Using Multi-Camera Systems in Robotics: Efficient Solutions to the NPnP Problem

Laurent Kneip, Paul Furgale, and Roland Siegwart
Autonomous Systems Lab, ETH Zurich, Switzerland

***Abstract*— This paper introduces two novel solutions to the generalized-camera exterior orientation problem, which has a vast number of potential applications in robotics: (i) a minimal solution requiring only three point correspondences, and (ii) *gPnP*, an efficient, non-iterative n-point solution with linear complexity in the number of points. Already existing minimal solutions require exhaustive algebraic derivations. In contrast, our novel minimal solution is solved in a straightforward manner using the Gröbner basis method. Existing n-point solutions are mostly based on iterative optimization schemes. Our n-point solution is non-iterative and outperforms existing algorithms in terms of computational efficiency. Our results present an evaluation against state-of-the-art single-camera algorithms, and a comparison of different multi-camera setups. It demonstrates the superior noise resilience achieved when using multi-camera configurations, and the efficiency of our algorithms. As a further contribution, we illustrate a possible robotic use-case of our non-perspective orientation computation algorithms by presenting visual odometry results on real data with a non-overlapping multi-camera configuration, including a comparison to a loosely coupled alternative.**

I. Introduction

The problem of computing the *exterior orientation* of a camera—its position and orientation given a set of correspondences between image observations and known 3D points—is one of the most fundamental in computer vision. It has a large number of potential applications such as camera calibration, augmented reality, object tracking, pose recovery, pose tracking, and visual SLAM, and thus turns out to be a key-component for vision-related tasks in robotics. However, when a camera has a restricted field of view, the solution may be poorly conditioned due to a bad distribution of observed points, or fail entirely due to lack of texture in the current viewing direction. Omnidirectional cameras perform better in this respect but the images they return are highly distorted. These are the main reasons why robotics system engineers are increasingly considering rigidly coupled multi-camera systems for mobile localization tasks. The availability of integrated algorithms that can jointly process the information from all cameras is however still limited. Such algorithms offer the advantage of handling multiple cameras as one [1], and thus allow for the reuse of standard single-camera computer vision pipelines. The present paper aims at solving part of this problematic, and presents a straight-forward extension of fundamental camera pose computation algorithms from computer vision to the non-perspective or multi-camera case, targeting a wide range of potential robotic applications.

The Perspective-n-Point (PnP) problem for a single camera originates from camera calibration [2], [3], [4], [5] and consists of recovering the camera position and orientation from n known 2D-3D correspondences. The first solutions have been presented outside the field of computer vision more than 150 years ago. Grunert (1846) [6] and Finsterwalder (1903) [7] were the first to derive minimal solutions aiming at solving the problem using three correspondences only. This problem is known in computer vision as the Perspective-*three*-Point (P3P) problem and leads to up to four solutions. Haralick et al. [8] reviewed the major direct solutions up to 1991 including the robust algorithm presented by Fischler and Bolles (1981) [3], who pointed out the importance of minimal solutions for robust hypothesize-and-test schemes when the measurements are affected by outliers (RANSAC). Different solutions to the P3P problem have been later proposed by Quan and Lan (1999) [5] and Gao et al. (2003) [9]. The most efficient solution has been presented in our previous work [10], and involves a parametrization avoiding the intermediate derivation of the point depths in the camera frame, and thus computes the camera pose in a single step.

The P3P problem is the minimal case of the PnP problem. The PnP problem was first solved by photogrammetrists (1963) [11], who also introduced the Direct Linear Transformation algorithm (DLT). The solutions of Fischler and Bolles [3], Horaud et al. [4], Quan and Lan [5], Fiore (2001) [12], and Ansar and Daniilidis (2003) [13] are other notable algorithms able to handle an arbitrary number of points. Lepetit et al. (2009) [14] present *ePnP*, the most efficient solution published to date as it is non-iterative and of linear complexity in the number of points.

The above-mentioned works are all designed for localization of a single perspective camera. In contrast, Chen and Chang (2004) [15] and Nistér and Stéwenius (2006) [16] developed minimal 3-point solutions for localization with a generalized camera leading to 8 possible solutions. This is known as the Non-Perspective-*three*-Point problem (NP3P), and the solutions are applicable to multi-camera systems. However, in both cases the derivation of the solution is not very intuitive and involves arduous algebraic reasoning and the numerical solution of an 8-th order polynomial. Regarding the Non-Perspective-n-Point problem (NPnP), a number of iterative solutions have been presented by Chen and Chang (2004) [15], Schweighofer and Pinz (2008) [17], and Tariq and Dellaert (2004) [18]. These iterative methods, however, are computationally expensive and depend on a critical initialization of the pose—either via perspective or non-perspective geometric solutions in a RANSAC scheme,

or temporal prediction in a pose tracking context. The major focus of these iterative methods thus lies on global optimization techniques rather than an improvement of the problem parametrization. A step forward in this direction was achieved by Ess et al. (2007) [19], who presented for the first time a non-iterative linear solution to the NPnP problem. However, the complexity is at least quadratic in the number of points.

The present paper starts off with a novel, intuitive parametrization of the multi-camera absolute pose computation problem, and presents a solution to the minimal NP3P case based on a straightforward application of the Gröbner basis approach. The major contribution then focuses on a novel solution to the NPnP problem with linear computational complexity in the number of points. To the best of our knowledge, this is the first non-iterative solution to the NPnP problem that achieves this level of efficiency. The paper is structured as follows: Section II outlines the synopsis of the problem. Section III presents our generalized minimal 3-point and linear n-point algorithms. In Section IV, we focus on a thorough comparison to equivalent state-of-the-art single camera models and an evaluation of different multi-camera setups. As an example, Section V shows for the first time visual odometry results on real data captured with a non-overlapping multi-camera rig, where all cameras are treated as one. Section VI finally concludes the paper.

