

# A Direct Least-Squares Solution to the PnP Problem with Unknown Focal Length

Yinqiang Zheng

National Institute of Informatics  
 Tokyo, JAPAN  
 yqzheng@nii.ac.jp

Laurent Kneip

Australian National University  
 Canberra, AUSTRALIA  
 laurent.kneip@anu.edu.au

## Abstract

*In this work, we propose a direct least-squares solution to the perspective-n-point (PnP) pose estimation problem of a partially uncalibrated camera, whose intrinsic parameters except the focal length are known. The basic idea is to construct a proper objective function with respect to the target variables and extract all its stationary points so as to find the global minimum. The advantages of our proposed solution over existing ones are that (i) the objective function is directly built upon the imaging equation, such that all the 3D-to-2D correspondences contribute equally to the minimized error, and that (ii) the proposed solution is noniterative, in the sense that the stationary points are retrieved by means of eigenvalue factorization and the common iterative refinement step is not needed. In addition, the proposed solution has  $O(n)$  complexity, and can be used to handle both planar and nonplanar 3D points. Experimental results show that the proposed solution is much more accurate than the existing state-of-the-art solutions, and is even comparable to the maximum likelihood estimation by minimizing the reprojection error.*

## 1. Introduction

To simultaneously estimate camera pose and focal length by using  $n$  ( $n \geq 4$ ) 3D points in the world frame and their image projections is an important problem in geometric computer vision, with widespread applications in camera tracking [7], incremental structure-from-motion [22] and geolocalization using community image collections [9]. It can be regarded as a natural extension to the classical absolute pose estimation problem with fully known camera calibration parameters, also known as the PnP problem. This extension is of tremendous practical value, since the other intrinsic parameters of modern digital cameras can usually be assumed to be known a priori. For example, the skewness is usually zero and the aspect ratio is one. The principle

point lies approximately in the image center.

Due to its importance, many algorithms [1, 3, 4, 11, 21, 24, 25, 28] have been proposed to solve the PnP problem with unknown focal length. Unfortunately, the existing state-of-the-art solutions for the general  $n$ -point case, like [21, 28], are still not satisfactory in estimation accuracy. The primary reason lies in that their objective functions are built upon some intermediate entities, and some point correspondences are harmfully privileged over the others. Although one can resort to an additional iterative refinement step to improve the estimation accuracy further, the potential risk of running into local minima or wandering around some extreme points justifies some more endeavors in developing an accurate but noniterative solution.

In this paper, we propose a direct least-squares solution to the PnP problem with unknown focal length in the general  $n \geq 4$  case. Similar to [28], our basic idea is to construct a proper objective function with respect to the target variables and extract all its stationary points so as to find the global minimum. However, our solution is quite different from the existing solutions in that (i) the objective function is directly built upon the imaging equation, such that all the 3D-to-2D correspondences are treated with balance, without privileging some points over the remaining, and that (ii) the proposed solution is noniterative, in the sense that the stationary points are retrieved by means of standard eigenvalue factorization and the common iterative refinement step is not needed.

The proposed solution has linear time complexity, and thus can be used to handle large scale problems. Unlike the existing work [21], our solution does not explicitly differentiate planar from nonplanar 3D points, thus is insensitive to 3D point configurations. We have experimentally verified that the proposed solution is more accurate and efficient than the current state-of-the-art solutions, whose accuracy is even comparable to that of the maximum likelihood estimation by minimizing the reprojection error, despite the fact that our cost function is algebraically meaningful only.

## 2. Related Works

In this section, we review the related works for the classical PnP problem and the extended one with unknown focal length. There are some other works addressing the pose estimation problem with even more unknown variables, such as aspect ratio [6], principle point [24] and radial distortion [2, 10, 17], which will not be discussed in detail.

### 2.1. Solutions for PnP

The PnP problem has been extensively studied in both the minimal [13, 19] and the overconstrained [8, 12, 18, 20, 27] case. The prominent solutions for the general  $n$ -point case have been extended to handle the PnP problem with unknown focal length, thus deserve to be carefully reviewed.

The EPnP solution [18] is the first widely known  $O(n)$  method for PnP. Its core idea is to choose four (three in the planar case) control points to represent all 3D points, and build a cost function with respect to (w.r.t.) the coordinates of those control points. A variable lifting strategy is used to find its approximate global minimum, which can be further refined via iterative optimization. Even ignoring the potential convergence issues, the final solution is not very accurate, since the control points are given more weight over the others. In addition to that, to explicitly differentiate planar 3D points from nonplanar ones would cause inaccuracy in the intermediate near-planar case. The RPnP solution [20] selects two 3D points as the control points, together with which a point triplet is constructed for each of the rest points. The objective function is w.r.t. the depth factors of the two control points, which are therefore privileged. Rather than working on some intermediate entities, the direct least-squares (DLS) solution [8] directly constructs an objective function on the basis of the projection equation, and retrieves all stationary points via the resultant technique. The OPnP solution [27] further conquers the degeneracy of rotation parametrization of DLS. Both DLS (when rotation degeneracy does not occur) and OPnP are much more accurate than EPnP and RPnP, since all the point correspondences are treated equally when constructing the cost function.

### 2.2. PnP with Unknown Focal Length

When the focal length is unknown, there are seven unknown variables, thus requiring at least three and a half image points from four 3D points. Most existing works for this minimal case, such as [1, 3, 24, 28], use four 3D-to-2D point correspondences, and ignore one constraint in the solving process. Recently, Wu [25] proposed an exactly minimal Gröbner basis (GB) solver by using three and a half points, and used the remaining half measurement for outlier removal. This solver is shown to be much faster than the existing 4-point solvers.

Given four or more points, the PnPf problem with unknown focal length is overconstrained in general. This fact triggered much research interest in developing an efficient and accurate solution for the general  $n$ -point case. Choi *et al.* [4] developed a branch-and-bound algorithm for this problem, whose computational speed is not satisfactory. To develop more efficient solutions with linear time complexity, researchers have tried to transplant the core techniques from existing PnP solutions. For example, Penate-Sanchez *et al.* [21] and Kanaeva *et al.* [11] borrowed the idea of virtual control points in EPnP [18], while Zheng *et al.* [28] followed the point triplet technique originated in RPnP [20]. Unfortunately, the unbalanced cost function in [18] and [20] leads to inferior noise resilience, and this behavior is emphasized in the more difficult case of unknown focal length. In addition to that, the exhaustive variable lifting technique in [21] is not effective enough, when the number of points is small (*e.g.*  $4 \leq n \leq 5$ ). The error accumulation issue in [20] becomes more severe, and one usually has to refine the results via iterative optimization.

