A unified PnP approach that minimizes geometric distance with global optimality (not stuck in local minima like iterative method), which outputs multiple solutions when necessary.Compared to OPnP, UPnP derives a geometric error term in the cost function.

GPnP referred to Kneip's work to extend EPnP to general non-central case, whereas, zheng's GPnPf problem

difference from OPNP:
1) geometric error
2) use quaternion constraints
to reduce the possible
solutions in Grobnor base decomposition, reducing the solutions from 81 to 16.

UPnP: An Optimal $O(n)$ Solution to the Absolute Pose Problem with Universal Applicability

Laurent Kneip[1], Hongdong Li[1], and Yongduek Seo[2]

[1] Research School of Engineering, Australian National University, Australia
[2] Department of Media Technology, Sogang University, Korea

Abstract. A large number of absolute pose algorithms have been presented in the literature. Common performance criteria are computational complexity, geometric optimality, global optimality, structural degeneracies, and the number of solutions. The ability to handle minimal sets of correspondences, resulting solution multiplicity, and generalized cameras are further desirable properties. This paper presents the first PnP solution that unifies all the above desirable properties within a single algorithm. We compare our result to state-of-the-art minimal, non-minimal, central, and non-central PnP algorithms, and demonstrate universal applicability, competitive noise resilience, and superior computational efficiency. Our algorithm is called *Unified PnP* (UPnP).

Keywords: PnP, Non-perspective PnP, Generalized absolute pose, linear complexity, global optimality, geometric optimality, DLS.

1 Introduction

The Perspective-n-Point (PnP) algorithm is a fundamental problem in geometric computer vision. Given a certain number of correspondences between 3D world points and 2D image measurements, the problem consists of fitting the absolute position and orientation of the camera to the measurement data. Our contribution is a PnP solution that unifies most desirable properties within one and the same algorithm. We call our method *Unified PnP* (UPnP), and the benefits are summarized as follows:

- *Universal applicability*: UPnP is applicable to both central and non-central camera systems (i.e. generalized cameras). In contrast, existing methods are often designed exclusively for the central case (e.g. [6], [17]).
- *Optimality*: Similarly to [16], we employ the object space error. However, we do not rely on convex relaxation techniques, which is why our solution is theoretically guaranteed to return a geometrical optimum. Likewise, UPnP is guaranteed to find the global optimum.
- *Linear complexity*: Similarly to many recent works (e.g. [11]), our algorithm solves the PnP problem with $O(n)$ (linear) complexity in the number of points. From a practical point of view, the $O(n)$-complexity argument is

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 127–142, 2014.

stronger than simple algebraic linearity of the solution. Despite of returning comparable results to [16] in terms of noise resilience, our method does not employ any iterative parts and therefore turns out to be **faster by about two orders of magnitude**.
- *Completeness*: The proposed solution is complete in the sense of returning multiple solutions. It therefore supports the minimal case, as well as other possible ambiguous-pose situations [15]. Moreover—in contrast to recent works such as [6] and [17]—our algorithm still does not return any spurious solutions. The returned number of solutions is precisely equal to the maximum number of solutions in the minimal case. Similarly to [17], we furthermore exploit 2-fold symmetry in the space of quaternions in order to avoid solution duplicates.
- *Homogeneity*: We parametrize rotations in terms of unit-quaternions—a non-minimal parametrization of rotations that is free of singularities and leads to homogeneous accuracy.

UPnP unifies all listed properties. It is inspired by several recent works, and—using first-order optimality conditions—solves the problem by a closed-form computation of all stationary points of the sum of squared object space errors. The conceptual innovations lie in the avoidance of a Lagrangian formulation, a geometrically consistent application of the Gröbner basis methodology, and a general technique to circumvent 2-fold symmetry in quaternion-based parametrizations.

The paper is structured as follows: The related work is presented in the following subsection. Section 2 then outlines the core theoretical contributions behind our approach. Section 3 finally contains a detailed comparison to existing algorithms, show-casing state-of-the-art noise resilience at superior computational efficiency.

1.1 Related Work

While an exhaustive review of the vast literature on the PnP problem goes beyond the extend of this introduction, we nonetheless note that—after more than 170 years of related research—still new solutions with interesting properties keep being discovered. The most recent advancement in the minimal case—the P3P problem—was presented in 2011 [9]. The P3P problem uses 3 correspondences and returns at most 4 solutions. One of the major recent achievements in the PnP case then consists of proving that the problem can be solved accurately in linear time with respect to the number of correspondences. The first solution to provide accurate results under linear complexity is EPnP [11] (2009). This algorithm is computationally efficient, however depends on a special variant with only 3 control points in the planar case, minimizes only an algebraic error, and fails in situations of solution multiplicity (i.e., in situations of pose-ambiguity such as for instance the minimal case). The first $O(n)$-successor that succeeds in all these criteria was presented in 2011 and is called DLS [6]. It performs measurement data compression in linear time and then computes all stationary points of the sum of squared object-space errors in closed-form, using polynomial resultant techniques. It achieves a least-squares geometric error in linear time,

Table 1. Comparison of properties of various $O(n)$ PnP algorithms. Note, however, that [16] contains an iterative convex relaxation part, which means that the effective computational complexity of SOS is in fact unbounded (hence the brackets).

	EPnP	DLS	OPnP	SOS	GPnP	UPnP
reference	[11]	[6]	[17]	[16]	[8]	this
year	2009	2011	2013	2008	2013	2014
central cameras	✓	✓	✓	✓		✓
non-central cameras				✓	✓	✓
geometric optimality		✓		✓		✓
linear complexity	✓	✓	✓	(✓)	✓	✓
multiple solutions		✓	✓			✓
singularity-free rotation param.	✓		✓	✓	✓	✓

however employs a singularity-affected rotation matrix parametrization [2]. The most recent contribution in $O(n)$-complexity PnP solvers is then given by the OPnP algorithm [17] (2013), which essentially replaces the Cayley parametrization by the singularity-free non-unit quaternion parametrization, thus leading to improved accuracy. They also exploit 2-fold symmetry in the solver, thus avoiding the duality of quaternion solutions. Although they achieve very good accuracy, we still note that—from a theoretical point of view—their algorithm again falls back to an algebraic error.

