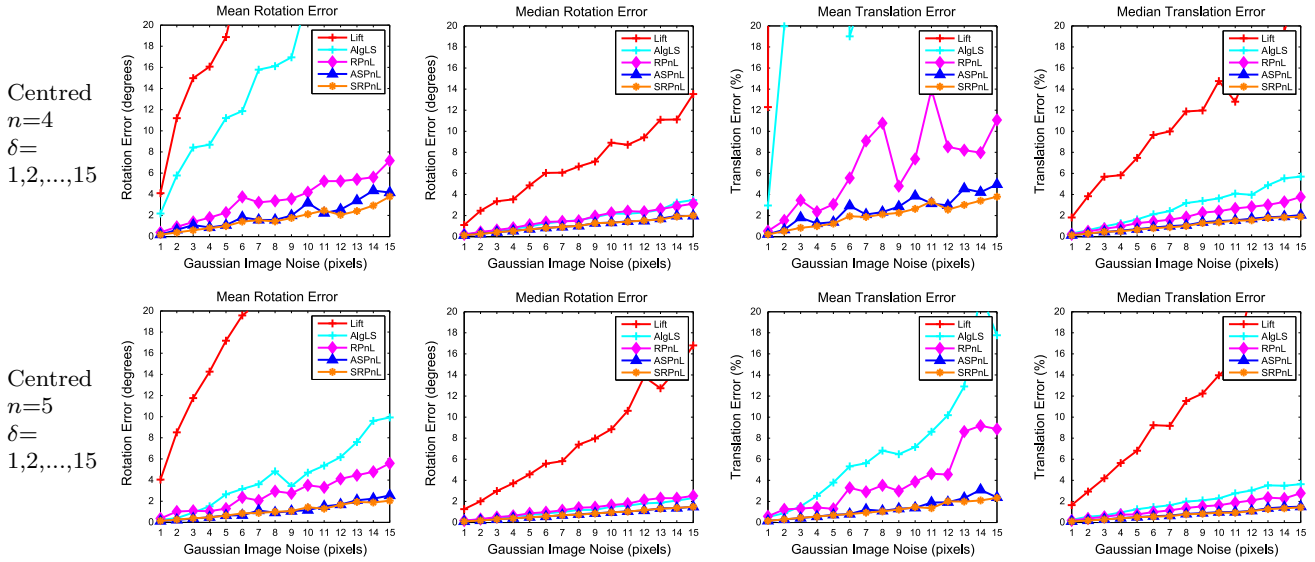


Fig. 5 Synthetic experimental results. n denotes the number of lines. δ denotes the standard deviation of image noise, and δ varies from 1 to 15 pixels. The robustness against small line set was tested in centered case

