A Robust $O(n)$ Solution to the Perspective-n-Point Problem

Shiqi Li, Chi Xu, and Ming Xie, *Member*, *IEEE*

Abstract—We propose a noniterative solution for the Perspective-n-Point ($\mathrm{P}n\mathrm{P}$) problem, which can robustly retrieve the optimum by solving a seventh order polynomial. The central idea consists of three steps: 1) to divide the reference points into 3-point subsets in order to achieve a series of fourth order polynomials, 2) to compute the sum of the square of the polynomials so as to form a cost function, and 3) to find the roots of the derivative of the cost function in order to determine the optimum. The advantages of the proposed method are as follows: First, it can stably deal with the planar case, ordinary 3D case, and quasi-singular case, and it is as accurate as the state-of-the-art iterative algorithms with much less computational time. Second, it is the first noniterative $\mathrm{P}n\mathrm{P}$ solution that can achieve more accurate results than the iterative algorithms when no redundant reference points can be used ($n \leq 5$). Third, large-size point sets can be handled efficiently because its computational complexity is $O(n)$.

Index Terms—Perspective-n-point problem, camera pose estimation, augmented reality.

1 Introduction

The term "Perspective-n-Point problem" ($\mathrm{P}n\mathrm{P}$) was coined by Fischler and Bolles [1] for the problem of determining the pose of a calibrated camera from n correspondences between 3D reference points and their 2D projections. It has many applications in computer vision [2], [3], [4], photogrammetry [5], robotics, augmented reality, etc. In practice, applications such as feature point-based camera tracking [6], [7] require solutions that can deal with both hundreds of feature points efficiently and a few feature points ($n \leq 5$) accurately.

The solutions for the $\mathrm{P}n\mathrm{P}$ problem are classified as iterative or noniterative methods. Noniterative methods are efficient, but their limitation is the instability in the presence of noise, especially when $n \leq 5$. The 3-point problem, which is the smallest subset of $\mathrm{P}n\mathrm{P}$, has had many closed form solutions since 1841 [1], [8], [9]. The stability of the 3-point problem is limited due to its multiple solutions [10], [11]. To derive a unique solution, more points should be considered. Fischler and Bolles [1] divided the n-point problem into 3-point subsets and eliminated the multiplicity by checking their consistency. Ameller et al. [12] proposed efficient direct solutions for the 3-point and the 4-point problems by SVD null space estimation. Zhi and Tang [13] presented a linear algorithm for 4-point problem which was easy to implement. Abidi and Chandra [14] proposed a solution for the coplanar 4-point problem from a perspective camera with unknown focal length. Bujnak et al. [15] generalized the solution of Abidi and Chandra to four nonplanar reference points. Triggs [16] proposed a novel method for camera calibration with 4 or 5-point set. The solution is considered as a combination of the null eigenvectors, and its central idea deeply affected the research on the $\mathrm{P}n\mathrm{P}$ problem. However, the existing noniterative solutions for fewer than 5 points tend to be unstable in practice because no redundant information is available.

The stability of the noniterative methods can be enhanced by introducing redundant points as additional information. The well-known Direct Linear Transformation (DLT) algorithm [17] achieves relatively accurate results from a large number of points. Quan and Lan [18] developed a family of linear solutions by taking advantage of data redundancy. Ansar and Daniilidis presented linear solutions for both n points and n lines [19]. However, as pointed out by Lepetit et al. [4], many noniterative methods are time-consuming for large-size point sets due to high computational complexity. For example, Ansar and Daniilidis' is $O(n^8)$ [19], Quan and Lan's is $O(n^5)$ [18], and Fiore et al.'s is $O(n^2)$ [20]. Schweighofer and Pinz [21] proposed a globally optimal $O(n)$ solution for large-size point sets using a semidefinite positive program, but it is not suitable for real-time application. The great work of Lepetit et al. [4] presented an efficient noniterative algorithm with linear complexity in n by expressing the solution as weighted sum of null eigenvectors. It is one of the most accurate noniterative solutions until now.

When the redundant points are unavailable, accurate results can be achieved by introducing iterative schemes based on the minimization of nonlinear cost functions [22], [23], [24], [25]. Although the iterative algorithms are more accurate than the noniterative ones, the drawbacks of the iterative methods are: 1) the instability due to the local minima of the cost functions, and 2) high computational cost.

The local minima of the $\mathrm{P}n\mathrm{P}$ problem are closely related to the configuration of the 3D point set. Let a matrix $M = [X_1\ X_2 \cdots X_n]^T$, where X_i is the 3D coordinate of the reference point and n is the size of the point set. According to the 3×3 matrix $M^T M$, we categorize the configuration of the reference points into three groups as follows:[1]

1. **Ordinary 3D case.** $Rank(M^T M) = 3$ and the smallest eigenvalue of $M^T M$ is not close to zero. In this case, the widely used iterative algorithm of Lu [22] can stably converge to the global optimum.
2. **Planar case.** $Rank(M^T M) = 2$. In this case, the reference points lie on a plane, and the cost function of $\mathrm{P}n\mathrm{P}$ has two distinct minima, which would lead to significant unstable results [26], [27]. Schweighofer and Pinz [27] enhanced the robustness of Lu [22] by taking local minima into account, and their method is one of the most robust and accurate algorithms for the planar case. The $Rank(M^T M) = 1$ *or* 0 case is not considered as camera pose is undetermined.
3. **Quasisingular case.** $Rank(M^T M) = 3$ and the ratio of the smallest eigenvalue to the largest one is very small (< 0.05), $M^T M$ is "quasisingular." For example, when the reference points distribute in a long-narrow region $[1,2] \times [1,2] \times [4,8]$ ("quasilinear") or a thin-flat region $[1,2] \times [-2,2] \times [4,8]$ ("quasiplanar"), the iterative algorithms would be disturbed by the local minima as in the planar case and the method for planar targets [27] cannot deal with this kind of configuration because the points are not coplanar.