An interesting fact is that—while searching for all stationary points—the DLS and OPnP algorithms find 27 and 40 solutions, respectively. In other words, despite of using more than the minimum amount of information, those algorithms return far more solutions than a minimal solver. It is true that many of the stationary points can be neglected because they are either complex or local maxima/saddle points, but still the computation at least intermediately reaches a seemingly too high level of complexity.

More recently, people have also started to consider the generalized PnP problem, which consists of estimating the position of a non-central or generalized camera given correspondences between arbitrary non-central rays in the camera frame and points in the world frame. [3], [14], and [8] present minimal solvers for the generalized PnP problem, proving that 3 correspondences are still enough and that the maximum number of possible solutions corresponds to 8. Regarding the generalized PnP problem, there has been less progress to date. [5] presents the first linear complexity solution, however minimizes only an algebraic error. It fails in situations of multiple solutions (e.g. the minimal case), and depends on a special variant for the planar case. The linear complexity solution presented in [16] (SOS) minimizes a geometric error, however again fails in the mentioned special cases, and depends on a computationally intensive, iterative convex relaxation technique. Yet another algebraic $O(n)$ solution to the generalized PnP problem has been discovered in 2013 [8] (GPnP), and essentially consists of a generalization of the EPnP algorithm to the non-central case. It thus comes with similar drawbacks.

Table 1 shows a summary of all relevant algorithms and their properties, including the proposed UPnP algorithm.

2 Theory

We now proceed to the theoretical part of our method. We start by recalling the geometry of the absolute pose problem in the generalized case, which covers the classical perspective situation as well. We then derive a cost-function in the space of quaternions reflecting the geometrical error as a function of absolute orientation. All local minima are found by a closed-form computation of all stationary points. This is achieved by computing a Gröbner basis over the first-order optimality conditions and the quaternion unit-norm constraint. We also present an alternative unit-norm constraint allowing us to exploit 2-fold symmetry in quaternion-space, thus reducing the number of solutions by a factor of two. We finally obtain an ideal number of solutions, and also introduce an easy way to verify second-order optimality and polish the final result.

2.1 Geometry of the Absolute Pose Problem

Let $\mathbf{p}_i \in \mathbb{R}^3$ describe a point in the world frame, $\mathbf{R} \in SO^3$ the rotation from the world frame to the camera frame, and $\mathbf{t} \in \mathbb{R}^3$ the position of the world origin seen from the camera frame. The measurements of $\mathbf{p}_i$ in the camera frame are given by non-central rays expressed by $\alpha_i \mathbf{f}_i + \mathbf{v}_i$, where $\mathbf{v}_i \in \mathbb{R}^3$ represents a point on the ray, $\mathbf{f}_i \in \mathbb{R}^3$ the normalized direction vector of the ray, and α_i the depth. The situation is explained in Figure 1. The non-central projection equation results to

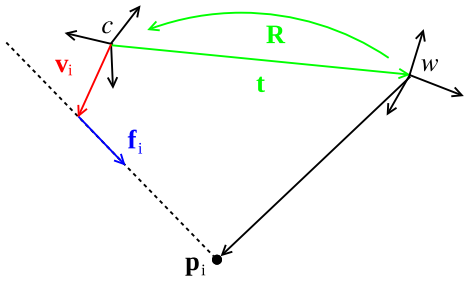

Fig. 1. Point measurements of a generalized camera

$$\alpha_i \mathbf{f}_i + \mathbf{v}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t}, i = 1, \ldots, n. \tag{1}$$

$\mathbf{R}$, $\mathbf{t}$, and α_i are the parameters to be computed from the inputs $\mathbf{f}_i$, $\mathbf{v}_i$, and $\mathbf{p}_i$. In case the generalized camera is given by a multi-camera system, the $\mathbf{v}_i$'s are simply the positions of the respective camera centers inside the main common frame. In this case, some $\mathbf{f}_i$'s may obviously have the same $\mathbf{v}$ for reflecting the non-centrality of their measurement. For a central camera, $\mathbf{v}_i = \mathbf{0},\ i = 1, \ldots n$.

Let $\mathbf{I}$ be the 3×3 identity matrix. We can stack all constraints into

$$\begin{bmatrix} \mathbf{f}_1 & & -\mathbf{I} \\ & \cdots & \vdots \\ & \mathbf{f}_n & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & & \\ & \cdots & \\ & & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{bmatrix} - \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_m \end{bmatrix} \Leftrightarrow \mathbf{A}\mathbf{x} = \mathbf{W}\mathbf{b} - \mathbf{w}. \tag{2}$$

2.2 Derivation of the Objective Function

The derivation of the objective function is based on the work of [6]:

- We start by applying block-wise matrix inversion to eliminate the unknown translation and point depths from the projection equations. *(Result 1)*
- The obtained expressions are then transformed into a residual that notably corresponds to the object-space error. *(Definition 1)*
- Factorization of the rotation matrix in the sum of squared object space errors then results in a fourth-order energy function of the quaternion variables. The measurement data inside this expression is compressed in form of linear-complexity summation terms. *(Results 2 & 3)*

We now proceed to the details of this derivation.