II. Synopsis of the Multi-Camera Exterior Orientation Problem

As an example, we consider in this paper the application of multi-camera exterior orientation computation to visual odometry. Exterior orientation algorithms represent a fundamental building block of geometric keyframe based egomotion computation pipelines, where the camera position is always derived with respect to a local point cloud. Note that a keyframe in the multi-camera sense denotes an entire set of keyframes (one per camera). The relative orientation of cameras required for triangulating new points can be derived from the exterior orientation of consecutive keyframes. This means that—apart from the bootstrapping phase—all geometric computations are achieved through the sole employment of exterior orientation algorithms.

The problem we are looking at is illustrated in Figure 1. It can be abstracted into a generalized form of the P3P algorithm [10], where the origins of the unit feature observation vectors, $\mathbf{f}_i$, are displaced from the rigid body origin by known vectors, $\mathbf{v}_{i0}$. The origins of $\mathbf{f}_i$ are equivalent to camera centers, and each $\mathbf{v}_{i0}$ thus represents the position of a camera inside the body frame. The variables we are interested in are the position, $\mathbf{t}$, of the rigid body in the world frame and the rotation, $\mathbf{R}$, from the body frame to the world frame. The observed points are expressed with $\mathbf{p}_{i0}$. Following the assumption of known extrinsic camera-to-body orientations, the unit feature observation vectors, $\mathbf{f}_i$, and the displacement vectors, $\mathbf{v}_{i0}$, from the body origin are expressed in the body frame. The depth of the features is denoted with n_i. Points expressed in the world and body frame are given the superscripts w and b, respectively. Note that the use of unit feature bearing vectors $\mathbf{f}_i$ is allowed under the assumption of calibrated cameras. Furthermore, the use of bearing vectors instead of normalized coordinates provides the generality of being applicable to any optical projection system.

III. Theory

The application of absolute orientation algorithms requires two variants. First, a minimal variant that uses only three points and thus can be employed in a hypothesize-and-test scheme. Second, an n-point solution that computes an optimal pose based on the identified inlier subset. This section highlights both approaches.

A. 3-point minimal solution

From Figure 1, we can easily derive the following system of equations

$$\begin{cases} \mathbf{R}(n_1\mathbf{f}_1 + \mathbf{v}_{10}) + \mathbf{t} = \mathbf{p}^w_{10} \\ \mathbf{R}(n_2\mathbf{f}_2 + \mathbf{v}_{20}) + \mathbf{t} = \mathbf{p}^w_{20} \\ \mathbf{R}(n_3\mathbf{f}_3 + \mathbf{v}_{30}) + \mathbf{t} = \mathbf{p}^w_{30} \end{cases} \Rightarrow \begin{cases} n_1\mathbf{f}_1 - n_2\mathbf{f}_2 + \mathbf{v}_{12} = \mathbf{R}^T\mathbf{p}^w_{12} \\ n_2\mathbf{f}_2 - n_3\mathbf{f}_3 + \mathbf{v}_{23} = \mathbf{R}^T\mathbf{p}^w_{23} \\ n_3\mathbf{f}_3 - n_1\mathbf{f}_1 + \mathbf{v}_{31} = \mathbf{R}^T\mathbf{p}^w_{31} \end{cases}, \tag{1}$$

with $\mathbf{v}_{ij} = \mathbf{v}_{i0} - \mathbf{v}_{j0}$ and $\mathbf{p}_{ij} = \mathbf{p}_{i0} - \mathbf{p}_{j0}$. Note that the position of the body center, $\mathbf{t}$, is easily removed by subtracting pairwise equations.

Despite the compact look, this equation system is arduous to solve by hand. It is a multivariate polynomial equation system commonly solved via the Gröbner basis method. A good introduction to the approach can be found in [20]. The method consists of defining a monomial ordering over the polynomial terms and then iteratively generating and reducing new polynomials inside the ideal (the so-called s-polynomials) until a set of polynomials with good criteria for solvability is obtained. The most well-known application of this technique in geometric vision is the 5-point essential matrix solution of Stewénius et al. [21]. Rather than having to apply the Gröbner method for each new set of coefficients, the sequence of reductions performed by the method is often constant for a specific problem. Hence, we may solve the system using randomly chosen coefficients in a prime field and *trace* the solution offline. This trace may then be applied online using coefficients emanating from real data. The final

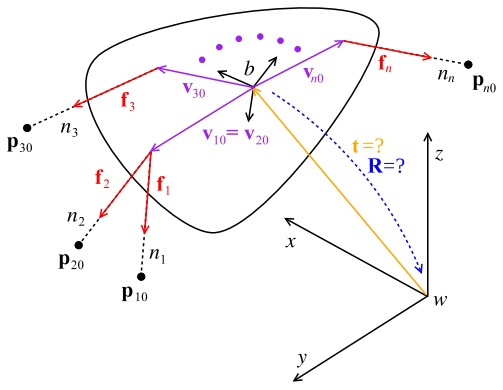


Fig. 1. Synopsis of the NPnP problem.

algorithm is fast since we do no longer need to check for all polynomial reductions, but directly generate and reduce only the necessary s-polynomials following a fixed paradigm.

