IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 10, 2020, accepted August 20, 2020, date of publication September 2, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021313

An Efficient Solution to the Perspective-n-Point Problem for Camera With Unknown Focal Length

BINGYI ZHOU, ZIQIANG CHEN, AND QINGHUA LIU
Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541000, China

Corresponding author: Ziqiang Chen (chenziqiang@guet.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61861011 and Grant 61871425, in part by the Guangxi Natural Science Foundation under Grant 2016GXNSFAA380036 and Grant 2018GXNSFAA138091, in part by the Science and Technology on Near-Surface Detection Laboratory Foundation under Grant TCGZ2017A010, and in part by the Guangxi Major Science and Technology Project under Grant AA17204093.

ABSTRACT To achieve lower time consumption in pose estimation of an uncalibrated camera, a novel and efficient method to the Perspective-n-Point problem (PnP) is proposed in this article. An uncalibrated camera refers to a camera whose focal length, one of the key parameters of PnP, is unknown. Compared to the traditional methods of PnP for uncalibrated cameras, the focal length is eliminated in our proposed method by projecting the given three-dimensional points vertically onto the image plane. Then, a new plane is determined by the projection points and the corresponding 3D points. Further, utilizing the orthogonal characteristic between the normal vector of the plane and the in-plane vector, we construct the PnP problem for an uncalibrated camera as a 20th-order polynomial system, which can easily be solved. As compared to recent PnP methods for uncalibrated cameras, our method has comparable accuracy at a lower computation cost when handling sets with between 8 and 40 points.

INDEX TERMS Perspective-n-point problem (PnP), pose estimation, uncalibrated camera.

I. INTRODUCTION

The Perspective-n-Point (PnP) problem is based on the perspective geometry of the pinhole imaging model. As an important part of computer vision, the PnP problem has been widely assessed and used to estimate the orientation and translation of the camera from n pairs of corresponding points. Due to the above, the PnP problem has also been defined as the ''Location Determination Problem'' (LDP) by Martin *et al.* [1].

In the traditional PnP problem, the focal length which can be obtained during camera calibration is usually used as a known parameter. This type of PnP problem is generally called the calibrated cases. Correspondingly, the PnP problem for the cameras with unknown intrinsic parameters is called the uncalibrated cases. In recent years, the PnP problem for uncalibrated cameras has attracted widespread attention from relevant researchers.

The PnP problem for calibrated cases has been extensively researched in the literature. Gao *et al.* [2] gave a standard to classify the P3P problem and proposed a method, named CASSC, with complete solution. Ke and Roumeliotis [3] proposed an algebraic solution by decomposing the rotation matrix into three directions and establishing a polynomial containing only one unknown. Compared to the CASSC, Ke's method performed better in terms of accuracy, robustness and running speed. Masselli and Zell [4] introduced an intermediate co-ordinate system and presented an algebraic solution which can be easily solved, thus obtaining a higher running speed. For the generic cases, Lepetit *et al.* [5] presented an effective method marked as EPnP_GN (Efficient Perspective-n-Point with Gaussian-Newton), whose time complexity is O(n) and which performs well in terms of both robustness and accuracy. However, all of these methods were established on one precondition, a known focal length [1], [5] or known directions from the three key points to the camera [3], [4], which is actually another representation of the focal length.

For the uncalibrated cases, Bujnak *et al.* [6] presented an approach called P4P to the absolute pose problem from four pairs of 2D-3D point correspondences and provided a polynomial system solution based on the Gröbner

The associate editor coordinating the review of this manuscript and approving it for publication was Shuhan Shen.

basis method. Wu [7] proposed a method named P3.5P. The very name suggests that it requires three pairs of complete points and either the x or y value of one pair of point as the input information. Compared to P4P, P3.5P performed better in terms of efficiency. Inspired by EPnP_GN, Penate-Sanchez *et al.* [8] formulated the PnP problem as an equation system, utilizing exhaustive linearization techniques to solve the underdetermined equation, called UPnP (Uncalibrated PnP). UPnP performed well in terms of accuracy; however, the time cost of UPnP is high.

In order to meet the application requirement of lower time cost, we present an efficient method in this article, which can be utilized to locate an illegal flying UAV (Unmanned Aerial Vehicle) by capturing the image signals from the UAV. In our approach, we project the 3D points vertically onto the image surface, which is perpendicular to the focal length f, thus eliminating the unknown variable f. Then, a relationship between the projection points and the 2D points is found. Using the properties of the normal vector, we establish a polynomial system to solve the rotation matrix. On this basis, the translation vector is further solved by using the distance constraint.

To evaluate the performance of our method, several experiments were conducted. For every case, we carried out statistical characteristic analyses in 500 experimental data. The results prove that our method achieves a comparable accuracy and higher stability at a lower computational cost, when the number of correspondences is between 8 and 40. In particular, the correct prediction rate of the proposed method can reach more than 95 percent for certain numbers of correspondences while that of UPnP is stable at roughly 83 percent.

II. PROPOSED METHOD

A. PROBLEM DEFINITION

Given ${}^{W}\boldsymbol{P}_i = (X_i, Y_i, Z_i)^{\mathrm{T}}, (i = 1, 2, 3, 4, 5, 6)$ as the positions in the world co-ordinate system of six known key points and the correspondences ${}^{C}\boldsymbol{P}_i = (x_i, y_i, f)^{\mathrm{T}}, (i = 1, 2, 3, 4, 5, 6)$ as the image positions in the camera co-ordinate system, where the value of focal length f is unknown. Defined the vector ${}^{C}\boldsymbol{n}$ as the normal vector of the image surface, which is obviously denoted as ${}^{C}\boldsymbol{n} = (0, 0, 1)^{\mathrm{T}}$ in the camera co-ordinate system. The image plane is parallel to the plane ${}^{c}Y^{c}O^{c}X$. Our objective is to determine the transformation matrix $[\boldsymbol{R} \mid \boldsymbol{t}]$ with respect to the world co-ordinate system. The rotation matrix can be parameterized as follows:

$$\boldsymbol{R} = \boldsymbol{Rot}(X, \alpha)\,\boldsymbol{Rot}(Y, \beta)\,\boldsymbol{Rot}(Z, \theta), \tag{1}$$

where $\boldsymbol{Rot}(X, \alpha), \boldsymbol{Rot}(Y, \beta)$ and $\boldsymbol{Rot}(Z, \theta)$ indicate the rotation matrices around the $X-axis$, $Y-axis$, and $Z-axis$ respectively.

From the geometry of the problem, as shown in Figure 1.(a), the point O, which refers to the image co-ordinate origin, and ${}^{C}\boldsymbol{P}_i\ (i = 1, 2, 3, 4, 5, 6)$ all lie on the image surface which is parallel to ${}^{c}Y^{c}O^{c}X$ surface and spaced with an unknown focal length f. The line $O^{c}O$ lies on the ${}^{c}Z$-axis.