Result 1: $\mathbf{x}$ appears linearly in (2) and can be eliminated by

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{W}\mathbf{b} - \mathbf{w}) = \begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}(\mathbf{W}\mathbf{b} - \mathbf{w}). \tag{3}$$

The pseudo-inverse of $\mathbf{A}$ is hence partitioned such that the depth parameters are a function of $\mathbf{U} = \begin{bmatrix}\mathbf{u}_1^T \dots \mathbf{u}_n^T\end{bmatrix}^T$, and the translation is a function of $\mathbf{V}$. Back-substitution results in the rotation-only projection equation

$$\left[\mathbf{u}_i^T(\mathbf{W}\mathbf{b} - \mathbf{w})\right]\mathbf{f}_i + \mathbf{v}_i = \mathbf{R}\mathbf{p}_i + \mathbf{V}(\mathbf{W}\mathbf{b} - \mathbf{w}). \tag{4}$$

Proof: The symbolic solution of $\mathbf{x}$ is mainly based on [6]. The derivation of the symbolic form of $\mathbf{U}$ and $\mathbf{V}$ is based on a) $\|\mathbf{f}_i\| = 1$, b) the Schur-complement, and c) block-wise matrix inversion. It results in

$$\mathbf{V}_{3\times 3n} = [\mathbf{V}_1, ..., \mathbf{V}_n], \text{ with}$$

$$\mathbf{V}_i = \mathbf{H}[\mathbf{f}_i\mathbf{f}_i^T - \mathbf{I}] \in \mathbb{R}^{3\times 3} \quad i = 1, ..., n, \text{ and} \tag{5}$$

$$\mathbf{H}_{3\times 3} = \left(n\mathbf{I} - \sum_{i=1}^{n}\mathbf{f}_i\mathbf{f}_i^T\right)^{-1} \tag{6}$$

$$\mathbf{U}_{n\times 3n} = \begin{bmatrix}\mathbf{f}_1^T & & \\ & \dots & \\ & & \mathbf{f}_n^T\end{bmatrix} + \begin{bmatrix}\mathbf{f}_1^T\\ \vdots \\ \mathbf{f}_n^T\end{bmatrix}\mathbf{V} = \begin{bmatrix}\mathbf{u}_1^T\\ \vdots \\ \mathbf{u}_n^T\end{bmatrix}, \text{ with}$$

$$\mathbf{u}_i^T = [\mathbf{u}_{i1}^T, \dots, \mathbf{u}_{in}^T]_{1\times 3n}, \quad i = 1, ..., n, \text{ and}$$

$$\mathbf{u}_{ij}^T = \mathbf{f}_i^T\delta(i,j) + \mathbf{f}_i^T\mathbf{V}_j \in \mathbb{R}^{1\times 3}, \quad i, j = 1, ..., n. \tag{7}$$

$\mathbf{u}_i^T$ represents row i of $\mathbf{U}$, and $\mathbf{u}_{ij}^T$ represents the 1×3 element of $\mathbf{U}$ in row i and column $3j$. We obtain $\alpha_i = \mathbf{u}_i^T(\mathbf{W}\mathbf{b} - \mathbf{w})$ and $\mathbf{t} = \mathbf{V}(\mathbf{W}\mathbf{b} - \mathbf{w})$, and back-substitution in (1) yields the rotation-only constraint (4).[1] ■

[1] It is worth noting here that the DLS mechanism is the only one to solve for the linear elements (i.e. depth and translation) in a homogeneous way. While this might be irrelevant for the central case, where we can assume that the z-coordinate in the camera frame is bigger than 1, there is no guarantee on any coordinate in the generalized camera situation. In the non-central case, the presented resolution therefore has better accuracy than the ones in [11] and [17].

Definition 1: The residual of an estimate for $\mathbf{R}$ is the *object space error*

$$\begin{aligned}\eta_i &= \left[\mathbf{u}_i^T(\mathbf{W}\mathbf{b}-\mathbf{w})\right]\mathbf{f}_i + \mathbf{v}_i - \mathbf{R}\mathbf{p}_i - \mathbf{V}(\mathbf{W}\mathbf{b}-\mathbf{w}) \\ &= \sum_{j=1}^{n}\mathbf{u}_{ij}^T\mathbf{R}\mathbf{p}_j\mathbf{f}_i - \sum_{j=1}^{n}\mathbf{V}_j\mathbf{R}\mathbf{p}_j - \mathbf{R}\mathbf{p}_i - \sum_{j=1}^{n}\mathbf{u}_{ij}^T\mathbf{v}_j\mathbf{f}_i + \sum_{j=1}^{n}\mathbf{V}_j\mathbf{v}_j + \mathbf{v}_i. \end{aligned} \quad (8)$$

Result 2: The residual vector of a correspondence can be expressed as $\eta_i = \mathcal{A}_i\mathbf{s} + \beta_i$, where the elements of $\mathbf{s}$ are quadratic functions of the quaternion variables, and $\mathcal{A}_i$ and β_i depend on the measurements only, and can be computed with linear complexity in the number of correspondences.

Proof: By substituting (7) in (8) and resolving the summations over terms including $\delta(i,j)$, we arrive at

$$\eta_i = \mathbf{f}_i^T\mathbf{R}\mathbf{p}_i\mathbf{f}_i - \mathbf{R}\mathbf{p}_i + \mathbf{f}_i^T\left[\sum_{j=1}^{n}\mathbf{V}_j\mathbf{R}\mathbf{p}_j\right]\mathbf{f}_i - \sum_{j=1}^{n}\mathbf{V}_j\mathbf{R}\mathbf{p}_j - \mathbf{f}_i^T\mathbf{v}_i\mathbf{f}_i + \mathbf{v}_i - \mathbf{f}_i^T\mathcal{J}\mathbf{f}_i + \mathcal{J}, \quad (9)$$

where $\mathcal{J} = \sum_{j=1}^{n}\mathbf{V}_j\mathbf{v}_j$ does not depend on any unknowns or i anymore, so it can be computed ahead with linear complexity (just like $\mathbf{H}$). All elements that do not depend on $\mathbf{R}$ can be summarized in

$$\beta_i = -(\mathbf{f}_i^T\mathbf{v}_i\mathbf{f}_i - \mathbf{v}_i + \mathbf{f}_i^T\mathcal{J}\mathbf{f}_i - \mathcal{J}) = -(\mathbf{f}_i\mathbf{f}_i^T - \mathbf{I})(\mathbf{v}_i + \mathcal{J}) \in \mathbb{R}^3. \quad (10)$$

We then adopt the singularity-free unit-quaternion parametrization $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$ such that $q_0^2 + q_1^2 + q_1^2 + q_3^2 = 1$. $\mathbf{R}$ in function of $\mathbf{q}$ is given by

$$\mathbf{R} = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_1q_2+2q_0q_3 & q_0^2-q_1^2+q_2^2-q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_2q_3+2q_0q_1 & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}. \quad (11)$$