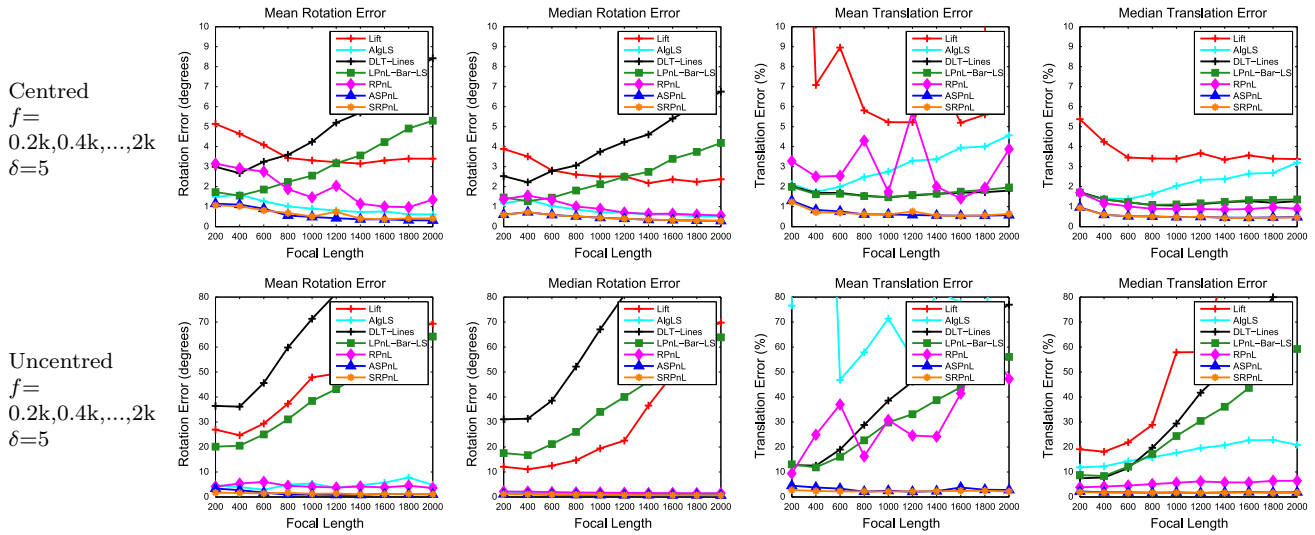


Fig. 6 Synthetic experimental results. f denotes the focal length, and δ denotes the standard deviation of image noise. The accuracies of the compared methods were tested in centered and uncentered cases by varing f from 200 to 2000 pixels

dant line correspondences. However, this method is not accurate enough in most cases, because it is a suboptimal method [30].

– *LPnL-Bar-LS* An efficient linear method, which parameterizes reference lines using barycentric coordinates, and uses homogeneous linear least squares to solve the PnL problem. This method also needs at least 6 lines [28].
– *ASPnL* An accurate subset-based PnL method, which is the improved version of **RPnL** method and represents the state-of-the-art method. A drawback of this method is that the computational time increases strongly for higher number of lines [28].

All methods are implemented via MATLAB, and are executed on a quad-core notebook with 2.5 GHz CPU and 4 GB RAM. The source code can be downloaded from https://sites.google.com/view/ping-wang-homepage.

4.1 Experiments with synthetic data

4.1.1 Synthetic data

We synthesized a virtual perspective camera with an image size of 640 × 480 pixels and generated n 3D reference lines, which are randomly distributed in the range of $[-2, 2] \times [-2, 2] \times [4, 8]$, in the camera frame. The focal length is