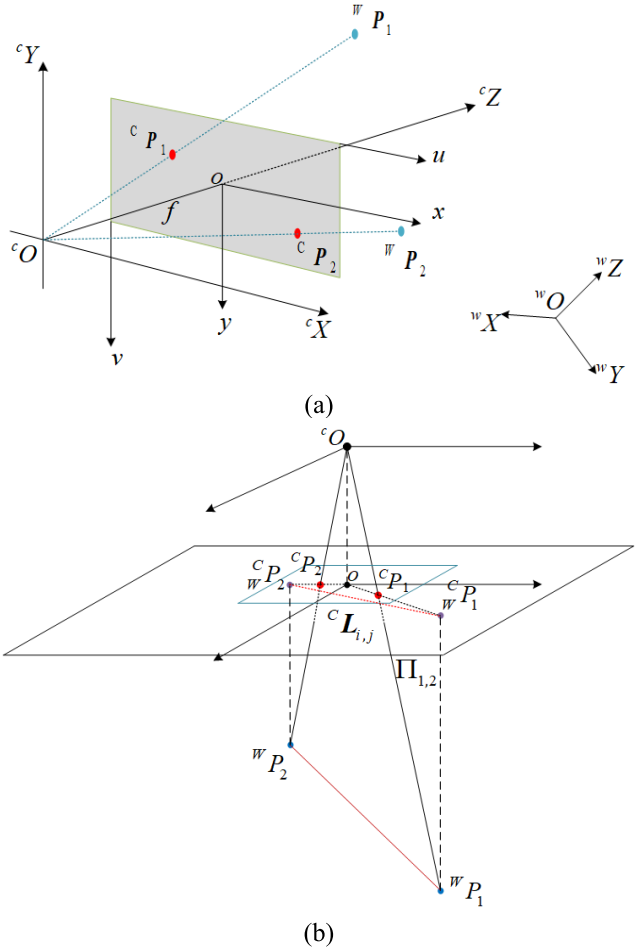


FIGURE 1. (a) The perspective geometry of the pinhole imaging model. $\{{}^{W}P\}$ denotes the 3D points in the world co-ordinate system, and $\{{}^{C}P\}$ are the corresponding pixel points in the camera co-ordinate system. (b) Projecting the points ${}^{W}P_i$ vertically onto the image plane, such that we have the corresponding orthogonal projections ${}^{C}_{W}P_i$ and the plane determined by the points ${}^{W}P_i, {}^{W}P_j, {}^{C}_{W}P_j$, and ${}^{C}_{W}P_i, (i \neq j)$, denoted by $\Pi_{i,j}, (i \neq j)$.

As illustrated in Figure 1.(b), we select one point ${}^{W}\boldsymbol{P}_1$ from ${}^{W}\boldsymbol{P}_i, (i = 1, 2, 3, 4, 5, 6)$ and then project the vectors $\overrightarrow{{}^{W}\boldsymbol{P}_1\,{}^{W}\boldsymbol{P}_i}, (i = 2, 3, 4, 5, 6)$ vertically onto the image plane to eliminate the unknown focal length, where the vectors $\overrightarrow{{}^{C}_{W}\boldsymbol{P}_1\,{}^{C}_{W}\boldsymbol{P}_i}, (i = 2, 3, 4, 5, 6)$ are the corresponding orthogonal projections, denoted as ${}^{C}\boldsymbol{L}_i$. Meanwhile, we define a plane $\Pi_{1,i}, (i = 2, 3, 4, 5, 6)$ determined by the points ${}^{W}\boldsymbol{P}_1$, ${}^{W}\boldsymbol{P}_i$, ${}^{C}_{W}\boldsymbol{P}_i$, and ${}^{C}_{W}\boldsymbol{P}_1$, such that we have the following:

$${}^{C}\boldsymbol{n}_i^{\mathrm{T}} \cdot {}^{C}\boldsymbol{L}_i = 0, (i = 2, 3, 4, 5, 6), \tag{2}$$

where ${}^{C}\boldsymbol{n}_i$ is a unit vector perpendicular to $\Pi_{1,i}$, which can be written as follows:

$${}^{C}\boldsymbol{n}_i = {}^{C}\boldsymbol{n} \times \left(\boldsymbol{R} \cdot \frac{\overrightarrow{{}^{W}\boldsymbol{P}_1\,{}^{W}\boldsymbol{P}_i}}{\left\| \overrightarrow{{}^{W}\boldsymbol{P}_1\,{}^{W}\boldsymbol{P}_i} \right\|} \right), (i = 2, 3, 4, 5, 6). \tag{3}$$

As depicted in Figure 2, the scalars $\lambda_i, (i = 2, 3, 4, 5, 6)$ stretch the vectors $\overrightarrow{\boldsymbol{O}^{C}\boldsymbol{P}_i}, (i = 2, 3, 4, 5, 6)$ to find vectors

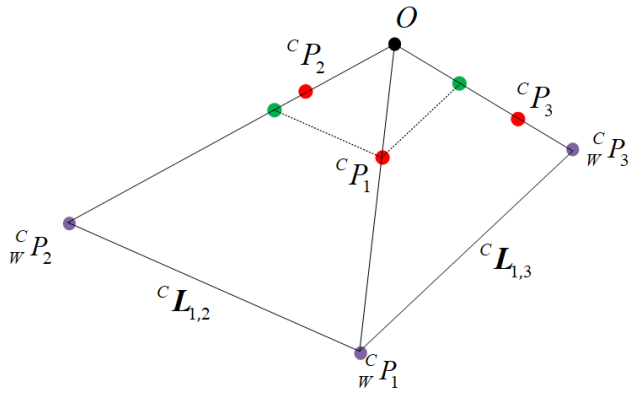


FIGURE 2. The relationships between 2D points $(x_i, y_i)^{\mathrm{T}}$ and the points $\left\{{}^{C}_{W}P\right\}$ projected vertically by the corresponding 3D points $\left\{{}^{W}P\right\}$.

which are in same directions as $^{C}\boldsymbol{L}_i$, $(i = 2, 3, 4, 5, 6)$, respectively, such that the $^{C}\boldsymbol{L}_i$ can be described as:

$$^{C}\boldsymbol{L}_i = -\mu\,^{C}\boldsymbol{v}_i, \tag{4}$$

where $\mu = \dfrac{\left\|\overrightarrow{O^{C}_{W}P_1}\right\|}{\left\|\overrightarrow{O^{C}P_1}\right\| \Big/ \left\|\overrightarrow{O^{C}P_1}\right\|}$, and $^{C}\boldsymbol{v}_i = \dfrac{\overrightarrow{O^{C}P_1}}{\left\|\overrightarrow{O^{C}P_1}\right\|} - \lambda_i \dfrac{\overrightarrow{O^{C}P_i}}{\left\|\overrightarrow{O^{C}P_i}\right\|}$.
Then, (2) can be transformed into

$$^{C}\boldsymbol{n}_i^{\mathrm{T}} \cdot {}^{C}\boldsymbol{v}_i = 0. \tag{5}$$

Equation (5) is a system of five equations with eight unknowns. In order to solve it, the associations between λ_2 and λ_i, $(i = 3, 4, 5, 6)$ are introduced, as illustrated in (6) – (9), as shown at the bottom of the page, to reduce the numbers of unknowns.

