2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

CPnP: Consistent Pose Estimator for Perspective-n-Point Problem with Bias Elimination

Guangyang Zeng[1], Shiyu Chen[1], Biqiang Mu[2], Guodong Shi[3], and Junfeng Wu[1,4]

***Abstract*— The Perspective-n-Point (PnP) problem has been widely studied in both computer vision and photogrammetry societies. With the development of feature extraction techniques, a large number of feature points might be available in a single shot. It is promising to devise a consistent estimator, i.e., the estimate can converge to the true camera pose as the number of points increases. To this end, we propose a consistent PnP solver, named *CPnP*, with bias elimination. Specifically, linear equations are constructed from the original projection model via measurement model modification and variable elimination, based on which a closed-form least-squares solution is obtained. We then analyze and subtract the asymptotic bias of this solution, resulting in a consistent estimate. Additionally, Gauss-Newton (GN) iterations are executed to refine the consistent solution. Our proposed estimator is efficient in terms of computations—it has $O(n)$ time complexity. Simulations and real dataset tests show that our proposed estimator is superior to some well-known ones for images with dense visual features, in terms of estimation precision and computing time.**

I. Introduction

Given n 2D-3D point correspondences, inferring the pose of a camera is referred to as the Perspective-n-Point (PnP) problem. It has widespread applications in robotics [1]–[3], computer vision [4], augmented reality [5], etc.

Most of the existing works set out from ideal geometric relationships to construct geometry-constrained equations. They do not explicitly take the projection noises into account and overlook the noise propagation in equation transformations. As such, they rarely analyze the property of the designed estimator from the perspective of statistics, such as the bias and covariance of the estimator, which are important metrics in estimation theory. It is noteworthy that with the development of feature extraction techniques, a large number of feature points might be available in a single shot. For example, Fig. 3 in the experiment part are four images from ETH3D dataset [6], with two in the outdoor scenario and two in the indoor scenario. All images contain thousands of feature points, exhibiting the huge potential to yield a precise pose estimate. By noting this, we argue that it is promising to revisit the PnP problem through the lens of statistics.

The works that took the projection noises into account include the CEPPnP [7], MLPnP [8], and EPnPU [4]. In these works, the covariance matrix of noises is utilized to improve the estimation precision, e.g., GN iterations. However, they did not analyze the statistical property of the proposed estimators. Actually, due to the nonlinear nature of projection equations, all of these estimators are biased, and thus not consistent. The definition of *consistent* here is that as the number of points increases, the estimate can converge in probability to the true value. To the best of our knowledge, devising a consistent PnP solver is still an open problem.

In this paper, we devise a **c**onsistent **PnP** (CPnP) solver in virtue of bias elimination. Specifically, linear equations are constructed from the original projection model via measurement model modification and variable elimination, and a closed-form least-squares solution is obtained. The least-squares solution is biased, hence we further draw lessons from [9] to give a consistent estimate of the variance of projection noises, based on which the asymptotic bias is eliminated, yielding a consistent estimate of the camera pose. Additionally, we perform constrained Gauss-Newton (GN) iterations to refine the consistent estimate. Our proposed PnP solver owns the attractive property that the estimate can converge to the true parameters as the number of points increases. In addition, the solver is efficient—its time complexity is $O(n)$, which is superior to most of the state-of-the-art algorithms. These two properties make our estimator favorable in the presence of numerous feature points. The CPnP solver can be integrated into visual localization systems for robot self-localization or object pose estimation.

We compare the proposed estimator with some well-known PnP solvers using both synthetic data and a benchmark dataset, named ETH3D [6]. The results show that when the number of feature points exceeds several hundred, our estimator has minimal RMSEs and almost the same CPU time as the EPnP algorithm. In addition, in the case of large noise intensity, the covariance of the proposed estimator can converge to 0, while the compared estimators have asymptotic biases, failing to converge to the true value.

The main contributions of this paper are two-fold:

(i). We revisit the PnP problem from the perspective of statistics, finding that the existing PnP solvers are generally biased, thus not consistent.

(ii). On the basis of a series of techniques, including variable elimination, noise estimation, and bias elimination, a consistent $O(n)$ estimator is proposed.

[1]School of Data Science, Chinese University of Hong Kong, Shenzhen, Shenzhen, P. R. China, zengguangyang@cuhk.edu.cn, shiyuchen@link.cuhk.edu.cn, junfengwu@cuhk.edu.cn.

[2]Key Laboratory of Systems and Control, Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, P. R. China, bqmu@amss.ac.cn.

[3]The Australian Center for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, NSW 2008, Sydney, guodong.shi@sydney.edu.au.

[4]Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen, P. R. China.

The implementation code can be found at https://github.com/SLAMLab-CUHKSZ/CPnP-A-Consistent-PnP-Solver.

II. Related Works

Some works considered a fixed number of points. The minimum number is three, and the resulting problem is referred to as P3P [10]–[12]. Apart from P3P solvers, there are also P4P [13] and P5P solutions [14]. Although these solvers have analytic solutions, they can be quite inaccurate in noisy circumstances. Most of the PnP solvers are able to handle arbitrary number of points, and they can be partitioned into two categories: non-iterative and iterative.

Early non-iterative solvers are generally computationally complex, e.g., [15] with $O(n^8)$, [16] with $O(n^5)$, and [17] with $O(n^2)$. EPnP [18] is a well-known $O(n)$ solver for the PnP problem and is utilized in many robot applications [19]–[21]. It defines four control points and represents other points as a linear combination of them. The coordinates of the control points are then estimated by linearization techniques. Apart from the linearization techniques, the polynomial solvers have become a mainstream, e.g., the RPnP [22], DLS [23], OPnP [24], EOPnP [25], and SRPnP [26]. These works all estimate the camera pose via solving polynomial equations with $O(n)$ complexity.