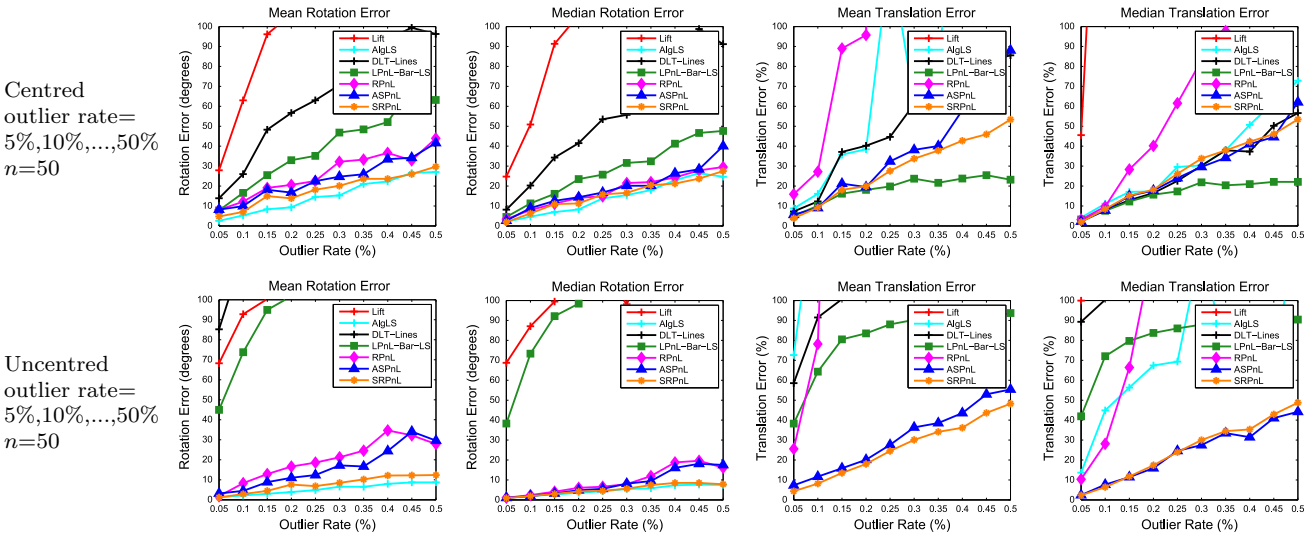


Fig. 7 Synthetic experimental results. n denotes the number of lines. The robustness against outliers was tested by varying outlier rate from 5 to 50%

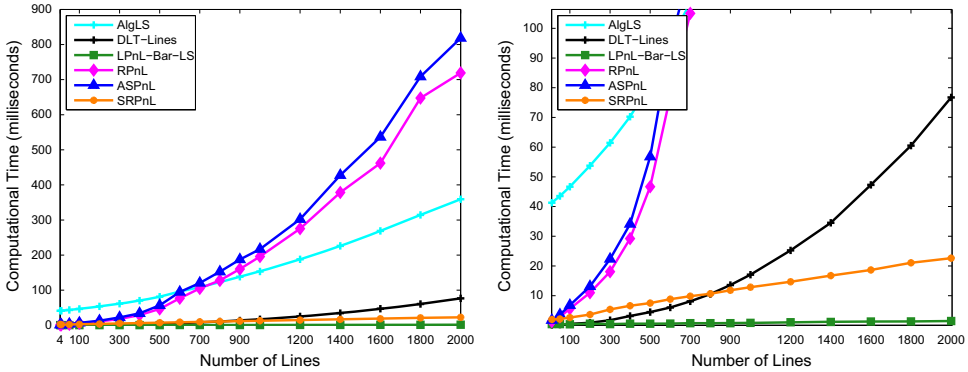


Fig. 8 The first shows a comparison of the computational efficiency of all methods. The second plot shows a zoomed-in version to clarify the comparison between the most efficient methods

chosen in the range from 200 to 2000 pixels. Then, we transformed these 3D lines into the world frame using the ground-truth of rotation R_{true} and translation T_{true}. Finally, we projected these 3D lines into the 2D image plane using the virtual calibrated camera. In the centered case, the 2D projections of the 3D lines spread in region $[0, 640] \times [0, 480]$ pixels on image plane. In the uncentered case, the 2D projections scatter in region $[0, 320] \times [0, 240]$ pixels on image plane. Depending on the experiment, a different level of white Gaussian noise was added to the 2D image plane.

The estimated rotation and translation were defined as R and T, respectively, and the errors of each were calculated as

$$E_{\text{rot}}(\text{degrees}) = \max_{k\in\{1,2,3\}} \cos^{-1}\left(r_{k,\text{true}}^T r_k\right) \times \frac{180}{\pi}$$
$$E_{\text{trans}}(\%) = \frac{\|t_{\text{true}} - t\|}{\|t\|} \times 100 \tag{22}$$

where $r_{k,\text{true}}$ and r_k are the k-th column of R_{true} and R, respectively. Note that this error metric might not be standard, but we use it in order to remain consistent with previous works.