According to the definition of $^{C}\boldsymbol{L}_i$, they also can be expressed as follows:

$$^{C}\boldsymbol{L}_i = \boldsymbol{R} \cdot \overrightarrow{{}^{W}P_1\,{}^{W}P_i} - {}^{C}\boldsymbol{n}^{\mathrm{T}} \cdot \boldsymbol{R} \cdot \overrightarrow{{}^{W}P_1\,{}^{W}P_i} \cdot {}^{C}\boldsymbol{n}. \tag{6}$$

Such that a proportional constraint based on similar triangles can be established, through (4) and (6):

$$\frac{\left\|{}^{C}\boldsymbol{v}_2\right\|}{\left\|{}^{C}\boldsymbol{L}_2\right\|} = \mu = \frac{\left\|{}^{C}\boldsymbol{v}_i\right\|}{\left\|{}^{C}\boldsymbol{L}_i\right\|}. \tag{7}$$

As the vectors $^{C}\boldsymbol{v}_i$ and $^{C}\boldsymbol{L}_i$ point in the same directions, respectively, the length relationship illustrated in (7) can be converted into a vector relationship, as follows:

$$^{C}\boldsymbol{v}_2^{\mathrm{T}} \cdot {}^{C}\boldsymbol{L}_i = {}^{C}\boldsymbol{v}_i^{\mathrm{T}} \cdot {}^{C}\boldsymbol{L}_2. \tag{8}$$

Substituting $^{C}\boldsymbol{v}_i = \dfrac{\overrightarrow{O^{C}P_1}}{\left\|\overrightarrow{O^{C}P_1}\right\|} - \lambda_i \dfrac{\overrightarrow{O^{C}P_i}}{\left\|\overrightarrow{O^{C}P_i}\right\|}$ into (8), λ_i can be written as:

Furthermore, it is worth noting that the vectors $^{C}\boldsymbol{n} \times {}^{C}\boldsymbol{v}_2$ and $^{C}\boldsymbol{n}_2$ are in the same direction, which was used to eliminate λ_2. Therefore, the scalar λ_2 can be written as:

$$\lambda_2 = (y_1 \cdot A_2 - x_1 \cdot B_2) \big/ (y_2 \cdot A_2 - x_2 \cdot B_2), \tag{10}$$

where

$$\begin{bmatrix} A_i \\ B_i \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \boldsymbol{R} \cdot \begin{bmatrix} X_1 - X_i \\ Y_1 - Y_i \\ Z_1 - Z_i \end{bmatrix} = \overrightarrow{{}^{C}_{W}\boldsymbol{P}_1\,{}^{C}_{W}\boldsymbol{P}_i},$$

for $i = 2, 3, 4, 5, 6$.

By substituting (9) and (10) into (5), we can obtain a system of equations with only three unknown quantities:

$$\begin{cases} A_3 \cdot B_3 \cdot [C_2 \cdot (x_3 \cdot B_3 - y_3 \cdot A_3) - C_3 \\ \qquad \cdot (x_2 \cdot B_2 - y_2 \cdot A_2)] = 0 \\ A_4 \cdot B_4 \cdot [C_2 \cdot (x_4 \cdot B_4 - y_4 \cdot A_4) - C_4 \\ \qquad \cdot (x_2 \cdot B_2 - y_2 \cdot A_2)] = 0 \\ A_5 \cdot B_5 \cdot [C_2 \cdot (x_5 \cdot B_5 - y_5 \cdot A_5) - C_5 \\ \qquad \cdot (x_2 \cdot B_2 - y_2 \cdot A_2)] = 0 \\ A_6 \cdot B_6 \cdot [C_2 \cdot (x_6 \cdot B_6 - y_6 \cdot A_6) - C_6 \\ \qquad \cdot (x_2 \cdot B_2 - y_2 \cdot A_2)] = 0, \end{cases} \tag{11}$$

with

$$\begin{cases} x_2 \cdot B_2 - y_2 \cdot A_2 \neq 0 \\ x_3 \cdot B_3 + y_3 \cdot A_3 \neq 0 \\ x_4 \cdot B_4 + y_4 \cdot A_4 \neq 0 \\ x_5 \cdot B_5 + y_5 \cdot A_5 \neq 0 \\ x_6 \cdot B_6 + y_6 \cdot A_6 \neq 0, \end{cases}$$

where $C_i = y_1 \cdot x_i - x_1 \cdot y_i$, $(i = 2, 3, 4, 5, 6)$.

The formula $A_i \cdot B_i = 0$, $(i = 3, 4, 5, 6)$ can be found only when $\overrightarrow{{}^{C}_{W}\boldsymbol{P}_1\,{}^{C}_{W}\boldsymbol{P}_i}$ is parallel to an axis, such that (11) can be written as follows:

$$\begin{cases} \sigma_1 \cos\alpha + \varepsilon_1 \sin\alpha + \tau_1 = 0 \\ \sigma_2 \cos\alpha + \varepsilon_2 \sin\alpha + \tau_2 = 0 \\ \sigma_3 \cos\alpha + \varepsilon_3 \sin\alpha + \tau_3 = 0 \\ \sigma_4 \cos\alpha + \varepsilon_4 \sin\alpha + \tau_4 = 0, \end{cases} \tag{12}$$

where

$$\begin{cases} \sigma_i = a_i \sin\theta + b_i \cos\theta \\ \varepsilon_i = a_i \sin\beta \cos\theta - b_i \sin\beta \sin\theta - c_i \cos\beta \\ \tau_i = d_i \cos\beta \cos\theta - e_i \cos\beta \sin\theta + f_i \sin\beta, \end{cases} \quad (i = 1, 2, 3, 4),$$

$$\lambda_i = \frac{\left[\left({}^{C}\boldsymbol{L}_2 - {}^{C}\boldsymbol{L}_i\right)^{\mathrm{T}} \cdot \overrightarrow{O^{C}P_1} \Big/ \left\|\overrightarrow{O^{C}P_1}\right\| + \lambda_2\,{}^{C}\boldsymbol{L}_i^{\mathrm{T}} \cdot \overrightarrow{O^{C}P_2} \Big/ \left\|\overrightarrow{O^{C}P_2}\right\|\right]}{{}^{C}\boldsymbol{L}_2^{\mathrm{T}} \cdot \overrightarrow{O^{C}P_i} \Big/ \left\|\overrightarrow{O^{C}P_i}\right\|}. \tag{9}$$

