A Direct Least-Squares (DLS) Method for P*n*P

Joel A. Hesch and Stergios I. Roumeliotis*
University of Minnesota
Minneapolis, MN 55455
{joel|stergios}@cs.umn.edu

Abstract

In this work, we present a Direct Least-Squares (DLS) method for computing all solutions of the perspective-n-point camera pose determination (PnP) problem in the general case ($n \geq 3$). Specifically, based on the camera measurement equations, we formulate a nonlinear least-squares cost function whose optimality conditions constitute a system of three third-order polynomials. Subsequently, we employ the multiplication matrix to determine all the roots of the system analytically, and hence all minima of the LS, without requiring iterations or an initial guess of the parameters. A key advantage of our method is scalability, since the order of the polynomial system that we solve is independent of the number of points. We compare the performance of our algorithm with the leading PnP approaches, both in simulation and experimentally, and demonstrate that DLS consistently achieves accuracy close to the Maximum-Likelihood Estimator (MLE).

1. Introduction

The task of determining the six-degrees-of-freedom (d.o.f.) camera position and orientation (pose) from observations of known points in the scene has numerous applications in computer vision and robotics. Examples include robot localization [12], spacecraft pose estimation during descent and landing [20], pose determination for model-based vision [17], as well as hand-eye calibration [4].

The perspective-n-point pose determination problem (PnP) has been studied for various numbers of points (from the minimum of 3, to the general case of n), and several different solution approaches exist, such as: (i) directly solving the nonlinear geometric constraint equations in the minimal case [6], (ii) formulating an overdetermined linear system of equations in the non-minimal case [1], and (iii) iteratively minimizing a nonlinear least-squares cost function, which accounts for the measurement noise [9].

Currently, no approach exists that directly provides all solutions for PnP ($n \geq 3$), in a Maximum-Likelihood sense, without the need for initialization or approximations in the problem treatment. Some authors have proposed methods which reach close to the global optimum, e.g., based on successive Linear Matrix Inequality (LMI) relaxations [15], transformation to a Semi-Definite Program (SDP) [23], or a geometric transformation of the problem [16]. However, these approaches are only applicable when PnP admits a unique solution, which can only be guaranteed when $n \geq 6$, and some approaches require special treatment (e.g., when all points are co-planar).

The proposed Direct Least-Squares (DLS) method seeks to overcome the limitations of the current approaches:

- It computes all pose solutions, as the minima of a nonlinear least-squares cost function, in the general case of $n \geq 3$ points.
- No initialization is required, and the performance is consistently better than competing methods and close to that of Maximum-Likelihood Estimator (MLE).
- The method is scalable, since the size of the nonlinear least-squares cost function which is minimized is not dependent on the number of points.

The rest of this paper is organized as follows: Section 2 provides an overview of the related work on PnP. We describe our proposed approach in Section 3, while we present simulation and experimental comparisons to alternative approaches in Section 4. Lastly, we provide our concluding remarks in Section 5.

2. Related Work

The minimal PnP problem (i.e., P3P) has typically been addressed by treating the geometric constraint equations as noise-free, and solving for the camera pose [6, 8]. Haralick *et al.* [10] provided a comparison of the classical P3P methods and an analysis of singular configurations. Direct solutions have also been proposed for the overdetermined case (i.e., PnP, $n \geq 4$). For instance, Horaud

*This work was supported by the University of Minnesota (DTC), and the National Science Foundation (IIS-0643680, IIS-0811946, IIS-0835637).

et al. [13] addressed the P4P problem by connecting the four known points to form three known lines, and exploiting the nonlinear line projection equations to compute the camera pose. Linear methods (e.g., based on lifting) also exist for both P4P and PnP [1, 16, 21, 22]. Significant work has also focused on characterizing the number of solutions for P3P [5, 7, 25], and PnP [14, 25].

A key drawback of the approaches which consider noise-free measurements is that they may return inaccurate or even erroneous solutions in the presence of noise. Hence, these analytic methods are most often employed as an initialization step for an MLE of the camera pose [24].

Several authors have addressed the PnP problem from a least-squares perspective, by iteratively minimizing a cost function which is the sum of the squared errors (either reprojection or geometric) for each point [9, 24]. These methods are more accurate, since they explicitly account for the measurement noise, and under certain noise assumptions, return the maximum-likelihood estimate of the camera pose. However, they can only compute one solution (out of possibly many), and require a good initial guess of the camera pose to converge.

Other approaches exist that seek to directly compute a global optimum without initialization. For instance, Kahl and Henrion [15] proposed a method based on a series of LMI relaxations, while Schweighofer and Pinz [23] presented an approach which first transforms the PnP problem into an SDP before optimizing for the camera pose. Unfortunately, these approaches do not provide a method for computing multiple solutions when they exist, and may require special treatment if the known points are co-planar.

In contrast to the above methods, we present a Direct Least-Squares (DLS) approach for PnP which accounts for the measurement noise, and admits all solutions to the problem without requiring iterations or an initial guess of the camera pose. Specifically, we reparametrize the constraint equations to obtain a polynomial cost function that only depends on the unknown orientation. We then solve the corresponding optimality conditions analytically, and recover all minima (pose hypotheses) of the LS problem directly.

3. Problem Formulation

3.1. Measurement Model

The camera observation of known points in the scene projected onto the image plane can be described by the spherical camera model:

$$\mathbf{z}_i = {}^S\bar{\mathbf{r}}_i + \boldsymbol{\eta}_i \tag{1}$$

$${}^S\mathbf{r}_i = {}^S_G\mathbf{C}\,{}^G\mathbf{r}_i + {}^S\mathbf{p}_G \tag{2}$$

where $\mathbf{z}_i$ is the measurement of the unit-vector direction, ${}^S\bar{\mathbf{r}}_i = \frac{{}^S\mathbf{r}_i}{||{}^S\mathbf{r}_i||}$, from the sensor frame $\{S\}$ towards point i, which is corrupted by noise $\boldsymbol{\eta}_i$. The point's coordinates in the sensing frame $\{S\}$ are a function of the known coordinates, ${}^G\mathbf{r}_i$, in the global frame $\{G\}$, as well as the unknown global-to-sensor transformation described by the rotation matrix ${}^S_G\mathbf{C}$ and translation vector ${}^S\mathbf{p}_G$. Figure 1 depicts the observation of three non-collinear points, which is the minimal case required in order to be able to solve the measurement equations and recover the camera pose.