As mentioned above, a direct least-squares solution can significantly improve the estimation accuracy of PnP. This motivates us to develop a DLS solution to the PnP problem with unknown focal length. The challenge lies in that, when the focal length is unknown, it is very difficult to solve the resulting polynomial system so as to find the global minimum. To resolve this issue, we adopt the rotation decomposition recently proposed in [25], and properly eliminate the variables to build a bivariate objective function, whose stationary points can be easily retrieved via standard eigenvalue factorization. Note that, although we are inspired by the rotation parametrization in [25], our methods for variable elimination and polynomial system solving are different.

## 3. Imaging Equation and Solution Symmetry

For a pinhole perspective camera with known skewness, aspect ratio and principle point, its intrinsic parameter matrix  $K$  can be simply described by  $K = \text{diag}\{f, f, 1\}$ , in which  $f$  denotes the focal length. Given  $n$  ( $n \geq 4$ ) 3D-to-2D point correspondences  $\{\mathbf{u}_i \leftrightarrow \mathbf{x}_i, 1 \leq i \leq n\}$ , where  $\mathbf{x}_i = [x_i, y_i, z_i]^T$  represents the  $i$ -th 3D point in the world frame, while  $\mathbf{u}_i = [u_i, v_i, 1]^T$  the  $i$ -th point in the normalized image frame, the target problem is to simultaneously estimate the rigid transformation  $\{R, \mathbf{t}\}$  and the camera focal length  $f$ . The pinhole imaging equation can be written into

$$\bar{\lambda}_i \mathbf{u}_i = \text{diag}\{f, f, 1\} (R \mathbf{x}_i + \bar{\mathbf{t}}), i = 1, 2, \dots, n, \quad (1)$$

where  $\bar{\lambda}_i$  represents the depth factor of the  $i$ -th point.

We adopt the standard quaternion expression to parameterize the rotation matrix  $R$ , because of its simplicity and generality. Specifically, for a unit-norm quaternion  $\mathbf{q} =$

$[q_1, q_2, q_3, q_4]^T$ ,  $R$  can be parameterized by

$$R(\mathbf{q}) = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2q_2q_3 - 2q_1q_4 & 2q_2q_4 + 2q_1q_3 \\ 2q_2q_3 + 2q_1q_4 & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2q_3q_4 - 2q_1q_2 \\ 2q_2q_4 - 2q_1q_3 & 2q_3q_4 + 2q_1q_2 & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix}. \quad (2)$$

The quaternion expression assumes a two-fold symmetry, because of the fact that  $\mathbf{q}$  and  $-\mathbf{q}$  denote the same rotation. As recognized in [25], when the focal length  $f$  is unknown, another two-fold symmetry arises, due to the fact that  $\text{diag}\{f, f, 1\}R = \text{diag}\{-f, -f, 1\}(\text{diag}\{-1, -1, 1\}R)$ . According to the Hamilton product law of quaternions,  $\text{diag}\{-1, -1, 1\}R$  can be represented by the quaternion  $[-q_4, -q_3, q_2, q_1]^T$ . In summary, when using the universal quaternion parametrization for rotation, the solutions assume four-fold symmetry in general, that is, the following quadruplets are equivalent

$$\begin{aligned} \{f, q_1, q_2, q_3, q_4\}, \quad & \{f, -q_1, -q_2, -q_3, -q_4\}, \\ \{-f, -q_4, -q_3, q_2, q_1\}, \quad & \{-f, q_4, q_3, -q_2, -q_1\}. \end{aligned}$$

Some existing works, like [14, 27], have tried to exploit such symmetry in the process of developing Gröbner basis solvers. Since it is generally difficult to design symmetric Gröbner basis solvers, we choose instead to eliminate the symmetry by decomposing the rotation and properly handle the resulting parametrization degeneracy and numerical instability.

When  $q_1^2 + q_4^2 \neq 0$ , the rotation matrix  $R(\mathbf{q})$  can be decomposed into the product of two rotations  $R(\mathbf{q}_\theta)$  and  $R(\mathbf{q}_\omega)$ . Specifically,  $\mathbf{q}_\theta = [q_1, 0, 0, q_4]^T$  and

$$R(\mathbf{q}_\theta) = \frac{1}{q_1^2 + q_4^2} \begin{bmatrix} q_1^2 - q_4^2 & -2q_1q_4 & 0 \\ 2q_1q_4 & q_1^2 - q_4^2 & 0 \\ 0 & 0 & q_1^2 + q_4^2 \end{bmatrix} = \begin{bmatrix} \tilde{x} & -\tilde{y} & 0 \\ \tilde{y} & \tilde{x} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where  $\tilde{x}^2 + \tilde{y}^2 = 1$ . As recognized in [25],

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} & -\tilde{y} & 0 \\ \tilde{y} & \tilde{x} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f\tilde{x} & -f\tilde{y} & 0 \\ f\tilde{y} & f\tilde{x} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & -y & 0 \\ y & x & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

which indicates that the focal length  $f$  happens to be merged into the two independent variables  $x$  and  $y$ , and the two-fold symmetry due to the sign ambiguity of  $f$  can be eliminated. In addition,  $\mathbf{q}_\omega = [\tilde{a}, \tilde{b}, \tilde{c}, 0]^T$ , where  $\tilde{a} = q_1^2 + q_4^2$ ,  $\tilde{b} = q_1q_2 + q_3q_4$ ,  $\tilde{c} = q_1q_3 - q_2q_4$ . The two-fold symmetry arising from the sign ambiguity of quaternion can be eliminated by dividing  $\mathbf{q}_\omega$  by  $\tilde{a}$ , resulting in  $\mathbf{q}_\omega = [1, b, c, 0]^T$ , such that

$$R(\mathbf{q}_\omega) = R(b, c) = \frac{1}{k} \begin{bmatrix} 1 + b^2 - c^2 & 2bc & 2c \\ 2bc & 1 - b^2 + c^2 & -2b \\ -2c & 2b & 1 - b^2 - c^2 \end{bmatrix}, \quad (5)$$

in which  $k = 1 + b^2 + c^2$ .