The computation of a Gröbner basis can be extremely long and is typically done by machine-generated code. The complexity depends to a large extent on the initial parametrization of the problem. It is influenced by: (1) the order of the equations; (2) the number of equations; (3) the number of unknowns; and (4) the chosen monomial ordering. In the present case, the system is best solved using the Cayley rotation matrix [22] parametrization $\mathbf{R}(w_1, w_2, w_3)$ and the *grevlex* monomial ordering. This leads to a system of 9 cubic equations in 6 unknowns. The produced Gröbner basis uses 8 base monomials: w_1^2, n_3, n_2, n_1, w_3, w_2, w_1, and 1. The generated code (∼8000 lines) performs only 39 s-polynomial reductions, operating on a 48x85 matrix. The SVD of the corresponding action matrix finally leads to 8 solutions for the rotation matrix and the depths of the features. The position is afterwards derived as $\mathbf{t} = \frac{1}{3}\sum_{i=1}^{3}\mathbf{p}_{i0}^{w} - \mathbf{R}(n_i\mathbf{f}_i + \mathbf{v}_{i0})$. Wrong solutions are easily found by putting a threshold on the imaginary parts of the singular values. In practice, a unique solution is obtained by considering a fourth point, computing the reprojection error of this point for all remaining valid solutions, and finally selecting the one with the smallest error.

B. gPnP: non-iterative n-point solution with linear complexity in the number of points

As presented already by Ess et al. [19], starting from 6 points and using the standard rotation matrix parametrization $\mathbf{R} = \begin{pmatrix}\mathbf{r}_1\\ \mathbf{r}_2\\ \mathbf{r}_3\end{pmatrix} = \begin{pmatrix}r_{11} & r_{12} & r_{13}\\ r_{21} & r_{22} & r_{23}\\ r_{31} & r_{32} & r_{33}\end{pmatrix}$, equation system (1) turns into a linear problem and becomes fully constrained with at least 15 linearly independent equations and 15 unknowns. Defining $\mathbf{n} = \begin{pmatrix}n_1 & n_2 & n_3 & n_4 & n_5 & n_6\end{pmatrix}^T$, $\mathbf{i}_1 = \begin{pmatrix}-1 & 0 & 0\end{pmatrix}^T$, $\mathbf{i}_2 = \begin{pmatrix}0 & -1 & 0\end{pmatrix}^T$, and $\mathbf{i}_3 = \begin{pmatrix}0 & 0 & -1\end{pmatrix}^T$, we obtain $\mathbf{A}\mathbf{s} = \mathbf{b}$, with $\mathbf{A} =$

$$\begin{pmatrix}\mathbf{f}_1 & -\mathbf{f}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{i}_1\mathbf{p}_{12}^{wT} & \mathbf{i}_2\mathbf{p}_{12}^{wT} & \mathbf{i}_3\mathbf{p}_{12}^{wT}\\ \mathbf{f}_1 & \mathbf{0} & -\mathbf{f}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{i}_1\mathbf{p}_{13}^{wT} & \mathbf{i}_2\mathbf{p}_{13}^{wT} & \mathbf{i}_3\mathbf{p}_{13}^{wT}\\ \mathbf{f}_1 & \mathbf{0} & \mathbf{0} & -\mathbf{f}_4 & \mathbf{0} & \mathbf{0} & \mathbf{i}_1\mathbf{p}_{14}^{wT} & \mathbf{i}_2\mathbf{p}_{14}^{wT} & \mathbf{i}_3\mathbf{p}_{14}^{wT}\\ \mathbf{f}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{f}_5 & \mathbf{0} & \mathbf{i}_1\mathbf{p}_{15}^{wT} & \mathbf{i}_2\mathbf{p}_{15}^{wT} & \mathbf{i}_3\mathbf{p}_{15}^{wT}\\ \mathbf{f}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{f}_6 & \mathbf{i}_1\mathbf{p}_{16}^{wT} & \mathbf{i}_2\mathbf{p}_{16}^{wT} & \mathbf{i}_3\mathbf{p}_{16}^{wT}\end{pmatrix},$$

$\mathbf{s} = (\mathbf{n}^T\ \mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3)^T$, and $\mathbf{b} = (\mathbf{v}_{12}^T\ \mathbf{v}_{13}^T\ \mathbf{v}_{14}^T\ \mathbf{v}_{15}^T\ \mathbf{v}_{16}^T)^T$. The complexity of this solution is not easily extendable to an arbitrary number of points since the dimensionality of the solution space is increasing by one each time we add another feature. This leads to an increasingly complicated computation of the pseudo-inverse of the matrix, which is at least of complexity $O(mn^2)$ for $m \times n$ matrices and $m > n$.

Instead, we extend ePnP—the $O(n)$ PnP solution presented in Lepetit et al. [14]—to the NPnP problem. We call our algorithm *gPnP*. The basic idea consists of expressing all n points as a weighted sum of four control points. We define the first control point to be the centroid of the point cloud $\mathbf{c}_0^w = \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_{i0}^w$. We then compute the principal components of the point cloud by singular value decomposition $\mathbf{USV}^*$ of the point data matrix. The other control points are then defined as $\mathbf{c}_j^w = \mathbf{c}_0^w + s_j\mathbf{u}_j, j \in \{1,2,3\}$, where s_j is the j-th singular value and $\mathbf{u}_j$ the j-th column of $\mathbf{U}$. Each point is then defined by a weighted sum of control points $\mathbf{p}_i^w = \sum_{j=0}^{3}\alpha_{ij}\mathbf{c}_j^w$. The weighting factors are invariant with respect to the coordinate frame, so we also have $\mathbf{p}_i^b = \sum_{j=0}^{3}\alpha_{ij}\mathbf{c}_j^b$. Moreover, we also have $\mathbf{p}_i^b = n_i\mathbf{f}_i + \mathbf{v}_{i0}$. Substitution leads to

$$\sum_{j=0}^{3}\alpha_{ij}\begin{pmatrix}c_{xj}^b\\ c_{yj}^b\\ c_{zj}^b\end{pmatrix} = n_i\begin{pmatrix}f_{xi}\\ f_{yi}\\ f_{zi}\end{pmatrix} + \begin{pmatrix}v_{xi0}\\ v_{yi0}\\ v_{zi0}\end{pmatrix}. \tag{2}$$

