Minimal Solvers for Generalized Pose and Scale Estimation from Two Rays and One Point

Federico Camposeco[1(✉)], Torsten Sattler[1], and Marc Pollefeys[1,2]

[1] Department of Computer Science, ETH Zurich, Zurich, Switzerland
{federico.camposeco,torsten.sattler,marc.pollefeys}@inf.ethz.ch
[2] Microsoft, Redmond, USA

Abstract. Estimating the poses of a moving camera with respect to a known 3D map is a key problem in robotics and Augmented Reality applications. Instead of solving for each pose individually, the trajectory can be considered as a generalized camera. Thus, all poses can be jointly estimated by solving a generalized PnP (gPnP) problem. In this paper, we show that the gPnP problem for camera trajectories permits an extremely efficient minimal solution when exploiting the fact that pose tracking allows us to locally triangulate 3D points. We present a problem formulation based on one point-point and two point-ray correspondences that encompasses both the case where the scale of the trajectory is known and where it is unknown. Our formulation leads to closed-form solutions that are orders of magnitude faster to compute than the current state-of-the-art, while resulting in a similar or better pose accuracy.

Keywords: Absolute camera pose · Pose solver · Generalized cameras

1 Introduction

Estimating the absolute pose of a camera, i.e., the position and orientation from which an image was taken, with respect to a given 3D map is a fundamental building block in many 3D computer vision applications such as Structure-from-Motion (SfM) [27], simultaneous localization and mapping (SLAM) [5], image-based localization [18,26,29,35], Augmented Reality (AR) [21,22], and visual navigation for autonomous vehicles [34]. Traditionally, research on camera pose estimation has mainly focused on individual cameras [8], potentially estimating the extrinsic parameters of the camera pose together with the parameters of its intrinsic calibration [2,10]. In the context of robotics applications such as autonomous drones and vehicles, it is desirable to use multi-camera systems that cover the full field-of-view around the robots. Multi-camera systems can be modelled as a generalized camera [25], i.e., a camera for which not all viewing rays intersect in a single center of projection. Accordingly, camera pose estimation for generalized cameras has started to receive attention lately [3,11,15,17,24,30,33].

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46454-1_13) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part V, LNCS 9909, pp. 202–218, 2016.
DOI: 10.1007/978-3-319-46454-1_13

In this paper, we consider a problem typically arising in AR or video registration against SfM models [14], where visual-inertial odometry (VIO) [9] or visual odometry (VO) [23] is used to track the pose of the camera over time while registering the trajectory against a previously build 3D map acting as a reference coordinate system for the virtual objects [21]. In this scenario, both the local pose tracking and the pose estimation with respect to the map need to be highly accurate. Instead of estimating the absolute pose w.r.t. the map for each image in the trajectory, the movement of the camera defines a generalized camera that can be used to obtain a more accurate and reliable pose estimate due to its larger field-of-view [1]. VIO, VO and SfM compute the trajectory by tracking features across views, which naturally leads to estimates of the corresponding 3D point coordinates in the local coordinate system of the trajectory.

The fact that 3D point positions are available for some features in the images has not been widely used—except for point registration techniques—by pose solvers for generalized cameras. Instead, state-of-the-art methods estimate the pose from three or more standard 2D-3D matches between 3D points in the map and corresponding 2D image features. In this paper, we show that using one known local 3D point coordinate significantly simplifies the pose estimation problem and leads to more efficient minimal solvers with a similar or better pose accuracy.

The above scenario leads to two variants of the generalized absolute pose problem: The scale of the local trajectory w.r.t. the map is either known or unknown. The former variant arises when the absolute scale can be estimated accurately, e.g., from inertial data in a VIO system. The latter variant is most relevant for purely visual odometry (VO) [5,6,23] systems, or for SfM methods that rely on building sub-reconstructions and merging them afterwards [31].

In this paper, we show that knowing the local 3D point position for one of the 2D-3D matches leads to a formulation that covers both problem variants, i.e., the know-scale variant is a special case and permits an even more efficient solution. In detail, this paper makes the following contributions. (i) we derive a joint formulation of the generalized absolute pose problem based on a known 3D point position and two matches between 3D points in the map and image observations. (ii) we develop two novel pose solvers for both cases; known and unknown scale. Whereas state-of-the-art approaches need to solve polynomials of degree 8 or higher, both our methods are solvable by radicals, requiring us to only solve polynomials of degree 2 or a polynomial of degree 4, respectively. As a result, both our solvers are significantly more efficient and also generate fewer solutions. (iii) we show through extensive experiments on both synthetic and real data that our solver is not only more efficient to compute, but also at least as stable and accurate as the current state-of-the-art.

The remainder of the paper is structured as follows. Section 2 reviews related work. Section 3 discusses the geometry of the absolute pose problem for generalized cameras with and without known scale. Section 4 derives our solvers, which are then evaluated in Sect. 5.