Finally, when  $q_1^2 + q_4^2 \neq 0$ , the imaging equation in eq.(1) can be reformulated into

$$\lambda_i \mathbf{u}_i = \begin{bmatrix} x & -y & 0 \\ y & x & 0 \\ 0 & 0 & 1 \end{bmatrix} R(b, c) \mathbf{x}_i + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \mathbf{x}_i + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \quad (6)$$

in which  $\lambda_i = k\bar{\lambda}_i$  and  $[t_1, t_2, t_3]^T = k\bar{\mathbf{t}}$ . Note that the aforementioned solution symmetry has been removed. We are going to develop the direct least-squares solution on the basis of eq.(6). It contains 7 unknown variables  $\{x, y, b, c, t_1, t_2, t_3\}$  that we intend to solve, in addition to the scaled depth factors  $\lambda_i$ .

Note that, when  $q_1^2 + q_4^2$  is (or close to) 0, eq.(6) is not applicable. A principled strategy to resolve this issue will be presented in Sec.4.5.

## 4. Direct Least-Squares Solution

In this section, we first show how to eliminate linear variables in a balanced way, and then present our method to build a proper cost function, whose stationary points can be easily retrieved by using eigenvalue factorization.

### 4.1. Variable Elimination

Based on the projection equation in eq.(6), we eliminate the depth factors  $\lambda_i$  by

$$\lambda_i = \mathbf{r}_3^T \mathbf{x}_i + t_3, i = 1, 2, \dots, n, \quad (7)$$

which can be plugged back into the first and second row of eq.(6), such that

$$\mathbf{r}_3^T (u_i \mathbf{x}_i) + u_i t_3 = \mathbf{r}_1^T \mathbf{x}_i + t_1, \quad \mathbf{r}_3^T (v_i \mathbf{x}_i) + v_i t_3 = \mathbf{r}_2^T \mathbf{x}_i + t_2. \quad (8)$$

By averaging each set of  $n$  equations in eq.(8), we can build the centralized projection equation

$$\begin{aligned} \mathbf{r}_3^T \left( \frac{1}{n} \sum_{i=1}^n u_i \mathbf{x}_i \right) + t_3 \left( \frac{1}{n} \sum_{i=1}^n u_i \right) &= \mathbf{r}_1^T \left( \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \right) + t_1, \\ \mathbf{r}_3^T \left( \frac{1}{n} \sum_{i=1}^n v_i \mathbf{x}_i \right) + t_3 \left( \frac{1}{n} \sum_{i=1}^n v_i \right) &= \mathbf{r}_2^T \left( \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \right) + t_2. \end{aligned} \quad (9)$$

The variables  $t_1$  and  $t_2$  can be eliminated by subtracting eq.(8) by eq.(9), which leads to the decentralized equation

$$\mathbf{r}_3^T \hat{u} \mathbf{x}_i + \hat{u}_i t_3 = \mathbf{r}_1^T \hat{\mathbf{x}}_i, \quad \mathbf{r}_3^T \hat{v} \mathbf{x}_i + \hat{v}_i t_3 = \mathbf{r}_2^T \hat{\mathbf{x}}_i, \quad (10)$$

in which  $\hat{u} \mathbf{x}_i$ ,  $\hat{u}_i$ ,  $\hat{v} \mathbf{x}_i$ ,  $\hat{v}_i$  and  $\hat{\mathbf{x}}_i$  denote the decentralized value of  $u_i \mathbf{x}_i$ ,  $u_i$ ,  $v_i \mathbf{x}_i$ ,  $v_i$  and  $\mathbf{x}_i$ , respectively. This decentralization operation serves in effect as data normalization, which is known to be critical in designing an algebraic cost function [7].

Now, we multiply the first equation of eq.(10) by  $\hat{u}_i$  and the second one by  $\hat{v}_i$ . By summing up all  $2n$  equations,  $t_3$  can be expressed by

$$t_3 = \frac{1}{s} \left( \mathbf{r}_3^T \sum_{i=1}^n (\hat{u}_i \hat{u}_i \hat{\mathbf{x}}_i + \hat{v}_i \hat{v}_i \hat{\mathbf{x}}_i) - \mathbf{r}_1^T \sum_{i=1}^n \hat{u}_i \hat{\mathbf{x}}_i - \mathbf{r}_2^T \sum_{i=1}^n \hat{v}_i \hat{\mathbf{x}}_i \right), \quad (11)$$

in which  $s = \sum_{i=1}^n (\hat{u}_i^2 + \hat{v}_i^2)$ . This elimination method makes sure that the elimination is stable, since  $s$  is definitely greater than 0.

To replace  $t_3$  in eq.(10) by using eq.(11) would lead to the following equation

$$\mathbf{r}_3^T \mathbf{p}_{3i} = \mathbf{r}_1^T \mathbf{p}_{1i} + \mathbf{r}_2^T \mathbf{p}_{2i}, \quad \mathbf{r}_3^T \mathbf{q}_{3i} = \mathbf{r}_1^T \mathbf{q}_{1i} + \mathbf{r}_2^T \mathbf{q}_{2i}, \quad (12)$$

in which  $\mathbf{p}_{ki}$  and  $\mathbf{q}_{ki}$ ,  $k = 1, 2, 3$ ,  $i = 1, 2, \dots, n$ , are three dimensional column vectors and can be easily calculated on the basis of eq.(10) and eq.(11).

There remain only four unknown variables  $\{x, y, b, c\}$  in eq.(12). Let us further recognize that the variables  $x$  and  $y$  are linear, although they are coupled with  $b$  and  $c$ . To project out  $x$  and  $y$  would lead to a bivariate equation system, which is significantly easier to handle than the four-variable system in eq.(12).

By organizing the equations in eq.(12) properly, we can obtain the following parametric linear system

$$M(b, c)[x, y]^T = \mathbf{e}(b, c), \quad (13)$$

in which  $M(b, c)$  is a  $2n \times 2$  matrix with parametric entries of  $b$  and  $c$ .  $\mathbf{e}(b, c)$  is a  $2n$ -dimensional column vector. In the following, we are going to denote  $M(b, c)$  and  $\mathbf{e}(b, c)$  as  $M$  and  $\mathbf{e}$  for simplicity.