Iterative methods solve PnP optimization problems in an iterative manner. One main issue of iterative methods is the set of the initial estimate—they can achieve excellent precision when they converge properly. LHM [27] and FP [28] initialized the estimate with a weak perspective approximation and refined the estimate via local iterations. SQPnP [29] set several initials and utilized a sequential quadratic programming method to compute each regional minimum. The aforementioned non-iterative solutions are often used as initial values, and iterative algorithms, e.g., Gauss-Newton (GN) iterations are applied to refine the estimate. For instance, EPnP [18] and MLPnP [8] both evaluated the performance by adding the GN iterations.

It is noteworthy that most of the existing works have not modelled the projection noises, based on which the solver can be optimized accordingly. The literature that took projection noise into account includes CEPPnP [7], MLPnP [8], PLUM [30], and EPnPU [4]. In these works, the covariance matrix of noises is utilized to improve the estimation precision. However, there is little research on the statistical properties of their proposed estimators. Actually, due to the inherent nonlinear nature of the camera projection model, all of these estimators are biased, and thus not consistent. In this paper, we take the noise model into account and devise an estimator with bias elimination. The proposed estimator is consistent, i.e., as the number of points increases, the estimate converges to the true value.

III. Problem Formulation

Notations: For a noisy quantity $\mathbf{a}$, we use $\mathbf{a}^o$ to denote its true value. For a vector $\mathbf{a}$, $[\mathbf{a}]_i$ is the i-th element of $\mathbf{a}$. $\mathbf{1}_{i\times j}$ and $\mathbf{0}_{i\times j}$ are $i \times j$ matrices whose elements are all 1 and 0.

As shown in Fig. 1, suppose there are n points whose coordinates in the world frame $\{W\}$ are $\mathbf{p}_i^w = [x_i^w \ y_i^w \ z_i^w]^\top, i = 1, \dots, n$. Denote their coordinates in the camera frame $\{C\}$ as $\mathbf{p}_i^c = [x_i^c \ y_i^c \ z_i^c]^\top, i = 1, \dots, n$. Then, given the rotation matrix $\mathbf{R}$ and transformation vector $\mathbf{t}$, it holds that $\mathbf{p}_i^c = \mathbf{R}\mathbf{p}_i^w + \mathbf{t}$. Further, for a calibrated pinhole camera with the intrinsic matrix

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

the ideal projection equation is

$$s_i \begin{bmatrix} \mathbf{q}_i \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R} \ \mathbf{t}] \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{q}_i$ is the 2D projection in the image plane of $\mathbf{p}_i^w$, and s_i is the scale factor. From the third line of (1), we have $s_i = \mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w + \mathbf{t})$, where $\mathbf{e}_i$ is the unit vector whose i-th element is 1. Considering projection noises, (1) can be rewritten as

$$\mathbf{q}_i = \frac{\mathbf{W}\mathbf{E}(\mathbf{R}\mathbf{p}_i^w + \mathbf{t})}{\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w + \mathbf{t})} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \boldsymbol{\epsilon}_i, \tag{2}$$

where $\mathbf{W} = \mathrm{diag}(f_x, f_y)$, $\mathbf{E} = [\mathbf{e}_1 \ \mathbf{e}_2]^\top$, and $\boldsymbol{\epsilon}_i$ is the projection noise. For $\boldsymbol{\epsilon}_i$, we assume that

Assumption 1. *The measurement noises* $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \sigma^2 I), i = 1, \dots, n$ *are i.i.d. with unknown variance* $\sigma^2 < \infty$.

Assumption 1 has been widely used in simulations for the PnP problem, e.g., [8], [18], [23]. Since the intrinsic matrix is known, for simplicity, we can obtain the shifted 2D projection $\mathbf{q}_i'$:

$$\mathbf{q}_i' = \mathbf{q}_i - \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \frac{\mathbf{W}\mathbf{E}(\mathbf{R}\mathbf{p}_i^w + \mathbf{t})}{\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w + \mathbf{t})} + \boldsymbol{\epsilon}_i. \tag{3}$$

The PnP problem involves estimating $\mathbf{R}$ and $\mathbf{t}$ from the n correspondences between $\mathbf{p}_i^w$ and $\mathbf{q}_i'$. A prevalent criterion to do so is to minimize the sum of squared reprojection errors:

$$\underset{\mathbf{R},\mathbf{t}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| \frac{\mathbf{W}\mathbf{E}(\mathbf{R}\mathbf{p}_i^w + \mathbf{t})}{\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w + \mathbf{t})} - \mathbf{q}_i' \right\|^2 \tag{4a}$$

$$\text{subject to} \quad \mathbf{R} \in \mathrm{SO}(3), \tag{4b}$$

The constrained least-squares problem (4) is a nonconvex optimization problem whose global solution is hard to obtain. In the following section, we relax the constraint (4b) and design a consistent estimator for $\mathbf{R}$ and $\mathbf{t}$.

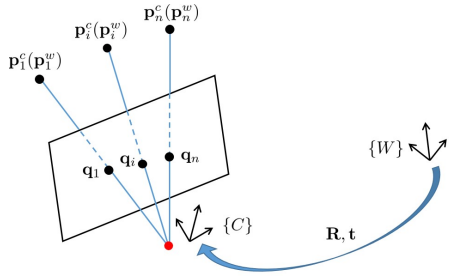


Fig. 1: Illustration of the PnP problem.

IV. Relaxed Least-Squares Solution: A Consistent Pose Estimate

In this section, we will first modify the original measurement equations (3) to obtain linear ones. We then conduct variable elimination to avoid scale ambiguity. Additionally, we give a consistent estimate of the covariance of projection noises, based on which a bias-eliminated least-squares solution is proposed. Further, the consistent estimate is refined with GN iterations.