2 Related Work

The problem of estimating the pose of a calibrated camera from n known 2D-3D correspondences is known as the n-Point-Pose or Perspective n Point (PnP) problem. The problem is typically solved by relating 3D map points to the viewing rays of their corresponding image measurements, i.e., the pose is estimated from *point-ray* correspondences. A computationally inexpensive, numerically stable and minimal solver is very desirable for RANSAC schemes, since it allows for a solution to be found fast and accurately. The P3P problem is the minimal case of the PnP problem, where only three point-ray correspondences are used to solve for the pose of the camera [8]. The solutions by Fischler and Bolles [7] and by Kneip *et al.* [13] are notable solvers of the P3P problem, where a quartic equation needs to be solved as part of the algorithm. Quartic equations can be solved by radicals non-iteratively, resulting in fast solvers that only require a 2 to 4 µs on a modern PC.

Solutions to the PnP problem only cover cameras whose viewing rays intersect in a single center of projection. The *generalized* PnP (gPnP) problem is the corresponding pose estimation problem for generalized cameras, i.e., cameras whose viewing rays do not intersect in a single center of projection. Minimal solvers for this problem require three point-ray correspondences (gP3P) and have been proposed by Nístér and Stéwenius [24], Kneip *et al.* [11] and Lee *et al.* [17]. The resulting solvers are noticeably more complex and require solving an octic polynomial, which cannot be solved non-iteratively by radicals. Consequently, gP3P solvers are significantly slower than P3P solvers. An iterative approach was proposed by Chen and Chang [3] as a special case of their gPnP solution.

Little work exists on the gPnP problem with *unknown scale*, referred to as the gPnP+s problem. The solver proposed by Ventura *et al.* [33] requires at least four point-ray correspondences (gP4P+s) and again leads to an octic polynomial. While mainly used as a minimal solver inside a RANSAC framework [7], their method can also use more correspondences to obtain a least squares solution. Kukelova *et al.* [15] recently proposed a gP4P+s solver that finds the coefficient to the octic very efficiently by circumventing any Gröbner basis computation. Compared to Ventura *et al.*, Kukelova's *et al.* speedup is 18.5, while ours is 47. Also, Kukelova's *et al.* method has a slightly worse accuracy than Ventura's *et al.*, while our solver has better accuracy w.r.t. Ventura's *et al.* Finally, in [30,31] Sweeney *et al.* proposed a more efficient scalable solution for n points that can also handle the so-called minimal case[1]. This is an $O(n)$ solution to the gPnP+s problem, minimizing an approximation of the reprojection error. While providing more accurate poses than [33], the solver from Sweeney *et al.* is also significantly slower.

In this work, we use two point-ray correspondences and one point-point match (obtained by triangulating points in the local frame of the camera trajectory) to simplify both the gPnP and the gPnP+s problem. Similar approaches have

[1] Estimating a similarity transformation with 7° of freedom (DOF) provides a solution to the gPnP+s problem while four point-ray correspondences provide 8 constraints.

been proposed in the context of *relative* generalized pose solvers [28], since the complexity of such problem is very high (64-degree polynomial). For example, Lee *et al.* [16] use the Ackermann motion constraint and shared observations between the cameras in a multi-camera system to reduce the problem to a six-degree polynomial. More related to our approach, Clipp *et al.* [4] simplify the relative generalized motion problem by triangulating one 4-view point, deriving a solution which requires the solution of 16-th degree polynomial. In contrast, our solver requires triangulating a point from two or more views and results in quadratic and quartic equations for the gPnP and gPnP+s problems.

3 Problem Statement

Consider the following problem: Given a 3D model, e.g., generated from SfM or SLAM, a trajectory of poses for a single camera or a multi-camera system, and n point-ray matches between features found in images from the trajectory and 3D points in the model, compute the position and orientation of the trajectory in the coordinate system of the model. The cameras in the trajectory form a generalized camera [25] and so this is an instance of the gPnP problem.

As mentioned in Sect. 2, a variant of the gPnP problem is the gPnP+s problem, where the internal scale of the generalized camera does not match the scale of the world points. In such cases it is required that the scale of the trajectory is estimated together with the pose. In this paper, we are interested in developing efficient minimal solvers for both problems, i.e., algorithms that compute a solution for the problems where the number of constraints *matches* the number of degrees of freedom (DOF) or unknowns. Such solvers are typically employed inside a RANSAC [7] loop, where using a minimal solver maximizes the probability of picking an all-inlier sample and thus reduces the number of necessary iterations. For solving the gPnP and gPnP+s problems, we assume that a 3D point position is known for at least one feature in the n-point sample drawn in each RANSAC step. Notice that this assumption is not restrictive: We are considering a camera trajectory generated from tracking features. These feature tracks can be triangulated to obtain 3D point positions in the local coordinate system of the trajectory. Triangulatable points are also easily available in multi-camera systems with visual overlap, where our solvers may be used even if there is no trajectory available.

In the following we discuss a mathematical representation of a generalized camera, and then describe the two versions of the generalized absolute pose problem.

