Pattern Recognition Letters 108 (2018) 31–37

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A simple, robust and fast method for the perspective-n-point Problem

Ping Wang[a], Guili Xu[a,*], Yuehua Cheng[b], Qida Yu[a]

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

A R T I C L E I N F O

Article history:
Received 18 March 2017
Available online 5 March 2018

MSC:
41A05
41A10
65D05
65D17

Keywords:
Perspective-n-point problem (PnP)
Absolute position and orientation
Camera pose estimation
Vision-based navigation
Computer vision

A B S T R A C T

In this work, we present a simple, robust and fast method to the perspective-n-point (PnP) problem for determining the position and orientation of a calibrated camera from known reference points. Our method transfers the pose estimation problem into an optimal problem, and only needs to solve a seventh-order and a fourth-order univariate polynomial, respectively, which makes the processes more easily understood and significantly improves the performance. Additionally, the number of solutions of the proposed method is substantially smaller than existing methods. Experiment results show that the proposed method can stably handle all 3D point configurations, including the ordinary 3D case, the quasi-singular case, and the planar case, and it offers accuracy comparable or better than that of the state-of-art methods, but at much lower computational cost.

1. Introduction

Determining the position and orientation of a calibrated camera from n 3D points and their 2D projections, which is also known as the perspective-n-point (PnP) problem [5], has numerous applications in computer vision and robotics. Examples include robot localization [6,8], augmented reality [17], structure-from-motion (SfM) [4], spacecraft pose estimation during descent and landing [18,23]. Considering the importance of the PnP problem, a large amount of research effort has been devoted to solving this problem over the past few decades. The existing PnP methods can be classified as iterative and non-iterative methods.

Classical iterative methods formulate the PnP problem into a non-linear least-squares problem [15], and then solve it using iterative optimization methods, i. e., Gauss–Newton and Levenberg–Marquardt [9] method. However, iterative methods are sensitive to the initialization, and are easily trapped into a local minimum, which will leads to poor accuracy, especially when no redundant points ($n \leq 6$) are available.

For non-iterative methods, the traditional methods apply linear operations to obtain solutions, i. e., the DLT [1] and HOMO method [16]. Non-iterative methods have an advantage of less computing costs, but are sensitive to noise. Quan and Lan [21] and Ansar and Daniilidis [2] presented two linear solutions for the PnP problem, with respective computational complexity $O(n^5)$ and $O(n^8)$. However, they are still inaccurate when n is small. On the contrary, they are very time-consuming when n is large. To overcome these problems, Lepetit et al. [12] introduced four virtual control points to represent the 3D reference points, and proposed the first linear complexity method, named EPnP, with respect to the number of the points. EPnP is computationally efficient, but is inaccurate for $n = 4$ or 5, due to its underlying linearization scheme. To improve accuracy, Li et al. [14] proposed another non-iterative $O(n)$ solution, named RPnP, which transfers the PnP problem into a suboptimal problem by solving a seventh-order polynomial. RPnP is very efficient and works well for both non-redundant ($n \leq 6$) and redundant points cases. Hesch and Roumeliotis [7] developed the first globally optimal method (called DLS) with complexity $O(n)$, which formulates the PnP problem into a multivariate polynomial system using the camera measurement equations, and employs the multiplication matrix to determine all roots of the system. Unfortunately, the accuracy of DLS is unstable because of the Cayley parameterization, which has a singularity for any 180-degree rotations. To resolve these drawbacks, Zheng et al. [26] proposed the OPnP method, which adopts the non-unit quaternion parameterization to replace the Cayley parameterization, and uses the Gröbner basis [11] to solve the PnP problem. To our knowledge, OPnP is one of the most accurate non-iterative methods until now. To ex-

* Corresponding author.
E-mail addresses: pingwangsky@gmail.com (P. Wang), guilixu2002@163.com (G. Xu).

https://doi.org/10.1016/j.patrec.2018.02.028
0167-8655/

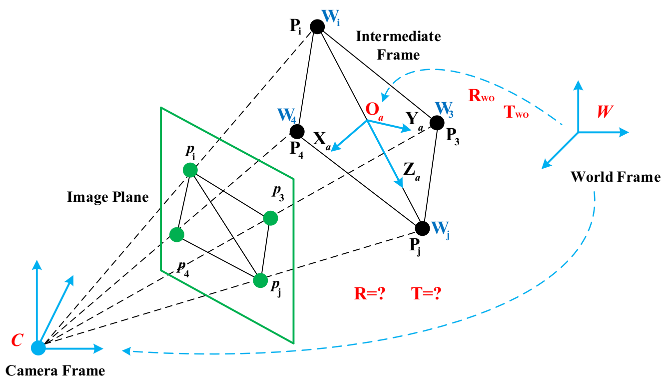

Fig. 1. Illustration of the P*n*P problem.

tend the scope of application, Kneip et al. [10] presented the UP*n*P method, which is applicable to both central and non-central camera systems [3]. However, its accuracy is worse than OP*n*P in some configurations. Recently, the P*n*P*f* [20,25] and P*n*P*fr* [19] methods were proposed to solve the pose estimation problem in the case of an uncalibrated camera. However, the accuracy of these methods is usually lower than the classical P*n*P method.

The most recent works, i. e., DLS, OP*n*P, and UP*n*P method, formulate the P*n*P problem as a minimization problem, and then solve it using the Gröbner basis technique. However, the Gröbner basis technique needs to construct a large elimination template, because of the large number of unknowns and the high maximum degree of the P*n*P problem. This process takes significant time and is difficult to assure reliability. All of these disadvantages will reduce overall performance and limit general understanding.