A. Modified Measurement equations

By multiplying both sides of (3) by $\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w+\mathbf{t})$ we obtain a modified linear measurement model:

$$\mathbf{W}\mathbf{E}(\mathbf{R}\mathbf{p}_i^w+\mathbf{t})-\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w+\mathbf{t})\mathbf{q}_i'+\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w+\mathbf{t})\boldsymbol{\epsilon}_i=0, \quad (5)$$

where $\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w+\mathbf{t})\boldsymbol{\epsilon}_i$ is the scaled noise term. Note that (5) is a linear equation with respect to $\mathbf{R}$ and $\mathbf{t}$. Let $\bar{\mathbf{L}}_i = \left[\mathbf{p}_i^{w\top}\ 1\right]\otimes\mathbf{I}_3 \in \mathbb{R}^{3\times 12}$. By vectorizing $[\mathbf{R}\ \mathbf{t}]$, i.e., $\mathbf{x} = \mathrm{vec}\left([\mathbf{R}\ \mathbf{t}]\right)$, we can concatenating (5) for all n points to obtain the following matrix form:

$$\mathbf{0}=\mathbf{M}\mathbf{x}+\boldsymbol{\epsilon}', \quad (6)$$

where $\mathbf{M}=[\mathbf{M}_1^\top\ \cdots\ \mathbf{M}_n^\top]^\top$, $\mathbf{M}_i=(\mathbf{W}\mathbf{E}-\mathbf{q}_i'\mathbf{e}_3^\top)\bar{\mathbf{L}}_i$ and $\boldsymbol{\epsilon}'=\left[\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_1^w+\mathbf{t})\boldsymbol{\epsilon}_1^\top\ \cdots\ \mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_n^w+\mathbf{t})\boldsymbol{\epsilon}_n^\top\right]^\top$.

For the modified measurement equation (6), on the one hand, the regressand is $\mathbf{0}$; on the other hand, due to the scale ambiguity, the regressor $\mathbf{M}$ is not full column rank. Therefore, the estimate of $\mathbf{x}$ cannot be calculated in a closed form. In the following, we will conduct variable elimination to avoid scale ambiguity and make the regressand a nonzero vector. After this procedure, the number of variables to be estimated is reduced from 12 to 11.

B. Variable Elimination

To eliminate the scale ambiguity, we introduce the following constraint:

$$\alpha\sum_{i=1}^{n}\mathbf{e}_3^\top(\mathbf{R}\mathbf{p}_i^w+\mathbf{t})=n, \quad (7)$$

where α is the scale factor. Let $\mathbf{R}=[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3]^\top$, and $\mathbf{t}=[t_1\ t_2\ t_3]^\top$. Define $\bar{\mathbf{p}}^w:=\sum_{i=1}^n\mathbf{p}_i^w/n$. From (7) we have

$$\alpha t_3=1-\alpha\bar{\mathbf{p}}^{w\top}\mathbf{r}_3. \quad (8)$$

Substituting t_3 into (5) yields

$$\mathbf{q}_i'=\alpha\mathbf{W}\mathbf{E}(\mathbf{R}\mathbf{p}_i^w+\mathbf{t})-\alpha\left(\mathbf{p}_i^w-\bar{\mathbf{p}}^w\right)^\top\mathbf{r}_3\mathbf{q}_i'+\boldsymbol{\varepsilon}_i, \quad (9)$$

where $\boldsymbol{\varepsilon}_i=(1+\alpha\left(\mathbf{p}_i^w-\bar{\mathbf{p}}^w\right)^\top\mathbf{r}_3)\boldsymbol{\epsilon}_i$. By the above variable elimination, the number of unknown variables is reduced from 12 to 11. We stack the 11 variables as follows:

$$\boldsymbol{\theta}:=[\hat{\mathbf{r}}_3^\top\ \hat{\mathbf{r}}_1^\top\ \hat{t}_1\ \hat{\mathbf{r}}_2^\top\ \hat{t}_2]^\top=\alpha[\mathbf{r}_3^\top\ \mathbf{r}_1^\top\ t_1\ \mathbf{r}_2^\top\ t_2]^\top. \quad (10)$$

Given the vector $\boldsymbol{\theta}$, the rotation matrix $\mathbf{R}$ and transformation vector $\mathbf{t}$ along with the scale factor α can be uniquely recovered; see (14)-(16). This is due to $\det(\mathbf{R})=\mathbf{1}$ since $\mathbf{R}\in\mathrm{SO}(3)$. The same variable elimination method is adopted in [24]. Compared with the prevalent strategy in DLT-based methods, i.e., setting the constraint $\|\mathbf{x}\|=1$, which leads to a nonlinear relationship among variables, the constraint in (7) owns the advantage that the resulting equation (8) has a linear form. This facilitates the construction of the linear system of equations (11) and the resulting ordinary least-squares solution (12). Let $\mathbf{q}_i'=[u_i\ v_i]^\top$, by concatenating (9) for all reference points, we obtain the following matrix form:

$$\mathbf{b}=\mathbf{A}\boldsymbol{\theta}+\boldsymbol{\varepsilon}, \quad (11)$$

where $\mathbf{b}=[\mathbf{q}_1'^\top\ \cdots\ \mathbf{q}_n'^\top]^\top$, $\boldsymbol{\varepsilon}=[\boldsymbol{\varepsilon}_1^\top\ \cdots\ \boldsymbol{\varepsilon}_n^\top]^\top$, and

$$\mathbf{A}=\begin{bmatrix}-u_1\left(\mathbf{p}_1^w-\bar{\mathbf{p}}^w\right)^\top & f_x\mathbf{p}_1^{w\top} & f_x & \mathbf{0}_{1\times 4}\\ -v_1\left(\mathbf{p}_1^w-\bar{\mathbf{p}}^w\right)^\top & \mathbf{0}_{1\times 4} & f_y\mathbf{p}_1^{w\top} & f_y\\ \vdots & \vdots & \vdots & \vdots\\ -u_n\left(\mathbf{p}_n^w-\bar{\mathbf{p}}^w\right)^\top & f_x\mathbf{p}_n^{w\top} & f_x & \mathbf{0}_{1\times 4}\\ -v_n\left(\mathbf{p}_n^w-\bar{\mathbf{p}}^w\right)^\top & \mathbf{0}_{1\times 4} & f_y\mathbf{p}_n^{w\top} & f_y\end{bmatrix}.$$

Compared with (6), (11) has the advantages that it is a nonhomogeneous system, and the matrix $\mathbf{A}$ has full column rank if $(\mathbf{p}_i^w)_{i=1}^n$ do not fall into any critical set given in Result 22.5 in [31]. A closed-form solution is given by

$$\hat{\boldsymbol{\theta}}_n^{\mathrm{B}}=\left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{b}. \quad (12)$$

It is noteworthy that the projections $\mathbf{q}_i'=[u_i\ v_i]^\top$ contain noises and thus the regressor $\mathbf{A}$ and the regressand $\mathbf{b}$ are correlated, which leads to biasedness of $\hat{\boldsymbol{\theta}}_n^{\mathrm{B}}$. In the following subsection, we give a consistent estimate of the variance of projection noises, based on which the asymptotic bias of $\hat{\boldsymbol{\theta}}_n^{\mathrm{B}}$ can be eliminated.