3.1 Generalized Cameras

In its most general definition, a generalized camera is a set of viewing rays which do not necessarily intersect in a single center of projection. Given a base frame $\{B\}$ for the generalized camera with origin $\mathbf{0} \in \mathbb{R}^3$, all viewing rays can be expressed using Plücker line coordinates [25] defined in the base frame.

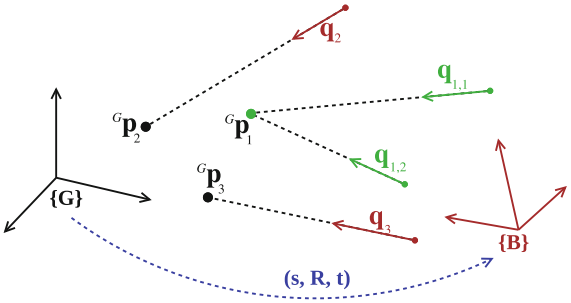


Fig. 1. The pose and scale problem for a generalized camera system. The origin of all rays $\mathbf{q}_i$ defined in base frame $\{B\}$. Notice that the point ${}^G\mathbf{p}_1$ can be triangulated from $\mathbf{q}_{1,1}$ and $\mathbf{q}_{1,2}$, highlighted in *green*. Note that for the known-scale scenario $s = 1$. (Color figure online)

A Plücker line is a pair of 3-vectors $\mathbf{q}$ and $\mathbf{q}'$, where $\mathbf{q}$ is a vector of any magnitude that points in the direction of the line and $\mathbf{q}' = \mathbf{q} \times \mathbf{p}$, where $\mathbf{p}$ is any point on the line (cf. Fig. 1). This definition implies that $\mathbf{q} \cdot \mathbf{q}' = 0$. Furthermore we enforce that $\mathbf{q} \cdot \mathbf{q} = 1$, which simplifies the terms that appear in our derivations. A 3D point ${}^B\mathbf{p}_i$ in the base frame $\{B\}$ of the generalized camera can be written as

$$ {}^B\mathbf{p}_i = \mathbf{q}_i \times \mathbf{q}'_i + \lambda_i \mathbf{q}_i, \tag{1} $$

where $\mathbf{q}_i$ is the (unit-length) ray defined in $\{B\}$ that points towards ${}^B\mathbf{p}_i$ and $\lambda_i \in \mathbb{R}_{>0}$ is the depth of the point along the Plücker line.

3.2 Generalized Pose Estimation with Unknown Scale (gP*n*P+s)

In the more general case, we aim to compute the similarity transform (pose and scale) between a generalized camera defined in the base frame $\{B\}$ and the global frame of reference $\{G\}$ based on n point-ray matches. This scenario arises more often in vision-only pipelines, e.g., during loop-closure or localization of a local SLAM or SfM trajectory—modeled as a generalized camera—against a known map of 3D landmarks [31]. As illustrated in Fig. 1, the transformation $(s, \mathbf{R}, \mathbf{t})$ maps the i-th point ${}^G\mathbf{p}_i$ from the global frame $\{G\}$ into the base frame via

$$ s\mathbf{R}\,{}^G\mathbf{p}_i + \mathbf{t} = {}^B\mathbf{p}_i = \mathbf{q}_i \times \mathbf{q}'_i + \lambda_i \mathbf{q}_i, \tag{2} $$

where $\mathbf{q}_i$ is the (unit-length) ray defined in $\{B\}$ that points towards ${}^B\mathbf{p}_i$ and λ_i is the depth of the point along the Plücker line.

If we directly use Eq. (2) to solve for the similarity transformation, at least 4 point-ray correspondences are required to find a solution [30,33]. However, this yields an overdetermined solution since 4 point-ray correspondences provide 8 constraints, while a similarity transformation has only 7° of freedom (DOF). This results in having to find the roots of an 8-th degree polynomial and obtaining up to 8 solutions.

Thus, we aim to derive a minimal solution to reduce the complexity of an overdetermined, least-square solution. If we instead consider the case where two of the 4 rays intersect in space, i.e., if we can triangulate the 3D position of *one* point in the base frame $\{B\}$, the gPnP+s problem can be solved by determining the similarity transformation from one point-point correspondence (fixing the three DOF of the translation) and two point-ray correspondences (fixing the remaining 4 DOF). Thus the DOF of the transformation match the number of constraints exactly. We will show in Sect. 4 that this minimal parametrization of the problem can be solved by finding the roots of a quartic, which can be obtained non-iteratively by radicals and yields up to 4 solutions. As a result, our solver is less computationally expensive as state-of-the-art solvers [30,33] and also exhibits fewer solutions. Notice that, in practice, the point we triangulate for the solution might already be known as part of a SLAM trajectory or a local SfM solution. In the case of multi-camera pose estimation however, we might need to explicitly triangulate for this point (e.g. using observations $\mathbf{q}_{1,1}$ and $\mathbf{q}_{1,2}$ as shown in Fig. 1). If so, we employ the method by [19], which is a very efficient (88 floating point operations) approximation to the L_2-optimal triangulation.