with

$$\begin{cases} \begin{bmatrix} a_{i-2} \\ b_{i-2} \\ c_{i-2} \end{bmatrix} = \begin{bmatrix} X_2 - X_i & X_1 - X_2 & X_1 - X_i \\ Y_2 - Y_i & Y_1 - Y_2 & Y_1 - Y_i \\ Z_2 - Z_i & Z_1 - Z_2 & Z_1 - Z_i \end{bmatrix} \cdot \begin{bmatrix} y_1 x_2 x_i \\ x_1 x_2 y_i \\ -x_1 y_2 x_i \end{bmatrix} \\ \begin{bmatrix} d_{i-2} \\ e_{i-2} \\ f_{i-2} \end{bmatrix} = \begin{bmatrix} X_2 - X_i & X_1 - X_2 & X_1 - X_i \\ Y_2 - Y_i & Y_1 - Y_2 & Y_1 - Y_i \\ Z_2 - Z_i & Z_1 - Z_2 & Z_1 - Z_i \end{bmatrix} \cdot \begin{bmatrix} x_1 y_2 y_i \\ y_1 y_2 x_i \\ -y_1 x_2 y_i \end{bmatrix}, \end{cases}$$

for $i = 3, 4, 5, 6$. By solving (12), we can obtain the following:

$$\begin{cases} \cos\alpha = \dfrac{\varepsilon_1\tau_2 - \tau_1\varepsilon_2}{\sigma_1\varepsilon_2 - \varepsilon_1\sigma_2} = \dfrac{\varepsilon_1\tau_3 - \tau_1\varepsilon_3}{\sigma_1\varepsilon_3 - \varepsilon_1\sigma_3} = \dfrac{\varepsilon_1\tau_4 - \tau_1\varepsilon_4}{\sigma_1\varepsilon_4 - \varepsilon_1\sigma_4} \\ \sin\alpha = \dfrac{\tau_1\sigma_2 - \sigma_1\tau_2}{\sigma_1\varepsilon_2 - \varepsilon_1\sigma_2} = \dfrac{\tau_1\sigma_3 - \sigma_1\tau_3}{\sigma_1\varepsilon_3 - \varepsilon_1\sigma_3} = \dfrac{\tau_1\sigma_4 - \sigma_1\tau_4}{\sigma_1\varepsilon_4 - \varepsilon_1\sigma_4}. \end{cases} \tag{13}$$

Rearranging (13), we get:

$$\begin{cases} \sigma_1(\tau_2\varepsilon_3 - \varepsilon_2\tau_3) - \sigma_2(\tau_1\varepsilon_3 - \varepsilon_1\tau_3) - \sigma_3(\tau_2\varepsilon_1 - \varepsilon_2\tau_1) = 0 \\ \sigma_1(\tau_2\varepsilon_4 - \varepsilon_2\tau_4) - \sigma_2(\tau_1\varepsilon_4 - \varepsilon_1\tau_4) - \sigma_4(\tau_2\varepsilon_1 - \varepsilon_2\tau_1) = 0 \\ (\varepsilon_1\tau_4 - \tau_1\varepsilon_4)^2 + (\tau_1\sigma_4 - \sigma_1\tau_4)^2 = (\sigma_1\varepsilon_4 - \varepsilon_1\sigma_4)^2. \end{cases} \tag{14}$$

Substituting $\sin\beta\cos\beta = \sin 2\beta/2$, $\sin^2\beta = (1 - \cos 2\beta)/2$, and $\cos^2\beta = (1 + \cos 2\beta)/2$ into the first two equations of (14), we have the following:

$$\begin{cases} \partial_1 \sin 2\beta + \partial_2 \cos 2\beta + \partial_3 = 0 \\ \partial_4 \sin 2\beta + \partial_5 \cos 2\beta + \partial_6 = 0 \end{cases} \tag{15}$$

Equation (15) can be transformed into:

$$\begin{cases} \cos 2\beta = \dfrac{\partial_3\partial_4 - \partial_1\partial_6}{\partial_1\partial_5 - \partial_2\partial_4} \\ \sin 2\beta = \dfrac{\partial_2\partial_6 - \partial_3\partial_5}{\partial_1\partial_5 - \partial_2\partial_4}. \end{cases} \tag{16}$$

Substituting (16) into $\cos^2 2\beta + \sin^2 2\beta = 1$, we have the following:

$$(\partial_3\partial_4 - \partial_1\partial_6)^2 + (\partial_2\partial_6 - \partial_3\partial_5)^2 = (\partial_1\partial_5 - \partial_2\partial_4)^2 \tag{17}$$

By substituting $\sin^2\theta + \cos^2\theta = 1$ into (17), we can obtain the following:

$$\sum_{i=0}^{5} k_{2i}\cos^{2i}\theta = \sin\theta \sum_{i=1}^{5} l_{2i}\cos^{2i-1}\theta. \tag{18}$$

Squaring both sides of (18), then substituting $\sin^2\theta = 1 - \cos^2\theta$ and making $x = \cos\theta$, the model can be expressed as a 20th-order polynomial:

$$f(x) = \sum_{i=0}^{10} \delta_{2i} x^{2i} = 0. \tag{19}$$

In this section, we first define a normal vector of a plane determined by two 3D points and their perpendicular projection points. Then, the vector established by these two projection points is represented by 2D points $(x_i, y_i)^{\mathrm{T}}$ through a proportional relationship. The PnP problem is transformed into solving the polynomial (19) whose odd-numbered coefficients are zero; that is, $\delta_i = 0$, $i = 1, 3, \ldots, 19$. For general configurations, (5) can be transformed to (19) and the rotation matrix can finally be determined.

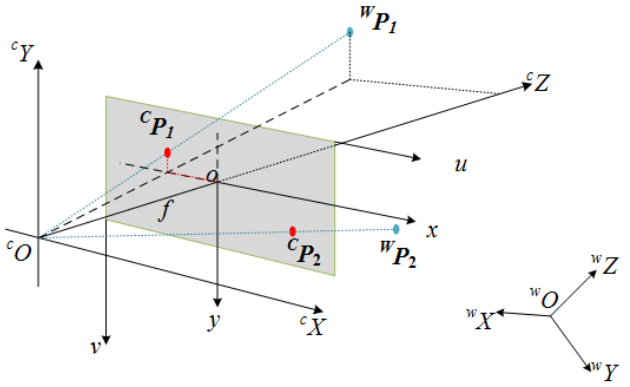


FIGURE 3. Similar triangles in the projection model.

B. SOLVING THE TRANSLATION VECTOR

Solving the translation vector $\boldsymbol{t} = [t_X, t_Y, t_Z]^{\mathrm{T}}$ and the focal length f by using the similar triangles constraint illustrated in Figure 3, which can be express as follows:

$$\boldsymbol{R}^{\mathrm{W}}\boldsymbol{P}_i + \boldsymbol{t} = s_i [x_i, y_i, f]^{\mathrm{T}}, \tag{20}$$

where s_i is the scaling of projection. Equation (20) is a system of equations with four unknown quantities. In order to solve it, we express ${}^{W}\boldsymbol{P}_i$ as:

$$\left[{}^{\mathrm{W}}\boldsymbol{P}_i^{\mathrm{T}}, 1\right] = \sum_{i=1}^{4} \varphi_i {}^{\mathrm{W}}\boldsymbol{C}_i^{\mathrm{T}},$$

where ${}^{\mathrm{W}}\boldsymbol{C}_1 = [0,0,0,1]^{\mathrm{T}}$, ${}^{\mathrm{W}}\boldsymbol{C}_2 = [1,0,0,1]^{\mathrm{T}}$., ${}^{\mathrm{W}}\boldsymbol{C}_3 = [0,1,0,1]^{\mathrm{T}}$, and ${}^{\mathrm{W}}\boldsymbol{C}_4 = [0,0,1,1]^{\mathrm{T}}$. Obviously, ${}^{\mathrm{C}}\boldsymbol{C}_i = [\mathrm{R}|\mathrm{t}]{}^{\mathrm{W}}\boldsymbol{C}_i$, where ${}^{\mathrm{C}}\boldsymbol{C}_i$ indicate the corresponding estimates in the camera co-ordinate system. According to the distance constraint, we can obtain the following:

$$\left\| {}^{C}\boldsymbol{C}_i - {}^{C}\boldsymbol{C}_j \right\|^2 = \left\| {}^{W}\boldsymbol{C}_i - {}^{W}\boldsymbol{C}_j \right\|^2, (i \neq j). \tag{21}$$

The translation vector and focal length can be solved by using (20) and (21).