4.1.2 The effect with the varying number of lines

The first simulated experiment investigated the performance of all methods with the varying number of reference lines for centered and uncentered cases. We varied the line number n from 4 to 20, and added zero-mean Gaussian noise with fixed deviation $\delta = 5$ pixels onto the image projections. At each n, 500 independent test sets were generated. We presented the mean and median rotation and translation error in Fig. 3. Generally speaking, the measured accuracy of **Lift** is very poor for all cases, even when the redundant lines are available. **DLT-Lines** and **LPnL-Bar-Ls** also offer poor estimation results for centered and uncentered cases, especially when n is small, because they are linear methods and ignore some nonlinear constraints. **AlgLs** is inaccurate on the mean rotation and translation error, due to the singularities of the Cayley parameterization. **RPnL** is also not accurate enough in most case, because it is a suboptimal method. Besides, **RPnL** sometimes produce a degenerate pose estimate very far from the ground-truth (See the mean translation error). In contrast, the **SRPnL** method stably reaches highly accurate

Table 1 Results of the methods on the VGG dataset in terms of average camera rotation error and average position error

	1	2	3	4	5	6	7	8	9	10
AlgLS										
$\Delta\theta$ (°)	0.969	0.475	1.031	1.119	4.289	1.510	3.244	1.031	0.969	0.475
ΔT (m)	0.452	0.267	0.558	0.432	1.486	0.745	1.656	0.558	0.452	0.266
DLT-Lines										
$\Delta\theta$ (°)	0.273	**0.066**	0.078	0.293	0.233	**0.081**	0.259	0.077	0.273	**0.066**
ΔT (m)	0.160	**0.041**	0.091	0.168	0.073	0.065	0.046	0.091	0.149	**0.041**
LPnL-Bar-LS										
$\Delta\theta$ (°)	0.109	0.186	0.172	0.344	0.088	0.408	0.913	0.172	0.108	0.187
ΔT (m)	0.056	0.148	0.096	0.175	0.026	0.179	0.372	0.096	0.056	0.148
RPnL										
$\Delta\theta$ (°)	0.474	0.288	0.156	1.379	1.570	0.829	2.580	0.156	0.474	0.288
ΔT (m)	0.206	0.144	0.056	0.645	0.711	0.439	1.403	0.055	0.206	0.144
ASPnL										
$\Delta\theta$ (°)	0.164	0.163	0.564	1.453	0.143	1.023	3.059	0.564	0.164	0.163
ΔT (m)	0.065	0.088	0.245	0.696	0.045	0.438	1.073	0.244	0.065	0.088
SRPnL										
$\Delta\theta$ (°)	**0.082**	0.151	**0.066**	**0.124**	**0.082**	0.138	**0.079**	**0.065**	**0.082**	0.151
ΔT (m)	**0.039**	0.076	**0.033**	**0.055**	**0.023**	**0.061**	**0.012**	**0.033**	**0.034**	0.076

Best results are in bold

results from $n = 4$ to 20, and offers accuracy comparable to the examined state-of-the-art method (**ASPnL**) for both centered and uncentered configurations.

4.1.3 The effect with the varying noise

The second simulated experiment tested the effects of noise on the accuracy of all methods. We fixed $n = 10$ and varied the noise deviation level δ from 1 to 15 pixels. At each noise level, we conducted 500 independent tests and reported the mean and median rotation and translation error for centered and uncentered cases. The results are shown in Fig. 4 and show that the mean rotation and translation error increased almost linearly with addition of noise for all methods. Similar to the previous experiment, the **SRPnL** method offers accuracy comparable or slightly better than that of the **ASPnL** method, and is still much better than others.

4.1.4 The small line set

Testing PnL methods are necessary when the input are noise small line sets, i.e., $n = 4$ or $n = 5$, because there are many applications, i.e., spacecraft pose estimation, satellite navigation [21], in which only 4 or 5 line features are provided. As can be seen in Fig. 5, only **SRPnL** and **ASPnL** can achieve highly accurate results when $n = 4$ or 5, and the mean rotation and translation error of our method is slightly better than **ASPnL**. The remaining methods are sensitive to the increased noise and have poor results. Note that **DLT-Lines** and **LPnL-Bar-Ls** can not be applied to $n = 4$ or 5, because they are linear method, which needs at least 6 lines.