3.3 Generalized Pose Estimation with Known Scale (gPnP)

The second scenario assumes that the scale of the preexisting map and the internal scale of the generalized camera are consistent, a situation that usually arises with multi-camera setups and VIO systems, where the scale of the map and the trajectory can be recovered. In this problem variant, the alignment from points in $\{B\}$ to points in $\{G\}$ is defined by a 6 DOF Euclidean transformation. Mathematically, this case is defined similar to Eq. (2), setting $s = 1$ instead of allowing an arbitrary scaling factor.

As discussed in Sect. 2, in the minimal instance this is known as the Generalized P3P problem, or gP3P [11], as we need at least 3 point-ray correspondences to get a finite number of solutions. Compared to the unknown-scale scenario, this has received more attention recently [3,11,12,17,24] due to its applicability in robotic systems, such as VIO trajectories and pre-calibrated multi-camera rigs. The gP3P problem has up to 8 solutions and can be solved by finding the roots of an 8-th degree polynomial.

In our setup, we assume a geometric situation similar to the general scenario (cf. Fig. 1), where one point is known in the base frame, and we aim to find the location of the two remaining points along their Plücker lines. In this case our solution is an overdetermined one—solving a 6 DOF problem with 7 constraints—and our solution is minimal only in the number of points used. Still, our solution to the gPnP problem is highly relevant for practical applications since it can be computed extremely efficiently by finding the roots of two quadratics. At the same time, our approach can outperform the minimal solutions in terms of accuracy and efficiency in the cases where the triangulation is accurate—which can easily be gauged by looking at the subtended angle of the two viewing rays.

4 Solution Methodology

Here we present our two solvers to address the problems presented in the previous section. For both solvers, we use the fact that we know the location of *one* point in the base frame of the generalized camera $\{B\}$, let us denote this point as ${}^{B}\mathbf{p}_1$. To simplify the expressions that will appear in both solvers, we translate the base frame $\{B\}$ to coincide with ${}^{B}\mathbf{p}_1$, such that in the new intermediate frame $\{B'\}$ points become ${}^{B'}\mathbf{p}_i = {}^{B}\mathbf{p}_i - {}^{B}\mathbf{p}_1\,,\ i = 1, 2, 3$.

For each problem we now have *one* point-point correspondence and *two* point-ray correspondences

$$\begin{aligned} {}^{B'}\mathbf{p}_1 &= s\mathbf{R}\,{}^{G}\mathbf{p}_1 + \mathbf{t} \\ \mathbf{q}_i \times \mathbf{q}'_i + \lambda_i \mathbf{q}_i &= s\mathbf{R}\,{}^{G}\mathbf{p}_i + \mathbf{t} \quad \text{for } i = 2, 3. \end{aligned} \tag{3}$$

For the pose and scale case, gPnP+s, we chose a scale-invariant constraint to get a set of equations that do not depend explicitly on s. If we regard the triplet of points in $\{B'\}$ and their counterparts in $\{G\}$ as triangles, we may use the notion of triangular similarity, which states that two triangles are similar if two of their angles are congruent or they have the same *side-length ratio* (cf. Fig. 2a). If the scale of the points is known (gP3P) then our correspondences in Eq. (3) are simplified by setting $s = 1$. In this case there is no need to use the ratio of lengths, instead we can directly enforce that the distances between points in $\{B'\}$ match the known distances in $\{G\}$ (cf. Fig. 2b).

For both problems we end up with a system of equations in λ_2 and λ_3, which when solved give us the location of the remaining points in $\{B'\}$. For each of these solutions we revert the translation offset from the triangulated point to obtain ${}^{B}\mathbf{p}_i = {}^{B'}\mathbf{p}_i + {}^{B}\mathbf{p}_1\,,\ i = 1, 2, 3$. We may then use this points to compute the rigid transformation between $\{B\}$ and $\{G\}$, for which we use the algorithm proposed in [32].

4.1 Minimal Solution for gP4P+s

Using the above notation, triangular similarity for our three correspondences may be written as

$$\triangle({}^{B'}\mathbf{p}_1, {}^{B'}\mathbf{p}_2, {}^{B'}\mathbf{p}_3) \sim \triangle({}^{G}\mathbf{p}_1, {}^{G}\mathbf{p}_2, {}^{G}\mathbf{p}_3), \tag{4}$$

which allows us to use either angular or length-ratio preservation between the two triangles as constraints. Using the ratio of the lengths we may write

$$\frac{\left\|{}^{B'}\mathbf{p}_2 - {}^{B'}\mathbf{p}_1\right\|^2}{\left\|{}^{B'}\mathbf{p}_3 - {}^{B'}\mathbf{p}_1\right\|^2} = \frac{s^2\left\|{}^{G}\mathbf{p}_2 - {}^{G}\mathbf{p}_1\right\|^2}{s^2\left\|{}^{G}\mathbf{p}_3 - {}^{G}\mathbf{p}_1\right\|^2} = \frac{D_{2,1}}{D_{3,1}} \quad \text{and} \tag{5a}$$