C. METHOD

A non-linear equation can be set up, according to (19), for every six correspondences, which is written as:

$$\begin{cases} f_1(x) = \sum_{i=0}^{10} \delta_{2i} x^{2i} = 0 \\ f_2(x) = \sum_{i=0}^{10} \delta_{2i} x^{2i} = 0 \\ \vdots \\ f_{n-5}(x) = \sum_{i=0}^{10} \delta_{2i} x^{2i} = 0. \end{cases} \tag{22}$$

To reduce the computational complexity of this non-linear system, we use the least squares residual to find the local minima of (22). We defined a cost function, J, as follows:

$$J = \sum\nolimits_{i=1}^{n-5} f_i^2(x). \tag{23}$$

TABLE 1. The steps of our proposed Method.

The Proposed Method

Input: ${}^{W}\boldsymbol{P}_i=\left(X_i,Y_i,Z_i\right)^T,\left(i=1,...,n\right)$ the world co-ordinate system;
${}^{C}\boldsymbol{P}_i=\left(x_i,y_i,0\right)^T,\left(i=1,...,n\right)$ the camera co-ordinate system
Output: $\boldsymbol{R}$, the rotation matrix; $\boldsymbol{t}$, the translation matrix

1: Choose five points from n key points
2: Initialize a 40×1 matrix to store the coefficients of $J1$
3: **for** i = {1, 2, …, n-5} **do**
4: Compute the coefficients of $f_i(x)\cdot f_i^{'}(x)$;
$J1=J1+f_i(x)\cdot f_i^{'}(x)$;
5: **end for**
6: Construct the coefficients matrix and solve it by saving the roots as alternative $\cos\theta$. Record the number of roots with the variable j;
7: Initialize $E_0=100$
8: **for** i = {1, …, j} **do**
9: Formulate $J2$ using the method which was used to formulate $J1$ in step 4, where the $f_i(x)$ in $J2$ is formed using (13);
Compute alternative $\cos\beta$ using the method mentioned in step 6, and record the number of roots with the variable m;
for $k=\{1,...,m\}$ **do**
Compute $\cos\alpha$ using (12);
Compute f and $\boldsymbol{t}$ using (20) and (21);
Compute $E=\sum_{j=1}^{n-1}\left(\boldsymbol{n}_j\boldsymbol{R}\overrightarrow{{}^{W}P_j\,{}^{W}P_{j+1}}\Big/\left\|\overrightarrow{{}^{W}P_j\,{}^{W}P_{j+1}}\right\|\right)\Big/(n-1)$,
where $\boldsymbol{n}_j=\left(x_j,y_j,f\right)\times\left(x_{j+1},y_{j+1},f\right)$;
if $E<E_0$ **do**
$E_0=E$;
if $E_0<1e-3\,\|\,(i==j\,\&\&k==m)$ **do**
Output $\boldsymbol{R}$ and $\boldsymbol{t}$;
break;
end if
end if
end for
10: **end for**

We obtain the minima of J by computing the roots of $J'=0$. Then, J' can be written as:

$$J'=\sum_{i=0}^{39}a_ix^i=0. \tag{24}$$

We solve (24) by the eigenvalue method [10]. First, we construct a matrix $\boldsymbol{A}$ with the coefficients of (24):

$$\boldsymbol{A}=\begin{bmatrix}-\frac{a_{38}}{a_{39}} & -\frac{a_{37}}{a_{39}} & \cdots & -\frac{a_1}{a_{39}} & -\frac{a_0}{a_{39}}\\ 1 & 0 & \cdots & 0 & 0\\ 0 & 1 & \cdots & 0 & 0\\ \cdots & \cdots & \ddots & \cdots & \cdots\\ 0 & 0 & \cdots & 1 & 0\end{bmatrix}. \tag{25}$$

According to the eigenvalue method, the roots of $|\lambda\boldsymbol{I}-\boldsymbol{A}|=0$ are also the roots of (24), where λ are the eigenvalues of $\boldsymbol{A}$. It is well known that $|\lambda\boldsymbol{I}-\boldsymbol{A}|=0$ can be easily solved by the matrix decomposition.

In this section, the PnP was further transformed into a sparse matrix decomposition.

The proposed method is summarized in Table 1.

TABLE 2. Information about the datasets.

Dataset No.	Image size(pixels)	Number of points	Noise	Focal length (pixels)
A		6:40	5	1200
B	640×480	20	1:15	1000
C		24	5	600:2850

III. EXPERIMENTS RESULTS

A. DATASETS

To assess the properties of our method, several experiments were carried out on six datasets. For each case with the same parameters, 500 synthetic data were provided. The image size of these data was 640 pixels ×480 pixels and the normalized depth of the 3D points was randomly distributed in the range [4,8]. The details of used datasets are show as follows:

1) Dataset A: The 3D points are randomly generated and then projected with focal length of 1200 pixels. The image noise level is $\sigma=5$. The number of points ranges from 6 to 40.
2) Dataset B: Varying image noise is considered in the 20 points cases with fixed focal length of $f=1000$ pixels. The noise level increases from 1 to 15.
3) Dataset C: The performances are compared in changing focal length values from 600 to 2850. In addition, the number of points is set to 24 points and the image noise level is $\sigma=5$.

The ground truth pose, corresponding to the estimates $\boldsymbol{R}$ and $\boldsymbol{t}$, are denoted as $\boldsymbol{R}_{true}$ and $\boldsymbol{t}_{true}$. As in [11] and [16], we used the angular distance to measure the rotation error, which can be expressed concisely as $E_{\boldsymbol{R}}=\max\left\{\text{acos}\left(\boldsymbol{R}_{true}(:,i),\boldsymbol{R}(:,i)\right)\times180/\pi\right\}(i=1,2,3)$, where acos $(\boldsymbol{R}_{true}(:,i),\boldsymbol{R}(:,i))$ refers to the angle between the ith column of $\boldsymbol{R}_{true}$ and $\boldsymbol{R}$. The translation error was defined as $E_{\boldsymbol{t}}=\|\boldsymbol{t}_{true}-\boldsymbol{t}\|/\|\boldsymbol{t}_{true}\|$. A result was determined to be true when $E_{\boldsymbol{R}}<5^\circ$ and $E_{\boldsymbol{t}}<5\%$.

