A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation

Xiao Xin LU
Department of Automation, Tsinghua University, Beijing, China
Email: xxluna2014@163.com

Abstract: As there is a rapid development of robotics in the field of automation engineering, ego-motion estimation has become a most challenging task. In this review, we presented a model to help describe the PnP problems, and introduced two most common solutions. The P3P solution is the smallest subset of control points that yields a finite number of solutions. The EPnP solution is to reduce the complexity by expressing the n 3D points as a weighted sum of four virtual control points. The former solution is widely applied while the latter is more used.

1. INTRODUCTION

Ego-motion estimation is a fundamental challenge in the field of automation engineering. For any robot or mobile device, avoiding dangerous situations such as collisions and unsafe conditions comes first. What’s more, if the robot has a purpose that relates to specific places in the environment, it must find those places. As a result, the technology of mobile robot navigation has become a central topic, which is also known as ego-motion estimation.

Various types of sensors are invented for the purpose of ego-motion examination. In the early 1970s, the U.S. Department of Defense (DoD) developed Global Position System (GPS) for timing and space-based navigation [1], but GPS can only be used in outdoor circumstances and provides positions with error. Some researchers use inertial navigation systems (INS) on mobile devices to achieve ego-motion estimation [2,3], but INSs suffer from drift accumulation as they calculate the position by performing mathematical integration with respect to time. The laser sensor, as an option for distance estimation method, shows a better accuracy than GPS and INS [4], but it is expensive and has certain limitations in applications. An ultrasonic method is available [5] but its efficiency highly depends on the material and surface. Among all the sensors developed for estimating robotic ego-motion, a vision-based method is a more effective solution for being more accurate, inexpensive and versatile.

By reconstructing the position of a camera tightly attached to a mobile device, the vision-based method achieves device’s ego-motion estimation [6]. In the field of computer vision, such problems have been studied for more than a century. In the early 1980s, Fischler proposed the Perspective-n-Point (PnP) problem [7], whose goal is to estimate the position and orientation of a calibrated camera from known 3D-to-2D point correspondences between a 3D model and their image projections. The PnP is a fundamental problem of many applications in the field of computer vision, among which robotic ego-motion estimating is a cause for concern.
The rest of this paper is organized as follows: Section 2 is divided into two parts, in which we introduce pinhole camera model and principles of two popular PnP solutions. In Section 3, we discuss the suitable application scenarios for the PnP problem solutions. Finally, conclusions are drawn in Section 4.

2. METHODS

2.1 Pinhole Camera Model

Among all the camera models proposed by researchers to describe the projection of a point from the 3-D world coordinate system to the 2-D image plane, pinhole model is the most popular.

Consider a point \( p \) in the world coordinate system \( \mathbb{W}: w_0, w_x, w_y, w_z \) and the homogeneous coordinate of \( p \) in \( \mathbb{W} \) is \( X_w = [x_w, y_w, z_w, 1]^T \). We can present the pinhole camera model as (1).

\[
m' = DK_0 M X_w
\]

where

\[
K_0 = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad D = \begin{pmatrix} \frac{1}{dx} & -\cot \theta \frac{1}{dy} & 0 \\ 0 & \frac{1}{\sin \theta dy} & 0 \\ 0 & 0 & 1 \end{pmatrix}
\]

and \( M = \begin{pmatrix} R & T \end{pmatrix} \).

In Fig. 1, we present a general procedure of the pinhole model. Initially, the matrix \( M \) transforms the coordinates of \( p \) in \( \mathbb{W} \) into the camera coordinate system. \( M \), \( R \) is a \( 3 \times 3 \) matrix that represents rotation of the camera and \( T \) is a \( 3 \times 1 \) vector that represents translation. Then \( K_0 \) projects point \( p \) to the image plane, where \( f \) is the camera focal length and \( (x_0, y_0) \) is the optic center of the plane. Lastly, \( D \) discretizes the 2-D coordinates in \( \mathbb{I} \), where \( (dx, dy) \) represents the size of each pixel in a CCD/CMOS image sensor with a intersection angle \( \theta \).