The classification of the configurations presented above is mainly inspired by Lepetit et al. [4]. Commonly, the configuration of the $\mathrm{P}n\mathrm{P}$ problem is classified as planar or nonplanar case. Recently, the novel work of Lepetit et al. [4] has shown that, when the reference points distribute in the region $[1,2] \times [1,2] \times [4,8]$ where the projections of the points cover only a small fraction of the image, the stability of existing methods degenerates significantly. They called this configuration "uncentered data." We find that in essence it is a kind of configuration between the "ordinary 3D case" and the "planar case" because the local minima of the points in the region $[1,2] \times [1,2] \times [4,8]$ are very similar to the planar points in the region $[2,2] \times [1,2] \times [4,8]$. What is more, for a more uncentered

1. Note that the 3×3 matrix $M^T M$ being used to determine the configuration is different from the 12×12 matrix in [4].

- *S. Li and C. Xu are with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China. E-mail: xuchi.hust@yahoo.com.cn, sqli@mail.hust.edu.cn.*
- *M. Xie is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798. E-mail: mmxie@ntu.edu.sg.*

Manuscript received 2 Dec. 2010; revised 4 Nov. 2011; accepted 13 Jan. 2012; published online 30 Jan. 2012.
Recommended for acceptance by A. Fitzgibbon.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-12-0917.
Digital Object Identifier no. 10.1109/TPAMI.2012.41.

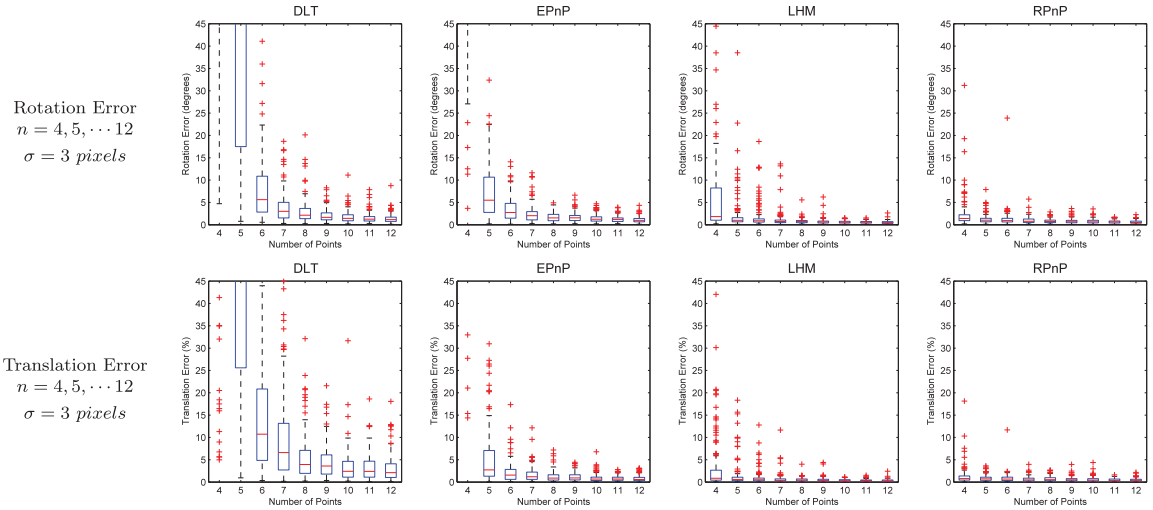

Fig. 1. The accuracy of our method $\mathbf{RP}n\mathbf{P}$ is compared to the state-of-the-art methods using the box-plot representation. (In the box-plot representation, the box denotes the first Q1 and the third Q3 quartiles, the red horizontal line denotes the median, the dashed vertical line denotes the statistical extent taken to be 1.5 (Q3-Q1) from the ends of the box, and the red crosses denote points lying outside of this range. In this paper, the box-plot representation is defined the same as in [4].) The rotation and translation errors are plotted as a function of the number of points n from 4 to 12. **DLT** [17] denotes the well-known direct linear transformation method, $\mathbf{EP}n\mathbf{P}$ [4] denotes one of the best noniterative solutions by Lepetit et al., and **LHM** [22] denotes the widely used iterative solution by Lu.

region $[1,2] \times [1,2] \times [7,8]$ in the ordinary 3D case, the iterative algorithm of Lu [22] works very well and stably converges to the global minimal.

Recently, an efficient noniterative pose estimator for 4-point square markers [28] has been reported to be as accurate as the best iterative solver. However, it can only deal with a specific kind of square target.