B. RESULTS AND PERFORMANCE EVALUATION

We assessed the performance of our proposed method and compared it against Adrian's method, UPnP_GN (UPnP with Gaussian-Newton) [8], and DLT [24]. Our proposed method, DLT and UPnP_GN were implemented in Matlab, although the UPnP_GN method used compiled C functions. The code for UPnP_GN was taken from Opencv3.4.1, which is available at https://github.com/opencv/opencv/ archive/3.4.1. zip.

1) ACCURACY

In this section, the results of EPnP[25] are included, which were obtained under known focal length, as a reference baseline. Obviously, EPnP performed better than the uncalibrated methods.

The first column in Figure 4 shows the performance of our method, compared to UPnP_GN, DLT and EPnP, with a fixed focal length $f=1200$ pixels. With increasing size of the point corresponding set, both the median rotation error and translation error of all methods showed a downward trend. The accuracy of our proposed method became closer to that of UPnP_GN as the number of points increased. For the cases

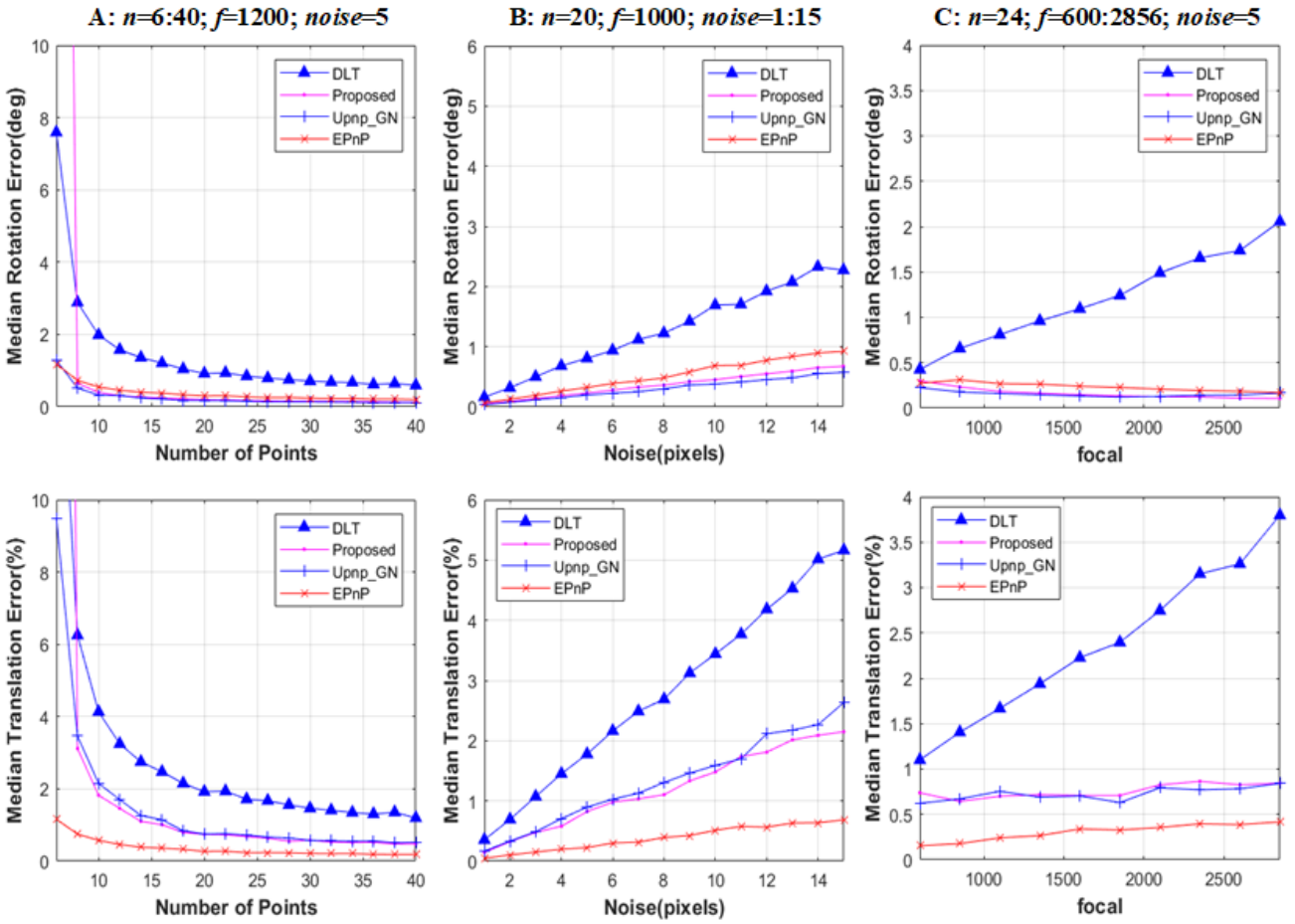


FIGURE 4. Experiments results on three different datasets. First column: median rotation error and median translation error for increasing number of 2D-3D correspondences. Second column: median rotation error and median translation error for increasing noise. Third column: median rotation error and median translation error for changing focal length. Every tick in the drawing indicates the statistical results of 500 experiments.

where $n > 8$, the accuracy of our method was roughly the same as that of UPnP_GN, while the accuracy of DLT was always inferior to UPnP_GN and our proposed method.

The second column in Figure 4 shows the robustness of three methods against noise at $n = 20$ and $f = 1000$. With increasing noise, the median rotation error and translation error gradually became larger. The accuracy of our proposed method was almost consistent with that of UPnP_GN. Affected by its linear characteristics, DLT performed weakly against noise.

The third column in Figure 4 illustrates the performances of the three approaches under changing focal length. Both the median rotation and translation error of DLT became larger and larger with increasing focal length. Our proposed method performed well, as did UPnP_GN, in terms of accuracy when the focal length was set between 600 to 2850.

Concisely, our proposed method was comparable to UPnP_GN, in terms of numerical accuracy, and better than DLT. This is because DLT defines the elements in the rotation matrix as nine variables, ignoring the constraints between variables. In contrast, our approach decomposes the rotation matrix into three rotation angles around the co-ordinate axes, as well as using the orthogonality of the rotation matrix to constrain the results.

2) STABILITY

The median errors are not enough to fully demonstrate the performances of these methods. Therefore, we further evaluated the stability of these methods by using the correct rate. The correct rate refers to the proportion of $E_R < 5°$ and $E_t < 5\%$ among 500 pose estimates.