$\mathbf{R}\mathbf{p}_i$ is a 3-vector of polynomials, each one having quadratic monomials in the quaternion parameters. Grouping those monomials in

$$\mathbf{s} = [q_0^2, q_1^2, q_2^2, q_3^2, q_0q_1, q_0q_2, q_0q_3, q_1q_2, q_1q_3, q_2q_3]^T \in \mathbb{R}^{10}, \quad (12)$$

we obtain $\mathbf{R}\mathbf{p}_i = \Phi(\mathbf{p}_i)_{3\times 10}\mathbf{s}$, where Φ is given by

$$\Phi(\mathbf{p}_i) = \begin{pmatrix} p_{ix} & p_{ix} & -p_{ix} & -p_{ix} & 0 & 2p_{iz} & -2p_{iy} & 2p_{iy} & 2p_{iz} & 0 \\ p_{iy} & -p_{iy} & p_{iy} & -p_{iy} & -2p_{iz} & 0 & 2p_{ix} & 2p_{ix} & 0 & 2p_{iz} \\ p_{iz} & -p_{iz} & -p_{iz} & p_{iz} & 2p_{iy} & -2p_{ix} & 0 & 0 & 2p_{ix} & 2p_{iy} \end{pmatrix}. \quad (13)$$

Using (13) and (10) in (9) results in

$$\eta_i = \mathbf{f}_i^T\Phi(\mathbf{p}_i)\mathbf{s}\mathbf{f}_i - \Phi(\mathbf{p}_i)\mathbf{s} + \mathbf{f}_i^T\left[\sum_{j=1}^{n}\mathbf{V}_j\Phi(\mathbf{p}_j)\right]\mathbf{s}\mathbf{f}_i - \left[\sum_{j=1}^{n}\mathbf{V}_j\Phi(\mathbf{p}_j)\right]\mathbf{s} + \beta_i, \quad (14)$$

and defining $\mathcal{G} = \sum_{j=1}^{n}\mathbf{V}_j\Phi(\mathbf{p}_j)$—another linear complexity term that can be computed ahead—we finally obtain

$$\begin{aligned}\eta_i =& \mathbf{f}_i^T \Phi(\mathbf{p}_i)\mathbf{s}\mathbf{f}_i - \Phi(\mathbf{p}_i)\mathbf{s} + \mathbf{f}_i^T \mathcal{G}\mathbf{s}\mathbf{f}_i - \mathcal{G}\mathbf{s} + \beta_i \\ =& \left[(\mathbf{f}_i\mathbf{f}_i^T - \mathbf{I})(\Phi(\mathbf{p}_i) + \mathcal{G})\right]\mathbf{s} + \beta_i = \mathcal{A}_i\mathbf{s} + \beta_i. \end{aligned} \quad \blacksquare \quad (15)$$

Result 3: The squared scalar residual for the i-th measurement is given by

$$\epsilon_i = \eta_i^T \eta_i = \tilde{\mathbf{s}}^T \begin{bmatrix} \mathcal{A}_i^T \mathcal{A}_i & \mathcal{A}_i^T \beta_i \\ \beta_i^T \mathcal{A}_i & \beta_i^T \beta_i \end{bmatrix} \tilde{\mathbf{s}}, \quad (16)$$

where $\tilde{\mathbf{s}} = \begin{bmatrix}\mathbf{s}^T 1\end{bmatrix}^T$. This notably corresponds to the squared object-space error (i.e. the squared spatial orthogonal distance between point and ray)[2]. The total error over all measurements finally results in

$$E = \sum_i \epsilon_i = \tilde{\mathbf{s}}^T \left\{ \sum_i \begin{bmatrix} \mathcal{A}_i^T \mathcal{A}_i & \mathcal{A}_i^T \beta_i \\ \beta_i^T \mathcal{A}_i & \beta_i^T \beta_i \end{bmatrix} \right\} \tilde{\mathbf{s}} = \tilde{\mathbf{s}}^T \mathbf{M} \tilde{\mathbf{s}}. \quad (17)$$

Since each $\mathcal{A}_i$ and β_i can be computed in $O(1)$ (assuming that $\mathbf{H}$, $\mathcal{J}$, and $\mathcal{G}$ are computed ahead), $\mathbf{M}$ is computed in $O(n)$. We call this error the compressed generalized object space error.

Proof: By induction. ■

2.3 Universal, Closed-Form Least-Squares Solution

The energy E is always positive, and its minimization corresponds to solving the generalized PnP problem with minimal geometric error. Following the concept presented in [6], the first-order optimality conditions constitute a system of polynomial equations that allows us to compute all stationary points in closed-form and constant time. They are given by the 4 equations

$$\frac{\partial E}{\partial q_j} = 2\tilde{\mathbf{s}}^T \mathbf{M} \cdot \frac{\partial \tilde{\mathbf{s}}}{\partial q_j} = 0, \; j = 0, \ldots, 3. \quad (18)$$

It is easy to recognize that (17) reduces to the central formulation presented in [6] in case $\mathbf{v}_i = \mathbf{0}\ \forall i \in \{1, \ldots, n\}$. Moreover, since all elements of $\mathbf{s}$ are quadratic in q_i, any $k \cdot \mathbf{q}_0, k \in \mathbb{R}$ represents a valid solution to the problem if $\mathbf{q}_0$ is also a solution. We avoid this infinity of solutions in the central case by adding the unit-norm constraint $\mathbf{q}^T\mathbf{q} - 1 = 0$. The solutions are finally computed using the Gröbner basis approach.