In this paper, a robust noniterative solution of PnP (which we refer to as RPnP) is presented. The proposed method works well for both nonredundant point sets ($n \leq 5$) and redundant point sets (see Fig. 1). RPnP retrieves correct results robustly in the three configurations mentioned above, and its computational complexity grows linearly with n.

The rest of the paper is organized as follows: The 2-point and 3-point constraints and Quan and Lan's outstanding work [18] closely related to ours are reviewed in Section 2. The details of the proposed method RPnP are presented in Section 3, and the experimental results using both synthetic data and real data are given in Section 4.

The source code of RPnP can be downloaded from http://xuchi.weebly.com/rpnp.html.

2 Background

2.1 The 2-Point Constraint

Given a calibrated camera and two reference points P_i and P_j with their corresponding 2D projections on the image plane as p_i and p_j (see Fig. 2), the correspondences $P_i \leftrightarrow p_i$ and $P_j \leftrightarrow p_j$ give a constraint on x_i and x_j, the unknown depths from the reference points to the camera center:

$$x_i^2 + x_j^2 - 2x_i x_j \cos\theta_{ij} - d_{ij}^2 = 0,$$

where d_{ij} is the known distance between P_i and P_j, and θ_{ij} is the viewing angle from the camera center to p_i and p_j [18].

2.2 The 3-Point Constraint

Given three reference points P_i, P_j, and P_k, we have three constraints by dividing them into 2-point subsets:

$$\begin{cases} x_i^2 + x_j^2 - 2x_i x_j \cos\theta_{ij} - d_{ij}^2 = 0, \\ x_i^2 + x_k^2 - 2x_i x_k \cos\theta_{ik} - d_{ik}^2 = 0, \\ x_k^2 + x_j^2 - 2x_k x_j \cos\theta_{kj} - d_{kj}^2 = 0, \end{cases}$$

with three unknown depth variables x_i, x_j, and x_k. The equation system can be equivalently converted into a fourth order polynomial:

$$f(x) = ax^4 + bx^3 + cx^2 + dx + e = 0.$$

The P3P solver in [29] is used here to form the P3P polynomial, which is robust to the vertex permutation problem[2] and the geometric singularity problem.[3] When the P3P polynomial is solved, the depths of the reference points can be determined.

2.3 The n-Point Problem

The camera pose is undetermined from 2 and 3-point sets due to the multiple solutions. To retrieve a unique solution, at least 4 points should be involved. For the n-point problem, Quan and Lan's linear solution, closely related to our work [18], is briefly reviewed here. By selecting a base point P_i, the n-point set is divided into $\frac{(n-1)(n-2)}{2}$ 3-point subsets such as $\{P_iP_jP_k \mid j \neq i, k \neq i, k \neq j, j \in \{1, \ldots, n\}, k \in \{1, \ldots, n\}\}$, and each subset yields a fourth order polynomial. Linearization technology is employed to convert these nonlinear equations into a linear equation system. The limitation of the linearization technology is the inconsistency between the elements of the variable vector solved from the linear equation when noise is involved, which would lead to instability in practice. Insightful analysis on this subject can be found in [16].

3 The RPnP Method

3.1 Selecting a Rotation Axis

Given a calibrated camera and n 3D reference points P_i $(i = 1, \ldots, n)$ which are projected onto the normalized image plane as p_i, the edges between the 3D reference points are $\{\overline{P_iP_j} \mid i > j, i \in \{1, \ldots, n\}, j \in \{1, \ldots, n\}\}$ (see Fig. 3). First, we select an edge $\overline{P_{i0}P_{j0}}$ from this set as a rotation axis, based on which a new orthogonal coordinate frame $O_aX_aY_aZ_a$ is created. The origin of $O_aX_aY_aZ_a$ is at the center of $\overline{P_{i0}P_{j0}}$, and the direction of its Z_a-axis is the same as

2. The vertex permutation problem denotes that the permutation of the triangle vertex can significantly affect the numerical stability of the P3P solution [9].

3. The geometric singularity problem is caused by some unstable geometric structure in which a small change in the position of the perspective center will lead to a big change in the result [9].

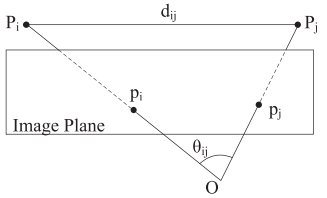

Fig. 2. The 2-point constraint.

that of the vector $\overrightarrow{P_{i0}P_{j0}}$. After creating $O_aX_aY_aZ_a$, the 3D coordinates of P_i in the world space are converted into the new orthogonal coordinate frame. The transformation from $O_aX_aY_aZ_a$ to the camera coordinate frame $O_cX_cY_cZ_c$ can be determined by the rotation axis Z_a, a rotation angle α around Z_a, and a translation vector $\overrightarrow{O_cO_a}$.

To select a rotation axis from $\{\overline{P_iP_j}\}$, n edges are randomly sampled and the edge with the longest projection length $\|p_ip_j\|$ is selected. Longer edges are less affected by noise added to the end points. The influence of the rotation axis selection step is evaluated in Section 4.

It is known that it is important for conventional PnP solvers (such as DLT) to center the 3D coordinates of P_i in the world space. However, the points are not required to be centered prior to RPnP because this step is equivalently done by converting the 3D coordinates of P_i into $O_aX_aY_aZ_a$.