Figures 5. (a), (b), and (c) show the stability of our proposed approach under Datasets A, B, and C, respectively. As shown in Figure 5. (a), the correct rates of all methods became higher and higher with an increasing number of points. UPnP_GN performed better than ours only when the number of correspondences was set to 6. For the cases where $n > 12$, the correct rate of our method and DLT gradually approached 100%, while the correct rate of UPnP_GN remained stable at 80%. Figure 5. (b) illustrates that the correct rates changed with growing noise. The correct rates of our method and DLT were stable, at roughly 100 percent, when $\sigma < 6$, where that of DLT was a little higher than ours; although, as shown in Figure 4, our proposed approach was more precise than DLT. When $\sigma > 7$, our proposed method performed better than both DLT and UPnP_GN. Figure 5. (c) displays the performances of three methods under different focal lengths which ranged from 600 pixels to 2850 pixels. In the tested focal length range, the correct

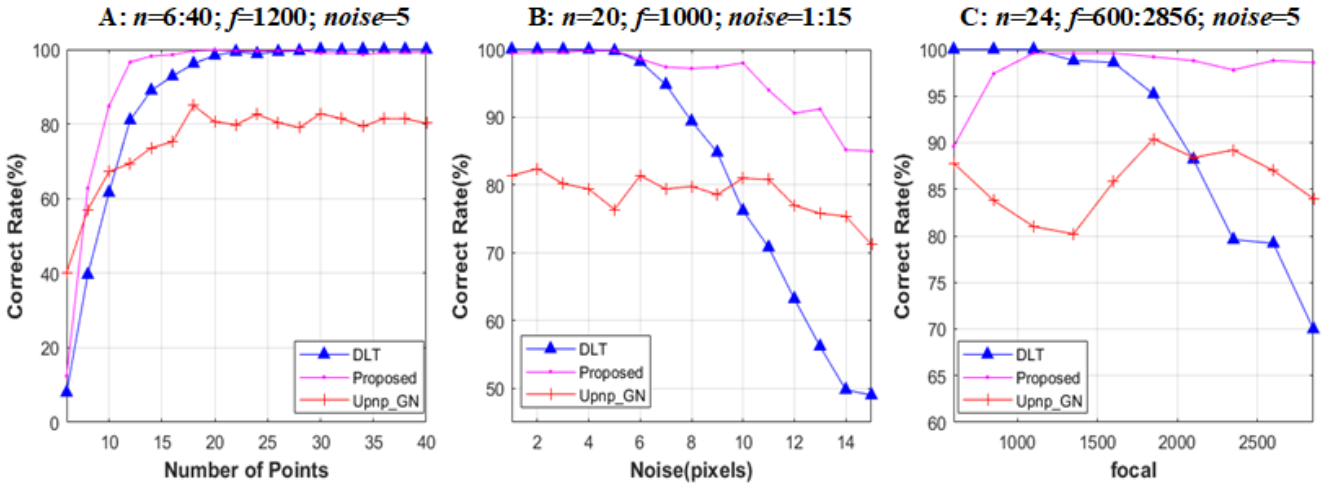


FIGURE 5. (a), (b) and (c) depict the correct rate on datasets A, B and C respectively. Every tick in the drawing indicates the ratio of $E_R < 5°$ and $E_t < 5\%$ among 500 experiments.

rate of our proposed method was always above 90 percent, better than that of UPnP_GN. For the cases with $f > 1000$, the proposed method performed better than both DLT and UPnP_GN.

Briefly, our proposed method is more stable than UPnP when $n \geq 8$. This is because our method formulated the solution of rotation matrix into a polynomial system without the unknown variable f by projecting the 3D points vertically onto the imaging surface and estimating the rotation matrix first. The strategy of first estimating the rotation matrix and then estimating the focal length and translation vector reduces the influence of unknown focal length on the estimation, as well as obtaining a more stable performance, (as illustrated in Figure 5). Conversely, UPnP_GN puts all unknowns together and establishes the same equation system as in EPnP, but contains the unknown focal length, which can cause the underdetermination of the equation system in some cases.

3) RUNNING SPEED

Figure 6 illustrates the running time of our method, DLT, and UPnP_GN for an increasing number of point pairs. The noise and focal length were set as $\sigma = 5$ and $f = 800$.

The DLT solver consumed less time than ours; however, as illustrated in Figure (4), DLT performed poorly in terms of accuracy.

As compared to UPnP_GN, our approach was faster when the number of correspondences was less than roughly 90. This is because our proposed method formulates the PnP problem as a polynomial system, which can be easily solved by decomposing a sparse matrix with 40 elements. This led to an increase in computational efficiency, while UPnP_GN uses exhaustive linearization techniques to solve the problem.

However, our proposed method happened to be slower with a large number of input points. This phenomenon was caused by increasing coefficient computation with the increase of amount of input points, which means that our method is more suitable when considering smaller point sets.

The above experiments and comparisons demonstrate that our proposed method achieves comparable numerical

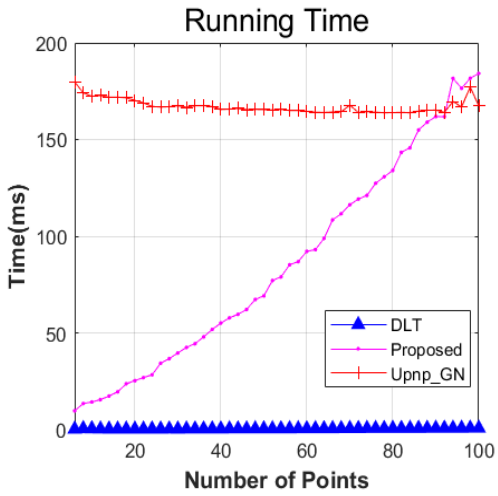


FIGURE 6. Running time with increasing number of points at fixed values of $\sigma = 5$, $f = 800$.

accuracy and higher stability at a lower computational cost when the number of correspondences located between 8 and 40. Particularly, the correct rate of our proposed method can reach more than 95 percent, while the correct rate of UPnP_GN finally increases to roughly 83 percent. In addition, our proposed method is more accurate than DLT while slower than it.

IV. REAL DATA

The proposed approach was evaluated on the image shown in Figure 7. (a), which was taken with a FUJIFILM FinePix HS35EXR digital camera. The ground truth position and rotation matrix were provided by using EPnP_GN[5], and twenty 2D–3D correspondences were manually selected.

We input twenty correspondences into our proposed method, and DLT in order to obtain the estimations. We projected 3D points onto the image plane using the estimates, and the reprojections are shown in Figure 7. (b). Finally, we computed the rotation errors and translation errors, and the results are listed in Table 3. The results prove the effectiveness of our method.