Our solver leads to 16 solutions only, which is substantially less than the 81 solutions reported in [17] (without considering 2-fold symmetry for the moment), and less than the 27 solutions reported in [6] despite of using a quaternion parametrization. The reason lies in the way we search for the Gröbner basis. Readers familiar with the procedure might recall that the Gröbner basis method

[2] The object-space error corresponds to the moment distance and has proven to perform very well for pose estimation. Good alternatives are given by angular (geodesic) distance, and the reprojection error. The latter one, however, does not make sense in the generalized camera situation, which does not bare a planar projective subspace.

requires to first solve the problem in a finite prime field, where exact zero cancellations are taking place[3]. $\mathbb{Z}_p$—the field of integers modulo a large prime number p—is a popular choice. The typical way consists of drawing random values in $\mathbb{Z}_p$ for the coefficients of all polynomials, and then proceeding to the Gröbner basis derivation. This might work in any situation where the polynomial coefficients are independent. In the present problem, however, this is not the case, and we can reduce the size of the Gröbner basis by chosing coefficients that inherently reflect the geometry of the problem. This requires to setup the entire random problem in $\mathbb{Z}_p$, which is done as follows:

- Chose random values in $\mathbb{Z}_p$ for $\mathbf{q}$, $\mathbf{t}$, $\mathbf{v}_i$, and $\mathbf{p}_i$.
- Derive the rotation matrix $\mathbf{R}$ and the measurement vectors $\mathbf{f}_i$.
- Derive the coefficients of the polynomials by applying Section 2.2 in $\mathbb{Z}_p$.

Beware that—by chosing random values in $\mathbb{Z}_p$—neither $\mathbf{q} \in \mathbb{Z}_p^4$ nor $\mathbf{f}_i \in \mathbb{Z}_p^3$ fulfill the unit-norm constraint unless we apply the square root in $\mathbb{Z}_p$. As an alternative, we simply derive $\mathbf{R}$ from $\mathbf{q}$ using the non-unit quaternion parametrization, and apply a modified version of the symbolic block-wise inversion of $(\mathbf{A}^T\mathbf{A})$ that accepts non-unit-norm $\mathbf{f}_i$ (presented in the supplemental material). Note that these changes only concern the derivation of the random coefficients before computing the Gröbner basis, we constantly use the unit-quaternion parametrization in the objective system of equations.

The norm constraint on the quaternion is not needed in the non-central case. A wrong quaternion-scale only scales the point cloud, and it seems intuitive that non-central measurements are no longer invariant with respect to such changes. Interestingly, including the same set of equations than in the central case (i.e. including the norm constraint) still in combination with properly posed problems in $\mathbb{Z}_p$ again reduces the number of solutions from 81 to 16. As can be observed, the reduction from 81 to 16 solutions is caused by different modifications in the central and the non-central case. While this might be just a coincidence given by analogies in the corresponding algebraic varieties, it requires a deeper investigation going beyond the scope of this paper.

In conclusion, we generate a solver from a properly-posed generalized P3P problem in $\mathbb{Z}_p$ as outlined above. We use first-order optimality conditions and the unit-norm constraint, and the obtained compact generalized P3P solver works for all scenarios, including P3P, PnP, and generalized PnP. We use our own Gröbner basis solver generator, which follows the idea presented in [10].

2.4 Comparison to a Lagrangian Formulation

The reader might ask why we did not chose a Lagrangian formulation with unit-norm constraint. We verified by experimentation that, in both the central and the non-central case, the number of solutions falls back to 80—even if chosing geometrically consistent coefficients. This is natural, since the Lagrangian formulation eventually computes the projections orthogonal to the level lines of all 80 stationary points onto the unit sphere in the space of quaternions.

[3] For details about the Gröbner basis method, the reader is referred to [4] and [10].

Our formulation is better than the Lagrangian one. In the unconstrained central case, the infinitly many solutions of $\frac{\partial E}{\partial q_j} = 0$ lie on one-dimensional varieties that correspond to radial lines intersecting with the origin. Constrained solutions are obtained by intersecting those varieties with the unit-norm sphere. It is intuitively clear that the gradient of the norm constraint and the gradient of (17) are orthogonal in the intersection points, which means that the Lagrangian multipliers have to be zero for any correct solution. In the non-central case, we again exploit the speciality of our optimization problem that the constraints appear to be fulfilled exactly by 16 of our 81 stationary points. This is clear from the fact that the number of solutions reduces from 81 to 16 if adding the unit-norm constraint. With $\frac{\partial E}{\partial q_j} = 0$, the partial derivatives of the Lagrangian by the quaternion variables therefore result in $\lambda q_j = 0$, where λ represents the Lagrangian multiplier. Since all quaternion variables can never be simultaneously zero, this implies that $\lambda = 0$.

We proved that λ ideally has to be zero, and thus that the Lagrangian formulation implicitly transitions to ours. Instead of computing the projection of a large variety (i.e. all stationary points), we provide a closed-form solution for a smaller variety (i.e. those 16 stationary points that idealy fulfill the norm constraint exactly). Moreover, our formulation leads to a substantially easier elimination template as well as a 5 times smaller action matrix (note that the theoretical complexity of an Eigen decomposition is $O(n^3)$, which means that in our case the action matrix decomposition is up to 125 times faster).

2.5 Elimination of Two-Fold Symmetry

Similar to [17], we employ the technique outlined in [1] to eliminate double roots in our polynomial equation system. The general idea consists of using only monomials with even total degree in order to create new equations in the Macaulay matrix. The pre-condition however consists of having monomials with either only even or only uneven total degrees in the initial equations. We do not yet fulfill this condition because the first-order optimality conditions only contain uneven total degrees, while the norm constraint contains only even ones. The condition is met by applying a small trick. We consider the squared unit-norm constraint $(\mathbf{q}^T\mathbf{q} - 1)^2 = 0$ depicted for 2D in Figure 2 instead of the standard unit-norm constraint. It can be observed that this function is stationary at $\mathbf{q} = \mathbf{0}$ and when the norm constraint is fulfilled. Adding all first-order derivatives of the squared norm constraint therefore results in the following system of 8 third-order equations which contains only uneven total degrees in the monomials