According to eq.(13),  $x$  and  $y$  can be projected out by  $[x, y]^T = M^\dagger \mathbf{e}$ , in which  $M^\dagger = (M^T M)^{-1} M^T$ . Plugging it back into eq.(13) leads to  $(MM^\dagger - I)\mathbf{e} = \mathbf{0}$ , in which  $I$  is the  $2n \times 2n$  entity matrix.

Considering that  $M^T M$  is positive definite (otherwise, there would be infinitely many solutions for  $x$  and  $y$ ), we multiply the square root of the determinant of  $M^T M$  at both sides of the above equation, and obtain

$$\sqrt{\det(M^T M)} (MM^\dagger - I)\mathbf{e} = \mathbf{0}, \quad (14)$$

upon which the objective function is built in the following section.

It is important to note that the aforementioned variable projection method is quite different from that in [25]. In [25], a homogeneous linear system similar to  $[M, \mathbf{e}][x, y, 1]^T = \mathbf{0}$  was constructed, which leads to some bivariate polynomials by calculating the determinant of all  $3 \times 3$  submatrices of  $[M, \mathbf{e}]$ . Effective as it is for the minimal problem, this method is not appropriate to the general  $n$ -point case, since the complexity to permute all submatrices is  $O(n^3)$ .

## 4.2. Objective Function and Global Minimization

In the presence of noise, equality in eq.(14) could not be completely satisfied. We therefore define the cost function as the least-square error of all  $2n$  equations in eq.(14) as follows

$$\begin{aligned} \psi(b, c) &= \det(M^T M) \mathbf{e}^T (MM^\dagger - I)^T (MM^\dagger - I) \mathbf{e} \\ &= -\det(M^T M) \mathbf{e}^T (MM^\dagger - I) \mathbf{e} \\ &= -\mathbf{e}^T [M(M^T M)^* M^T - \det(M^T M) I] \mathbf{e}, \end{aligned} \quad (15)$$

in which  $(M^T M)^*$  denotes the adjoint of  $M^T M$ . It is worthy of noting that, although  $M$  is parametric w.r.t.  $b$  and  $c$ ,  $(M^T M)^*$  and  $\det(M^T M)$  can be easily calculated, since  $M^T M$  is of size  $2 \times 2$ .

The objective function  $\psi(b, c)$  is a bivariate polynomial of degree 12. To find its global minimum, we try to find all its stationary points by solving the following partial derivative equations

$$\partial\psi(b, c)/\partial b = 0, \quad \partial\psi(b, c)/\partial c = 0, \quad (16)$$

which constitute an 11-degree bivariate polynomial system with 72 monomials in each equation.

## 4.3. Polynomial Eigenvalue Solver

A bivariate polynomial system can be easily solved by setting one variable as the hidden variable and constructing a polynomial eigenvalue factorization problem [16] guided by the parametric Sylvester resultant. As for our bivariate system, by for example hiding  $b$ , it can be written into<sup>1</sup>

$$(A_0 + bA_1 + \dots + b^{11}A_{11})\boldsymbol{\beta} = \mathbf{0}, \quad (17)$$

in which  $b$  is the polynomial eigenvalue and  $\boldsymbol{\beta} = [c^{20}, c^{19}, \dots, c, 1]^T$  the polynomial eigenvector.  $A_j$ ,  $j = 0, 1, \dots, 11$ , denote the  $21 \times 21$  coefficient matrices. One can easily solve a polynomial eigenvalue factorization problem by, for example, using *polyeig* in MATLAB.

As shown in Fig. 4, the polynomial eigenvalue factorization solver shows outstanding numerical precision. However, it gives 210 (real and complex) solutions in general, which is much greater than the maximum number (121) of solutions for a 11-degree bivariate system. Therefore, this solver is not desirable in terms of computational efficiency.

## 4.4. Gröbner Basis Solver

We have also tried to develop a more efficient solver by using the Gröbner basis technique [5]. The adapted Gröbner basis technique in geometric computer vision uses a random instance of the target polynomial system, and tries to record

<sup>1</sup>In our implementation, to hide  $b$  or  $c$  is decided online, according to which choice leads to a smaller condition number of  $A_0$ .