3.2 Determination of Rotation Axis Using Least-Squares Residual

We divide the n reference points into $(n-2)$ subsets, each of which contains three points such as $\{P_{i0}P_{j0}P_k \mid k \neq i0, k \neq j0\}$. By using the 3-point constraint, each subset yields one polynomial of order 4 as follows:

$$\begin{cases} f_1(x) = a_1x^4 + b_1x^3 + c_1x^2 + d_1x + e_1 = 0, \\ f_2(x) = a_2x^4 + b_2x^3 + c_2x^2 + d_2x + e_2 = 0, \\ \cdots, \\ f_{n-2}(x) = a_{n-2}x^4 + b_{n-2}x^3 \\ \qquad\qquad + c_{n-2}x^2 + d_{n-2}x + e_{n-2} = 0. \end{cases} \tag{1}$$

Instead of directly solving the nonlinear equation system (1) by the linearization technique [18], which would lead to an inconsistent result from redundant equations, we explore the local minima of the equation system in terms of least-squares residual. First, a cost function F is defined as the square sum of the polynomials in (1), and $F = \sum_{i=1}^{n-2} f_i^2(x)$. The minima of F can 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 can be easily solved by the eigenvalue method [30]. As soon as x is determined, the depths of P_{i0} and P_{j0} can be calculated according to [29], and then the rotation axis of $O_aX_aY_aZ_a$ can be determined as $Z_a = \overrightarrow{P_{i0}P_{j0}}/\|P_{i0}P_{j0}\|$.

The eighth order polynomial F has at most four minima. A brief proof is as follows.

Assuming F has m stationary points, in which there are m_1 minima and m_2 maxima, $m_1 + m_2 \leq m$. As there exists at least one maximum between two minima, we have $m_1 - 1 \leq m_2$. As the stationary points of F are the real roots of F', we have $m \leq 7$. Therefore, $2m_1 - 1 \leq m_1 + m_2 \leq 7$, and we have $m_1 \leq 4$.

For each minimum, the camera pose will be estimated by solving the rotation angle and translation vector, which will be described in Section 3.3.

3.3 Equations for Solving the Rotation Angle and Translation Vector

When the Z_a-axis of $O_aX_aY_aZ_a$ is determined, the rotation matrix from $O_aX_aY_aZ_a$ to the camera $O_cX_cY_cZ_c$ can be expressed as

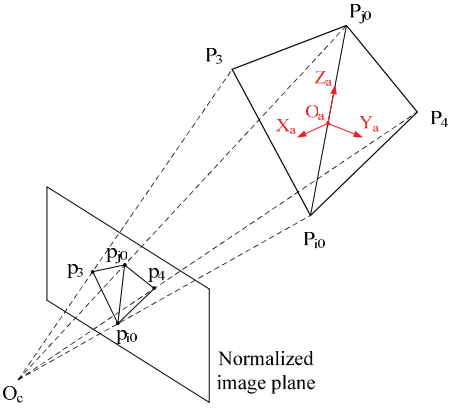

Fig. 3. The projection of the reference points.

$$R = R'rot(Z,\alpha) = \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix} \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where R' is an arbitrary rotation matrix whose third column $[r_7\ r_8\ r_9]^T$ equals the rotation axis Z_a and R' should meet the orthogonal constraint of the rotation matrix. $rot(Z,\alpha)$ denotes a rotation of α degree around the Z-axis with $c = \cos\alpha$ and $s = \sin\alpha$.

The projection from the 3D points to the 2D normalized image plane can be expressed as follows:

$$\lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix} \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \tag{3}$$

where (u_i, v_i) are the normalized coordinates of image point p_i and $t = [t_x\ t_y\ t_z]^T$ is the translation vector.

We arrange the terms of (3) to a $2n \times 6$ homogenous linear equation system with an unknown variables vector $[c\ s\ t_x\ t_y\ t_z\ 1]^T$,

$$[\, A_{2n\times 1} \quad B_{2n\times 1} \quad C_{2n\times 4} \,] \begin{bmatrix} c \\ s \\ t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} = 0, \tag{4}$$

where

$$A_{2n\times 1} = \begin{bmatrix} u_1X_1r_3 - Y_1r_4 - X_1r_1 + u_1Y_1r_6 \\ v_1X_1r_3 - Y_1r_5 - X_1r_2 + v_1Y_1r_6 \\ \cdots \\ \cdots \\ u_nX_nr_3 - Y_nr_4 - X_nr_1 + u_nY_nr_6 \\ v_nX_nr_3 - Y_nr_5 - X_nr_2 + v_nY_nr_6 \end{bmatrix},$$

$$B_{2n\times 1} = \begin{bmatrix} Y_1r_1 + u_1X_1r_6 - u_1Y_1r_3 - X_1r_4 \\ Y_1r_2 + v_1X_1r_6 - v_1Y_1r_3 - X_1r_5 \\ \cdots \\ \cdots \\ Y_nr_1 + u_nX_nr_6 - u_nY_nr_3 - X_nr_4 \\ Y_nr_2 + v_nX_nr_6 - v_nY_nr_3 - X_nr_5 \end{bmatrix},$$

$$C_{2n\times 4} = \begin{bmatrix} -1 & 0 & u_1 & u_1r_9Z_1 - r_7Z_1 \\ 0 & -1 & v_1 & v_1r_9Z_1 - r_8Z_1 \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ -1 & 0 & u_n & u_nr_9Z_n - r_7Z_n \\ 0 & -1 & v_n & v_nr_9Z_n - r_8Z_n \end{bmatrix}.$$