$$\frac{\left\|{}^{B'}\mathbf{p}_3 - {}^{B'}\mathbf{p}_2\right\|^2}{\left\|{}^{B'}\mathbf{p}_2 - {}^{B'}\mathbf{p}_1\right\|^2} = \frac{s^2\left\|{}^{G}\mathbf{p}_3 - {}^{G}\mathbf{p}_2\right\|^2}{s^2\left\|{}^{G}\mathbf{p}_2 - {}^{G}\mathbf{p}_1\right\|^2} = \frac{D_{3,2}}{D_{2,1}} \tag{5b}$$

where $D_{i,j}$ is the known squared distance between points ${}^{G}\mathbf{p}_i$ and point ${}^{G}\mathbf{p}_j$. Using this results in a very succinct equation system since ${}^{B'}\mathbf{p}_1 = \mathbf{0}$:

$$\left\| {}^{B'}\mathbf{p}_i - {}^{B'}\mathbf{p}_1 \right\|^2 = \left\| {}^{B'}\mathbf{p}_i \right\| \text{ for } i = 2, 3. \tag{6}$$

Consequently, Eq. (5a) may be then simplified to

$$\|\mathbf{q}_2 \times \mathbf{q}_2' + \lambda_2 \mathbf{q}_2\|^2 - \tfrac{D_{2,1}}{D_{3,1}} \|\mathbf{q}_3 \times \mathbf{q}_3' + \lambda_3 \mathbf{q}_3\|^2 = 0, \tag{7}$$

and since $(\mathbf{q}_i \times \mathbf{q}_i') \cdot \mathbf{q}_i = 0$, we arrive at

$$\lambda_2^2 - \tfrac{D_{2,1}}{D_{3,1}} \lambda_3^2 + \|\mathbf{q}_2 \times \mathbf{q}_2'\|^2 - \tfrac{D_{2,1}}{D_{3,1}} \|\mathbf{q}_3 \times \mathbf{q}_3'\|^2 = 0. \tag{8}$$

The constraint from Eq. (5b) has a more general form, and no simplification occurs. With this, we may write our constraints as

$$\lambda_2^2 + k_1 \lambda_3^2 + k_2 = 0 \tag{9a}$$

$$\lambda_2^2 + k_3 \lambda_2 \lambda_3 + k_4 \lambda_3^2 + k_5 \lambda_2 + k_6 \lambda_3 + k_7 = 0, \tag{9b}$$

where k_i, $i = 1, .., 7$ depends only on the measurements and the known locations of the points in $\{G\}$.

Equations (9) are two quadratic equations with real coefficients (i.e. conic sections) on λ_2 and λ_3, which in general can be solved using a *quartic* univariate polynomial. In fact, the system is small enough that we can generate a Gröbner basis w.r.t. the lexicographic order symbolically. This yields a triangular system where we can get λ_3 as the solution to a quartic and λ_2 linearly afterwards,

$$\begin{aligned} &\left(k_1^2 + k_3^2 k_1 - 2k_4 k_1 + k_4^2\right) \lambda_3^4 + 2\left(k_1 k_3 k_5 - k_1 k_6 + k_4 k_6\right) \lambda_3^3 + \\ &\quad \left(k_2 k_3^2 + k_1 k_5^2 + k_6^2 + 2k_1 k_2 - 2k_2 k_4 - 2k_1 k_7 + 2k_4 k_7\right) \lambda_3^2 + \\ &\quad \left(2k_2 k_3 k_5 - 2k_2 k_6 + 2k_6 k_7\right) \lambda_3 + k_2^2 + k_2 k_5^2 + k_7^2 - 2k_2 k_7 = 0 \end{aligned} \tag{10a}$$

$$\left(k_4 - k_1\right) \lambda_3^2 + k_6 \lambda_3 + \lambda_2 \left(k_3 \lambda_3 + k_5\right) - k_2 + k_7 = 0. \tag{10b}$$

4.2 Solution for gP3P

Solving for the known-scale scenario is, as noted in Sect. 3.3, an overdetermined problem. In fact, one can solve for each depth, λ_2 and λ_3 independently. Since the scale is known, we can directly enforce that the distance of the known point ${}^{B'}\mathbf{p}_1$ to either point ${}^{B'}\mathbf{p}_i$ with $i = 2, 3$, be preserved by the Euclidean transformation. This results in the constraints

$$f_i\left(\lambda_i\right) \triangleq \left\| {}^{B'}\mathbf{p}_i - {}^{B'}\mathbf{p}_1 \right\|^2 = \left\| {}^{G}\mathbf{p}_i - {}^{G}\mathbf{p}_1 \right\|^2 = D_{i,1} \text{ with } i = 2, 3, \tag{11}$$

where we have defined f_i as the squared distance from point i to the known point. The constraints from Eq. (11) can be visualized as the intersection of a ray in