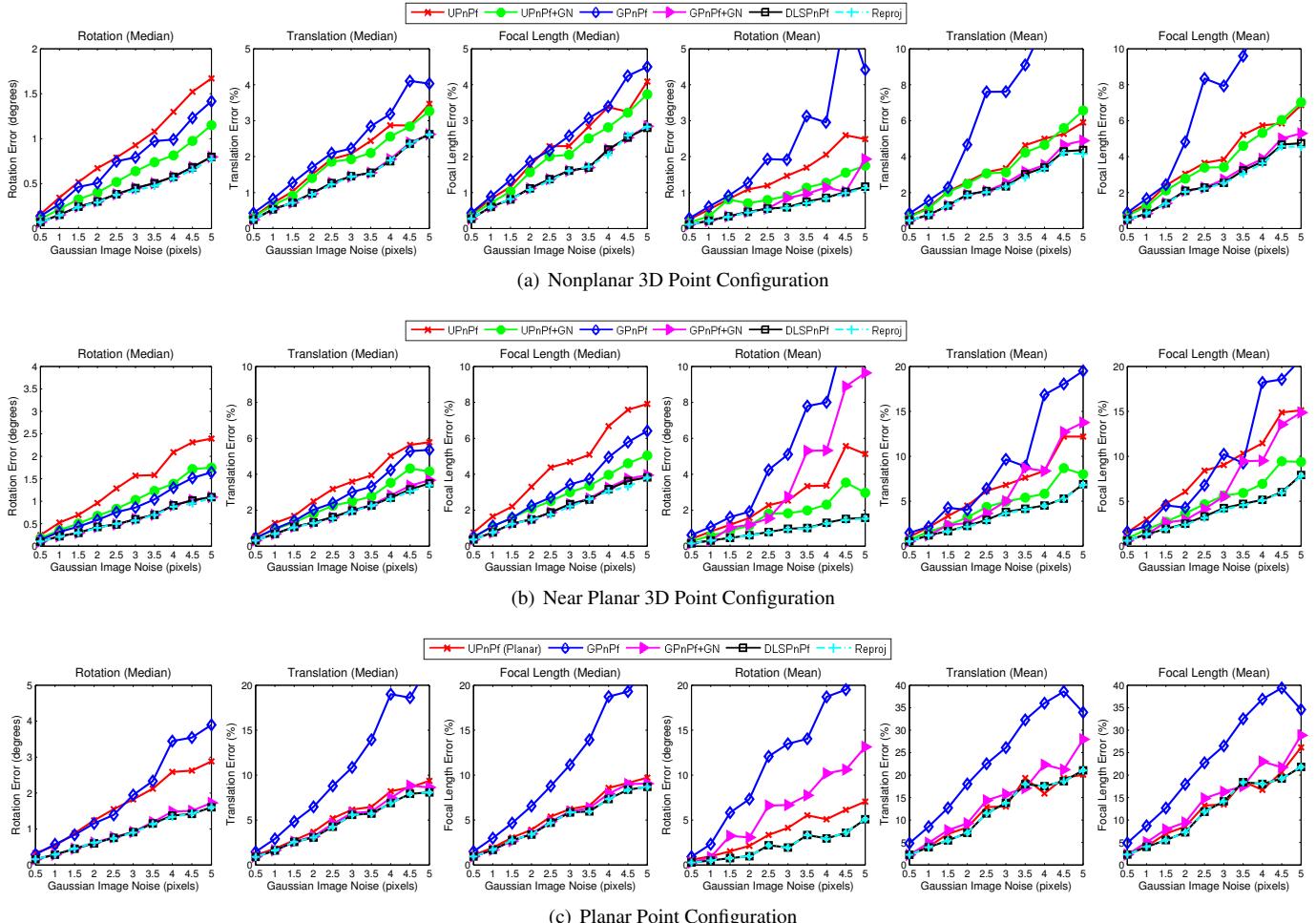


Figure 1. Estimation accuracy w.r.t. varying noise levels. The number of points  $n$  is 10. The performance under nonplanar, near planar and planar point configurations is shown in (a), (b) and (c), respectively. For each row, median errors among 500 trials for rotation, translation and focal length are shown in the 1st, 2nd and 3rd column, while the mean errors are indicated in the 4th, 5th and 6th column, respectively.

the path for monomial elimination. This elimination path is recorded in the so-called elimination template, from which one can construct the action matrix. The solutions to the original polynomial system can be found by eigenfactorization of the action matrix [23].

By using completely random coefficients for our bivariate system, the automatic solver generator [15] is able to generate a GB solver with 121 solutions. Unfortunately, the solver fails to work for any instance arising from a real pose estimation problem. We have found that the reason lies in the fact that the coefficients of our bivariate system are correlated, and the number of solutions is 61, rather than 121.

Inspired by this observation, we have generated consistent integer coefficients that preserve intrinsic correlations. By using such a consistent instance, we have been able to develop a GB solver, whose elimination template is of size  $130 \times 251$ , while action matrix is of  $61 \times 61$ . This solver takes about 3ms in our setting, which is more than 15 times faster than the polynomial eigenvalue solver. Due to its efficiency,

we are going to use the GB solver for all the experiments, unless explicitly stated otherwise.

#### 4.5. Avoiding (Near-)Degenerate Cases

Let us recall that the rotation decomposition applies only when  $q_1^2 + q_4^2 > 0$ . When  $q_1 = q_4 = 0$ , the decomposition is not unique [25]. In addition, when the magnitude of  $q_1$  and  $q_4$  is much smaller than  $q_2$  and  $q_3$ , to divide  $q_1^2 + q_4^2$  would cause numerical instability.

Here, we suggest a principled remedy to resolve those two issues on the basis of the following observation

$$R\mathbf{x}_i = (R\text{diag}\{1, -1, -1\})(\text{diag}\{1, -1, -1\}\mathbf{x}_i) = \tilde{R}\tilde{\mathbf{x}}_i, \quad (18)$$

in which  $\text{diag}\{1, -1, -1\}$  denotes the 180 degrees rotation around the  $x$ -axis.

Due to the Hamilton law, the quaternion of  $\tilde{R} = R\text{diag}\{1, -1, -1\}$  is  $[q_2, -q_1, -q_4, q_3]^T$ , from which we observe that the location of  $\{q_1, q_4\}$  and that of  $\{q_2, q_3\}$  are