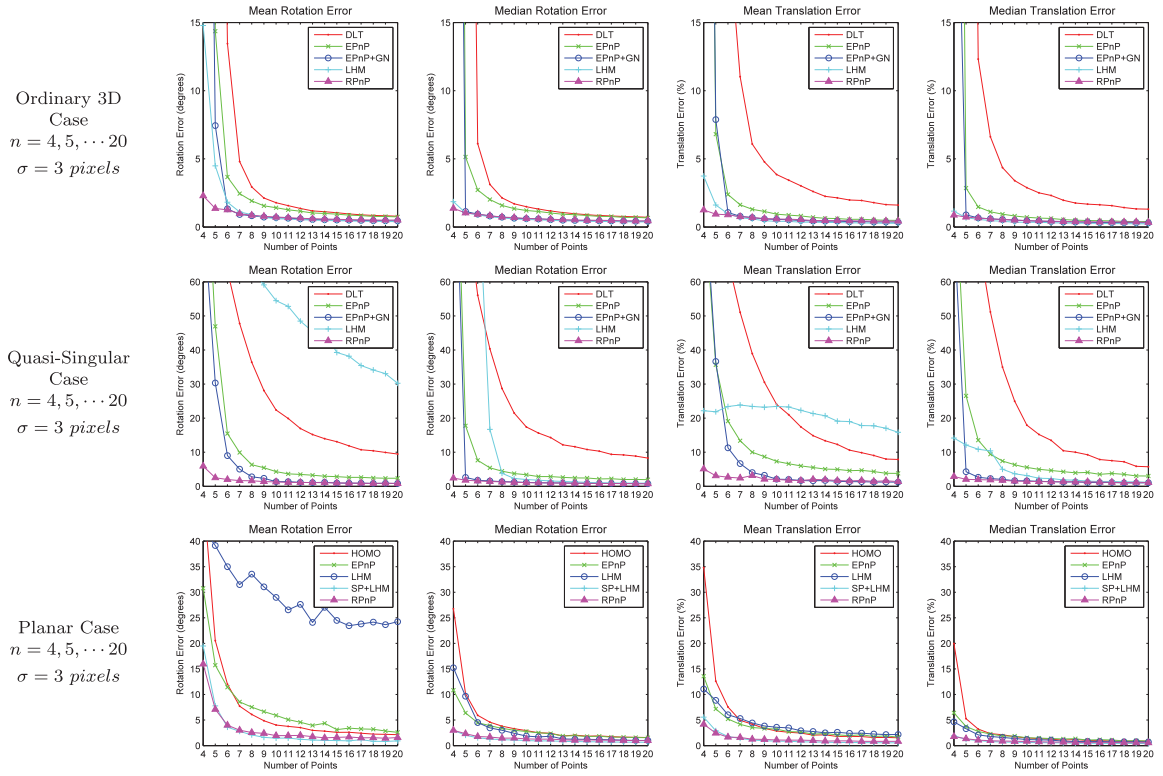

Fig. 4. The mean and median rotation and translation errors in the ordinary 3D case (first row), the quasisingular case (second row), and the planar case (third row).

The unknown variables c, s, t_x, t_y, and t_z can be retrieved by solving this linear equation system using Singular Value Decomposition (SVD) [30].

3.4 Determination of Camera Pose

As the solution of (4) may not exactly meet the triangular constraint $c^2 + s^2 = 1$ due to the noise in practice, the orthogonal constraint is imposed on the rotation matrix R in (2). To normalize R, we estimate the 3D coordinates of reference points in the camera coordinate frame $O_c X_c Y_c Z_c$ using the unnormalized R and t, and then the normalized camera pose can be retrieved by a standard 3D alignment scheme [31].

The cost function F defined in (1) has at most four local minima, and we estimate the camera pose from each local minimum and select the result with the least reprojection residual as the optimum of the solution.

4 Results

4.1 Experiments with Synthetic Data

We used the experimental configuration of [4] to compare the results of our system to the existing state of the art. Given a virtual perspective camera with image size 640×480 pixels and focal length 800 pixels, the 3D reference points were randomly generated in the camera coordinate frame. In the ordinary 3D case, the reference points were uniformly distributed in the range $[-2, 2] \times [-2, 2] \times [4, 8]$. In the quasisingular case, the reference points were uniformly distributed in the range $[1, 2] \times [1, 2] \times [4, 8]$. Different levels of Gaussian noise were added to the projected image points, and for each noise level 1,000 test data sets were generated. The source code can be downloaded from http://xuchi.weebly.com/rpnp.html .

In the ordinary 3D case and the quasisingular case, the following methods were compared:

1. **DLT.** The well-known direct linear transformation method [17].
2. **EPnP.** The efficient $O(n)$ noniterative solution of PnP by Lepetit et al. [4], which achieves excellent results when $n > 5$. It is one of the best noniterative solutions.
3. **EPnP+GN.** The EPnP method followed with a Gauss-Newton optimizer.
4. **LHM.** One of the best iterative solutions of PnP by Lu [22]. It is globally convergent in the ordinary 3D case, but not that stable in the quasisingular and the planar cases.