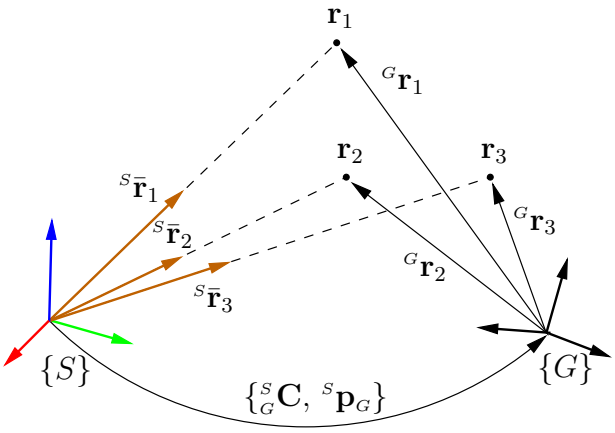


Figure 1. This figure depicts the observations of points $\mathbf{r}_i$, $i = 1, 2, 3$ via the unit-vector directions ${}^S\bar{\mathbf{r}}_i$ from the origin of the camera frame $\{S\}$ towards each point. The distance from $\{S\}$ to each point is $\alpha_i = ||{}^S\mathbf{r}_i||$. The vector ${}^S\mathbf{p}_G$ is the origin of $\{G\}$ with respect to $\{S\}$, the rotation matrix from $\{G\}$ to $\{S\}$ is ${}^S_G\mathbf{C}$, and ${}^G\mathbf{r}_i$ is the position of each point in $\{G\}$.

3.2. Cost function

PnP can be formulated as the following constrained nonlinear least-squares minimization problem:

$$\begin{aligned} \{\alpha_i^*,\ {}^S_G\mathbf{C}^*,\ {}^S\mathbf{p}_G^*\} &= \arg\min J \\ \text{subject to} \quad {}^S_G\mathbf{C}^T\,{}^S_G\mathbf{C} &= \mathbf{I}_3,\ \ \det({}^S_G\mathbf{C}) = 1 \\ \alpha_i &= ||{}^S_G\mathbf{C}\,{}^G\mathbf{r}_i + {}^S\mathbf{p}_G|| \end{aligned} \tag{3}$$

where the cost function J is the sum of the squared measurement errors, i.e.,

$$J = \sum_{i=1}^{n} ||\mathbf{z}_i - {}^S\bar{\mathbf{r}}_i||^2 = \sum_{i=1}^{n} ||\mathbf{z}_i - \frac{1}{\alpha_i}\left({}^S_G\mathbf{C}\,{}^G\mathbf{r}_i + {}^S\mathbf{p}_G\right)||^2.$$

Unfortunately, J is nonlinear in the unknown quantities, and computing all of its local minima is quite challenging. One approach is to select an initial guess for the parameter vector and employ an iterative minimization technique, such as Gauss-Newton, to numerically compute a single local minimum of J. A clear limitation of this approach is that it can only converge to one of the minima of the cost function, and even with multiple restarts, we are not guaranteed to obtain all minima of J. An alternative approach is to attempt to analytically solve the system of equations provided

by the Karush-Khun-Tucker (KKT) optimality conditions of (3) for the unknown quantities. However, this method is also challenging since the KKT conditions form a nonlinear system of equations in $6+n$ unknowns (3 from ${}^S_G\mathbf{C}$, 3 from ${}^G\mathbf{p}_S$, and n from α_i, $i=1,\dots,n$).[1] A third strategy is to relax the original optimization problem [see (3)] and manipulate the measurement equations to reduce the number of unknowns. This leads to a modified LS problem for the reduced set of parameters, which can be solved analytically.

In this paper, we follow the third approach, which is described in Sects. 3.3-3.5. Before discussing our method in detail, we first provide a brief overview. We satisfy the constraints in the following way: (i) We employ the Cayley-Gibbs-Rodriguez (CGR) parametrization of the rotation matrix ${}^S_G\mathbf{C}$ and utilize the three CGR parameters as unconstrained optimization variables. In this way we satisfy the rotation matrix constraints, ${}^S_G\mathbf{C}^T\,{}^S_G\mathbf{C}=\mathbf{I}$ and $\det({}^S_G\mathbf{C})=1$, exactly. (ii) We relax the scale constraint $\alpha_i=||{}^S_G\mathbf{C}\,{}^G\mathbf{r}_i+{}^S\mathbf{p}_G||$, treating each α_i as a free parameter. Note that this relaxation is reasonable since solving the optimality conditions results in $\alpha_i^*=\mathbf{z}_i^T({}^S_G\mathbf{C}\,{}^G\mathbf{r}_i+{}^S\mathbf{p}_G)$, which exactly satisfies the constraint when the measurements are noise free (see Appendix A). Subsequently, in order to reduce the number of unknown parameters in the LS cost function, we manipulate the measurement equations, and express ${}^S\mathbf{p}_G$ and α_i as functions of the unknown rotation ${}^S_G\mathbf{C}$. We then directly solve a modified LS problem to obtain all rotation hypotheses (local minima), from which we recover the scale α_i and translation ${}^S\mathbf{p}_G$.