The matrix \( K = DK_0 \) contains all the intrinsic parameters of the camera, while the matrix \( M \) contains the camera ego-motions, which are also the extrinsic parameters. For a calibrated camera, we can estimate its extrinsic parameters using a set of 3-D points in \( \mathbb{W} \) and their corresponding 2-D projections in the image. We call it a PnP problem.

Fig. 1. The general steps of the pinhole model. This figure is modified from (Armangu, 2003).
2.2 Solutions for PnP Problems

The camera pose is made up of 6 degrees-of-freedom (DOF) which consists 3D rotation (roll, pitch, and yaw) and 3D translation of the camera concerning the world. Therefore, it is necessary to get information of at least three pairs of corresponding points to solve a PnP problem. Most of the available solutions are applicable for the common case in which \( n > 3 \), while solutions particularly applicable for \( n = 3 \) cases also exist.

Before introducing the solutions, it is necessary to discuss some common preliminary aspects of all solutions to the PnP problem. It is a general assumption that the camera is already calibrated in most solutions. Therefore, we regard its intrinsic properties already known, such as the focal length, principal image point, skew parameter, and other parameters mentioned above. However, the assumption does not adapt to some methods like UPnP and the Direct Linear Transform (DLT) applied to the projection model. It is another important premise for PnP solutions that we cannot choose coplanar point correspondences. Furthermore, a PnP problem can have a set of solutions, and choosing one solution in them requires post-processing of the solution set. In addition, the impact of noisy data can be reduced by using more point correspondences to solve PnP. It is also common to select in the set of point correspondences in a way of Random sample consensus (RANSAC) to make the solution robust to outliers [8]. However, in most methods we assume that the data is noise-free. In the following part of this section, we will introduce two of the most common methods for solving the PnP problem. One is called P3P and another EPnP.

1) P3P Solution

P3P problem, the smallest subset of control points, yields a finite number of solutions. Grunert [9] first investigated the problem in 1841. Then in 1903, Finsterwalder [10] noticed that there can be no more than four solutions for a calibrated camera, and then with a fourth point it can be disambiguated. In the previous studies, researchers have already come up with many solutions to this problem. The solutions can be classified into iterative, non-iterative, linear, and non-linear ones. In 1991, Haralick et al. [11] reviewed six of the most common direct solutions to the problem up to 1991 and as a conclusion gave the analytical solution with resultant computation. Later, Quan and Lan proposed different solution to the P3P problem in 1991[12], and Gao et al. proposed another in 2003 [13]. We now introduce a common P3P solution based on these studies.

Gao et al in [13] considered the problem illustrated in Fig. 2. Let \( P \) be the Center of Perspective, and \( A, B, C \) the control points. Let \( |PA| = X, |PB| = Y, |PC| = Z, \alpha = \angle BPC, \beta = \angle APC, \gamma = \angle APB, p = 2 \cos \alpha, q = 2 \cos \beta, r = 2 \cos \gamma, |AB| = c', |BC| = a', |AC| = b'. \) Firstly we can get the P3P equation system (3) from the triangles \( PBC, PAC, \) and \( PAB \):

\[
\begin{align*}
Y^2 + Z^2 - YZp - a'^2 &= 0 \\
Z^2 + X^2 - XZq - b'^2 &= 0 \\
X^2 + Y^2 - XYr - c'^2 &= 0
\end{align*}
\] (3)

If the following conditions (4), also called “reality conditions” are satisfied, a set of solutions for \( A, B, C \) is physical. These conditions are basic assumption of this paper.