In the planar case, the following methods were compared:

1. **HOMO.** The homography method for planar targets [32].
2. **EPnP.** The EPnP solver for the planar case. EPnP+GN is not considered because it does not improve the accuracy in the planar case [4].
3. **LHM.**
4. **SP + LHM.** An iterative algorithm of Schweighofer and Pinz [27] initialized with a weak perspective assumption, which is one of the most robust and accurate solutions for the planar case.

As can be seen in Fig. 4, in the ordinary 3D case, RPnP stably reaches the correct result for $n \geq 4$ and its accuracy is of the same level as the iterative algorithms. In the quasisingular case, LHM suffers from the local minima problem, EPnP and EPnP+GN achieve excellent results when $n > 6$, and RPnP stably reaches highly accurate results from $n = 4$ to 20. In the planar case, RPnP stably reaches the global optimum, and it matches the accuracy of the iterative scheme SP + LHM with much less computational time.

One significant advantage of RPnP compared to the existing state-of-the-art methods is that RPnP can achieve more accurate results than the iterative algorithms when no redundant points are available. As can be seen in Fig. 5, only RPnP and LHM can achieve effective results when $n = 4$ in the ordinary 3D case, and the mean rotation and translation errors of RPnP are significantly better than that of LHM. When $n = 5$, EPnP, EPnP+GN, LHM,

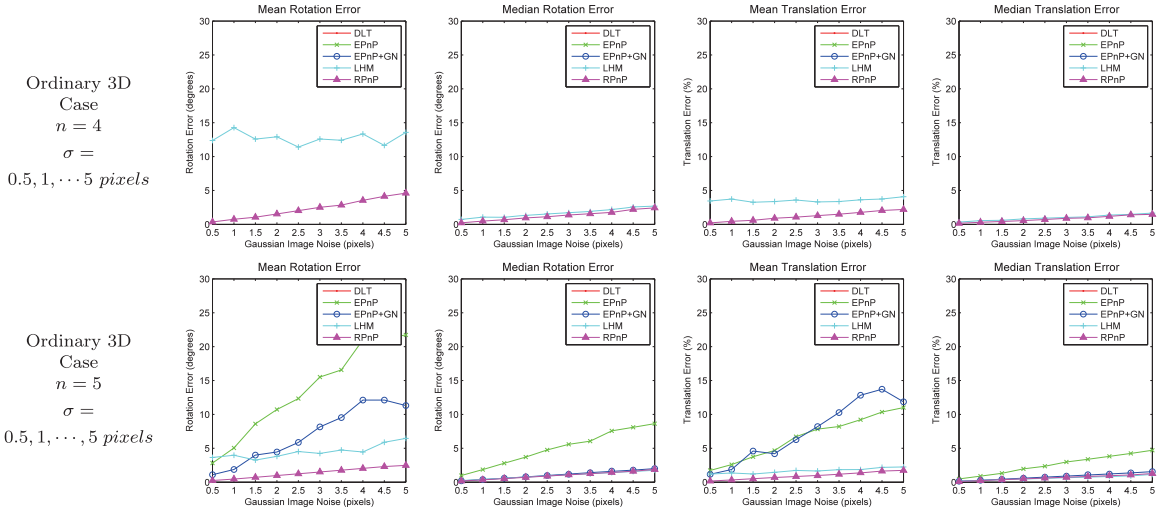

Fig. 5. The mean and median rotation and translation errors for $n = 4$ (first row) and $n = 5$ (second row).

and RPnP achieve effective results, and RPnP is still much better than others in accuracy.

RPnP is highly efficient and its computational time grows linearly with n. As can be seen in Fig. 6, the average execution times are plotted as a function of the number of points n from 4 to 100. The method was implemented in MATLAB and 1,000 test runs were performed. The efficiency of RPnP is significantly better than that of LHM, and is also better than that of EPnP.

As the rotation axis of RPnP is randomly selected, as described in Section 3.1, the influence of the random selection step is shown in Fig. 7. The selection is performed in three different ways: 1) **RPnP1** denotes to randomly select an edge as the rotation axis; 2) **RPnP*** denotes to select an edge with the longest projection length from all the $\frac{n(n-1)}{2}$ edges in $\{\overline{P_iP_j}\}$; 3) **RPnP** denotes the default setting which randomly samples n edges and selects the one with the longest projection length. The experiment was performed in the ordinary 3D case, noise level $\sigma = 3$, $n = 4, \ldots, 20$. The result of DLT is also plotted as a reference. We can see that RPnP* is the most accurate, and RPnP is as accurate as RPnP*. The rotation and translation error of RPnP1 is a little bigger than that of RPnP and RPnP*, but RPnP1 can still achieve accurate results. The complexity of the rotation axis selection step of RPnP is $O(n)$, and it takes only a small fraction (about 6.8 percent) of the total computational time. The computational time of the rotation axis determination (Section 3.2) and the camera pose estimation (Sections 3.3 and 3.4) steps are 40.7 and 52.5 percent, respectively.