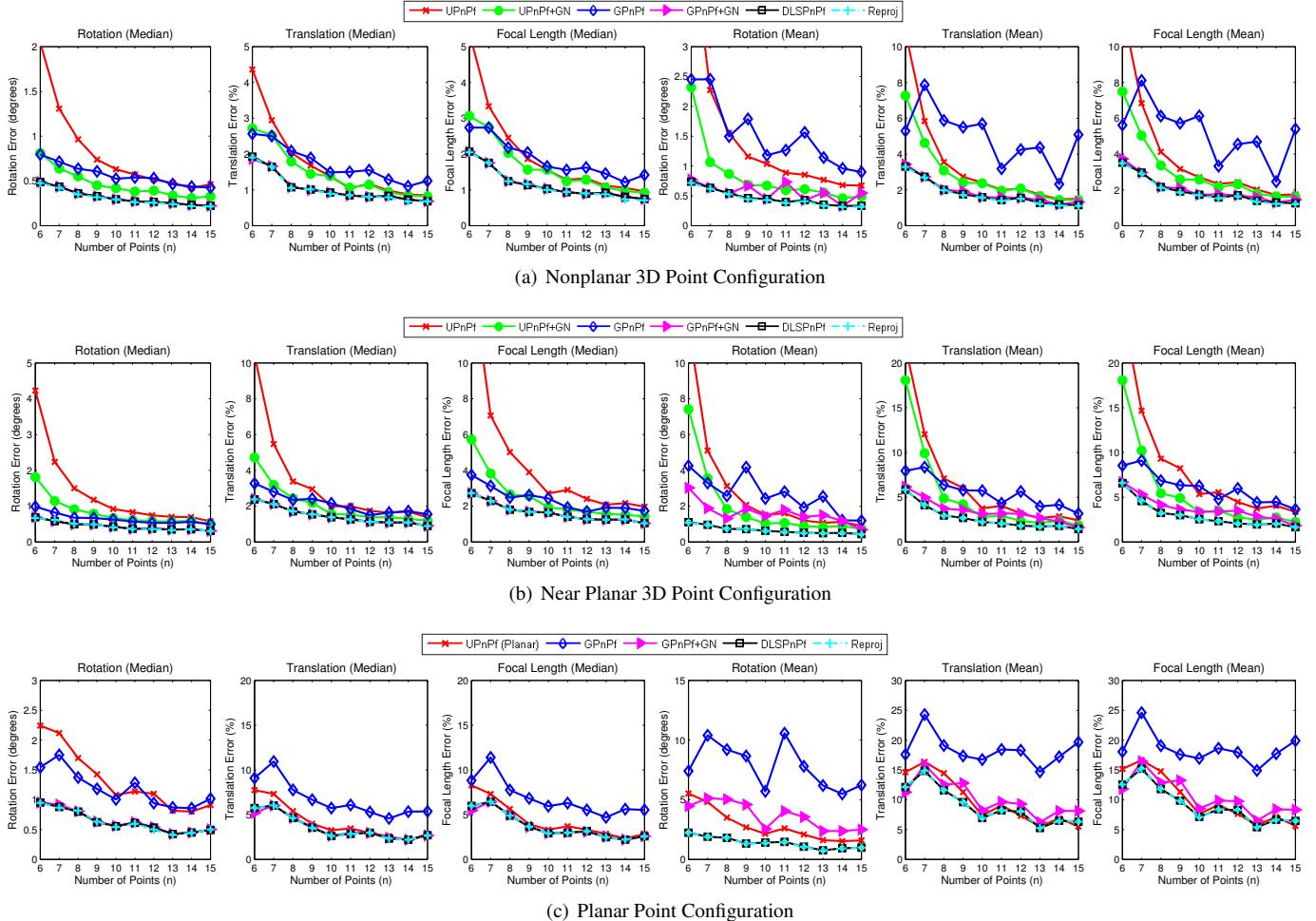


Figure 2. Estimation accuracy w.r.t. varying number of points. The noise level  $\delta$  is 2 pixels. The performance under nonplanar, near planar and planar point configurations is shown in (a), (b) and (c), respectively. For each row, median errors among 500 trials for rotation, translation and focal length are shown in the 1st, 2nd and 3rd column, while the mean errors in the 4th, 5th and 6th column, respectively.

switched. When  $q_1^2 + q_4^2$  is 0 (or close to 0), the decomposition of  $R$  is infeasible (or unstable), yet that of  $\tilde{R}$  should be feasible (or stable), since  $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$  has unit norm scale. As a result, our strategy to resolve the potential issues of rotation decomposition is to rotate the 3D points around the  $x$ -axis by 180 degrees, and solve the whole problem again.

Now the computational procedure can be summarized as follows: (1). Construct and solve the bivariate system in eq.(16) via the GB technique; (2). For each real solution of  $\{b, c\}$ , find  $x$  and  $y$  by solving eq.(13), from which the focal length and rotation can be recovered. (3). Recover the translation vector by using eq.(11) and eq.(9); (4). Rotate the 3D points around the  $x$ -axis by 180 degrees and repeat step (1-3).

In our implementation, we evaluate the reprojection error of each solution, and choose the one with smallest reprojection error as our final solution.

## 5. Experimental Results

In this section, we compare our direct least-squares solution (referred to as *DLSPnPf*) with the state-of-the-art solutions, including *UPnPf* and its iterative variant *UPnPf+GN* in [21], as well as *GPnPf* and its iterative variant *GPnPf+GN* [28]. To disclose the difference between reprojection error minimization and our minimization criterion in eq.(15), we also minimize the reprojection error and use the solution from *DLSPnPf* for initialization. This is denoted by *Reproj*.

We implement *DLSPnPf* in MATLAB and use the publicly available source codes of *UPnPf*, *UPnPf+GN*, *GPnPf* and *GPnPf+GN*. The comparison is conducted on a laptop with a 2.6GHz CPU and 12GB RAM.

### 5.1. Synthetic Data

We synthesize a pinhole camera with zero skew and unit aspect ratio, whose resolution is 800×640 pixels. The principle point is assumed to be at the image cen-

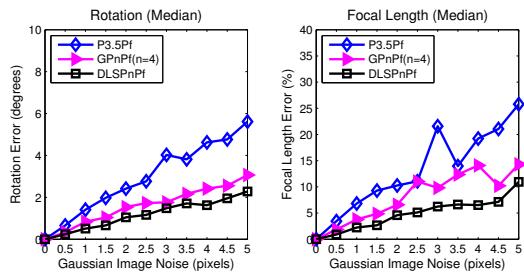


Figure 3. Comparison of estimation accuracy w.r.t. varying noise levels in the minimal 4-point case. The median rotation error and focal length error among 500 trials are shown in the left and right subfigure, respectively.

ter. The focal length is randomly chosen in the range from 200 to 2000 pixels. Considering that the methods might behave differently under different 3D point configurations, we randomly synthesize  $n$  points in the box of  $[-2,2] \times [-2,2] \times [4,8]$  for nonplanar 3D point configuration, while in  $[-2,2] \times [1,2] \times [4,8]$  for near planar point configuration. Ground-truth rotations and translations are generated, which are used to transform the 3D points into the world frame. We evaluate the absolute error (in degrees) for rotation, while the relative error (%) for focal length and translation.

### 5.1.1 Varying Noise Levels

Here, the number of points is fixed to 10. Zero-mean Gaussian noise with standard deviation  $\delta$  is added to disturb the image points. We vary the noise level  $\delta$  from 0.5 to 5 pixels. At each  $\delta$ , 500 independent trials are conducted, and the median and mean error for rotation, translation and focal length are reported in Fig.1. We examine the competing solutions under the nonplanar 3D, near planar and planar point configurations. Note that  $UPnPf+GN$  is not included in the planar case, since the Gauss-Newton iterative refinement step is not needed there.