C. Least-Squares Solution with Bias Elimination

First, we use the method in [9] to obtain a consistent estimate of σ^2, denoted as $\hat{\sigma}^2$. The details are presented in Appendix A in the full version of this paper [32]. Given the consistent $\hat{\sigma}^2$, we are on the point to eliminate the bias of the solution given in (12). Specifically, define

$$\mathbf{G}=\begin{bmatrix}-\left(\mathbf{p}_1^w-\bar{\mathbf{p}}^w\right)^\top & \mathbf{0}_{1\times 8}\\ -\left(\mathbf{p}_1^w-\bar{\mathbf{p}}^w\right)^\top & \mathbf{0}_{1\times 8}\\ \vdots & \vdots\\ -\left(\mathbf{p}_n^w-\bar{\mathbf{p}}^w\right)^\top & \mathbf{0}_{1\times 8}\\ -\left(\mathbf{p}_n^w-\bar{\mathbf{p}}^w\right)^\top & \mathbf{0}_{1\times 8}\end{bmatrix}\in\mathbb{R}^{2n\times 11}.$$

The bias-eliminated solution is given as

$$\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}=\left(\mathbf{A}^\top\mathbf{A}-\hat{\sigma}^2\mathbf{G}^\top\mathbf{G}\right)^{-1}\left(\mathbf{A}^\top\mathbf{b}-\hat{\sigma}^2\mathbf{G}^\top\mathbf{1}_{2n\times 1}\right). \quad (13)$$

To ensure the consistency of $\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}$, we make the following assumption.

Assumption 2. *The reference points $(\mathbf{p}_i^w)_{i=1}^n$ do not concentrate on any critical set given in Result 22.5 in [31].*

Assumption 2 is an assumption on the asymptotic distribution of 3D feature points. It can be satisfied in general, guaranteeing that the camera matrix $\mathbf{K}[\mathbf{R}\ \mathbf{t}]$ in (1) is unique [31]. The following theorem presents the asymptotic property of

the bias-eliminated solution $\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}$. The proof of is available in Appendix B in the full version of this paper [32].

Theorem 1. *Given Assumptions 1-2, the estimate $\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}$ is consistent, i.e.,*

$$\operatorname*{plim}_{n\to\infty} \hat{\boldsymbol{\theta}}_n^{\mathrm{BE}} = \boldsymbol{\theta}^o.$$

Given $\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}$, according to (10), the estimate of $\mathbf{R}$ and $\mathbf{t}$ can be calculated as follows:

$$\hat{\alpha} = \left(\det\left(\left[\left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{4:6} \quad \left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{8:10} \quad \left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{1:3}\right]^\top\right)\right)^{1/3}, \tag{14}$$

$$\hat{\mathbf{R}}_n^{\mathrm{BE}} = \left[\left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{4:6} \quad \left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{8:10} \quad \left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{1:3}\right]^\top \Big/ \hat{\alpha}, \tag{15}$$

$$\hat{\mathbf{t}}_n^{\mathrm{BE}} = \left[\left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{7} \quad \left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{11} \quad 1 - \bar{\mathbf{p}}^{w\top}\left[\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}\right]_{1:3}\right]^\top \Big/ \hat{\alpha}. \tag{16}$$

Since $\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}$ is consistent, so are $\hat{\mathbf{R}}_n^{\mathrm{BE}}$ and $\hat{\mathbf{t}}_n^{\mathrm{BE}}$.

Note that the matrix $\hat{\mathbf{R}}_n^{\mathrm{BE}}$ does not necessarily fall within $\mathrm{SO}(3)$. Hence, we should further project it into $\mathrm{SO}(3)$. Let the SVD of $\hat{\mathbf{R}}_n^{\mathrm{BE}}$ be $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, then the following projection gives the closest element in $\mathrm{SO}(3)$ to $\hat{\mathbf{R}}_n^{\mathrm{BE}}$ in terms of the Frobenius norm [33]:

$$\pi(\hat{\mathbf{R}}_n^{\mathrm{BE}}) = \mathbf{U}\mathrm{diag}([1 \ 1 \ \det(\mathbf{U}\mathbf{V}^{\mathrm{T}})]^{\mathrm{T}})\mathbf{V}^{\mathrm{T}}. \tag{17}$$

With a little abuse of notations, we still denote the projected matrix as $\hat{\mathbf{R}}_n^{\mathrm{BE}}$.