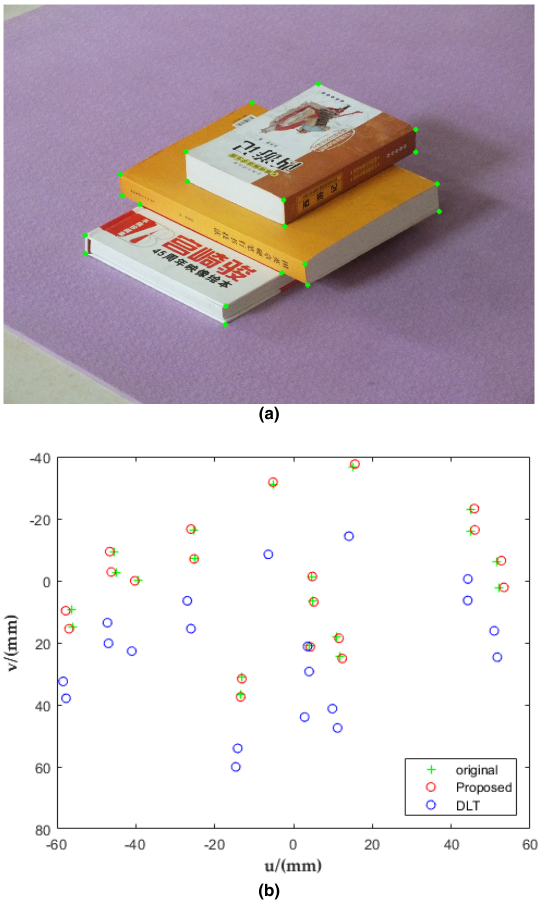


FIGURE 7. (a) The real image. (b) the reprojection by estimated R and t provided by our proposed method and DLT.

TABLE 3. Rotation error and translation error.

Method	E_R (*deg*)	E_t (%)
Our Proposed	0.1512	0.1400
DLT	1.2998	2.2383

V. CONCLUSION

In comparison to previous studies, our proposed method exhibits a principal advantage. We found that our method has sufficient effectiveness to reliably provide pose estimations for the uncalibrated cameras.

We improved the computation speed by formulating the PnP problem into a polynomial system which could easily be solved. Our proposed method obtained more stable estimation by vertically projecting the 3D points onto the image surface, thus reducing the interference arising from the unknown focal length. Further research on the relationship between accuracy and computation speed is necessary for the application of pose estimation methods for uncalibrated cameras.

REFERENCES

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.

[2] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003, doi: 10.1109/tpami.2003.1217599.

[3] T. Ke and S. I. Roumeliotis, "An efficient algebraic solution to the Perspective-Three-Point problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4618–4626.

[4] A. Masselli and A. Zell, "A new geometric approach for faster solving the Perspective-Three-Point problem," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Stockholm, Sweden, Aug. 2014, pp. 2119–2124.

[5] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009, doi: 10.1007/s11263-008-0152-6.

[6] M. Bujnak, Z. Kukelova, and T. Pajdla, "A general solution to the P4P problem for camera with unknown focal length," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AA, USA, Jun. 2008, pp. 1–8.

[7] C. Wu, "P3.5P: Pose estimation with unknown focal length," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2440–2448.

[8] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer, "Exhaustive linearization for robust camera pose and focal length estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2387–2400, Oct. 2013, doi: 10.1109/TPAMI.2013.36.

[9] M. Bujnak, Z. Kukelova, and T. Pajdla, "New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Queenstown, New Zealand, 2010, pp. 11–24.

[10] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, "Eigensystem-s," in *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007, pp. 583–589.

[11] C. Xu, L. Zhang, L. Cheng, and R. Koch, "Pose estimation from line correspondences: A complete analysis and a series of solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1209–1222, Jun. 2017, doi: 10.1109/TPAMI.2016.2582162.

[12] L. Zhou, J. Ye, and M. Kaess, "A stable algebraic camera pose estimation for minimal configurations of 2D/3D point and line correspondences," in *Proc. ACCV*, Perth, WA, Australia, May 2019, pp. 273–288.

[13] Y. Kuang and K. Astrom, "Pose estimation with unknown focal length using points, directions and lines," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 529–539.

[14] L. Li, W. Li, and H. Xu, "Camera pose estimation with uncertain reference point based on orthogonal iterative," *Laser Optoelectronics Prog.*, vol. 56, no. 1, pp. 251–259, Jan. 2019, doi: 10.3788/LOP56.011503.

[15] A. Kudriashov, T. Buratowski, and M. Giergiel, "Hybrid AMCL-EKF filt-ering for SLAM-based pose estimation in rough Terrain," in *Mechanism and Machine Science*. Cham, Switzerland: Springer, 2019, pp. 2819–2828.

[16] L. Ferraz, X. Binefa, and F. Moreno-Noguer, "Very fast solution to the PNP problem with algebraic outlier rejection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 501–508.

[17] G. Bourmaud and R. Megret, "Robust large scale monocular visual SLAM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1638–1647.

[18] Y. Wang and Y.-F. Xu, "Unsupervised learning of accurate camera pose and depth from video sequences with Kalman filter," *IEEE Access*, vol. 7, pp. 32796–32804, Mar. 2019, doi: 10.1109/access.2019.2903871.

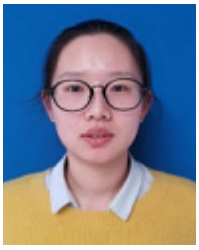
[19] J. Cui, C. Min, X. Bai, and J. Cui, "An improved pose estimation method based on projection vector with noise error uncertainty," *IEEE Photon. J.*, vol. 11, no. 2, Apr. 2019, Art. no. 6801016, doi: 10.1109/JPHOT.2019.2901811.

[20] P. Chen, "An iterative camera pose estimation algorithm based on EPnP," in *Proc. Chin. Intell. Automat. Conf.*, Tianjin, China, 2017, pp. 415–422.

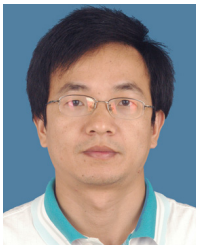
[21] P. Wang, C. Gao, W. Guo, M. Li, F. Zha, and Y. Cong, "Robust and fast mapping based on direct methods," in *Proc. 2nd Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Hefei Tai'an, China, Aug. 2017, pp. 592–595.

[22] S. Li, C. Xu, and M. Xie, "A robust O(n) solution to the Perspective-n-Point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1444–1450, Jul. 2012, doi: 10.1109/TPAMI.2012.41.

[23] J. K. Lee and K. J. Yoon, "Joint estimation of camera orientation and vanishing points from an image sequence in a non-manhattan World," *Int. J. Comput. Vis.*, vol. 127., no. 1, pp. 1426–1442, Jul. 2019, doi: 10.1007/s11263-019-01196-y.
[24] R. Hartley and A. Zisserman, "Multiple view geometry," in *Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[25] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative O(n) solution to the PnP problem," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.



BINGYI ZHOU received the B.E. degree in electronic information engineering from the Henan University of Technology, Henan, China, in 2016. She is currently pursuing the master's degree with the Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology. Her current research interests include communication signal processing and camera pose estimation.



ZIQIANG CHEN was born in Hunan, China, in 1973. He received the B.S. degree from Xidian University, in 1994, the M.S. degree from the Guilin University of Electronic Technology, in 2003, and the Ph.D. degree from Xidian University, in 2013. He is currently an Assistant Professor with the Guilin University of Electronic Technology. His main research interest includes video signal processing.

QINGHUA LIU was born in Sichuan, China, in 1974. She received the B.Sc. degree from Sichuan Normal University, in 1995, the M.Sc. degree from the Guilin University of Electronic Technology, in 2001, and the Ph.D. degree from Xidian University, in 2014. She is currently a Professor with the Guilin University of Electronic Technology. Her main research interest includes radar signal processing.