3.3. Modified measurement equations

We first consider the noise-free geometric constraints which appear in the measurement model (1),

$$\alpha_i\,{}^S\bar{\mathbf{r}}_i={}^S_G\mathbf{C}\,{}^G\mathbf{r}_i+{}^S\mathbf{p}_G,\ i=1,\dots,n. \tag{4}$$

This system of equations contains unknown quantities (α_i, ${}^S_G\mathbf{C}$, ${}^S\mathbf{p}_G$), and quantities which are either known perfectly (${}^G\mathbf{r}_i$), or are measured by the camera (${}^S\bar{\mathbf{r}}_i$). We would like to reparametrize this system of equations in terms of fewer unknowns. Since both the scale and translation parameters appear linearly, they are good candidates for reduction. We can rewrite (4) in matrix-vector form as

$$\underbrace{\begin{bmatrix}{}^S\bar{\mathbf{r}}_1 & & -\mathbf{I}\\ & \ddots & \vdots\\ & {}^S\bar{\mathbf{r}}_n & -\mathbf{I}\end{bmatrix}}_{\mathbf{A}}\underbrace{\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_n\\ {}^S\mathbf{p}_G\end{bmatrix}}_{\mathbf{x}}=\underbrace{\begin{bmatrix}{}^S_G\mathbf{C} & & \\ & \ddots & \\ & & {}^S_G\mathbf{C}\end{bmatrix}}_{\mathbf{W}}\underbrace{\begin{bmatrix}{}^G\mathbf{r}_1\\ \vdots\\ {}^G\mathbf{r}_n\end{bmatrix}}_{\mathbf{b}}$$
$$\Leftrightarrow\ \mathbf{A}\mathbf{x}=\mathbf{W}\mathbf{b} \tag{5}$$

where $\mathbf{A}$ and $\mathbf{b}$ comprise quantities that are known or measured, $\mathbf{x}$ is the vector of unknowns which we wish to eliminate from the system of equations, and $\mathbf{W}$ is a block diagonal matrix of the unknown rotational matrix. From (5), we can express ${}^S\mathbf{p}_G$ and α_i, $i=1,\dots,n$ in terms of the other system quantities as

$$\mathbf{x}=\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b}=\begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}\mathbf{W}\mathbf{b} \tag{6}$$

where we have partitioned $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ into $\mathbf{U}$ and $\mathbf{V}$ such that the scale parameters are a function of $\mathbf{U}$ and the translation is a function of $\mathbf{V}$. Exploiting the sparse structure of $\mathbf{A}$, $\mathbf{U}$ and $\mathbf{V}$ in (6) are computed in closed form (see Appendix A).

We note that both ${}^S\mathbf{p}_G$ and α_i are *linear* functions of the unknown rotation matrix ${}^S_G\mathbf{C}$, i.e.,

$$\alpha_i=\mathbf{u}_i^T\mathbf{W}\mathbf{b},\quad i=1,\dots,n \tag{7}$$
$${}^S\mathbf{p}_G=\mathbf{V}\mathbf{W}\mathbf{b}, \tag{8}$$

where $\mathbf{u}_i^T$ corresponds to the i-th row of matrix $\mathbf{U}$ [see (6)]. Hence, we can rewrite the constraint equations (4) as

$$\underbrace{\mathbf{u}_i^T\mathbf{W}\mathbf{b}}_{\alpha_i}\,{}^S\bar{\mathbf{r}}_i={}^S_G\mathbf{C}\,{}^G\mathbf{r}_i+\underbrace{\mathbf{V}\mathbf{W}\mathbf{b}}_{{}^S\mathbf{p}_G},\quad i=1,\dots,n. \tag{9}$$

At this point, we have reduced the number of unknown parameters from $6+n$ down to 3. Furthermore, we express the rotation matrix in terms of the CGR parameters $\mathbf{s}=\begin{bmatrix}s_1 & s_2 & s_3\end{bmatrix}^T$, where

$${}^S_G\mathbf{C}=\frac{\bar{\mathbf{C}}}{1+\mathbf{s}^T\mathbf{s}} \tag{10}$$
$$\bar{\mathbf{C}}\triangleq\left(\left(1-\mathbf{s}^T\mathbf{s}\right)\mathbf{I}_3+2\lfloor\mathbf{s}\times\rfloor+2\mathbf{s}\mathbf{s}^T\right), \tag{11}$$

where $\mathbf{I}_3$ denotes the 3×3 identity matrix, and $\lfloor\mathbf{s}\times\rfloor$ is the skew-symmetric matrix parametrized by $\mathbf{s}$. Using the CGR parameters will allow us to formulate a LS minimization problem in $\mathbf{s}$ that automatically satisfies the rotation matrix constraints, i.e., ${}^S_G\mathbf{C}^T\,{}^S_G\mathbf{C}=\mathbf{I}$, $\det({}^S_G\mathbf{C})=1$. We can explicitly show the dependence of (9) on $\mathbf{s}$, i.e.,

$$\mathbf{u}_i^T\mathbf{W}\left({}^S_G\mathbf{C}(\mathbf{s})\right)\mathbf{b}\,{}^S\bar{\mathbf{r}}_i={}^S_G\mathbf{C}(\mathbf{s})\,{}^G\mathbf{r}_i+\mathbf{V}\mathbf{W}\left({}^S_G\mathbf{C}\,(\mathbf{s})\right)\mathbf{b}. \tag{12}$$

[1] Note that in this case, the KKT conditions can be written as a system of polynomial equations whose degree and number of variables depend linearly on the number of measurements. Given the doubly exponential (in the degree and number of variables) complexity of current methods for solving polynomial systems, this approach is only practical for small-scale problems.

Note that ${}^{S}_{G}\mathbf{C}(\mathbf{s})$ appears *linearly* in this equation. This allows one further simplification, specifically, we can cancel the denominator $1+\mathbf{s}^T\mathbf{s}$ from the constraint equation (12) [see (10)], i.e.,

$$\mathbf{u}_i^T\mathbf{W}\big(\bar{\mathbf{C}}(\mathbf{s})\big)\,\mathbf{b}\,{}^{S}\bar{\mathbf{r}}_i = \bar{\mathbf{C}}(\mathbf{s})\,{}^{G}\mathbf{r}_i + \mathbf{V}\mathbf{W}\big(\bar{\mathbf{C}}(\mathbf{s})\big)\,\mathbf{b}, \quad (13)$$

which renders constraints that are quadratic in $\mathbf{s}$.