D. Constrained Gauss-Newton Iteration

With the consistent preliminary estimates $\hat{\mathbf{R}}_n^{\mathrm{BE}}$ and $\hat{\mathbf{t}}_n^{\mathrm{BE}}$, local methods such as the GN iteration associated with measurement equation (3) can be applied to further improve the estimation precision. Let $\mathbf{L}_i = \mathbf{p}_i^{w\top} \otimes \mathbf{I}_3 \in \mathbb{R}^{3\times 9}$. The measurement equation (3) can be rephrased as

$$\mathbf{q}_i' = \frac{\mathbf{W}\mathbf{E}\left(\mathbf{L}_i\mathrm{vec}(\mathbf{R}) + \mathbf{t}\right)}{\mathbf{e}_3^\top\left(\mathbf{L}_i\mathrm{vec}(\mathbf{R}) + \mathbf{t}\right)} + \boldsymbol{\epsilon}_i. \tag{18}$$

Note that the rotation matrix $\mathbf{R}$ needs to be in $\mathrm{SO}(3)$. Hence, we cannot directly use the Jacobian obtained by taking derivatives with respect to $\mathbf{R}$ in the GN iterations. Note that we can represent the rotation matrix $\mathbf{R}$ in the vicinity of a given matrix $\mathbf{R}_0$ as $\mathbf{R} = \mathbf{R}_0\exp(\mathbf{s}^\wedge)$, where $\mathbf{R}_0 \in \mathrm{SO}(3)$ and $\mathbf{s}^\wedge$ is a skew-symmetric matrix generated by the "hat" operation:

$$\mathbf{s}^\wedge = \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix}$$

with $\mathbf{s} = [s_1 \ s_2 \ s_3]^\top$. We calculate the Jacobian associated with $\mathbf{s}$ to guarantee the updated estimate of the rotation matrix is still in $\mathrm{SO}(3)$. Define

$$\boldsymbol{\Psi} := \frac{\partial \mathrm{vec}(\exp(\mathbf{s}^\wedge))}{\partial \mathbf{s}^\top}\Big|_{\mathbf{s}=\mathbf{0}},$$

and

$$\begin{aligned} f_i(\mathbf{R},\mathbf{s},\mathbf{t}) &:= \frac{\mathbf{W}\mathbf{E}\left(\mathbf{L}_i\mathrm{vec}(\mathbf{R}\exp(\mathbf{s}^\wedge)) + \mathbf{t}\right)}{\mathbf{e}_3^\top\left(\mathbf{L}_i\mathrm{vec}(\mathbf{R}\exp(\mathbf{s}^\wedge)) + \mathbf{t}\right)}, \\ g_i(\mathbf{R},\mathbf{s},\mathbf{t}) &:= \mathbf{W}\mathbf{E}\left(\mathbf{L}_i\mathrm{vec}(\mathbf{R}\exp(\mathbf{s}^\wedge)) + \mathbf{t}\right), \\ h_i(\mathbf{R},\mathbf{s},\mathbf{t}) &:= \mathbf{e}_3^\top\left(\mathbf{L}_i\mathrm{vec}(\mathbf{R}\exp(\mathbf{s}^\wedge)) + \mathbf{t}\right). \end{aligned}$$

The derivatives are

$$\begin{aligned} \frac{\partial f_i(\mathbf{R},\mathbf{s},\mathbf{t})}{\partial \mathbf{s}^\top}\Big|_{\mathbf{s}=0} &= \frac{(h_i(\mathbf{R},0,\mathbf{t})\mathbf{W}\mathbf{E} - g_i(\mathbf{R},0,\mathbf{t})\mathbf{e}_3^\top)\mathbf{p}_i^{w\top}\otimes\mathbf{R}\boldsymbol{\Psi}}{h_i(\mathbf{R},0,\mathbf{t})^2} \\ \frac{\partial f_i(\mathbf{R},\mathbf{s},\mathbf{t})}{\partial \mathbf{t}^\top}\Big|_{\mathbf{s}=0} &= \frac{h_i(\mathbf{R},0,\mathbf{t})\mathbf{W}\mathbf{E} - g_i(\mathbf{R},0,\mathbf{t})\mathbf{e}_3^\top}{h_i(\mathbf{R},0,\mathbf{t})^2}. \end{aligned}$$

Then we can obtain the Jacobian matrix $\mathbf{J}(\mathbf{R},\mathbf{t})$ as follows:

$$\mathbf{J}(\mathbf{R},\mathbf{t}) = \begin{bmatrix} \vdots & \vdots \\ \frac{\partial f_i(\mathbf{R},\mathbf{s},\mathbf{t})}{\partial \mathbf{s}^\top}\Big|_{\mathbf{s}=0} & \frac{\partial f_i(\mathbf{R},\mathbf{s},\mathbf{t})}{\partial \mathbf{t}^\top}\Big|_{\mathbf{s}=0} \\ \vdots & \vdots \end{bmatrix} \in \mathbb{R}^{2n\times 6}.$$

Denote the GN refinement of $\hat{\mathbf{s}}_n$ and $\hat{\mathbf{t}}_n$ by $\hat{\mathbf{s}}_n^{\mathrm{GN}}$ and $\hat{\mathbf{t}}_n^{\mathrm{GN}}$, respectively. Given the initial consistent estimate $\hat{\mathbf{R}}_n^{\mathrm{BE}}$ and $\hat{\mathbf{t}}_n^{\mathrm{BE}}$, we have

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{s}}_n^{\mathrm{GN}} \\ \hat{\mathbf{t}}_n^{\mathrm{GN}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{t}}_n^{\mathrm{BE}} \end{bmatrix} + \left(\mathbf{J}^\top(\hat{\mathbf{R}}_n^{\mathrm{BE}},\hat{\mathbf{t}}_n^{\mathrm{BE}})\mathbf{J}(\hat{\mathbf{R}}_n^{\mathrm{BE}},\hat{\mathbf{t}}_n^{\mathrm{BE}})\right)^{-1} \\ \mathbf{J}^\top(\hat{\mathbf{R}}_n^{\mathrm{BE}},\hat{\mathbf{t}}_n^{\mathrm{BE}})\left(\mathbf{b} - f(\hat{\mathbf{R}}_n^{\mathrm{BE}},\hat{\mathbf{t}}_n^{\mathrm{BE}})\right), \end{aligned} \tag{19}$$

where $f(\hat{\mathbf{R}},\hat{\mathbf{t}}) = [f_1(\hat{\mathbf{R}},0,\hat{\mathbf{t}})^\top \ \cdots \ f_n(\hat{\mathbf{R}},0,\hat{\mathbf{t}})^\top]^\top$. As such,

$$\hat{\mathbf{R}}_n^{\mathrm{GN}} = \hat{\mathbf{R}}_n^{\mathrm{BE}}\exp\left(\hat{\mathbf{s}}_n^{\mathrm{GN}\wedge}\right). \tag{20}$$

We remark here that our devised PnP solver has overall $O(n)$ time complexity. Specifically, it can be verified that the $O(n)$ calculations in our algorithm include the computation of the centroid of 3D reference points $\bar{\mathbf{p}}^w$, the estimation of noise variance, the calculation of the consistent estimate $\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}$, and the constrained GN iterations. The other operations consume constant time. Therefore, our PnP solver is efficient and favorable in large sample applications. The whole algorithm is summarized in Algorithm 1.