\[
\begin{align*}
Y^2 + Z^2 - YZp - a'^2 &= 0 \\
Z^2 + X^2 - XZq - b'^2 &= 0 \\
X^2 + Y^2 - XYr - c'^2 &= 0
\end{align*}
\]
\[
\left\{ \begin{array}{l}
X > 0, Y > 0, Z > 0, a' > 0, b' > 0, c' > 0 \\
a' + b' > c', a' + b' > c', a' + b' > c'
\end{array} \right. \\
0 < \alpha, \beta, \gamma < \pi, 0 < \alpha + \beta + \gamma < 2\pi \\
\alpha + \beta > \gamma, \alpha + \gamma > \beta, \gamma + \beta > \alpha \\
l_0 = p^2 + q^2 + r^2 - pq - r - 1 \neq 0
\] (4)

To simplify the equation system, let \( X = xZ, Y = yZ, |AB| = \sqrt{v}Z, |BC| = \sqrt{w}Z, |AC| = \sqrt{vZ} \).
As \( Z = |PC| \neq 0 \), we get the following equivalent equation system:
\[
\left\{ \begin{array}{l}
y^2 + 1 - yp - av = 0 \\
x^2 + 1 - qx - bv = 0 \\
x^2 + y^2 - xyr - v = 0
\end{array} \right. \\
(5)
\]

During camera calibration, we can make use of many symmetrical objects in practical cases and reduce the complexity of the P3P problem this way. Suppose that we calibrate the camera using a regular triangle, which means \( \Delta ABC \) is a regular triangle. Thus the equation system in (3) can be simplified as
\[
\left\{ \begin{array}{l}
Y^2 + Z^2 - YZp - a'^2 = 0 \\
Z^2 + X^2 - XZq - a'^2 = 0 \\
X^2 + Y^2 - XYr - a'^2 = 0
\end{array} \right. \\
(6)
\]
Eliminating \( a' \) from (6) and combining with (5), we get
\[
\left\{ \begin{array}{l}
y^2 - rxy - 1 + qx = 0 \\
y^2 - py - x^2 + qx = 0
\end{array} \right. \\
(7)
\]

which has the same number of physical solutions with (3). We now simplify the P3P problem aimed at finding the positive solutions of two quadratic equations. Therefore, we get the following result: The P3P problem has either an infinite number of solutions or at most four physical solutions. This result was known before only for the “main part” of the P3P problem.

Next we use Wu-Ritt’s zero decomposition method [14] which is considered a general method to solve systems of algebraic equations to solve the P3P equation system (7). The central idea of this algorithm is using the union of zero sets of equations in triangular form to represent the zero set of a polynomial equation system, which means \( \Delta ABC \) is a regular triangle. Thus the equation system in (3) can be simplified as
\[
\left\{ \begin{array}{l}
Y^2 + Z^2 - YZp - a'^2 = 0 \\
Z^2 + X^2 - XZq - a'^2 = 0 \\
X^2 + Y^2 - XYr - a'^2 = 0
\end{array} \right. \\
(6)
\]

Inferring to [14], solutions for an equation system in triangular form are well-determined. For example, we can easily reduce the solution of an equation system in triangular form to the solution of univariate equations. For a polynomial set \( PS \) and a polynomial \( I \), let \( \text{Zero}(PS) \) be the set of solutions of the equation system \( PS = 0 \), and \( \text{Zero}(PS/I) = \text{Zero}(PS) - \text{Zero}(I) \). Based on the “reality conditions” listed in (4), we can use \( l_0 \neq 0 \) to simplify the computation. That is to say, we consider \( \text{Zero}(ES/I_0) \).