To summarize, we began with the original geometric constraint relationship between a known point coordinate ${}^{G}\mathbf{r}_i$ and its noise-free observation ${}^{S}\bar{\mathbf{r}}_i$, and reparametrized the geometric constraint to be only a function of the unknown rotation matrix ${}^{S}_{G}\mathbf{C}$. To do so, we treated the unknown scales $\alpha_i, i=1,\ldots,n$, as independent variables, relaxing the original problem formulation (3). Subsequently, we employed the CGR parameters to express orientation, and as a final step, we canceled the denominator from the CGR rotation matrix. Hence, this approach results in constraints which are quadratic in the elements of $\mathbf{s}$.

3.4. Modified cost function

We employ the modified measurement constraint (13) to formulate a LS minimization problem for computing the optimal CGR rotation parameters $\mathbf{s}$. Recalling that the measured unit-vector direction towards each point is $\mathbf{z}_i = \bar{\mathbf{r}}_i + \boldsymbol{\eta}_i$, we rewrite the measurement constraints as

$$\mathbf{u}_i^T\mathbf{W}(\bar{\mathbf{C}}(\mathbf{s}))\,\mathbf{b}\,(\mathbf{z}_i-\boldsymbol{\eta}_i) = \bar{\mathbf{C}}(\mathbf{s})\,{}^{G}\mathbf{r}_i + \mathbf{V}\mathbf{W}(\bar{\mathbf{C}}(\mathbf{s}))\,\mathbf{b} \quad (14)$$

$$\Rightarrow\ \mathbf{u}_i^T\mathbf{W}(\bar{\mathbf{C}}(\mathbf{s}))\,\mathbf{b}\,\mathbf{z}_i - \bar{\mathbf{C}}(\mathbf{s})\,{}^{G}\mathbf{r}_i - \mathbf{V}\mathbf{W}(\bar{\mathbf{C}}(\mathbf{s}))\,\mathbf{b} = \boldsymbol{\eta}_i' \quad (15)$$

where $\boldsymbol{\eta}_i'$ is a zero-mean noise term that is a function of $\boldsymbol{\eta}_i$, but whose covariance depends on the system parameters, and both $\mathbf{u}_i$ and $\mathbf{V}$ are evaluated at ${}^{S}\bar{\mathbf{r}}_i = \mathbf{z}_i$.

Based on (15), the pose-determination problem can be reformulated as the following *unconstrained* least-squares minimization problem

$$\{s_1^*, s_2^*, s_3^*\} = \arg\min J' \quad (16)$$

where the cost function J' is the sum of the squared constraint errors from (15), i.e.,

$$\begin{aligned} J' &= \sum_{i=1}^{n} ||\mathbf{u}_i^T\mathbf{W}(\bar{\mathbf{C}}(\mathbf{s}))\,\mathbf{b}\,\mathbf{z}_i - \bar{\mathbf{C}}(\mathbf{s})\,{}^{G}\mathbf{r}_i - \mathbf{V}\mathbf{W}(\bar{\mathbf{C}}(\mathbf{s}))\,\mathbf{b}||^2 \\ &= \sum_{i=1}^{n} \boldsymbol{\eta}_i'^T\boldsymbol{\eta}_i'. \end{aligned} \quad (17)$$

Note that each summand in J' is quartic in the elements of $\mathbf{s}$, and J' contains all monomials up to degree four, i.e., $\{1, s_1, s_2, s_3, s_1s_2, s_1s_3, s_2s_3, \ldots, s_1^4, s_2^4, s_3^4\}$.

Since J' is a fourth-order polynomial, the corresponding optimality conditions form a system of three third-order polynomials. What we show next, is how to employ the Macaulay matrix to directly compute all of the critical points of J' by finding the roots of the polynomial system.

A key benefit of our proposed approach is that the polynomial system we solve is of a constant degree, independent of the number of points in the PnP problem. Changing the number of points only affects the coefficients appearing in the system. Thus, we need only compute the Macaulay matrix symbolically once. Subsequently, we simply form the elements of the Macaulay matrix from the data (an operation which is linear in the number of points), and directly find the roots via the eigen decomposition of the Schur complement of the Macaulay matrix (see Sect. 3.5).[2]

3.5. Directly computing the local minima

What follows next is a brief overview of how we employ the Macaulay matrix [18, 19] to directly determine the roots of a system of polynomial equations. We refer the interested reader to "Using Algebraic Geometry" by Cox *et al.* [3] for a more complete perspective.

Since J' is a fourth-order polynomial function in three unknowns, the corresponding optimality conditions form a system of polynomial equations, i.e.,

$$\nabla_{s_i} J' = F_i = 0, \quad i = 1,2,3. \quad (18)$$

Each F_i is a polynomial of degree three in the variables s_1, s_2, s_3. The Bézout bound (i.e., the maximum number of possible solutions) for this system of equations is 27. Under mild conditions [3], which are met for general PnP instantiations, the Bézout bound is reached.

Our goal is to compute the *multiplication matrix* from which we can directly obtain all the solutions to our system via eigen decomposition [2]. We obtain the multiplication matrix by first constructing the Macaulay resultant matrix. To do so, we augment our polynomial system with an additional linear equation, which is generally non-zero at the roots of our system, i.e., $F_0 = u_0 + u_1s_1 + u_2s_2 + u_3s_3$, where each u_j, $j = 0,\ldots,3$ is randomly generated. We denote the set of all monomials up to degree 7 as

$$S = \{\mathbf{s}^{\boldsymbol{\gamma}} : \textstyle\sum_j \gamma_j \leq 7\} \quad (19)$$

where we use the notation $\mathbf{s}^{\boldsymbol{\gamma}} \triangleq s_1^{\gamma_1}s_2^{\gamma_2}s_3^{\gamma_3}$, $\gamma_i \in \mathbb{Z}_{\geq 0}$, to denote a specific monomial. The set S is important, since, using S we can expand our original system of polynomials to obtain a square system that has the same number of equations as monomials. To do so, we first partition S into four subsets, such that S_3 contains all monomials that can be divided by s_3^3, S_2 contains all monomials that can be divided by s_2^3 but not s_3^3, S_1 contains all monomials that can be divided by s_1^3 but not by s_2^3 or s_3^3, and S_0 contains the remaining monomials, i.e.,