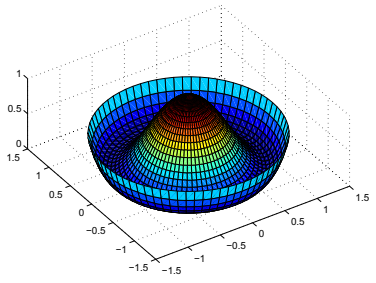

Fig. 2. The squared norm constraint in 2D

$$\begin{cases} \tilde{\mathbf{s}}^T \mathbf{M} \cdot \frac{\partial \tilde{\mathbf{s}}}{\partial q_j} = 0, \ j = 0, \ldots, 3 \\ (\mathbf{q}^T \mathbf{q} - 1) q_j = 0, \ j = 0, \ldots, 3 \end{cases}. \tag{19}$$

Since there is one additional solution now ($\mathbf{q} = \mathbf{0}$), the size of the action matrix for this system becomes 17×17. Applying the technique for eliminating 2-fold symmetry however reduces the number of solutions to 8. The size of our final elimination template shrinks to 141×149, which is substantially smaller than all elimination templates mentioned in [17]. The action matrix has a size of 8×8 only, leading to very efficient Eigenvector decomposition. The result is a drastic reduction in execution time of geometric error minimizers. Moreover, we note that the number of solutions now elegantly agrees with the maximum number of solutions for the generalized P3P problem.

2.6 Second-Order Optimality and Root Polishing

The above method only computes complex stationary points of (17). Disambiguation is done by checking the energy $\mathbf{E}$. Besides, one can easily solve the chirality ambiguity by checking the sign of the resulting depth parameters. However, there exist further sanity checks that do not depend on the number of correspondences. For instance, the magnitude of the imaginary part of each Eigenvalue from the Action matrix decomposition should be very small. In order to also verify second-order optimality, we apply another small trick. Let $\mathbf{R}_0(\mathbf{q}_0)$ be a solution of (17). We can easily compensate for this rotation by replacing all $\mathbf{p}_i$ by $\mathbf{R}_0 \mathbf{p}_i$, and recomputing $\mathbf{M}$. The stationary point now lies at identity rotation. The well-conditioned 3D subspace of rotations around $\mathbf{q}_0 = [1\ 0\ 0\ 0]^T$ is now entered by simply switching to the Cayley rotation parametrization $\mathbf{c} = [c_1\ c_2\ c_3]^T$. $\mathbf{s}$ as a function of the Cayley parameters is given by

$$\mathbf{s} = \frac{1}{1 + c_1^2 + c_2^2 + c_3^2} [1, c_1^2, c_2^2, c_3^2, c_1, c_2, c_3, c_1 c_2, c_1 c_3, c_2 c_3]^T \in \mathbb{R}^{10}. \tag{20}$$

It is almost trivial and very efficient to compute the 3×1 Jacobian J_E and the 3×3 Hessian H_E of (17) around $\mathbf{c} = \mathbf{0}$. This allows to easily verify the presence of a local minimum by checking for positive-definitness of $H_E|_{\mathbf{c}=\mathbf{0}}$, as well as perform a single Newton step for root pollishing which is given by $\delta_{\mathbf{c}} = -(H_E|_{\mathbf{c}=\mathbf{0}})^{-1} \cdot J_E|_{\mathbf{c}=\mathbf{0}}$. More information is provided in the supplemental material.

3 Experimental Evaluation

In this section, we compare UPnP to both central and non-central state-of-the-art PnP solvers using simulation experiments. We reuse the experimental evaluation toolbox from [17], and only extend it by an additional algorithm for the non-central case, plus the proposed UPnP algorithm. We start by looking at the central case, and include experiments on regular, planar, near-singular,

and minimal situations. Near-singular configurations are defined as world point distributions with low variance. We then show a comparative study for the non-central case, again including both the n-point and the minimal situation. We conclude the section by evaluating the computational efficiency. Due to space limitations, we constrain our results to the rotation errors only. However, the conclusions wouldn't change if we would instead evaluate the translational error. The absolute rotation error between the ground truth rotation $\mathbf{R}_{\text{true}}$ and the estimated rotation $\mathbf{R}$ is given by $\epsilon_{\text{rot}} = max_{k\in\{1,2,3\}} \cos^{-1}(\mathbf{r}_{k,\text{true}}^T \mathbf{r}_k)$, where $\mathbf{r}_{k,\text{true}}$ and $\mathbf{r}_k$ are the k-th column of $\mathbf{R}_{\text{true}}$ and $\mathbf{R}$, respectively. This error metric might not be standard, but we use it in order to remain consistent with [17].

3.1 The Central Case

In the central case, we create random experiments by assuming a virtual perspective camera with focal length 800, and add different levels of Gaussian noise to the measurements in the image plane. Up to 20 random world points are then generated with varying distribution depending on the type of experiment. In the normal case, the points are distributed such that they lie in the range $[-2, 2] \times [-2, 2] \times [4, 8]$ in the camera frame, and then transformed into the world frame by assuming a random transformation. In the planar case, they are picked in the range $[-2, 2] \times [-2, 2]$ in the plane $z = 0$ in the world frame, and then transformed into the camera frame using a random transformation. In the quasi-singular case, they are again defined in the camera frame with a distribution of $[1, 2] \times [1, 2] \times [4, 8]$.

The comparison algorithms in the central case are the exact same algorithms than the ones used in [17]. They are given by EPnP+GN (or EPnP in the planar case) [11], RPnP [12], DLS [6] (plus its non-degenerate version DLS+++), SDP [16], LHM [13] (plus the planar variant SP+LHM), and OPnP itself [17]. We name the SDP algorithm here in agreement with [17], but note however that the original, more accurate name of this method is SOS. As indicated in Figures 3 and 4, our algorithm leads to state-of-the-art noise resilience in all situations. Figure 4 also shows a comparison to [9], proving that the minimal case is still accurately solved.

The careful reader might notice that we plot only the median error in the minimal, planar, and quasi-singular cases. The reason is that the universal applicability and the superior computational efficiency naturally lead to reduced robustness in special situations. As clearly underlined by the low median errors, the solver is however able to generally find a good solution in any special situation too. Failures are easily pruned when all sanity checks are enabled. For the present experiments, however, we configured UPnP to always return at least one solution.