We can decompose \( \text{Zero}(ES/I_0) \) into 10 disjoint components with Wu-Ritt’s zero decomposition method [14] :
\[
\text{Zero}(ES/I_0) = \bigcup_{i=1}^{10} C_i \\
(9)
\]

In the above formula, \( C_i = \text{Zero}(TS_i/T_i), \) \( i = 1, \cdots, 9 \) and \( C_{10} = \text{Zero}(TS_{10}/T_{10}) \cup \text{Zero}(TS_{11}/T_{11}) \), where \( TS_i \) are polynomial equations in triangular form and \( T_i \) are polynomials. From this decomposition, we get the following observations:

a. This decomposition provides a complete set of analytical solutions for the P3P problem for the reason that the solutions for each triangular set are well determined.

b. From Table 1 and the analysis following it, we can easily get the observation that under the reality condition (2) there are at most four distinct solutions, which was proven previously only for the main component.

c. It is proved that the above decomposition provides a complete and robust way to find the solutions to the P3P problem.
Considering the above discussion about (9), we sort out a general procedure to solve the P3P problem as the following:

- Calculate $p, q, r, a'$ from the camera intrinsic parameter matrix, the coordinates of the control points and corresponding image points.
- Determine the decomposition of the equation system (7).
- Determine the number of solutions.
- Determine the final solution to the P3P problem.

2) EPnP Solution

In most of the cases, there are usually four or more correspondences in the PnP problem, and it is acknowledged that larger point sets have redundancy decreasing the interference of noise. But larger point sets also bring a problem: the procedure of processing is much more complex. Considering the complexity, it is necessary to increase the efficiency of the PnP algorithms.

An algorithm proposed by Quan and Lan in 1999 is based on singular value decomposition (SVD) involving $O(n^3)$ operations [15]. Because this method is on account of the fact that the inter-point distances constraints introduce quadratic terms, the linearization of the equations generates additional parameters making the system more complex. A different method proposed by Fiore in 2001 avoids the necessity for these constraints by initially forming a set of linear equations from which the world to camera rotation and translation parameters are eliminated, which allows directly recovering the point depths while ignoring inter-point distances [16]. This procedure reduces the complexity of the estimation of the camera pose to $O(n^2)$, making it possible for the algorithm to perform in a real-time way while $n$ is large. Efficient PnP (EPnP) method was then proposed in 2009 by Lepetit et al. Its central idea is expressing the $n$ 3D points as a weighted sum of four virtual control points [17], which reduces the complexity to $O(n)$, at the same time keeping the estimation accuracy good. In the following part, we are going to introduce the EPnP solution in detail.

Assume that we already know 2D image projections and 3D coordinates in the world coordinate system of a set of $n$ reference points. The next step is to retrieve their coordinates in the camera coordinate system as most of the proposed solutions to the PnP problem. As mentioned above, the central idea is to express all the coordinates of the given points as a weighted sum of four non-coplanar virtual control points, which means that the coordinates of the control points in the camera coordinate system become unknown in the problem. In large $n$'s cases, this is a much smaller number of unknowns that the $n$ depth values that traditional approaches have to deal with and is key to his efficient implementation. Let the $n$ reference points in the world coordinate system be $p_i, i = 1, \ldots, n$, and the 4 control points be $c_j, j = 1, \ldots, 4$. Specify that the point coordinates are expressed in the world and camera coordinate system by using the $^w$ and $^c$ superscript respectively. We can express each reference point as a weighted sum of the control points

$$p_i^w = \sum_{j=1}^4 \alpha_{ij} c_j^w, \quad \text{with} \quad \sum_{j=1}^4 \alpha_{ij} = 1,$$

where the $\alpha_{ij}$ are homogeneous barycentric coordinates which are uniquely defined and can easily be estimated. The relation remains same in the camera coordinate system:

$$p_i^c = \sum_{j=1}^4 \alpha_{ij} c_j^c.$$

Theoretically, we can choose the control points at random. In practice, however, an increase in the stability of our method has been found while taking the centroid of the reference points as one, and selecting the rest in a way that they form a basis aligned with the principal directions of the data. This

| $C_i, i =$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----------|---|---|---|---|---|---|---|---|---|---|
| No. of Solutions | 4 | 3 | 2 | 2 | 1 | 1 | 2 | 4 | 3 |

TABLE I. THE MAXIMAL NUMBER OF SOLUTIONS FOR EACH COMPONENT
makes sense because it equals conditioning the linear system of equations. We will introduce the system below in a way of normalizing the point coordinates, which is nearly the same to the one we recommend for the classic DLT algorithm [18].