[2] We compute the Schur complement of a sparse 120×120 matrix, followed by the eigen decomposition of a non-sparse 27×27 matrix. The total time to complete both operations in Matlab is approximately 15 ms.

$$S_0 = \{1, s_1, s_1^2, s_2, s_1 s_2, s_1^2 s_2, s_2^2, s_1 s_2^2, s_1^2 s_2^2, s_3, s_1 s_3, s_1^2 s_3, s_2 s_3, s_1 s_2 s_3, s_1^2 s_2 s_3, s_2^2 s_3, s_1 s_2^2 s_3, s_1^2 s_2^2 s_3, s_3^2, s_1 s_3^2, s_1^2 s_3^2, s_2 s_3^2, s_1 s_2 s_3^2, s_1^2 s_2 s_3^2, s_2^2 s_3^2, s_1 s_2^2 s_3^2, s_1^2 s_2^2 s_3^2\}.$$

Note that the second, fourth, and tenth elements of S_0 are the three CGR rotation parameters $\{s_1, s_2, s_3\}$; a fact that we will exploit later.

We next form an extended system of equations by multiplying F_0 with each of the monomials in S_0, and multiplying F_i with each of the monomials in S_i divided by s_i^3, $i = 1, 2, 3$. We denote polynomials obtained from extending F_i as $G_{i,j}$, $j = 1, \ldots, |S_i|$. Thus, the extended set of polynomial equations is

$$\begin{bmatrix} G_{0,1} \\ G_{0,2} \\ \vdots \\ G_{1,1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{0,1}^T \\ \mathbf{c}_{0,2}^T \\ \vdots \\ \mathbf{c}_{1,1}^T \\ \vdots \end{bmatrix} \underline{\mathbf{s}}^{\boldsymbol{\gamma}} = \mathbf{M}\underline{\mathbf{s}}^{\boldsymbol{\gamma}} = \mathbf{M} \begin{bmatrix} \underline{\mathbf{s}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{s}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{20}$$

where each polynomial $G_{i,j}$ is expressed as an inner product between the coefficient vector, $\mathbf{c}_{i,j}^T$, and the vector of all monomials $\underline{\mathbf{s}}^{\boldsymbol{\gamma}}$, i.e., $G_{i,j} = \mathbf{c}_{i,j}^T \underline{\mathbf{s}}^{\boldsymbol{\gamma}}$. The Macaulay matrix $\mathbf{M}$ is formed by stacking the coefficient vectors. Finally, we partition $\underline{\mathbf{s}}^{\boldsymbol{\gamma}}$ such that $\underline{\mathbf{s}}^{\boldsymbol{\alpha}}$ comprises monomials in S_0, and $\underline{\mathbf{s}}^{\boldsymbol{\beta}}$ contains the remaining monomials.

If we evaluate (20) at a root, $\mathbf{p} = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T$, of the original system (18), then all polynomials $G_{i,j}$ extended from F_i, $i = 1, 2, 3$ will be zero, since $F_i(\mathbf{p}) = 0$ by definition. However, F_0 and hence $G_{0,j}$, $j = 1, \ldots, |S_0|$ will not generally be zero, i.e.,

$$\begin{bmatrix} G_{0,1}(\mathbf{p}) \\ \vdots \\ G_{0,|S_0|}(\mathbf{p}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \Leftrightarrow \begin{bmatrix} F_0(\mathbf{p})\underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = \mathbf{M} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{21}$$

where $\underline{\mathbf{p}}^{\boldsymbol{\alpha}}$ and $\underline{\mathbf{p}}^{\boldsymbol{\beta}}$ denote the monomial vectors evaluated at $\mathbf{p}$, i.e., $\underline{\mathbf{s}}^{\boldsymbol{\alpha}}(\mathbf{p}) = \underline{\mathbf{p}}^{\boldsymbol{\alpha}}$ and $\underline{\mathbf{s}}^{\boldsymbol{\beta}}(\mathbf{p}) = \underline{\mathbf{p}}^{\boldsymbol{\beta}}$. Based on this observation, we partition $\mathbf{M}$ into four blocks where $\mathbf{M}_{00}$ is of dimension $|S_0| \times |S_0|$, and rewrite (21) as

$$\begin{bmatrix} F_0(\mathbf{p})\underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{00} & \mathbf{M}_{01} \\ \mathbf{M}_{10} & \mathbf{M}_{11} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix}. \tag{22}$$

Finally, exploiting the Schur complement, we obtain

$$F_0(\mathbf{p})\underline{\mathbf{p}}^{\boldsymbol{\alpha}} = \mathcal{M}_{F_0}\underline{\mathbf{p}}^{\boldsymbol{\alpha}} \tag{23}$$

where $\mathcal{M}_{F_0} = \mathbf{M}_{00} - \mathbf{M}_{01}\mathbf{M}_{11}^{-1}\mathbf{M}_{10}$ is the multiplication matrix corresponding to F_0. From (23) we see that $F_0(\mathbf{p})$ is an eigenvalue of $\mathcal{M}_{F_0}$ with corresponding eigenvector $\underline{\mathbf{p}}^{\boldsymbol{\alpha}}$. We can directly obtain all 27 solutions to our system of equations (18) via eigen decomposition, since the eigenvectors of $\mathcal{M}_{F_0}$ are the monomials of S_0 evaluated at each of the 27 roots. Since the first element in S_0 is 1, we normalize each eigenvector by its first element, and read off the solution for s_i, $i = 1, 2, 3$, from the second, fourth, and tenth elements of the eigenvector.