space parametrized by λ_i, and a sphere centered around ${}^{B'}\mathbf{p}_1$ with radius $\sqrt{D_{i,1}}$, for $i = 2, 3$ (cf. Fig. 2). However, for some cases the ray will not intersect the sphere because of noise in the image. If this happens, both solutions to Eq. (11) will be complex and we will have no solutions. Instead we *minimize* the error of the ray to the surface of the sphere. The distance to the sphere surface is

$$d_i(\lambda_i) = \left(f_i(\lambda_i) - D_{i,1}\right)^2, \tag{12}$$

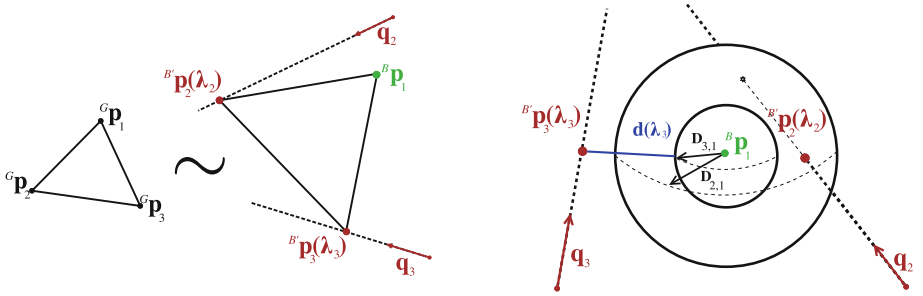
and we may find its critical points by finding λ_i such that

$$\frac{\partial d_i'(\lambda_i)}{\partial \lambda_i} = 0, \text{ for } i = 2, 3. \tag{13}$$

The constraints in Eq. (13) are univariate cubic equations in λ_2 and λ_3. However, it can be shown that they are reducible to

$$\lambda_i \left(\lambda_i^2 + \|\mathbf{q}_i \times \mathbf{q'}_i\|^2 - D_{i,1}\right) = 0 \text{ for } i = 2, 3, \tag{14}$$

which can be solved using only *one* square root. If the solution of the square root is real, then the ray intersects the sphere in two places. Otherwise, the closest point to the sphere is at $\lambda_i = 0$. This results in up to 4 real solutions, however, we do not need to output all solutions. In order to discard as many (λ_2, λ_3) pairs as possible, we use the remaining distance of the point triplet, $D_{3,2}$. We discard all solutions for which the distance $\left(\|f_3(\lambda_3) - f_2(\lambda_2)\| - \left\|{}^{G}\mathbf{p}_3 - {}^{G}\mathbf{p}_2\right\|\right)^2$ is larger than a threshold ($0.1D_{3,2}$ in our real-world experiments), leaving out all but one solution in practically all cases.



(a) Unknown Scale Constraint

(b) Known Scale Constraint

Fig. 2. Illustration of the geometry of the constraints. In (a) we intend to find the values λ_2 and λ_3 for which the triangle that we form becomes *similar* to the triangle formed by $({}^{B'}\mathbf{p}_1, {}^{B'}\mathbf{p}_2, {}^{B'}\mathbf{p}_3)$. In (b), for the given point in the base frame, ${}^{B'}\mathbf{p}_1$, our goal is to find the depth along the direction $\mathbf{q}_i$ such that the distance from that point along the ray to the sphere centered at ${}^{B'}\mathbf{p}_1$ with radius $\sqrt{D_{i,1}}$ is minimized. Notice that $\mathbf{q}_3$ has a direction which cannot intersect the sphere and so our results is the closest point to the sphere (the point where $d(\lambda_3)$ is smallest).

5 Evaluation

To evaluate our methods, we use synthetic data to evaluate their numerical stability, sensitivity to measurement noise and to triangulation accuracy. Additionally, our methods' accuracy was evaluated using 11 sequences of real-world data [33], where a SLAM camera trajectory is registered to an SfM model. For both of these evaluation modes, we compared against the following methods:

Absolute Orientation. This method [32] registers two 3D point sets via a similarity transform. The method is very simple and requires only linear operations and returns only one solution, however, it needs at least three points in the $\{B\}$ frame, so at least six point-ray correspondences are needed.

gP+s. Solves the GP4P+s problem as proposed by Ventura in [33] by finding the roots of an octic polynomial and returns up to 8 solutions.

gDLS. Scalable n point method for the GPnP+s problem proposed in [30]. It is designed to handle cases with several point-ray correspondences and minimizes the reprojection error globally, returning up to 27 solutions. For our evaluations, we used it with only 4 point-ray correspondences.

gP3P Chen. Chen's method [3] is the earliest solution to the gP3P problem. This solver is iterative in nature and may return up to 16 solutions.

gP3P Lee. One of the latest methods to tackle the GPnP problem presented in [17]. Similar to ours, this method represents ray-point correspondences as Plücker lines and solves for points along those lines. It includes a closed-form minimal solution the the absolute orientation problem, needed for the last step of aligning points in $\{B\}$ and $\{G\}$. The solution requires finding the roots of an octic and may return up to 8 feasible configurations.

gP3P Kneip. A minimal method from [11] that notably solves for the rotation of $\{B\}$ directly, and thus requires no last step that aligns two points sets. Similarly, it requires to solve an octic and returns up to 8 solutions as well.

g1P2R+s. Our Generalized 1 Point, 2 Rays plus scale solver (cf. Sect. 4.1). For our solver we need to find the roots of a quartic and we return up to four solutions.

g1P2R. Our Generalized 1 Point, 2 Rays solver (cf. Sect. 4.2). For this solver we need to compute two square roots and we return only one solution.