We now can pick any row (at best in the dimension where the bearing vector coordinates are highest) in order to find an expression for n_i. Let's say this is the third dimension. We obtain $n_i = (\sum_{j=0}^{3}\frac{\alpha_{ij}}{f_{zi}}c_{zj}^b) - \frac{v_{zi0}}{f_{zi}}$. Backsubstitution into the two first equations leads to

$$\begin{cases}\sum_{j=0}^{3}(\alpha_{ij}f_{zi}c_{xj}^b - \alpha_{ij}f_{xi}c_{zj}^b) = f_{zi}v_{xi0} - f_{xi}v_{zi0}\\ \sum_{j=0}^{3}(\alpha_{ij}f_{zi}c_{yj}^b - \alpha_{ij}f_{yi}c_{zj}^b) = f_{zi}v_{yi0} - f_{yi}v_{zi0}\end{cases} \tag{3}$$

For n points, this turns into a linear system as follows

$$\begin{pmatrix}\mathbf{D}_{10} & \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{D}_{13}\\ \mathbf{D}_{20} & \mathbf{D}_{21} & \mathbf{D}_{22} & \mathbf{D}_{23}\\ \cdot & \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot & \cdot\\ \mathbf{D}_{n0} & \mathbf{D}_{n1} & \mathbf{D}_{n2} & \mathbf{D}_{n3}\end{pmatrix}\begin{pmatrix}\mathbf{c}_0^b\\ \mathbf{c}_1^b\\ \mathbf{c}_2^b\\ \mathbf{c}_3^b\end{pmatrix} = \begin{pmatrix}\mathbf{E}(\mathbf{f}_1 \times \mathbf{v}_{10})\\ \mathbf{E}(\mathbf{f}_2 \times \mathbf{v}_{20})\\ \cdot\\ \cdot\\ \mathbf{E}(\mathbf{f}_n \times \mathbf{v}_{n0})\end{pmatrix}, \tag{4}$$

where $\mathbf{D}_{ij} = \alpha_{ij}\begin{pmatrix}f_{zi}\mathbf{I}_2 & \mathbf{J}\mathbf{f}_i\end{pmatrix}$, $\mathbf{I}_2$ is the 2-by-2 identity matrix, $\mathbf{J} = \begin{pmatrix}-1 & 0\\ 0 & -1\\ 0 & 0\end{pmatrix}$, and $\mathbf{E} = \begin{pmatrix}0 & 1 & 0\\ -1 & 0 & 0\end{pmatrix}$. This problem is of the form $\mathbf{A}\mathbf{s} = \mathbf{b}$, and the least-squares solution is given by $\mathbf{s} = \mathbf{A}^+\mathbf{b}$, where $\mathbf{A}^+$ denotes the Moore-Penrose pseudo-inverse of matrix $\mathbf{A}$ and $\mathbf{s} = (\mathbf{c}_0^{bT}\ \mathbf{c}_1^{bT}\ \mathbf{c}_2^{bT}\ \mathbf{c}_3^{bT})^T$. Having always a solution space dimensionality of 12, this solution is of linear complexity in the number of points. $\mathbf{A}$ has full rank in the noise-free case. It has been verified that this is guaranteed as long as all points do not originate from the same camera. However, similar to the ePnP algorithm, the situation becomes more complicated under noise. In this case, $\mathbf{A}$ might be rank deficient and the general least-squares solution becomes

$$\hat{\mathbf{s}}_{LS} = \mathbf{A}^+\mathbf{b} + \left[\mathbf{I}_n - \mathbf{A}^+\mathbf{A}\right]\mathbf{y}, \tag{5}$$

where $\mathbf{y}$ is an arbitrary vector in $\mathbb{R}_n$. In other words, the solution is the sum of $\mathbf{A}^+\mathbf{b}$ and a varying number of right-most nullspace vectors (the ones corresponding to the smallest singular values) multiplied by some unknown factors. In our experimental section, we consider noise levels up to 10 pixels in standard deviation, and we verify that a good maximum number of right-most nullspace vectors to consider is 5. This leads to a total number of 6 possible cases. $\mathcal{N}_i(\mathbf{A})$ describes the i-th right-most nullspace vector of $\mathbf{A}$.

- case 0: The solution is simply given by $\mathbf{s}_0 = \mathbf{A}^+\mathbf{b}$. This is the case if no noise is added to the measurements.
- case 1: The solution is given by $\mathbf{s}_1 = \mathbf{A}^+\mathbf{b} + \lambda_1\mathcal{N}_1(\mathbf{A})$. The unknown factor λ_1 is found by imposing the constraint that the distances between the control points expressed in the world and body frames need to be

preserved. Using $c_{ij} = (\mathbf{c}_i^w - \mathbf{c}_j^w)^T \cdot (\mathbf{c}_i^w - \mathbf{c}_j^w)$ and 3 distance constraints, we obtain