Through this procedure we obtain 27 critical points, which include real and imaginary minima, maxima, and saddle points of the cost function (17). In practice, we have only observed up to 4 real local minima that place the points in front of the center of perspectivity. In almost all cases, when $n \geq 6$ we obtain a single real minimum of the function. After obtaining the minima, we evaluate the cost function to find the optimal orientation, and compute the corresponding translation from (8). Additional details about the DLS PnP algorithm implementation are available as supplemental material [11].

4. Simulation and Experimental Results

4.1. Simulations

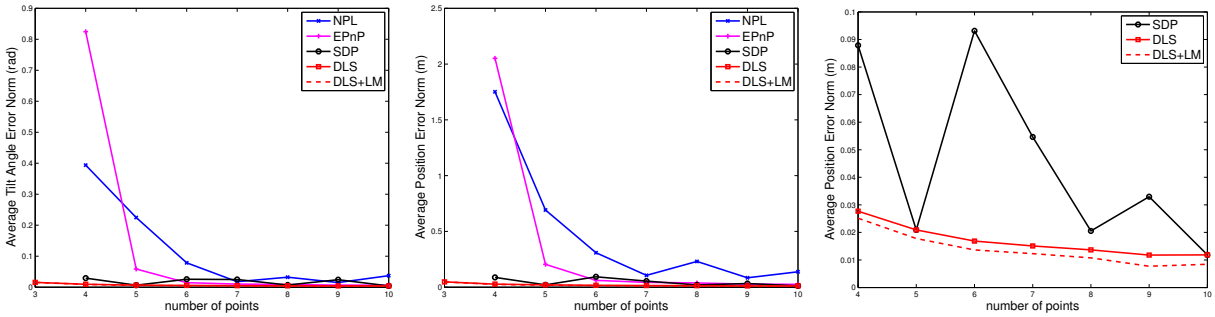
We hereafter present simulation results which compare the accuracy of our method to the leading PnP approaches:

- **NPL:** The N-Point Linear (NPL) method of Ansar and Daniilidis [1].
- **EP*n*P:** The approach of Lepitit *et al.* [16].
- **SDP:** The Semi Definite Program (SDP) approach of Schweighofer and Pinz [23].
- **DLS:** The Direct Least-Squares (DLS) solution presented in this paper. An open source implementation of DLS is available at `www.umn.edu/~joel`
- **DLS-LM:** Maximum-likelihood estimate, computed using iterative Levenberg-Marquardt (LM) minimization of the sum of the squared reprojection errors, initialized with DLS.

To test the NPL, EPnP, and SDP methods, we obtained the authors' own Matlab implementations, which were either provided via e-mail request or publicly available on the web.

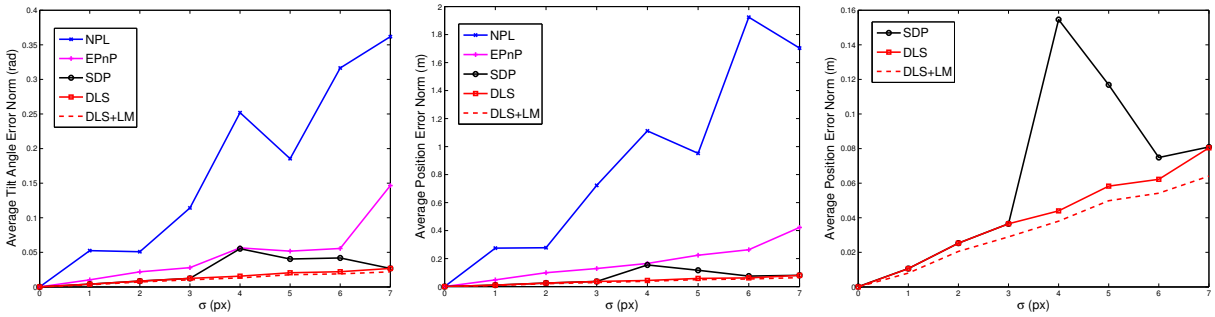
We first examine the performance of the above algorithms versus number of points. We randomly distribute points within the field of view ($45° \times 45°$) of an internally calibrated camera (focal length 600 px), at distances between 0.5 and 5.5 meters. We perturb each image measurement (point projection on the image plane) by independent zero-mean Gaussian noise ($\sigma = 1.5$ px along both u and v axes). We vary the number of points from 3 to 10, noting that for the methods which require a unique solution to work (i.e., NPL, EPnP, and SDP), we only show results for 4 or more points (when a unique solution is probable).

Figure 2 shows the results comparing the five approaches based on their average error norm computed over 100 tri-



(a) (b) (c)

Figure 2. Accuracy comparison depicted as the average error norm, over 100 trials for each number of points, for orientation 2(a) and position 2(b). The results for just SDP, DLS, and DLS+LM are depicted in 2(c).



(a) (b) (c)

Figure 3. Accuracy comparison depicted as the average error norm, over 100 trials for each value of σ, for orientation 4(a) and position 4(b). The results for just SDP, DLS, and DLS+LM are depicted in 3(c).

als. We compute the position error norm as $||{}^{S}\mathbf{p}_{G,true} - {}^{S}\mathbf{p}_{G,est}||$, while we compute the tilt-angle (orientation) error norm as $||\boldsymbol{\delta\theta}|| = 2||\tilde{\mathbf{s}}||$, where $\tilde{\mathbf{s}}$ is the CGR parameter obtained from ${}^{S}_{G}\tilde{\mathbf{C}} = {}^{S}_{G}\mathbf{C}^{T}_{true}\,{}^{S}_{G}\mathbf{C}_{est}$. We see that DLS performs consistently better than other approaches, and obtains results close to the MLE estimate (DLS-LM). The SDP method treats strictly planar scenes differently than non-planar scenes [23], by using two different SDP relaxations. However in some cases, when the points are close to a coplanar configuration, neither SDP approach provides accurate results [e.g., $n = 6$ in Fig. 2(c), the average error is larger due to a few nearly coplanar cases out of the 100 trials]. We also note that NPL is least accurate since it sometimes returns imaginary solutions (due to recovery of the original parameters after lifting). In these instances, we compute a real solution by projecting the imaginary solution back onto the real axis.