Let $A$ be the camera intrinsic calibration matrix and $\{u_i\}$ be the 2D projections of the $\{p_i\}$ reference points, $i = 1, ..., n$, and we get

$$\forall i, w_i \left[ \begin{array}{c} u_i \\ 1 \end{array} \right] = Ap_c^i = A \sum_{j=1}^{4} \alpha_{ij} c_j^c,$$

where the $w_i$ are scalar projective parameters. With the parameters including the specific 3D coordinates $[x^c_j, y^c_j, z^c_j]$ of each $c_j^c$ control point, the 2D coordinates $[u_i, v_i]$ of the $u_i$ projections, the $f_u, f_v$ focal length coefficients and the $(u_c, v_c)$ principal point appearing in the $A$ matrix, (10) can be expanded into

$$\forall i, w_i \left[ \begin{array}{c} u_i \\ v_i \end{array} \right] = \left[ \begin{array}{ccc} f_u & 0 & u_c \\ 0 & f_v & v_c \end{array} \right] \sum_{j=1}^{4} \alpha_{ij} \left[ \begin{array}{c} x^c_j \\ y^c_j \\ z^c_j \end{array} \right].$$

The 12 control point coordinates $\{(x^c_j, y^c_j, z^c_j)\}$, $j = 1, ..., 4$, and the $n$ projective parameters $\{w_i\}$, $i = 1, ..., n$ are parameters of this linear system unknown. It can be inferred from the last row of (11) that $w_i = \sum_{j=3}^{4} \alpha_{ij} z^c_j$. Thus, substitute this expression in the first two rows, and for each reference point we get two linear equations:

$$\sum_{j=1}^{4} \alpha_{ij} f_u x^c_j + \alpha_{ij} (u_c - u_i) z^c_j = 0,$$

$$\sum_{j=1}^{4} \alpha_{ij} f_v y^c_j + \alpha_{ij} (v_c - v_i) z^c_j = 0.$$

Given that there are no projective parameter $w_i$ in those equations, we generate a linear system based on the concatenation of the two equations for all $n$ reference points:

$$Mx = 0$$

where $x = [c_1^cT, c_2^cT, c_3^cT, c_4^cT]^T$ is a 12-vector made of the unknowns, and $M$ is a $2n \times 12$ matrix which is generated by arranging the coefficients of (12) and (13) for each reference point. Because of difference that (12) and (13) do not involve the image referential system, there is no need to normalize the 2D projections as in the case of DLT. Therefore the solution belongs to the null space, or kernel, of $M$.

3. APPLICATION SCENARIOS

Up to now, ego-motion estimation has been playing an important role in many fields, such as computer animation, automation, augmented reality, image analysis, automated cartography, photogrammetry, robotics, model-based machine vision systems. As an effective method to determine the pose of a calibrated camera, the PnP problem provides quantities of solutions to solve the problem, leading to a wide range of application in many fields as well.

The P3P solutions contributed in the field of robotics. Even early in 1991, William J. Wolfe et al [18] came up with a geometric explanation of the camera-triangle configuration and provided a justification for the common thought that there were two solutions in many cases. In the field of computer vision, the algorithm of EPnP inspired an updated method faster and more accurate proposed by Adrian Penate-Sanchez and Juan Andrade-Cetto in 2013 [19]. In practice, Iryna Skrypnyk and David G. Lowe in 2004 [20] designed a system to achieve feature point-based camera tracking in a robust way.