4.1.5 The effect with the varying focal length

In this test, the focal length f was varied from 200 to 2000 pixels, and the standard deviation of image noise was fixed to 5 pixels. Figure 6 shows the result of evaluation. Most of the methods are not stable for centered and uncentered cases with varying focal length. On the contrary, the accuracy of **SRPnL** is comparable to **ASPnL** for both centered and uncentered cases, which is much better than other methods.

4.1.6 The robustness against outliers

This section tested the robustness to outlying correspondences for all methods. We used $n = 50$ lines and varied the outlier rate from 5 to 50%. As indicated in Fig. 7, **Lift** yields unstable results with low accuracy in overall evaluations. Note that the mean and median translation errors of **Lift** are out of the range. **AlgLS** and **RPnL** have high rotation accuracy but with low translation accuracy. **DLT-Lines** and **LPnL-Bar-LS** are stable in centered case, but are not stable in uncentered case. In contrast, **SRPnL** achieves reliable results in centered and uncentered cases, slightly outperforming **ASPnL**.

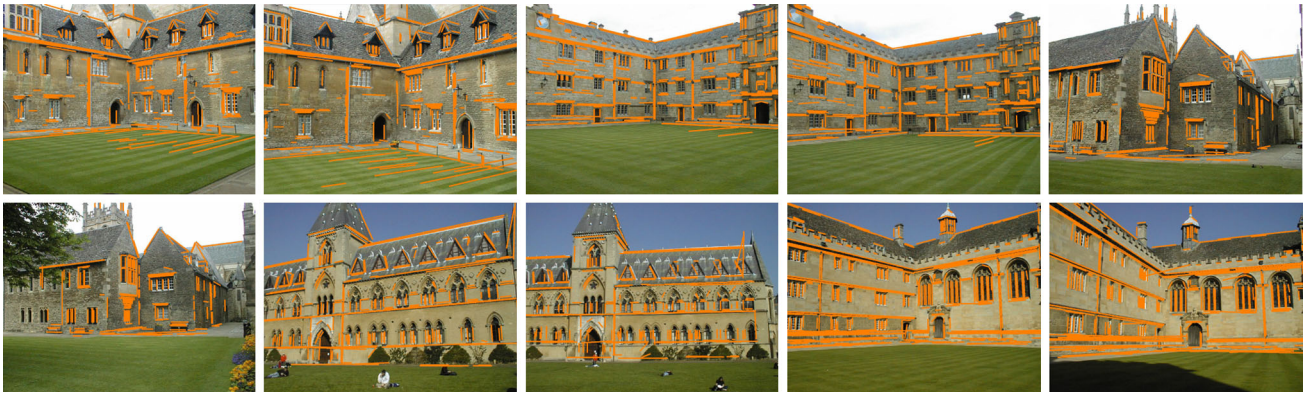


Fig. 9 Images from the VGG dataset overlaid with reprojections of 3D line segments using our estimated camera pose

4.1.7 Computational efficiency

Figure 8 shows the computational time with varying $4 \leq n \leq 2000$ and fixed $\delta = 2$. For each n, we conducted 500 tests and showed the average running time in milliseconds (ms). As evident, our method has high efficiency and is even faster than linear **DLT-Lines** method for $n \geq 800$ lines. This is primarily due to two reasons: our method only needs to solve a fifteenth-order and fourth-order univariate polynomial; and the vectorization strategy is implemented, instead of the iterative way. This result indicates that our method is suitable for real-time applications such as augmented reality and visual SLAM, where $n \geq 100$ is not a rare situation. **LPnP-Bar-Ls** is faster than our method, However, our method is still very competitive, especially considering its high accuracy and general applicability. Faster performance can also be acquired for the **SRPnL** by using a C++ implementation, which will be published upon completion.