We also examine the performance of the five approaches as a function of the pixel noise. We vary the pixel noise standard deviation between $\sigma = 0$ px and $\sigma = 7$ px, noting that we only permit noise between $\pm 3\sigma$ (to prevent outliers). Figure 3 displays the results of the average error norm over 100 trials for position and orientation. We note that DLS again outperforms the existing methods and is very close to the MLE estimate (DLS-LM).

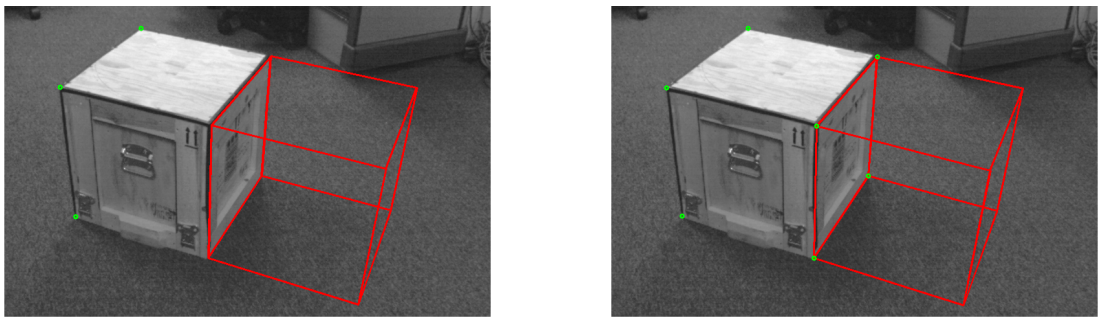
4.2. Experiments

We evaluated our method experimentally with observations of 7 known points at the corners of a cube. We computed the camera pose with each method (using 3, 4, and 7 known points), and compared the resulting pose value to the MLE estimate obtained using all 7 points. Table 1 lists the errors for orientation and position for each method.

Figure 4 depicts the visual results of the experiment. We show the back-projection of the known global points on the image as green circles, for DLS3 [Fig. 4(a)], and DLS7 [Fig. 4(b)]. In order to further validate the results visually, we also back-project a virtual box (of identical dimensions as the real box) next to the real box. Additional trials are included in the supplemental material [11].

4.3. Processing time comparison

The speed of the four direct methods was evaluated in Matlab 7.8 running on a Linux (kernel 2.6.32) computer



(a) (b)

Figure 4. The solution computed using DLS with 3 known points is depicted in 4(a), where the green circles represent the three known points back-projected onto the image using the computed transformation. 4(b) is the result obtained using DLS with 7 known points. In both cases, we also back-project a virtual cube, placed next to the real one, to aid visual verification of the result.

n-points	Ori. Error Norm (rad)	Pos. Error Norm (m)
NPL4	2.87×10^{-3}	8.67×10^{-3}
NPL7	2.12×10^{-3}	2.42×10^{-3}
EPnP4	2.49×10^{-2}	2.33×10^{-2}
EPnP7	1.24×10^{-2}	3.41×10^{-3}
SDP4	4.26×10^{-3}	9.82×10^{-3}
SDP7	3.86×10^{-4}	3.49×10^{-4}
DLS3	5.41×10^{-3}	1.02×10^{-2}
DLS4	4.28×10^{-3}	9.83×10^{-3}
DLS7	4.29×10^{-4}	3.35×10^{-4}

Table 1. The orientation and position errors for different numbers of points. Errors are computed with respect to the MLE estimate of the camera pose computed using all 7 points.

with a 2.4 GHz Intel Core 2 Duo processor. NPL and EPnP were the fastest algorithms, requiring approximately 10 ms and 5 ms, respectively, to solve a four-point problem. Our algorithm required approximately 15 ms to compute all local minima of the LS cost function using the Macaulay resultants method. The slowest approach was SDP which required approximately 200 ms to solve the semi-definite program (using SeDuMi). Since the implementations are Matlab-based and not optimized for speed, we provide these only as "ball-park" figures for performance. Part of our ongoing work is to compare the run-time of these methods using efficient C/C++ implementations.

5. Conclusion

In this work, we have presented a Direct Least-Squares (DLS) method for PnP which has several advantages compared to existing approaches. First, it is flexible in that it can handle any number of points from the minimal case of 3, to the general case of $n \geq 4$. It computes all pose solutions analytically, as the minima of a nonlinear least-squares cost function, without the need for initialization. Instead, using a reformulation of the geometric constraints, we obtain LS optimality conditions that form a system of three third-order polynomials, which are solved efficiently using the multiplication matrix.

We have validated the proposed method alongside three leading PnP algorithms as well as the MLE, both in simulation and experimentally. Compared to existing approaches, DLS is consistently more accurate, attaining performance close to the MLE. DLS is also efficient, since the order of the polynomial system that it solves is independent of the number of measurements. Lastly, in contrast to other techniques which seek to obtain a single global optimum (e.g., SDP and EPnP) DLS has the unique characteristic that it analytically computes all minima of the LS cost function.