$$\begin{cases} \left\| (\mathbf{I}_3 \quad -\mathbf{I}_3 \quad \mathbf{0}_3 \quad \mathbf{0}_3) \ \mathbf{s}_1(\lambda_1) \right\|^2 = c_{01} \\ \left\| (\mathbf{I}_3 \quad \mathbf{0}_3 \quad -\mathbf{I}_3 \quad \mathbf{0}_3) \ \mathbf{s}_1(\lambda_1) \right\|^2 = c_{02} \\ \left\| (\mathbf{I}_3 \quad \mathbf{0}_3 \quad \mathbf{0}_3 \quad -\mathbf{I}_3) \ \mathbf{s}_1(\lambda_1) \right\|^2 = c_{03} \end{cases}, \tag{6}$$

which results in 3 quadratic equations in the monomials λ_1^2, λ_1, and 1. This is enough equations to find a unique solution. However, this also highlights the main difficulty of the gPnP compared to the ePnP algorithm. We no longer end up with only even powers of the unknown λ_1. This means that the gPnP solution is substantially more complicated than ePnP, even for a single added nullspace vector. When considering more nullspace vectors, the number of monomials increases drastically, which means that straightforward solution techniques using linearization can not be used. Instead, we propose to consistently use the Gröbner basis method for finding the linear combination factors of nullspace vectors. In the present case, this leads to a 5×3 matrix and one s-polynomial reduction.

- case 2: The solution with 2 nullspace vectors is given by $\mathbf{s}_2 = \mathbf{A}^+\mathbf{b} + \sum_{g=1}^{2} \lambda_g \mathcal{N}_g(\mathbf{A})$. The three distance constraints from (6) with $\mathbf{s}_1$ substituted by $\mathbf{s}_2(\lambda_1, \lambda_2)$ are still sufficient for finding the unknown linear combination factors λ_1 and λ_2. The Gröbner matrix in this case is 10×6 and solved via 8 s-polynomial reduction steps.
- case 3: In the case of 3 nullspace vectors the solution is $\mathbf{s}_3 = \mathbf{A}^+\mathbf{b} + \sum_{g=1}^{3} \lambda_g \mathcal{N}_g(\mathbf{A})$. Now we need the additional distance constraint, c_{12}, with $\mathbf{s}_1$ substituted by $\mathbf{s}_3(\lambda_1, \lambda_2, \lambda_3)$, namely

$$\left\| (\mathbf{0}_3 \quad \mathbf{I}_3 \quad -\mathbf{I}_3 \quad \mathbf{0}_3) \ \mathbf{s}_3(\lambda_1, \lambda_2, \lambda_3) \right\|^2 = c_{12}. \tag{7}$$

  The resulting system of 4 equations is solved using a Gröbner matrix of 15×18 and 59 s-polynomial reduction steps.
- case 4: $\mathbf{s}_4 = \mathbf{A}^+\mathbf{b} + \sum_{g=1}^{4} \lambda_g \mathcal{N}_g(\mathbf{A})$. The solution requires the consideration of the additional distance constraint, c_{13},

$$\left\| (\mathbf{0}_3 \quad \mathbf{I}_3 \quad \mathbf{0}_3 \quad -\mathbf{I}_3) \ \mathbf{s}_4(\lambda_1, ..., \lambda_4) \right\|^2 = c_{13}. \tag{8}$$

  The system of five equations leads to a Gröbner matrix of 25×37 and is solved via 240 s-polynomial reduction steps.
- case 5: $\mathbf{s}_5 = \mathbf{A}^+\mathbf{b} + \sum_{g=1}^{5} \lambda_g \mathcal{N}_g(\mathbf{A})$. The solution requires the consideration of the last available distance constraint c_{23}

$$\left\| (\mathbf{0}_3 \quad \mathbf{0}_3 \quad \mathbf{I}_3 \quad -\mathbf{I}_3) \ \mathbf{s}_5(\lambda_1, ..., \lambda_5) \right\|^2 = c_{23}. \tag{9}$$

  The system of six equations leads to a Gröbner matrix of 44×80 and is solved via 936 s-polynomial reduction steps.

The Gröbner basis computations are considerably exhaustive, especially for an increased number of nullspace vectors. This means that they do not deliver optimal linear combination factors under noise. In order to tackle this problem, cases 1 to 5 are followed by a polishing scheme that consists of a nonlinear optimization of λ_i over the six distance-conservation constraints of the original control points. This is not to be confused with a batch optimization of the reprojection error of all points into all cameras; The complexity of this operation is independent of the number of involved points, n, and thus approximately constant. $\mathbf{R}$ is finally derived by control point alignment: $(\mathbf{U}, \mathbf{D}, \mathbf{V}) = \text{SVD}(\sum_{j=0}^{3} (\mathbf{c}_j^b - \bar{\mathbf{c}}^b)(\mathbf{c}_j^w - \bar{\mathbf{c}}^w)^T) \Rightarrow \mathbf{R} = \mathbf{V}\mathbf{U}^T$. The position of the body center is then given by $\mathbf{t} = \frac{1}{4}\{(\sum_{j=0}^{3} \mathbf{c}_j^w) - \mathbf{R}(\sum_{j=0}^{3} \mathbf{c}_j^b)\}$. Knowing the pose of the body, we can now properly select the best solution based on the smallest reprojection error of the control points. The reprojection errors are evaluated as a function of the dot-products between unit bearing vectors from the body frame. Finally, the depth of each point can be retrieved by reusing the weight factors in order to recompute the points in the body frame, and then transforming them into the corresponding camera frame: $n_i = ||(\sum_{j=0}^{3} \alpha_{ij} \mathbf{c}_j^b) - \mathbf{v}_{i0}||$.

IV. Simulation Results

This section presents experimental results on the algorithms presented in Section III. After introducing the experiment outline, we present analyses of noise resilience, numerical accuracy, and computational efficiency. A comparison of both the minimal and the n-point solutions to their most efficient, state-of-the-art single camera equivalents is included.