From Fig.1, we can observe that both  $UPnPf$  and  $GPnPf$  are not sufficiently accurate, and they indeed rely heavily on their respective iterative refinement  $UPnPf+GN$  and  $GPnPf+GN$  to improve accuracy. In terms of median errors,  $GPnPf+GN$  can offer highly accurate estimation, which is usually better than  $UPnPf+GN$ . The reason is that the objective function in  $UPnPf+GN$  is build upon some virtual control points, and those anchor points give more weights to the minimized cost. However, when observing the mean errors,  $GPnPf+GN$  is usually worse than  $UPnPf+GN$ , especially for near planar and planar points with high noise levels ( $\delta > 2$  pixels). This reveals the fact that  $GPnPf+GN$  might get trapped into a poor local minimum from time to time, which is not obvious by examining the median error only. The risk of local minima or even divergence is a common problem in iterative optimization of a nonconvex problem. Fortunately, due to its noniterative nature,  $DL-$

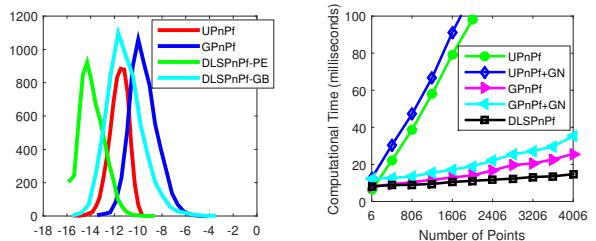


Figure 4. Numerical precision. Figure 5. Running time.

$SPnPf$  always has the highest accuracy, irrespective of the point configuration and the error criterion. Its accuracy is even comparable to that of *Reproj*. This verifies the validity of our derived cost function in eq.(15), although it is algebraically meaningful only.

Note that the translation and focal length error for all competing solutions significantly increase under the planar point configuration, which means that translation and focal length are hard to be disambiguated then. This becomes understandable by considering the extreme case that all 3D points lie on a plane that is parallel to the image plane, in which these two components can not be separated.

### 5.1.2 Varying Number of Points

Now, the noise level  $\delta$  is fixed to be 2 pixels. Considering that  $UPnPf$  does not work with 4 or 5 points, we vary  $n$  from 6 to 15. Similarly, we conduct 500 independent tests for each  $n$ . The median and mean errors of rotation, translation and focal length are shown in Fig.2. Again, although  $GPnPf+GN$  works well in most trials, it occasionally returns poor solutions, as verified by comparing the median and the mean errors. This issue becomes more apparent under the challenging near planar and planar point configurations. Similar to the observations in Sec.5.1.1,  $DLSPnPf$  is the most accurate solution in terms of both mean and median errors, and works consistently well under all examined point configurations.

### 5.1.3 Comparison with Minimal Solvers

Here, we compare  $DLSPnPf$  with the state-of-the-art 4-point solver [28] ( $GPnPf$  ( $n=4$ )) and the exactly minimal solver in [25] ( $P3.5Pf$ ). We randomly generate 4 nonplanar 3D points and corrupt their image projections by zero-mean Gaussian noise with standard deviation from 0 to 5 pixels. For the  $P3.5Pf$  solver, we simply ignore the  $v$ -axis measurement of the 4th point. The median error among 500 independent trials of rotation and focal length are shown in Fig.3. We can observe that  $DLSPnPf$  provides more accurate estimation, since all constraints have been utilized in a balanced way. Considering that the minimal solvers are much faster,  $DLSPnPf$  in its current form can not replace them in RANSAC-type applications. Instead, due to its accuracy and noniterativeness,  $DLSPnPf$  is currently suited

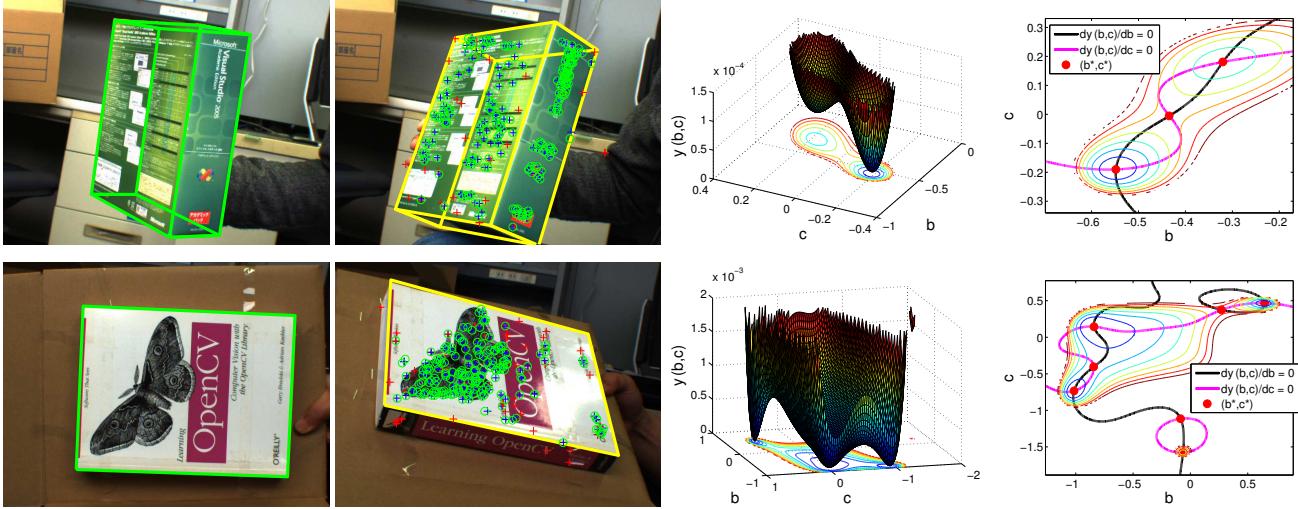


Figure 6. Pose and focal length estimation of a 3D box (1st row) and a planar book cover (2nd row). In each row, the first image is the reference, while the second image is the input. The input image has been augmented by using the projected contour calculated from the pose and focal length parameters of our direct least-squares solution. The third image visualizes the surface of our objective function, while the fourth image shows the intersection of the two derivative polynomials, in which red dots denote stationary points.

for outlier-free applications, like model-based pose tracking using fiducial markers and pose bundle adjustment, after outliers have been removed via RANSAC or other robust matching techniques [26].