5.1 Synthetic Data Evaluation

For our synthetic data evaluation we first generate four cameras randomly placed in the cube $[-1, 1] \times [-1, 1] \times [-1, 1]$ around the origin. Then, 3D points in $\{B\}$ are sampled randomly from the volume $[-1, 1] \times [-1, 1] \times [2, 6]$. The point-ray correspondences are then generated by projecting all points to all cameras. Each method, however, is given the exact amount of correspondences it requires, e.g. gP3P only gets the first three point-ray correspondences. After this, a random rotation and translation is then applied to cameras and observations. Finally, if evaluating an unknown scale solver, a random scale between 0.5 and 20 is applied to the world points. The experiments were executed several thousand times (the exact number depends on the evaluation mode) in order to obtain a meaningful statistic of the accuracy under varying conditions as explained next.

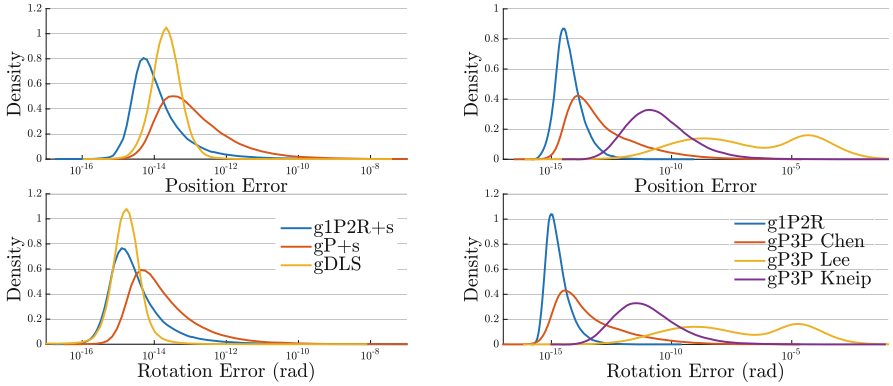
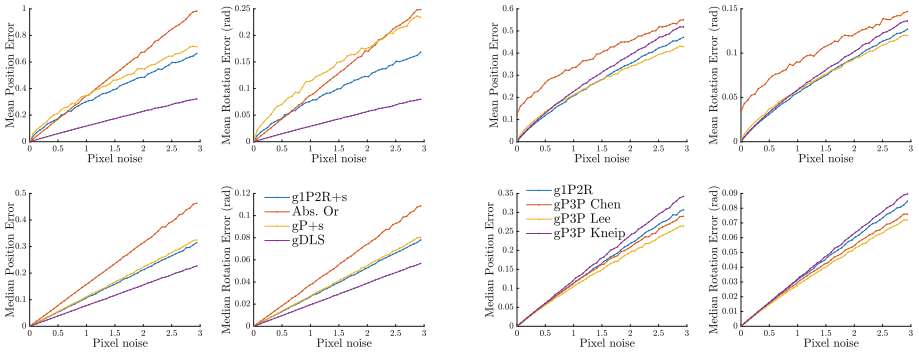


Fig. 3. Kernel-smoothed histograms of the numerical stability of the algorithms tested, gPnP+s algorithms on the *left* and gPnP on the *right*. Each algorithm was ran 10^5 times under noiseless conditions. Because of the lower computational complexity of our methods (*blue lines*), we achieve a very high numerical stability. (Color figure online)

Numerical Stability. One of the benefits of having a less complex solution to a particular problem is that there is less opportunity for numerical errors and instabilities to accumulate. This is specially true for the solvers presented in this paper, since they are both in closed-form. To evaluate this, the point-ray correspondences are left uncorrupted with noise. As seen in Fig. 3, the numerical errors are very small, and most often outperform the stability of other methods in their category. A 32-bit floating point implementation might even prove accurate enough and might increase performance even further.

Measurement Noise Resilience. To compare the accuracy of our solutions in the presence of measurement noise, we add Gaussian pixel noise using a focal length of 800 and an image size of 640×480. After each method is executed, we compare their rotational and translational accuracy with ground-truth. Figure 4 shows the median error of all trials for increasing pixel noise. For the unknown scale scenario, our method outperforms gP+s in rotational and translational precision. However, g1P2R+s is not as accurate as gDLS for any noise level. We emphasize here that gDLS optimizes the reprojection error over all four correspondences, and has a vastly larger computational cost (cf. Table 1). gDLS is better suited as a refinement step and is compared here as a baseline for accuracy. In the case of g1P2R, we manage to get precisions comparable to other state-of-the-art methods. Notably, we outperform Kneip's gP3P in most metrics. This might be due to the fact that other solvers absorb some of the errors in the point-ray correspondences when they align the obtained points to the world points as a post-processing step, whereas Kneip's solver computes the pose directly.