4.2 Experiments with real images

We also tested our and other methods using ten real images from the VGG Multiview Dataset. VGG dataset contains image sequences of buildings with detected 2D line segments, their reconstructed 3D line segments, and camera projection matrices. Each method was run on these images, and the mean camera orientation error $\Delta\theta = |\Delta\theta|$ and the mean position error $\Delta T = \|T_{\text{estimation}} - T_{\text{true}}\|$ are given in Table 1. The top results for each image were typeset in bold, from which we can observe that our **SRPnL** method provided accuracy and precision comparable or better than other state-of-the-art methods. In addition, the our estimated camera poses were used to reproject the 3D lines onto the images to visualize the results. As shown in Fig. 9, it demonstrates that the proposed method can recover the camera pose.

5 Conclusion

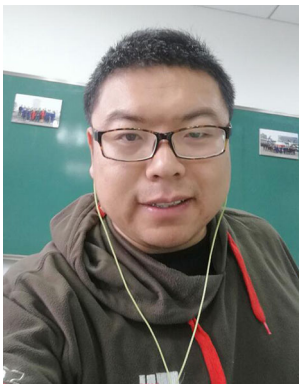
In this work, we developed a novel closed-form solution to the perspective-n-line pose problem. The key process of our method is to solve univariate polynomials, and the derivations of our method are easy to understand. The experiment results also demonstrated that our method is substantially faster, yet equally accurate and robust compared with the state-of-the-art methods. Future work involves examination of the degenerate line configurations. To facilitate further improvement, we have made the source code publicly available.

Acknowledgements This study was supported by the National Natural Science Foundation of China (No. 61473148), and the Foundation of China Shipbuilding Industry Company Limited (No. 6141B0405103).

References

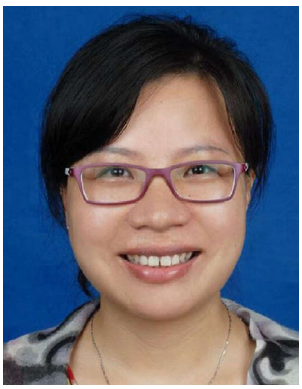
1. Abdelaziz, Y.I.: Direct linear transformation from comparator coordinates in close-range photogrammetry. In: Asp Symposium on Close-Range Photogrammetry in Illinois (1971)
2. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 578–589 (2003)
3. Atiya, S., Hager, G.D.: Real-time vision-based robot localization. IEEE Trans. Rob. Autom. **9**(6), 785–800 (1993)
4. Billinghurst, M., Clark, A., Lee, G., et al.: A survey of augmented reality. Found. Trends® Human–Comput. Interact. **8**(2–3), 73–272 (2015)
5. Foulds, L.R.: Optimization Techniques: an Introduction. Springer (2012)
6. Caglioti, V.: The planar three-line junction perspective problem with application to the recognition of polygonal patterns. Pattern Recognit. **26**(11), 1603–1618 (1993)
7. Chen, H.H.: Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6), 530–541 (1991)
8. Dhome, M., Richetin, M., Lapreste, J.T.: Determination of the attitude of 3D objects from a single perspective view. IEEE Trans. Pattern Anal. Mach. Intell. **11**(12), 1265–1278 (1989)