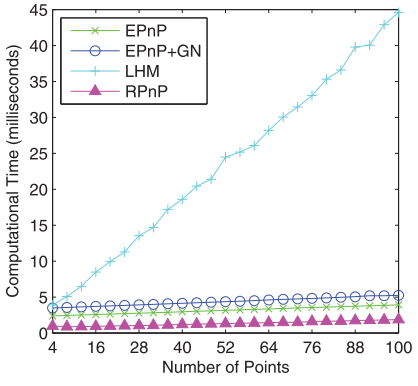

Fig. 6. The average computational time of the compared methods. The computational times of a MATLAB implementation on a standard PC are plotted as a function of the number of the points n.

The average numbers of sampling iterations in the RANSAC scheme [1] are given in Table 1. k-point $(k = 3, \ldots, 7)$ subsets are used for random sampling. The experiment was performed in the ordinary 3D case, noise level $\sigma = 3$, $n = 50$, and the percentage of outliers varies between 10 and 50 percent. The RANSAC iteration

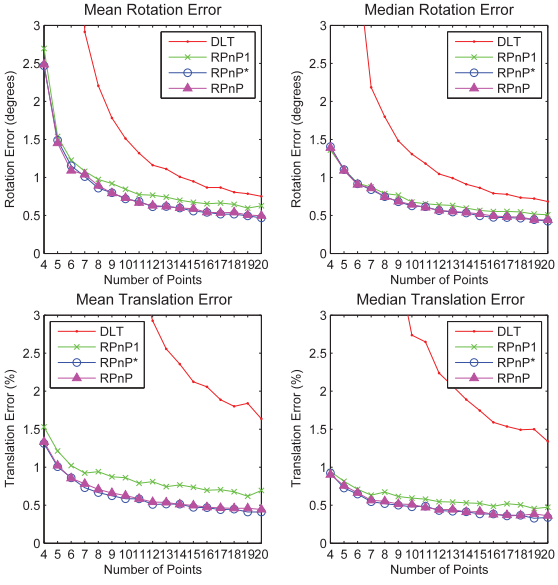

Fig. 7. The influence of the rotation axis selection step. The selection is performed in three different ways, and their accuracies are plotted as a function of the number of the points n.

TABLE 1
Mean RANSAC Iterations Required

Outliers	3pt	4pt	5pt	6pt	7pt
10%	2.2	1.7	1.7	1.9	2.1
20%	3.4	2.7	3.3	4.0	4.8
30%	5.4	4.9	6.4	9.3	13.0
40%	9.0	10.2	15.0	24.7	$41.6^{(0.2\%)}$
50%	20.8	24.4	$38.5^{(0.3\%)}$	$78.1^{(5.5\%)}$	$134.8^{(25\%)}$

*Note: k*pt $(k = 3, \ldots, 7)$ *denotes the number of points in the subset for random sampling. The value in the parentheses denotes the failure rate after 200 iterations of RANSAC. The data without this value denotes that the failure rate is 0.*

Fig. 8. Top left and bottom left: The calibrated reference images. Others: The input images. The objects are augmented with their contours using RPnP for pose estimation.

stops when the inliers are more than 30 percent. RPnP is used to estimate the camera pose for $k \geq 4$, and the P3P solver in [29] is used for $k = 3$. The difference among the cases $k = 3, \ldots, 7$ is not obvious when the percentage of outliers is small, but k becomes a critical factor when the outliers are more than 30 percent. For example, when the percentage of outliers is 50 percent, the average number of sampling iterations of 3 and 4-point subsets are 20.8 and 24.4, respectively, and their failure rates are 0; Meanwhile, for the 7-point subset, 134.8 trails are required on average, and the failure rate reaches 25 percent. The accuracies for $k = 3, \ldots, 7$ are almost the same because when a true hypothesis is found, not only the sampled subset but also all the inliers found are used to calculate the camera pose. Therefore, it is better to choose a small point subset ($k = 3$ *or* 4) for random sampling.

4.2 Experiments with Real Images

The results of the experiments on real images are shown in Fig. 8. The feature points of the input images were matched with that of the reference image using SIFT [33]. The RANSAC scheme was used to solve the camera pose from the matched point pairs, and the size of the sampling subset was 4. RPnP was employed by the RANSAC scheme for pose estimation. The red marks "+" in the input images denote the feature points matched with the reference image using the SIFT method, and the green marks "o" denote the reprojection of the verified feature points using the estimated camera pose.

5 Conclusions

In this paper, we have shown the details of a new proposed method to solve the PnP problem in a noniterative way. Our proposed RPnP method can effectively cope with data sets which could be planar, nonplanar, and quasisingular. The RPnP method with much less computational complexity is as accurate as the state-of-the-art iterative algorithms.

The robustness of the PnP solution is closely related to its local minima. In order to enhance the robustness, two privileged points are selected and a cost function is formed by summing the square of the P3P polynomials so that the optimum can be found. When the rotation axis is determined, two of the degrees of freedom of the camera pose are fixed and only a remaining rotation and three translation parameters are to be solved. It will largely enhance the numerical accuracy of the equation system because the number of unknown variables is reduced. From this viewpoint, RPnP is slightly different from the traditional PnP solutions because not all the n points contribute equally to the final result.

Until now, the PnP problem has relied on the time consuming iterative schemes for high accuracy when redundant points are not available ($n \leq 5$). The proposed method RPnP robustly handles both the nonredundant ($n \leq 5$) and redundant ($n > 5$) cases with much more efficiency than the iterative algorithms. RPnP is suitable for applications that need to handle both small and large point sets, such as the feature point-based object tracking application in computer vision and augmented reality.