A. Experiment outline

Besides being applicable to non-central cameras that do not have a single effective viewpoint, the non-perspective pose algorithms offer the advantage of being applicable to a rigidly coupled system of multiple central cameras. In this way, the camera system can be treated as one single camera, allowing the reuse of a single-camera structure-from-motion pipeline. With this concept in mind, we evaluate different multi-camera setups and compare the results to perspective localization with a single camera using state-of-the-art methods. As a reference minimal solution for a single camera, we use the novel P3P algorithm presented in [10], which shows superior behavior in accuracy and efficiency compared to alternative solutions. The reference for n-point perspective localization is given by the efficient non-iterative linear-complexity ePnP algorithm [14] from which we also derived the basic idea for our gPnP approach. The disambiguation of the multiple solutions returned by the minimal algorithms is done each time by considering a fourth point and picking the solution that leads to its smallest reprojection error. Note that a direct comparison between central and non-central algorithms is not possible, since the latter turn out to become degenerate in case all $\mathbf{v}_{i0}$ turn out to be the same.

The different setups are illustrated in Figure 2. All virtual cameras have a distance of 1m to the fictive body center and camera calibration parameters taken from a real camera. The focal length equals to 400, the resolution to 640×480, the principal point to $(320, 240)$, and the field of view to $77^\circ \times$

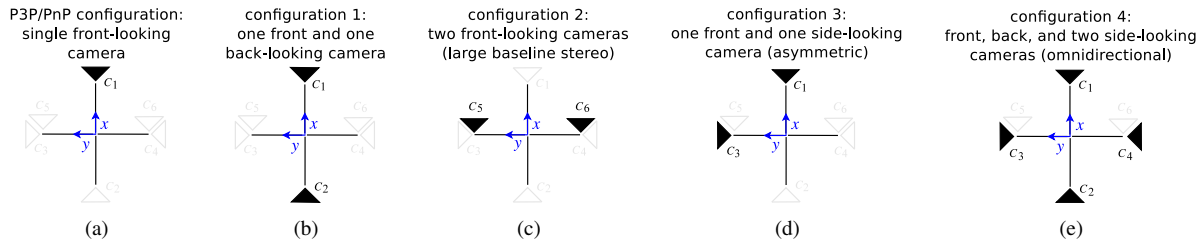


Fig. 2. The five configurations that are explored in the simulation experiments.

62°. For each experimental iteration, we generate 50 random visible points in each camera with a uniformly varying depth between 10m and 20m. The single camera algorithms are evaluated by considering only the points from camera C1. The generalized multi-camera algorithms are evaluated using four different configurations: 1) C1 and C2: two cameras facing opposite directions, 2) C5 and C6: two cameras facing the same direction (classical stereo configuration with large baseline), 3) C1 and C3: asymmetric configuration with two cameras facing orthogonal directions, and 4) C1, C2, C3, and C4: four cameras facing in all four directions. The ground-truth pose for each experimental run is simply kept at $\mathbf{t} = \mathbf{0}$ and $\mathbf{R} = \mathbf{I}_3$. All experiments have been performed on a standard 2.8GHz Intel Core 2 Duo CPU.

B. Noise resilience

In order to evaluate the resilience to noise, we add Gaussian noise with zero mean and standard deviations reaching from 0 to 10 pixels to our measurements in the image plane. We execute 10,000 iterations for each combination of noise level, algorithm, and camera-configuration. The resulting plots show the mean and median norm of translation and rotation error vectors. For a rotation error, we use the rotation angle, in radians, between the solution and the ground truth. Because the true orientation is at identity, we can compute this as the angle component of the axis/angle decomposition of R.

Figure 3 shows the results for the minimal algorithms. It can be seen that configuration 2 (two front-looking cameras) behaves worst and very similar to the single front-looking camera. Configuration 1 (symmetric) provides lower median error than configuration 3 (asymmetric). However, configuration 1 shows a slightly elevated mean translation error, which leads to the conclusion that the translation computation is less robust in this case. Configuration 4 with four cameras pointing in all directions shows best behavior.

As indicated in Figure 4, the behavior for the n-point solutions is not much different. We can see that, again, configuration 2 behaves worst, this time even slightly worse than the ePnP algorithm. The median error of configurations 1, 3 and 4 are much smaller with configuration 4 again being the best configuration. Looking at the mean errors, we can see that configurations 1, 2 and 3 are deficient in terms of robustness, and mostly lead to even higher errors than the ePnP, especially in the rotational degrees of freedom. However, configuration 4 again behaves best for the mean error too.

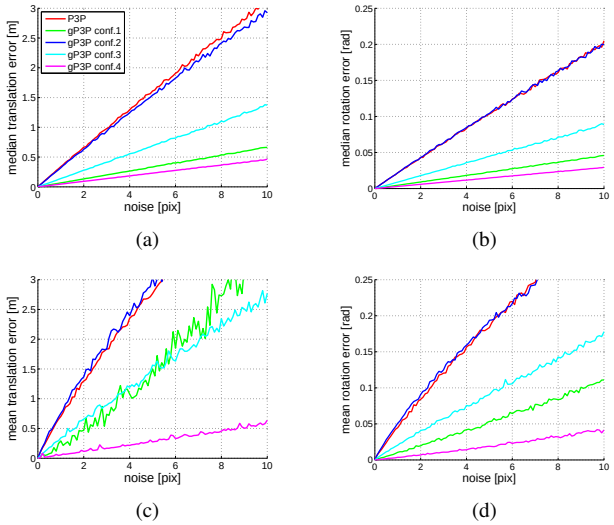


Fig. 3. Mean and median error of translation (a, c) and rotation (b, d) for the non-linear minimal algorithms. As indicated in Figure 2, the reference P3P algorithm (P3P, red) [10] uses only one camera, whereas our new NP3P algorithm (gP3P) is tested with four different multi-camera configurations.