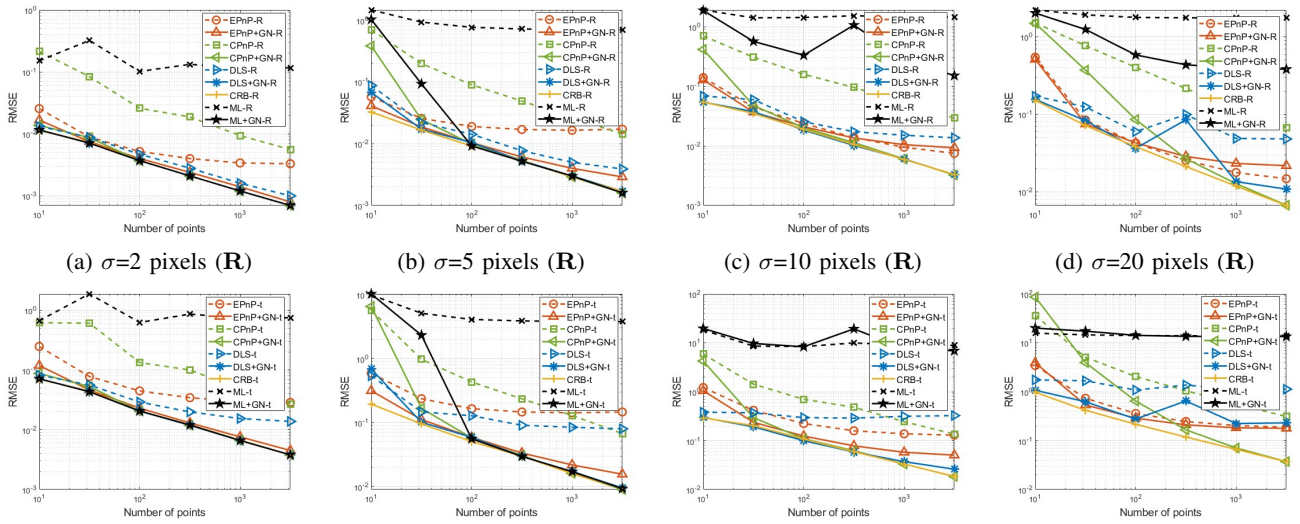
Algorithm 1 CPnP: Consistent PnP Pose Estimator

Input: 3D points $(\mathbf{p}_i^w)_{i=1}^n$ and 2D projections $(\mathbf{q}_i)_{i=1}^n$.
Output: the estimates of $\mathbf{R}$ and $\mathbf{t}$.

1: Calculate $(\mathbf{q}_i')_{i=1}^n$ according to (3);
2: Estimate the variance of projection noises;
3: Calculate the bias-eliminated solution $\hat{\boldsymbol{\theta}}_n^{\mathrm{BE}}$ in (13);
4: Recover $\hat{\mathbf{R}}_n^{\mathrm{BE}}$ and $\hat{\mathbf{t}}_n^{\mathrm{BE}}$ according to (14)-(16);
5: Project the rotation matrix into $\mathrm{SO}(3)$ using (17);
6: Refine the estimate using the GN iterations (19) and (20).

V. Experiments

In this section, we compare our algorithm, referred to as CPnP, with some well-known PnP solvers, including EPnP [18], MLPnP [8], and DLS [23]. All the algorithms are implemented in Matlab via a laptop with Apple M1 Pro, where we directly use the open source codes for the three comparison solvers. Since the DLS codes do not include GN iterations, we use the proposed GN iterations (19) and (20) for the DLS solver. The results will be presented in terms of estimation accuracy and computing time.



(a) σ=2 pixels ($\mathbf{R}$) (b) σ=5 pixels ($\mathbf{R}$) (c) σ=10 pixels ($\mathbf{R}$) (d) σ=20 pixels ($\mathbf{R}$)

(e) σ=2 pixels ($\mathbf{t}$) (f) σ=5 pixels ($\mathbf{t}$) (g) σ=10 pixels ($\mathbf{t}$) (h) σ=20 pixels ($\mathbf{t}$)

Fig. 2: RMSE comparison among different PnP solvers with synthetic data.

A. Experiments with Synthetic Data

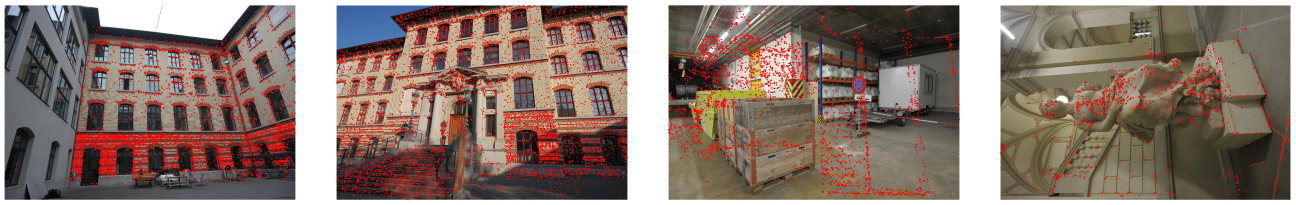
In simulations, the Euler angles of the camera are set as $[\pi/3\ \pi/3\ \pi/3]^\top$, and the translation vector is $[2\ 6\ 6]^\top$. For the intrinsic parameters, the focal length is set as $f_x = f_y = 50$mm (800 pixels), and the size of the image plane is 640×480 pixels. The principle point lies in the top-left corner of the image plane and the principle point offset is $u_0 = 320$ pixels, $v_0 = 240$ pixels. For the 3D points under the camera frame, we randomly generate them from the region $[-2, 2] \times [-2, 2] \times [4, 16]$ m. After filtering out the points outside the range of the image plane, the remaining 3D points are projected onto the image plane by the projection equation (2). Specifically, the projection noise $\boldsymbol{\epsilon}_i$ is Gaussian noise whose standard deviation is σ pixels. For each fixed σ and number of points n, we execute $T = 1000$ Monte Carlo tests to evaluate the root-mean-square errors (RMSE) of each estimator.