3.2 The Non-central Case

In the non-central case, we assume a multi-camera system with 4 virtual spherical cameras with focal length 800, and add noise in the tangential plane of each

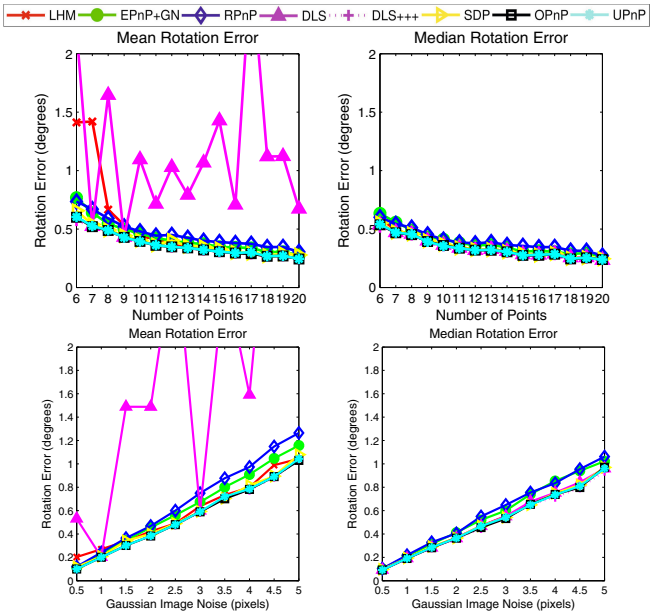

Fig. 3. Comparison of various central state-of-the-art PnP solvers w.r.t. varying point numbers (1st row, noise level=2 pixels) and varying noise levels (2nd row, n=10) in case of normal 3D points

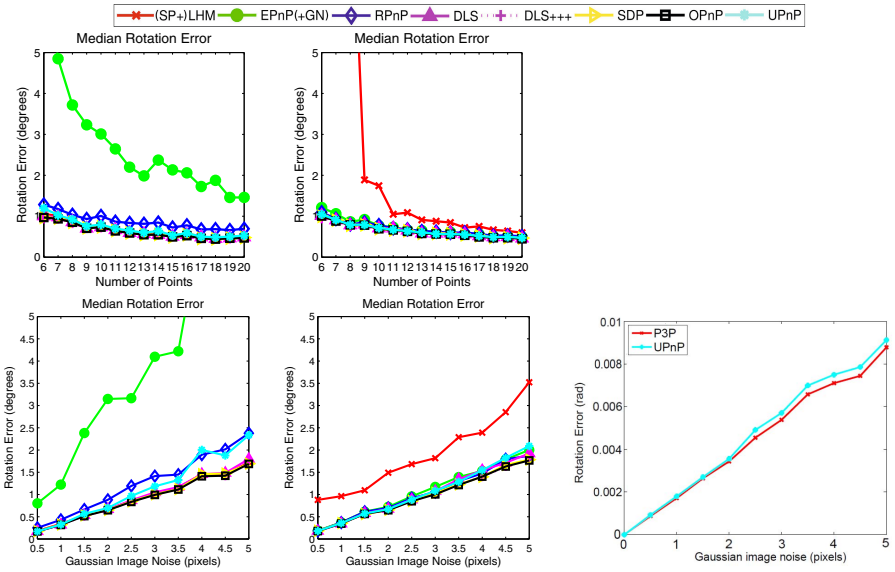

Fig. 4. Comparison of various central state-of-the-art PnP solvers w.r.t. varying point numbers (1st row, noise level=2 pixels) and varying noise levels (2nd row, n=10) in the case of planar (1st column) and near singular (2nd column) point configurations. The last figure indicates a comparison to the state-of-the-art minimal solver.

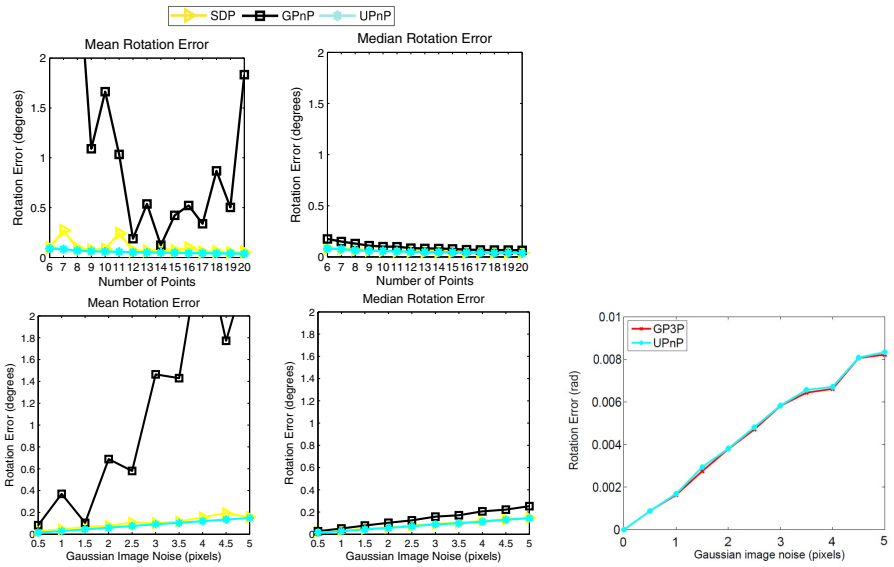

Fig. 5. Comparison of various noncentral state-of-the-art PnP solvers w.r.t. varying point numbers (1st row, noise level=2 pixels) and varying noise levels (2nd row, n=10) in case of normal 3D points. The last figure indicates a comparison to a minimal solver

measurement. The points are now always defined in the world frame, and the camera to world transformation is picked such that the distance between the camera and the world origin does not exceed 2. The world points are picked such that they have an average distance between 4 and 8 from the world frame origin, and are evenly distributed in all directions.