A. Appendix

Employing the expression for $\mathbf{A}$ from (5) we have:

$$\mathbf{A}^T\mathbf{A} = \left[\begin{array}{ccc|c} 1 & & & -{}^S\bar{\mathbf{r}}_1^T \\ & \ddots & & \vdots \\ & & 1 & -{}^S\bar{\mathbf{r}}_n^T \\ \hline -{}^S\bar{\mathbf{r}}_1 & \dots & -{}^S\bar{\mathbf{r}}_n & n\mathbf{I} \end{array}\right] \tag{24}$$

where we have exploited the fact that ${}^S\bar{\mathbf{r}}_i^T{}^S\bar{\mathbf{r}}_i = 1$. Using block-matrix inversion yields

$$\left(\mathbf{A}^T\mathbf{A}\right)^{-1} = \left[\begin{array}{c|c} \mathcal{E} & \mathcal{F} \\ \hline \mathcal{G} & \mathcal{H} \end{array}\right] \tag{25}$$

$$\mathcal{E} = \mathbf{I} + \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T \\ \vdots \\ {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} \mathcal{H} \begin{bmatrix} {}^S\bar{\mathbf{r}}_1 \dots {}^S\bar{\mathbf{r}}_n \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T \\ \vdots \\ {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} \mathcal{H}$$

$$\mathcal{G} = \mathcal{H} \begin{bmatrix} {}^S\bar{\mathbf{r}}_1 \dots {}^S\bar{\mathbf{r}}_n \end{bmatrix}, \quad \mathcal{H} = \left(n\mathbf{I} - \sum_{i=1}^{n} {}^S\bar{\mathbf{r}}_i{}^S\bar{\mathbf{r}}_i^T \right)^{-1}.$$

Next, we compute the block matrices $\mathbf{U}$ and $\mathbf{V}$ in (6) by post-multiplying the above expression with $\mathbf{A}^T$, i.e.,

$$\begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T \tag{26}$$

$$\mathbf{U} = \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T & & \\ & \ddots & \\ & & {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} + \begin{bmatrix} {}^S\bar{\mathbf{r}}_1^T \\ \vdots \\ {}^S\bar{\mathbf{r}}_n^T \end{bmatrix} \mathbf{V} \tag{27}$$

$$\mathbf{V} = \mathcal{H}\left[{}^S\bar{\mathbf{r}}_1\,{}^S\bar{\mathbf{r}}_1^T - \mathbf{I} \quad \dots \quad {}^S\bar{\mathbf{r}}_n\,{}^S\bar{\mathbf{r}}_n^T - \mathbf{I}\right] \tag{28}$$

where $\mathbf{U}$ is $n \times 3n$ and $\mathbf{V}$ is $3 \times 3n$. Based on (5), (6), and (26), we compute both the scale and the translation as a function of the unknown rotation matrix:

$${}^S\mathbf{p}_G = \mathcal{H}\sum_{i=1}^{n}\left({}^S\bar{\mathbf{r}}_i\,{}^S\bar{\mathbf{r}}_i^T - \mathbf{I}\right){}^S_G\mathbf{C}\,{}^G\mathbf{r}_i \tag{29}$$

$$\alpha_i = {}^S\bar{\mathbf{r}}_i^T\left({}^S_G\mathbf{C}\,{}^G\mathbf{r}_i + {}^S\mathbf{p}_G\right). \tag{30}$$

References

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):578–589, May 2003. 1, 2, 5

[2] W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Proc. of the Int. Conf. on Numerical Mathematics*, pages 11–30, Singapore, 1988. 4

[3] D. A. Cox, J. B. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer-Verlag, New York, NY, 2nd edition, 2005. 4

[4] K. Daniilidis. Hand-eye calibration using dual quaternions. *Int. Journal of Robotics Research*, 18(3):286–298, Mar. 1999. 1

[5] J.-C. Faugére, G. Moroz, F. Rouillier, and M. S. E. Din. Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities. In *Proc. of the Int. symposium on Symbolic and Algebraic Computation*, pages 79–86, Hagenberg, Austria, July 20–23, 2008. 2

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applicatlons to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981. 1

[7] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(8):930–943, Aug. 2003. 2

[8] J. A. Grunert. Das pothenotische problem in erweiterter gestalt nebst bber seine anwendugen in der geodäsie. *Grunerts Archiv für Mathematik und Physik*, pages 238–248, 1841. Band 1. 1

[9] R. M. Haralick, H. Joo, C. nan Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1426–1446, Nov./Dec. 1989. 1, 2

[10] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int. Journal of Computer Vision*, 13(3):331–356, Dec. 1994. 1

[11] J. A. Hesch and S. I. Roumeliotis. A practical guide to DLS PnP. Technical Report 2011-001, University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Aug. 2011. 5, 6

[12] S. Hijikata, K. Terabayashi, and K. Umeda. A simple indoor self-localization system using infrared LEDs. In *Proc. of the Int. Conf. on Networked Sensing Systems*, pages 1–7, Pittsburgh, PA, June 17–19, 2009. 1

[13] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 500–507, San Diego, CA, June 4–8, 1989. 2

[14] Z. Y. Hu and F. C. Wu. A note on the number of solutions of the noncoplanar P4P problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):550–555, Apr. 2002. 2

[15] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. *Int. Journal of Computer Vision*, 74(1):3–15, Aug. 2007. 1, 2

[16] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate $O(n)$ solution to the PnP problem. *Int. Journal of Computer Vision*, 81(2):155–166, June 2008. 1, 2, 5

[17] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991. 1

[18] F. S. Macaulay. On some formulæ in elimination. *London Mathematics Society*, 35:3–27, May 1902. 4

[19] F. M. Mirzaei and S. I. Roumeliotis. Globally optimal pose estimation from line-to-line correspondences. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 5581–5588, Shanghai, China, May 9–13, 2011. 4

[20] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Trans. on Robotics*, 25(2):264–280, Apr. 2009. [Best Journal Paper Award]. 1

[21] L. Quan and Z. Lan. Linear N-point camera pose determination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):774–780, Aug. 1999. 2

[22] G. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. In *Proc. of the Int. symposium on Symbolic and Algebraic Computation*, pages 215–223, Philadelphia, PA, Aug. 3–6, 2003. 2

[23] G. Schweighofer and A. Pinz. Globally optimal $O(n)$ solution to the PnP problem for general comera models. In *Proc. of the Brittish Machine Vision Conf.*, pages 1–10, Leeds, United Kindom, Sept. 1–4, 2008. 1, 2, 5, 6

[24] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883, pages 298–372. Springer-Verlag, 2000. 2

[25] Y. Wu and Z. Hu. PnP problem revisited. *Journal of Mathematical Imaging and Vision*, 24(1):131–141, Jan. 2006. 2