9. Dornaika, F., Garcia, C.: Pose estimation using point and line correspondences. Real Time Imaging **5**(3), 215–230 (1999)
10. Flannery, B.P., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, Cambridge (1987)
11. Gander, W.: Least Squares Fit of Point Clouds. Springer, Berlin (1997)
12. Hesch, J.A., Roumeliotis, S.I.: A direct least-squares (DLS) method for P*n*P. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 383–390. IEEE (2011)
13. Horaud, R.: Iterative Pose Computation from Line Correspondences. Elsevier, Amsterdam (1999)
14. Kneip, L., Li, H., Seo, Y.: UPnP: an optimal $O(n)$ solution to the absolute pose problem with universal applicability. In: European Conference on Computer Vision, pp. 127–142. Springer (2014)
15. Kukelova, Z., Bujnak, M., Pajdla, T.: Automatic generator of minimal problem solvers. In: European Conference on Computer Vision, pp. 302–315 (2008)
16. Kumar, R., Hanson, A.R.: Robust Methods for Estimating Pose and a Sensitivity Analysis. Academic Press, Cambridge (1994)
17. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: an accurate $O(n)$ solution to the P*n*P problem. Int. J. Comput. Vis. **81**(2), 155 (2009)
18. Li, S., Xu, C., Xie, M.: A robust $O(n)$ solution to the perspective-n-point problem. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1444–1450 (2012)
19. Lu, C.P., Hager, G.D., Mjolsness, E.: Fast and globally convergent pose estimation from video images. IEEE Trans. Pattern Anal. Mach. Intell. **22**(6), 610–622 (2000)
20. Mirzaei, F.M., Roumeliotis, S.I.: Globally optimal pose estimation from line correspondences. In: IEEE International Conference on Robotics and Automation, pp. 5581–5588 (2011)
21. Mourikis, A.I., Trawny, N., Roumeliotis, S.I., Johnson, A.E., Ansar, A., Matthies, L.: Vision-aided inertial navigation for spacecraft entry, descent, and landing. IEEE Trans. Robot. **25**(2), 264–280 (2009)
22. Nakano, G.: Globally optimal DLS method for P*n*P problem with Cayley parameterization. In: BMVC, pp. 78–1 (2015)
23. Přibyl, B., Zemčík, P., Čadík, M.: Camera pose estimation from lines using Plücker coordinates. In: Proceedings of the British Machine Vision Conference, pp. 45.1–45.12 (2015)
24. Qin, L.J., Feng, Z.: A new method for pose estimation from line correspondences. Acta Autom. Sin. **34**(2), 130–134 (2008)
25. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104–4113 (2016)
26. Silva, M., Ferreira, R., Gaspar, J.: Camera calibration using a color-depth camera: points and lines based DLT including radial distortion. In: Proceedings of IROS Workshop in Color-Depth Camera Fusion in Robotics (2012)
27. Wang, P., Xu, G., Cheng, Y., Yu, Q.: A simple, robust and fast method for the perspective-n-point problem. Pattern Recognit. Lett. **108**, 31–37 (2018)
28. Xu, C., Zhang, L., Cheng, L., Koch, R.: Pose estimation from line correspondences: a complete analysis and a series of solutions. IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1209–1222 (2017)
29. Zhang, L., Koch, R.: Hand-held monocular slam based on line segments. In: Machine Vision and Image Processing Conference, pp. 7–14 (2011)
30. Zhang, L., Xu, C., Lee, K.M., Koch, R.: Robust and efficient pose estimation from line correspondences. In: Asian Conference on Computer Vision, pp. 217–230 (2012)
31. Zheng, Y., Kuang, Y., Sugimoto, S., Åström, K., Okutomi, M.: Revisiting the P*n*P problem: a fast, general and optimal solution. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 2344–2351. IEEE (2013)

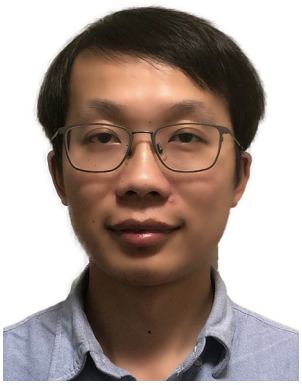


Ping Wang received the B.S. degree in electrical and information engineering in 2011 and the M.S. degree in control theory and control in 2014. He is currently pursuing the Ph.D. degree with Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include computer vision, machine learning, signal processing and intelligence algorithms.

Guili Xu is a professor at College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He received the Ph.D. degree from the Jiangsu University in 2002. His current research interests are the photoelectric measure, computer vision and intelligent system.



Yuehua Cheng is an associated professor at College of Astronautics, Nanjing University of Aeronautics and Astronautics. She received her PhD degree concentrating in control theory and control engineering from Nanjing University of Aeronautics and Astronautics in 2012. Her current research interest is fault diagnosis and lifespan prediction applied on satellite attitude control system.



Qida Yu is currently pursuing the Ph.D. degree with Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research interests are in the areas of digital image processing and intelligent system.