We compare our algorithm against SDP [16] and GPnP [8]. SDP (i.e. SOS) is the only alternative algorithm that is also applicable to both the central and the non-central case. As indicated in Figure 5, UPnP maintains state-of-the-art noise resilience in the non-central case too, slightly outperforming SDP. The figure also shows a comparison to the minimal solver of [8], proving that the minimal case is still accurately solved.

3.3 Computational Efficiency

Figure 6 illustrates the computational efficiency, highlighting the clear advantage of our method. For 100 points, our method outperforms OPnP by a factor of 10, and SDP—the only general alternative that can handle both the central and the non-central case—by a factor of 150. We have to emphasize that the core part of our algorithm is implemented in a mex-file. Some algorithms, such as EPnP, have a more efficient pendant in C++ too, but we sticked here to a baseline implementation. All experiments have been executed on an Intel Core 2 Duo with 2.8 GHz.

Note: Our algorithm is publically available within the open-source library OpenGV[7], and all results can easily be reproduced.

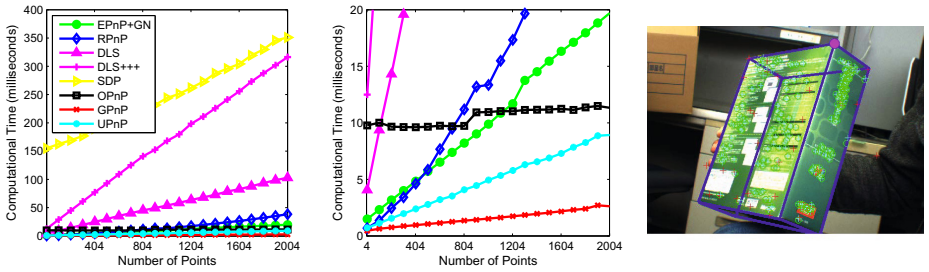

Fig. 6. The first plot shows a comparison of the computational efficiency of all algorithms. The second plot shows a zoomed-in version to clarify the comparison between the most efficient algorithms. Despite the fact that UPnP is geometrically optimal and completely general, the computational efficiency stays among the fastest ones. The last figure shows an image that is augmented by the contours of a box after recomputing the pose of the camera from SIFT feature correspondences to a reference image with known point depths.

3.4 Results on Real Data

We again repeat the experiments from [17], and recompute the pose of the camera in front of a box from matched SIFT feature correspondences. An augmented image with the contours of the box is indicated in Figure 6, showing the similarly visually pleasing results.

4 Conclusion

The scientific relevance of the presented material is given by the fact that we provide for the first time a completely general and highly computationally efficient solution to one of the most fundamental problems in geometric vision. We present a non-iterative minimization of a geometric error in linear time, and are able to handle the minimal, non-minimal, central, and non-central cases, as well as any special situations in which multiple solutions can appear. With an execution time of only a couple of ms for hundreds of points, we outperform the state-of-the art generalized geometric error minimizer by about two orders of magnitude, while having improved noise resilience. Besides our generalized formulation, the conceptual cornerstones of our method that lead to a reduced number of solutions are a geometrically consistent application of the Gröbner basis method, as well as the avoidance of Lagrangian multipliers in the special case of optimization problems that are known to fulfill the constraints exactly in the noise-free case. We furthermore provide an alternative to L2-norm constraints with only uneven terms, which potentially eases the general exploitation of p-fold symmetries in polynomial solvers.

Acknowledgment. The research leading to these results has received funding from ARC grants DP120103896 and DP130104567. We furthermore want to thank Stergios Roumeliotis, Joel Hesch, and Erik Ask for their supportive feedback.

References

1. Ask, E., Yubin, K., Astrom, K.: Exploiting p-fold symmetries for faster polynomial equation solving. In: Proceedings of the International Conference on Pattern Recognition (ICPR), Tsukuba, Japan (2012)
2. Cayley, A.: 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)
3. Chen, C.S., Chang, W.Y.: On pose recovery for generalized visual sensors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26(7), 848–861 (2004)
4. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3rd edn. Undergraduate Texts in Mathematics. Springer-Verlag New York, Inc., Secaucus (2007)
5. Ess, A., Neubeck, A., Van Gool, L.: Generalised linear pose estimation. In: Proceedings of the British Machine Vision Conference (BMVC), pp. 22.1–22.10. Warwick, UK (2007)
6. Hesch, J.A., Roumeliotis, S.I.: A Direct Least-Squares (DLS) Method for PnP. In: Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain (2011)
7. Kneip, L., Furgale, P.: OpenGV: A Unified and Generalized Approach to Real-Time Calibrated Geometric Vision. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hongkong (2014)
8. Kneip, L., Furgale, P., Siegwart, R.: Using Multi-Camera Systems in Robotics: Efficient Solutions to the NPnP Problem. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany (2013)
9. Kneip, L., Scaramuzza, D., Siegwart, R.: 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)
10. Kukelova, Z., Bujnak, M., Pajdla, T.: Automatic generator of minimal problem solvers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 302–315. Springer, Heidelberg (2008)
11. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate O(n) solution to the PnP problem. International Journal of Computer Vision (IJCV) 81(2), 578–589 (2009)
12. Li, S., Xu, C., Xie, M.: A robust O(n) solution to the perspective-n-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 34(7), 1444–1450 (2012)
13. Lu, C., Hager, G., Mjolsness, E.: Fast and globally convergent pose estimation from video images. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 22(6), 610–622 (2000)
14. Nistér, D., Stewénius, H.: A minimal solution to the generalized 3-point pose problem. Journal of Mathematical Imaging and Vision (JMIV) 27(1), 67–79 (2006)

15. Schweighofer, G., Pinz, A.: Robust pose estimation from a planar target. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28(12), 2024–2030 (2006)
16. Schweighofer, G., Pinz, A.: 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)
17. Zheng, Y., Kuang, Y., Sugimoto, S., Astrom, K., Okutomi, M.: Revisiting the PnP problem: A fast, general and optimal solution. In: Proceedings of the International Conference on Computer Vision (ICCV), Sydney, Australia (2013)