In contrast to previous methods, in this paper we propose a simple, robust, and fast solution to the classical P*n*P problem. Our method transfers the P*n*P problem into an optimal one, that only needs to solve a seventh-order and fourth-order univariate polynomial without using the Gröbner basis technique. The number of the solutions for our method is substantially smaller than existing globally optimal methods, i. e., the DLS and OP*n*P method. All of these make the processes more easily applicable and significantly improve the performance. The experiment results show that our method can stably address all 3D point configurations, including the ordinary 3D case, quasi-singular case, and planar case. It also offers accuracy comparable to the leading methods, but at much lower computational cost.

The rest of the paper is organized as follows. Section 2 presents the derivations of our method. Section 3 provides a thorough analysis of the proposed method by simulated experiments. Section 4 shows the real tests. Section 5, finally, concludes the work.

2. Proposed method

As shown in Fig. 1, suppose a reference point W_i whose coordinates in the world frame and the normalized image plane are $W_i = [X_i^w, Y_i^w, Z_i^w]^T$ and $f_i = [u_i, v_i, 1]^T$, respectively. Note that the superscript indicates the different coordinate frame, i. e., w indicates the world frame. Our goal is to retrieve the rotation matrix R and the translation vector t between the world frame and the camera frame using n $(n \geq 4)$ reference points when the camera is calibrated.

2.1. Building an object frame

The first step involves the definition of a new, intermediate object frame from the 3D reference points. As shown in Fig. 1, we choose the center of $\overrightarrow{W_iW_j}$ as the origint O_a, and create an intermediate frame $[O_a - \vec{X}_a, \vec{Y}_a, \vec{Z}_a]$, where

$$\vec{X}_a = \frac{W_j - O_a}{\|W_j - O_a\|}$$
$$\vec{Z}_a = \frac{\vec{X}_a \times [0, 1, 0]^T}{\|\vec{X}_a \times [0, 1, 0]^T\|}$$
$$\vec{Y}_a = \frac{\vec{Z}_a \times \vec{X}_a}{\|\vec{Z}_a \times \vec{X}_a\|}, \tag{1}$$

if $|[0, 1, 0]^T\vec{X}_a| \leq |[0, 0, 1]^T\vec{X}_a|$, and

$$\vec{X}_a = \frac{W_j - O_a}{\|W_j - O_a\|}$$
$$\vec{Y}_a = \frac{[0, 0, 1]^T \times \vec{X}_a}{\|[0, 0, 1]^T \times \vec{X}_a\|}$$
$$\vec{Z}_a = \frac{\vec{X}_a \times \vec{Y}_a}{\|\vec{X}_a \times \vec{Y}_a\|}, \tag{2}$$

if $|[0, 1, 0]^T\vec{X}_a| > |[0, 0, 1]^T\vec{X}_a|$.

Via the transformation matrix $T_{wo} = [\vec{X}_a, \vec{Y}_a, \vec{Z}_a]^T$, the reference point $W_i = [X_i^w, Y_i^w, Z_i^w]^T$ can be easily transformed into the intermediate frame using

$$P_i = T_{wo}(W_i - O_a) \qquad i = 1, 2, \ldots, n, \tag{3}$$

where $P_i = [X_i^p, Y_i^p, Z_i^p]^T$, and the superscript p indicates the intermediate object frame.

2.2. Determining a rotation axis using least-square residual

Every remaining point together with the P_i and P_j forms a 3-point subsets. By using the P3P (perspective-three-point) constraint [13], each subset can build a fourth-order polynomial as follows:

$$\begin{cases} f_1(x) = a_1x^4 + b_1x^3 + c_1x^2 + d_1x + e_1 = 0 \\ f_2(x) = a_2x^4 + b_2x^3 + c_2x^2 + d_2x + e_2 = 0 \\ \cdots \\ f_{n-2}(x) = a_{n-2}x^4 + b_{n-2}x^3 + c_{n-2}x^2 + \\ \qquad d_{n-2}x + e_{n-2} = 0 \end{cases}. \tag{4}$$

Instead of directly solving a series of fourth-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 seventh-order polynomial, which has at most 4 minima, and can be easily solved by the eigenvalue method [22]. Once the minimal of F is determined, the depths of P_i and P_j can be calculated according to the P3P constraint [13], and then the rotation axis Z_a can be calculated as $Z_a = \overrightarrow{P_iP_j}/\|P_iP_j\|$.

2.3. Retrieving the pose by solving an optimal problem

When the Z_a-axis of $[O_a - \vec{X}_a, \vec{Y}_a, \vec{Z}_a]$ is determined, the transformation from the intermediate object frame $[O_a - \vec{X}_a, \vec{Y}_a, \vec{Z}_a]$ to the camera frame $[O_c - \vec{X}_c, \vec{Y}_c, \vec{Z}_c]$ can be expressed as

$$\lambda_i f_i = R_c P_i + t_c \qquad i = 1, 2, \ldots, n, \tag{5}$$

where

$$R_c = R_1 R_2 = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & x & -y \\ 0 & y & x \end{bmatrix}$$

and $t_c = [t_1, t_2, t_3]^T$. R_1 is an arbitrary rotation matrix whose third column $[r_3, r_6, r_9]^T$ equals the rotation axis Z_a, and R_1 should meet the orthogonal constraint of the rotation matrix. R_2 denotes a rotation of α degrees around the $Z - axis$, with $x = cos\alpha$ and $y = sin\alpha$.