#### 5.1.4 Numerical Precision

To verify that our strategy to avoid (near-)degeneracy in rotation decomposition is effective and that our solvers are numerically stable, we randomly synthesize 6000 nondegenerate, near degenerate and degenerate rotations (2000 for each type). We synthesize 10 noise-free points, and measure the  $\log_{10}$  value of the relative error between the estimated focal length and its corresponding ground truth. We also add  $UPnPf$  and  $GPnPf$  for comparison. The error histogram over all 6000 trials is shown in Fig.4, from which we observe that the numerical precision of our solution with the GB solver, i.e.  $DLSPnPf\text{-}GB$ , is comparable to those of  $UPnPf$  and  $GPnPf$ , although it is lower than the polynomial eigenvalue solver  $DLSPnPf\text{-}PE$ . This verifies the effectiveness of our strategy to avoid rotation degeneracy and the stability of our GB solver to handle high-degree polynomials.

#### 5.1.5 Computational Efficiency

It is easy to verify that the time complexity of  $DLSPnPf$  is  $O(n)$ , because of the matrix multiplication operation in eq.(15). Now we compare the running time of competing solutions by varying  $n$  from 6 to 4006. For each  $n$ , 500 independent trials are conducted, and the mean running time in milliseconds (ms) is reported in Fig.5. We can see that

$DLSPnPf$  is faster than  $UPnPf$  and  $GPnPf$ , especially when the number of points is large.

## 5.2. Real Images

We have also evaluated our direct least-squares solution by using real images provided in [27, 28]. We assume that the camera has zero-skew, unit aspect ratio and a centered principle point. Some tentative correspondences are established by matching the input image and the reference one. Outliers are removed via RANSAC with a threshold of 2 pixels. In Fig.6, we show two representative results of pose and focal length estimation for a box and a planar book cover. Considering that the ground-truth parameters are not provided, we do not conduct quantitative comparison. Instead, we draw the surface of the objective function in eq.(15) and the intersection of the two derivative polynomials in eq.(16), from which one can obtain a clearer sense of the difficulty of the problem due to its strong nonconvexity.

## 6. Conclusion

We have developed a direct least-squares solution for the perspective- $n$ -point pose problem of a partially uncalibrated camera with unknown focal length. Unlike existing solutions, which would give more weights to certain anchor points, our cost function is built directly upon the imaging equation, and all point correspondences are treated equally. Our proposed solution is noniterative and insensitive to the 3D point configuration. It can be applied to handle large problems because of its linear time complexity. The proposed solution has proven to be more accurate and efficient than the current state-of-the-art solutions.

## References

- [1] M. Abidi and T. Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: algorithm and evaluation. *TPAMI*, 17(5):534–538, 1995. [1](#) [2](#)
- [2] M. Bujnak. Algebraic solutions to absolute pose problems. *PhD Thesis, Czech Technical University*, 2012. [2](#)
- [3] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the p4p problem for camera with unknown focal length. *Proc. CVPR*, pages 1–8, 2008. [1](#) [2](#)
- [4] K. Choi, S. Lee, and Y. Seo. A branch-and-bound algorithm for globally optimal camera pose and focal length. *Image and Vision Computing*, 28(9):1369–1376, 2010. [1](#) [2](#)
- [5] D. Cox, J. Little, , and D. O’Shea. *Using Algebraic Geometry*. Springer-Verlag, 2005. [4](#)
- [6] Y. Guo. A novel solution to the P4P problem for an uncalibrated camera. *Journal of Mathematical Imaging and Vision*, 45(2):186–198, 2013. [2](#)
- [7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2nd edition, 2003. [1](#) [3](#)
- [8] J. Hesch and R. Roumeliotis. A direct least-squares (dls) method for PnP. *Proc. ICCV*, pages 383–390, 2011. [2](#)
- [9] A. Irschara1, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. *Proc. CVPR*, pages 2599–2606, 2009. [1](#)
- [10] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. *Proc. CVPR*, pages 2419–2426, 2009. [2](#)
- [11] E. Kanaeva, L. Gurevich, and A. Vakhitov. Camera pose and focal length estimation using regularized distance constraints. *Proc. BMVC*, 2015. [1](#) [2](#)
- [12] L. Kneip, H. Li, and Y. Seo. UPnP: An optimal O(n) solution to the absolute pose problem with universal applicability. *Proc. ECCV*, pages 127–142, 2014. [2](#)
- [13] 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. *Proc. CVPR*, pages 2969–2976, 2011. [2](#)
- [14] Y. Kuang, Y. Zheng, and K. Astrom. Partial symmetry in polynomial systems and its applications in computer vision. *Proc. CVPR*, pages 438–445, 2014. [3](#)
- [15] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. *Proc. ECCV*, pages 302–315, 2008. [5](#)
- [16] Z. Kukelova, M. Bujnak, and T. Pajdla. Polynomial eigenvalue solutions to minimal problems in computer vision. *TPAMI*, 34(7):1381–1393, 2012. [4](#)
- [17] Z. Kukelova, M. Bujnak, and T. Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. *Proc. ICCV*, pages 2816–2823, 2013. [2](#)
- [18] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *IJCV*, 81(2):155–166, 2008. [2](#)
- [19] S. Li and C. Xu. A stable direct solution of perspective-three-point problem. *IJPRAI*, 25(5):627–642, 2011. [2](#)
- [20] S. Li, C. Xu, and M. Xie. A robust O(n) solution to the perspective-n-point problem. *TPAMI*, 34(7):1444–1450, 2012. [2](#)
- [21] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *TPAMI*, 35(10):2387–2400, 2013. [1](#) [2](#) [6](#)
- [22] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3D. *ACM TOG (SIGGRAPH 2006)*, pages 835–846, 2006. [1](#)
- [23] H. Stewenius. *Grobner Basis Methods for Minimal Problems in Computer Vision*. PhD Thesis, Lund University, 2005. [5](#)
- [24] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. *Proc. ICCV*, pages 278–284, 1999. [1](#) [2](#)
- [25] C. Wu. P3.5P: Pose estimation with unknown focal length. *Proc. CVPR*, 2015. [1](#) [2](#) [3](#) [4](#) [5](#) [7](#)
- [26] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu. Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE TPAMI*, 2015. [8](#)
- [27] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisting the PnP problem: A fast, general and optimal solution. *Proc. ICCV*, pages 2344–2351, 2013. [2](#) [3](#) [8](#)
- [28] Y. Zheng, S. Sugimoto, I. Sato, and M. Okutomi. A general and simple method for camera pose and focal length determination. *Proc. CVPR*, pages 430–437, 2014. [1](#) [2](#) [6](#) [7](#) [8](#)