4. CONCLUSION

In this paper, we presented the most popular model of pinhole camera to describe the projection of a point from the 3-D world coordinate system to the 2-D image plane. Thus, we introduced some preliminary aspects of solutions for PnP problems. Among all of the solutions, we reviewed two common methods based on the pinhole camera model in detail. The P3P solution, a method with 3 pairs of corresponding
points, is the smallest subset of control points that yields a finite number of solutions. A common P3P solution is based on publications of Gao et al and Wu-Ritt’s zero decomposition method, providing a complete and robust way to find the solutions to the P3P problem. As point sets become larger, four or more correspondences bring complexity in processing. To enhance the efficiency of solutions, Efficient PnP (EPnP) method is proposed. Expressing the n 3D points as a weighted sum of four virtual control points, the EPnP solution reduces the complexity to \(O(n)\). Given that in most of the cases we have to deal with a relatively complex situation, the EPnP solution is now used more than the former one.

REFERENCES

[1] Mcneff J G. The global positioning system[J]. IEEE Transactions on Microwave Theory & Techniques, 2002, 50(3):645-652.

[2] Syed Z F, Aggarwal P, Goodall C, et al. A new multi-position calibration method for MEMS inertial navigation systems[J]. Measurement Science & Technology, 2007, 18(7):1897.

[3] Nouredin A, Karamat T B, Eberts M D, et al. Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications. IEEE Trans Veh Technol[J]. IEEE Transactions on Vehicular Technology, 2009, 58(3):1077-1096.

[4] Aqel M O A, Marhaban M H, Saripan M I, et al. Review of visual odometry: types, approaches, challenges, and applications[J]. Springerplus, 2016, 5(1):1897.

[5] Choi B S, Lee J W, Lee J J, et al. A Hierarchical Algorithm for Indoor Mobile Robot Localization Using RFID Sensor Fusion[J]. Industrial Electronics IEEE Transactions on, 2011, 58(6):2226-2235.

[6] Scaramuzza D, Fraundorfer F. Visual Odometry [Tutorial][J]. Robotics & Automation Magazine IEEE, 2011, 18(4):80-92.

[7] Fischler M A. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the Acm, 1980, 24(6):381-395.

[8] Strutz T. Data Fitting and Uncertainty (2nd edition)[M]// Data Fitting and Uncertainty. Vieweg+Teubner, 2016.

[9] J. A. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst uber seine Anwendungen in Geod asie. In Gruenerts Archiv fur Mathematik und Physik, 1841

[10] S. Finsterwalder and W. Scheufele. Das Ruckw arteinschneiden im Raum. Verlag Herbert Wichmann, Berlin, Germany, 1937

[11] R. Haralick, C. Lee, K. Ottenberg, and M. Nolle. Analysis and solutions of the three point perspective pose estimation problem. In IEEE International Conference on Computer Vision and Pattern Recognition, Maui, USA, 1991.

[12] L. Quan and Z. Lan. Linear n-point camera pose determination. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(8):774–780, 1999.

[13] X. Gao, X. Hou, J. Tang, and H. Cheng. Complete solution classification for the perspective-three-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(8):930–943, 2003.

[14] Wu W J. Basic principles of mechanical theorem proving in elementary geometries[J]. Journal of Automated Reasoning, 1986, 2(3):221-252.

[15] Quan L, Lan Z. Linear n-point camera pose determination[J]. IEEE Transactions on pattern analysis and machine intelligence, 1999, 21(8):774-780.

[16] Fiore P D. Efficient linear solution of exterior orientation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(2):140-148.

[17] Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate o (n) solution to the pnp problem[J]. International journal of computer vision, 2009, 81(2):155-166.

[18] Mathis D, Magee M. The perspective view of three points[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 1991, 13(1):66-73.
[19] Penate-Sanchez A, Andrade-Cetto J, Moreno-Noguer F. Exhaustive Linearization for Robust Camera Pose and Focal Length Estimation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(10):2387-2400.

[20] Skrypnyk I, Lowe D G. Scene modelling, recognition and tracking with invariant image features[C]// IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE, 2007:110-119.