We set $\sigma = 2, 5, 10, 20$ pixels, respectively, and for each choice of σ, the number of points varies from 10 to 3000. The RMSE comparison among different PnP solvers is presented in Fig. 2. "CRB" in the figure offers the theoretical lower bound, named Cramer-Rao bound, for the RMSE of any unbiased estimator. For the derivation of the constrained CRB, one can refer to [34]. We can see from the figure that when the intensity of noises is small, all estimators with GN iterations can achieve the CRB as the number of points increases. This is because in that case, the biases of the other estimators are negligible, and the GN iterations can make the estimate converge to the global minimizer of the original ML problem. Nevertheless, with the increase of noise intensity, all the other PnP solvers have relatively large biases. As a result, the initial estimate may not locate within the attraction region of the global minimum, and the GN iterations may only obtain local minima. However, since our proposed estimator has undergone bias elimination, it is consistent—as the number of points goes large, the estimate can converge to the true camera pose. With the initial consistent estimate, the GN iterations can further obtain the global minimum of the ML problem (4), achieving the theoretical lower bound. It can be seen that when the number of points exceeds several hundred, our estimator has minimal RMSEs. When the point number is small, the proposed estimator may be inferior to the compared ones, since we have relaxed the SO(3) constraint in the first step.

B. Experiments with Real Images

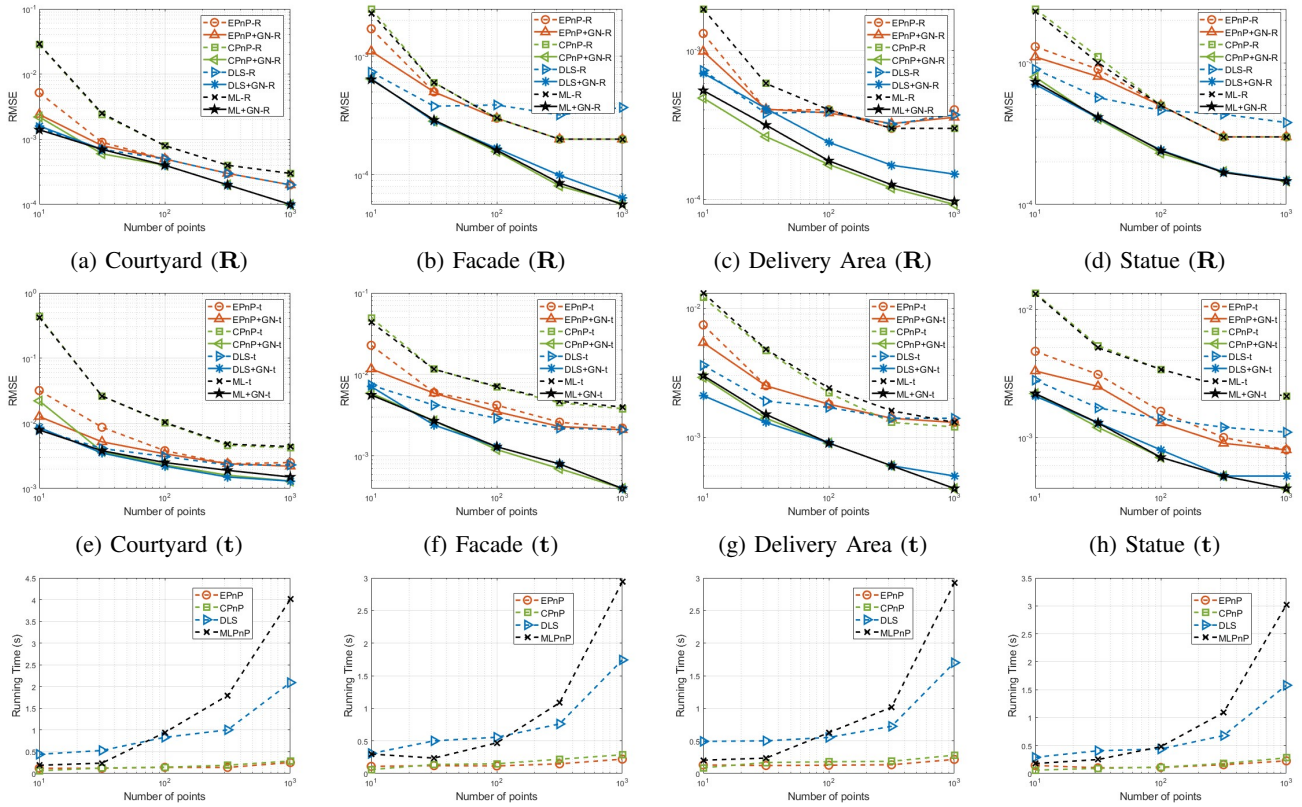
We use the ETH3D dataset [6] to test the performances of our developed algorithm. This dataset provides original images, the intrinsic parameters of the cameras, the coordinates of 3D points and the corresponding 2D projections, and the true pose of cameras associated with each image. As shown in Fig. 3, the evaluation is done in four images—two are outdoor scenarios, and two are indoor scenarios. The red points are the 2D points given in the dataset. For each image, we remove the 2D points that have no corresponding 3D points. We then randomly select a certain number of point correspondences from the cleaned dataset to estimate the camera pose. For each number of points, totally $T = 50$ random selections are conducted to calculate the RMSE.