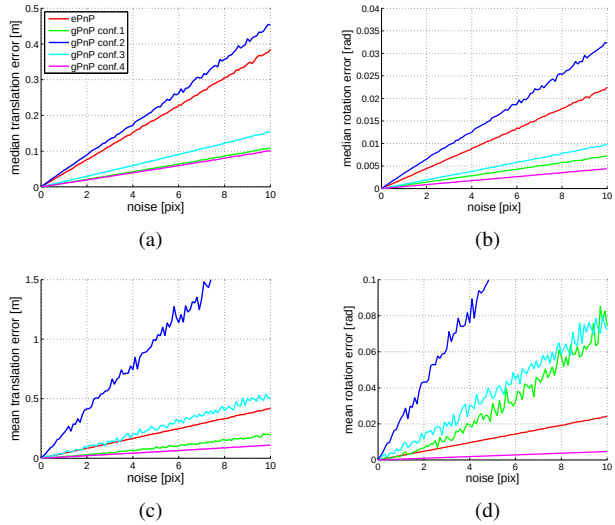
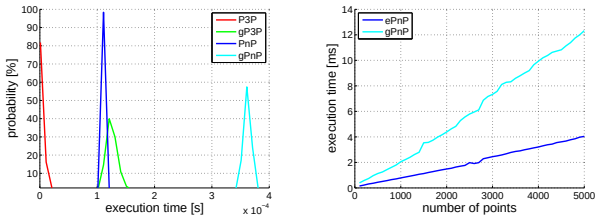


Fig. 4. Mean and median error of translation (a, c) and rotation (b, d) for the linear n-point algorithms. As indicated in Figure 2, the reference PnP algorithm (ePnP, red) [14] uses only one camera, whereas our new NPnP algorithm (gPnP) is tested with four different multi-camera configurations.

C. Efficiency and numerical precision

Figure 5(a) shows the execution time of the minimal solutions and the n-point algorithms with 100 points. The generalized algorithms have higher complexity and thus also higher execution times. They, however, still remain real-time capable. Looking at Figure 5(b), we observe the major benefit of the ePnP and the gPnP algorithms, namely that their execution time stays linear as a function of the number of used points. The higher slope indicates that the per-point time consumption is higher for the gPnP algorithm. This is however not problem, since we have to bear in mind that the algorithm also solves a different, more complicated problem. Moreover, the low execution time of 12 ms for 5000 points proves that the gPnP algorithm is perfectly suited for real-time applications. Table I shows that the numerical accuracy (median error for zero noise) of the gP3P algorithm is worse than P3P. This is clearly related to the difference in computational complexity. The numerical accuracy of the gPnP algorithm, however, remains competitive with the ePnP algorithm. The reason is that the linear-combination-factor-polishing compensates for errors originating from the more-complicated Gröbner basis solutions.



(a) (b)

Fig. 5. Comparison of the execution time of the different algorithms (a) and execution time of the ePnP and gPnP algorithms in function of the number of points (b).

TABLE I

NUMERICAL ACCURACY (MEDIAN ERROR). [M] OR [RAD]

	P3P	gP3P	ePnP	gPnP
trans.	1.7392e-014	1.1829e-011	2.6246e-014	7.1394e-014
rot.	8.1712e-016	2.2107e-013	1.1096e-015	6.4384e-016

V. CASE STUDY: VISUAL ODOMETRY

In order to underline the potential of the presented algorithms in robotics, we integrated them into a structure-from-motion pipeline as outlined in Section II. We extended the framework presented in our earlier work [23], which operates on two non-overlapping cameras facing opposite directions, as indicated in Figure 6. The performance is evaluated on the same dataset as in [23], which has been collected by moving the camera rig in a room equipped with a Vicon motion capture system offering ground truth data. The algorithm works as follows:

- Bootstrapping is done using the exact pipeline presented in our previous work, which executes single camera structure-from-motion in each of the two cameras separately, and then fuses the individual relative displacements in order to determine the visual scale factors and recover results in metric scale.
- Once the scale is converged, we switch to our novel algorithms treating all cameras as one. As illustrated in Figure 7, we achieve stable egomotion computation with our novel approach.
- The point clouds from both sides are unified into a single one, and we use the non-perspective 3-point and n-point algorithms presented in this paper for robust exterior orientation computation.
- Each time the cameras experience enough median disparity between feature correspondences, a new point cloud is triangulated. This procedure is followed by bundle adjustment over two keyframes only, meaning 2 pairs of views including the rigid-coupling constraint in between.

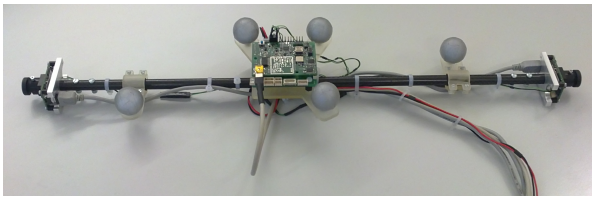


Fig. 6. Stereo-camera rig with two cameras facing opposite directions used in our experiments. The cameras are synchronized and capture WVGA images at 10Hz.