Acknowledgments

The authors would like to thank Dr. Vincent Lepetit, Dr. Moreno-Noguer, and the anonymous reviewers for their helpful advice. The source code of the RPnP method can be downloaded from http://xuchi.weebly.com/rpnp.html. The corresponding author for this paper is Chi Xu.

References

[1] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381-395, 1981.
[2] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2003.
[4] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *Int'l J. Computer Vision*, vol. 81, no. 2, pp. 155-166, 2008.
[5] J. McGlone, E. Mikhail, and J. Bethel, *Manual of Photogrammetry*, fifth ed. Am. Soc. for Photogrammetry and Remote Sensing, 2004.
[6] V. Lepetit and P. Fua, "Keypoint Recognition Using Randomized Trees," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1465-1479, Sept. 2006.
[7] I. Skrypnyk and D. Lowe, "Scene Modelling, Recognition and Tracking with Invariant Image Features," *Proc. IEEE/ACM Third Int'l Symp. Mixed and Augmented Reality*, pp. 110-119, 2004.
[8] D. DeMenthon and L. Davis, "Exact and Approximate Solutions of the Perspective-Three-Pointproblem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1100-1105, Nov. 1992.
[9] B. Haralick, C. Lee, K. Ottenberg, and M.N. lle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem," *Int'l J. Computer Vision*, vol. 13, no. 3, pp. 331-356, 1994.
[10] X. Gao, X. Hou, J. Tang, and H. Cheng, "Complete Solution Classification for the Perspective-Three-Point Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930-943, Aug. 2003.
[11] W. Wolfe, D. Mathis, C. Sklair, and M. Magee, "The Perspective View of Three Points," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, pp. 66-73, Jan. 1991.
[12] M. Ameller, B. Triggs, and L. Quan, "Camera Pose Revisited-New Linear Algorithms," *Proc. European Conf. Computer Vision*, 2000.
[13] L. Zhi and J. Tang, "A Complete Linear 4-Point Algorithm for Camera Pose Determination," *AMSS, Academia Sinica*, vol. 21, pp. 239-249, 2002.

[14] M. Abidi and T. Chandra, "A New Efficient and Direct Solution for Pose Estimation Using Quadrangular Targets: Algorithm and Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 5, pp. 534-538, May 1995.
[15] M. Bujnak, Z. Kukelova, and T. Pajdla, "A General Solution to the P4P Problem for Camera with Unknown Focal Length," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.
[16] B. Triggs, "Camera Pose and Calibration from 4 or 5 Known 3D Points," *Proc. Seventh IEEE Int'l Conf. Computer Vision,* pp. 278-284, 1999.
[17] Y. Abdel-Aziz and H. Karara, "Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry," *Proc. ASP/UI Symp. Close-Range Photogrammetry,* pp. 1-18, 1971.
[18] L. Quan and Z. Lan, "Linear n-Point Camera Pose Determination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 8, pp. 774-780, Aug. 1999.
[19] A. Ansar and K. Daniilidis, "Linear Pose Estimation from Points or Lines," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 578-589, May 2003.
[20] P. Fiore, B. Syst, and N. Merrimack, "Efficient Linear Solution of Exterior Orientation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 140-148, Feb. 2001.
[21] G. Schweighofer and A. Pinz, "Globally Optimal o(n) Solution to the PnP Problem for General Camera Models," *Proc. British Machine Vision Conf.,* 2008.
[22] C. Lu, "Fast and Globally Convergent Pose Estimation from Video Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 6, pp. 610-622, June 2000.
[23] D. DeMenthon and L. Davis, "Model-Based Object Pose in 25 Lines of Code," *Int'l J. Computer Vision,* vol. 15, no. 1, pp. 123-141, 1995.
[24] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, "Object Pose: The Link between Weak Perspective, Paraperspective, and Full Perspective," *Int'l J. Computer Vision,* vol. 22, no. 2, pp. 173-189, 1997.
[25] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 11, pp. 1330-1334, Nov. 2000.
[26] D. Oberkampf, D. DeMenthon, and L. Davis, "Iterative Pose Estimation Using Coplanar Feature Points," *Computer Vision and Image Understanding,* vol. 63, no. 3, pp. 495-511, 1996.
[27] G. Schweighofer and A. Pinz, "Robust Pose Estimation from a Planar Target," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 2024-2030, Dec. 2006.
[28] S. Li and C. Xu, "Efficient Lookup Table Based Camera Pose Estimation for Augmented Reality," *Computer Animation and Virtual Worlds,* vol. 22, no. 1, pp. 47-58, 2011.
[29] S. Li and C. Xu, "A Stable Direct Solution of Perspective-Three-Point Problem," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 25, no. 5, pp. 627-642, 2011.
[30] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes: The Art of Scientific Computing.* Cambridge Univ. Press, 2007.
[31] S. Umeyama, "Least-Squares Estimation of Transformation Parameters between Two Point Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 4, pp. 376-380, Apr. 1991.
[32] S. Malik, G. Roth, and C. McDonald, "Robust 2D Tracking for Real-Time Augmented Reality," *Proc. Conf. Vision Interface,* vol. 1, no. 2, p. 12, 2002.
[33] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.