We vary the number of points from 10 to 1000. The RMSE and CPU time comparisons are shown in Fig. 4. Our estimator has the smallest RMSE when the number of points is large. The trend of the lines of CPnP shows the consistent property of our proposed algorithm. It is worth noting that since the points may not be sufficient enough to support $T = 50$ Monte Carlo repetitions, the slope has slowed down a little when the selected number of points is 1000. Regarding the comparison of time complexity, we record the CPU time of each algorithm in 50 Monte Carlo runs. The CPU time of different estimators is comparable when the number of points is below 100. They deviate when the number further



(a) Outdoor: Courtyard (b) Outdoor: Facade (c) Indoor: Delivery Area (d) Indoor: Statue

Fig. 3: Several images in ETH3D Benchmark [6], among which two are outdoor scenarios and two are indoor scenarios.



(a) Courtyard (**R**) (b) Facade (**R**) (c) Delivery Area (**R**) (d) Statue (**R**)

(e) Courtyard (**t**) (f) Facade (**t**) (g) Delivery Area (**t**) (h) Statue (**t**)

(i) Courtyard (CPU time) (j) Facade (CPU time) (k) Delivery Area (CPU time) (l) Statue (CPU time)

Fig. 4: RMSE and CPU time comparisons among different PnP solvers with real images.

increased, with MLPnP increasing most sharply and DLS the second. Our proposed CPnP remains remarkably stable and maintains low CPU time as the EPnP does. In the case of 1000 points, the proposed CPnP consumes less than one second in total 50 estimations, showing the applicability in real-time applications.

VI. Conclusion

In this paper, we revisited the PnP problem from the view of statistics. On the basis of the consistent estimate of noise variance, a consistent estimator for the camera pose was proposed via a bias-eliminated closed-form solution. Constrained GN iterations were executed to refine the initial estimate. Our proposed CPnP estimator has two advantages: it is consistent; its time complexity is $O(n)$. Experiments using both synthetic data and benchmark datasets showed that the proposed CPnP is superior to the other estimators for images with dense visual features, in terms of RMSE and CPU time. In future work, we hope to theoretically prove the asymptotic efficiency of our two-step estimator. In addition, we will introduce robust schemes, e.g., the RANSAC algorithms, to remove some outliers, which may further improve the performance of our algorithm.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under grant no. 62273288, and in part by Shenzhen Science and Technology Program JCYJ20220818103000001.

REFERENCES

[1] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[2] L. Meng, J. Chen, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva, "Backtracking regression forests for accurate camera relocalization," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6886–6893.

[3] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4503–4508.

[4] A. Vakhitov, L. Ferraz, A. Agudo, and F. Moreno-Noguer, "Uncertainty-aware camera pose estimation from points and lines," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4659–4668.

[5] H. Zhou, T. Zhang, and J. Jagadeesan, "Re-weighting and 1-point ransac-based p n n p solution to handle outliers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 3022–3033, 2018.

[6] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3260–3269.

[7] L. Ferraz Colomina, X. Binefa, and F. Moreno-Noguer, "Leveraging feature uncertainty in the pnp problem," in *Proceedings of British Machine Vision Conference (BMVC)*, 2014, pp. 1–13.

[8] S. Urban, J. Leitloff, and S. Hinz, "Mlpnp-a real-time maximum likelihood solution to the perspective-n-point problem," *arXiv:1607.08112*, 2016.

[9] B. Mu, E.-W. Bai, W. X. Zheng, and Q. Zhu, "A globally consistent nonlinear least squares estimator for identification of nonlinear rational systems," *Automatica*, vol. 77, pp. 322–335, 2017.

[10] D. DeMenthon and L. S. Davis, "Exact and approximate solutions of the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1100–1105, 1992.

[11] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2969–2976.

[12] S. Li and C. Xu, "A stable direct solution of perspective-three-point problem," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 5, pp. 627–642, 2011.

[13] M. Bujnak, Z. Kukelova, and T. Pajdla, "A general solution to the p4p problem for camera with unknown focal length," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[14] B. Triggs, "Camera pose and calibration from 4 or 5 known 3d points," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 1999, pp. 278–284.

[15] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578–589, 2003.

[16] L. Quan and Z. Lan, "Linear n-point camera pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 774–780, 1999.

[17] P. D. Fiore, "Efficient linear solution of exterior orientation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 140–148, 2001.

[18] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[19] E. Boukas, A. Gasteratos, and G. Visentin, "Towards orbital based global rover localization," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2874–2881.

[20] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9426–9432.

[21] J. Lambrecht, P. Grosenick, and M. Meusel, "Optimizing keypoint-based single-shot camera-to-robot pose estimation through shape segmentation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 843–13 849.

[22] S. Li, C. Xu, and M. Xie, "A robust o (n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.

[23] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) method for pnp," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 383–390.

[24] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the pnp problem: A fast, general and optimal solution," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2344–2351.

[25] L. Zhou and M. Kaess, "An efficient and accurate algorithm for the perspecitve-n-point problem," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6245–6252.

[26] P. Wang, G. Xu, Y. Cheng, and Q. Yu, "A simple, robust and fast method for the perspective-n-point problem," *Pattern Recognition Letters*, vol. 108, pp. 31–37, 2018.

[27] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000.

[28] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2011–2018.

[29] G. Terzakis and M. Lourakis, "A consistently fast and globally optimal solution to the perspective-n-point problem," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020, pp. 478–494.

[30] H. Li, J. Yao, X. Lu, and J. Wu, "Combining points and lines for camera pose estimation and optimization in monocular visual odometry," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1289–1296.

[31] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[32] G. Zeng, S. Chen, B. Mu, G. Shi, and J. Wu, "Cpnp: Consistent pose estimator for perspective-n-point problem with bias elimination," *arXiv:2209.05824*, 2022.

[33] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 698–700, 1987.

[34] P. Stoica and B. C. Ng, "On the cramér-rao bound under parametric constraints," *IEEE Signal Processing Letters*, vol. 5, no. 7, pp. 177–179, 1998.