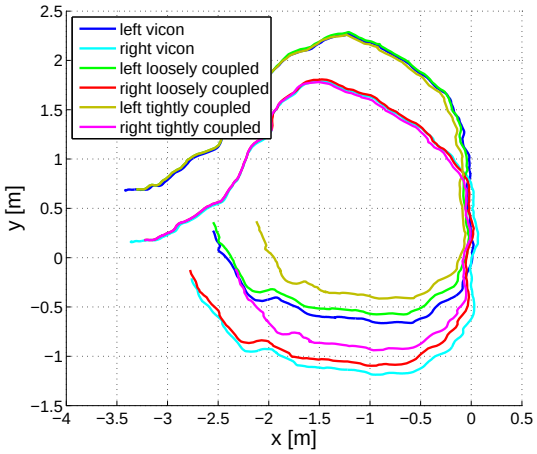


Fig. 7. Top view on visual odometry results obtained on a real indoor dataset. The plot shows a comparison between ground truth, results obtained using the IMU-supported framework in [23], and results obtained using a minimal single-camera like vision-only pipeline based on the proposed algorithms.

Note that the higher drift is not a sign of bad performance, but rather the natural consequence of our minimal approach. The original loosely coupled framework presented in [23] employs the visual odometry algorithm presented in [24] extended by windowed bundle adjustment over 10 keyframes. Moreover, it employs IMU information in order to retrieve relative rotation between successive camera frames, and performs explicit scale propagation and scale estimation over multiple keyframes in order to achieve low drift and

stable metric scale recovery. In contrast, the presented tightly coupled pipeline is very minimalistic and operates with point clouds optimized over two keyframes only. It also does not use any IMU information. Metric scale is implicitly recovered by including the rigid coupling constraint into our minimal non-linear optimization. However, as explained in [23], the observability of metric scale in the non-overlapping case is affected by motion singularities. This also explains the drift in the results: The metric scale is at times badly observable, which leads to a drifting translation magnitude during non-linear optimization. Similar to [23], the observability of the metric scale can easily be improved in a future implementation that performs joint windowed bundle adjustment over both cameras and multiple keyframes, including the rigid coupling constraint.

In conclusion, the fact that we can robustly compute the egomotion of the rig in a minimal, "single-camera like" fashion disposing of a number of elements such as scale estimation, scale propagation, windowed bundle adjustment, and IMU information certainly proves the potential given by our novel algorithms, and leads to significantly lower implementational and computational complexity. To the best of our knowledge, this marks the first visual odometry results on real data treating cameras with non-overlapping fields of view as one. Due to the generality of the employed geometric algorithms, the presented approach can be used for an arbitrary number of cameras.

VI. Conclusion

In this paper, we presented two novel solutions to the non-perspective camera pose computation problem. The first solution is minimal and solved in a straightforward manner using the original parametrization and the Gröbner basis method. The second solution is a linear-complexity and non-iterative n-point algorithm for generalized cameras outperforming existing solutions in terms of efficiency. Our simulation results show that using these algorithms in conjunction with rigidly coupled multi-camera systems easily outperforms well-established single camera solutions. The best results are obtained with camera-systems pointing into all viewing directions. The algorithms are robust and real-time compliant. Finally—as one of numerous potential applications in robotics—, we show a minimal single-camera-like visual odometry pipeline for multi-camera systems based on the presented algorithms. This leads for the first time to real results implementing the generalized concept of "treating multiple cameras as one" on a non-overlapping multi-camera rig.

ACKNOWLEDGMENT

The research leading to these results has received funding from the EU project V-Charge (FP7-269916), and the Swiss National Science Foundation under agreement n. 200021 125017/1.

References

[1] R. Pless. Using many cameras as one. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 587–593, Madison, WI, USA, 2003.

[2] M.A. Abidi and T. Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(5):534–538, 1995.

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

[4] R. Horaud, B. Conio, and O. Leboulleux. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics, and Image Processing*, 47(1):33–44, 1989.

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

[6] J.A. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendungen in Geodäsie. In *Grunerts Archiv für Mathematik und Physik*, 1841.

[7] S. Finsterwalder and W. Scheufele. *Das Rückwärtseinschneiden im Raum*. Verlag Herbert Wichmann, Berlin, Germany, 1937.

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

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

[10] 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 the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, USA, 2011.

[11] I. Sutherland. Sketchpad: A man machine graphical communications system, 1963. Technical Report 296, MIT Lincoln Laboratories.

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

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

[14] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate O(n) solution to the pnp problem. *International Journal of Computer Vision (IJCV)*, 81(2):578–589, 2009.

[15] Chu-Song Chen and Wen-Yan Chang. On pose recovery for generalized visual sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(7):848–861, 2004.

[16] D. Nistér and H. Stewénius. A minimal solution to the generalized 3-point pose problem. *Journal of Mathematical Imaging and Vision (JMIV)*, 27(1):67–79, 2006.

[17] G. Schweighofer and A. Pinz. Globally optimal O(n) solution to the pnp problem for general camera models. In *Proceedings of the British Machine Vision Conference (BMVC)*, Leeds, UK, 2008.

[18] Sarah Tariq and Frank Dellaert. A multi-camera 6-DOF pose tracker. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 296–297, Arlington, VA, USA, 2004.

[19] A. Ess, A. Neubeck, and L. Van Gool. Generalised linear pose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 22.1–22.10, Warwick, UK, 2007.

[20] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[21] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.

[22] Arthur Cayley. About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic. *Reine Angewandte Mathematik*, 32, 1846.

[23] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart. Real-Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, USA, 2012.

[24] L. Kneip, M. Chli, and R. Siegwart. Robust real-time visual odometry with a single camera and an IMU. In *Proceedings of the British Machine Vision Conference (BMVC)*, Dundee, Scotland, 2011.