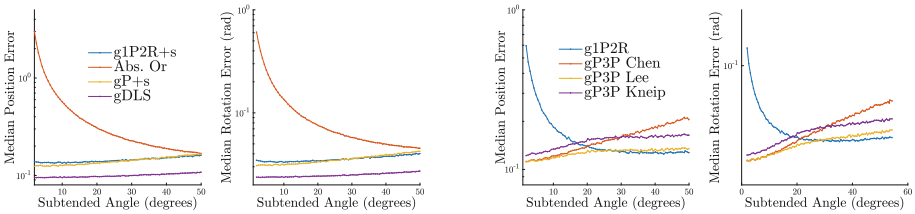


(a) Unknown Scale (b) Known Scale

Fig. 4. Average (*top rows*) and median (*bottom rows*) translational and rotational errors from 10^5 trials per pixel noise level. The median subtended angle of the triangulated point for our algorithms was of 14.5°. Notice that our unknown-scale solver (*blue line*) performs better than gP+s for all noise levels. Our known-scale algorithm (*blue line*) is not as resilient to noise as other minimal solvers, however it performs comparably and under higher subtended angles (cf. Fig. 5) it even outperforms them. (Color figure online)

Sensitivity to the Quality of Triangulation. The main concern with the algorithms presented here might be their dependency on the quality of the triangulated point in the base frame. To address this and find the point in which the reliability of our methods might decay due to triangulation errors, we exhaustively tested a wide range of subtended angles for the triangulated point that is used as a part of our solvers. It is known that the accuracy with which a triangulated point can be obtained largely depends on the subtended angle. Note, however, that in many of our target applications triangulated points in the local base frame are already available as part of the VIO/VO/SfM trajectory and one can safely assume that they will have enough accuracy (this assumption is validated with real-world data in Sect. 5.2). Figure 5 shows the accuracy of each method for a constant value of pixel noise (1 pixel standard deviation) while we vary the point configuration such that the subtended angle of the triangulated point changes. Using this, we can see that after approximately 30°, our solvers are likely to yield comparable or better results than other state-of-the-art methods, while taking only a fraction of the time to compute as it will be shown next. Notice that, since triangulation errors impact Absolute Orientation more dramatically, its performance does not become reliable until a very high subtended angle.

Runtime Analysis. To give an estimate of the computational cost of our algorithms compared to its alternatives, we generated the same random instances of synthetic scenes with fixed pixel noise of 1 pixel standard deviation. We compared against those methods which have available C++ implementations, adding



(a) Unknown Scale (b) Known Scale

Fig. 5. Median rotational and translational errors for a pixel noise of 1. For each of the 10^6 trials, the subtended angle of the triangulated point was varied. The accuracy of our pose and scale solver is comparable to gP+s [33] throughout. However, g1P2R has an acceptable accuracy only after the triangulated point has a subtended angle of more than 15°. After 30°, it tends to outperform all other methods.

to those our own C++ implementations of [33] and of [17]. Since our solutions are solvable by radicals, we vastly outperform all other competing methods by at least one order of magnitude (cf. Table 1).

Table 1. Runtime comparison of the algorithms used for our evaluations. Notice that both of our solvers are at least one order of magnitude faster than their counterparts. Timings are reported for C++ implementations running on an Intel i7 at 2.5 GHz.

Method	gDLS	gP+s	gP3P Kneip	g1P2R+s	g1P2R
Microseconds	432.78	98.31	41.01	2.07	0.86

5.2 Real Data Comparison

To validate the performance of our method in real-world scenarios, we used the dataset from [33]. The dataset consists of 12 SLAM sequences of a scene with local poses of cameras in a trajectory and ground-truth obtained from an ART-2 optical tracker, from which the first 11 sequences were used. Additionally, the dataset includes a full SfM reconstruction of the scene. This allows us to register each SLAM sequence against the SfM data via a similarity transform. SIFT [20] keypoints were used to get a set of putative matches between all frames in a sequence and the 3D map using exhaustive search and Lowe's ratio test.

All algorithms we compared against were used within RANSAC. The resulting similarity transform from RANSAC with the highest number of inliers was directly used to transform all the SLAM poses in the sequence. Using these corrected poses, positional accuracy against ground-truth from the tracker was extracted (cf. Fig. 6). To get a robust measure of accuracy, we executed RANSAC 1000 times and took the median positional error for all methods. In order to also

evaluate all the known-scale methods, we computed the true scale using the provided ground-truth SLAM poses and scaled each trajectory accordingly. Notice that for our solvers we do not have a simple inlier/outlier partition. Our methods use two modes of data points, one triangulated point and two point-ray observations. In order to accommodate this, our RANSAC stopping criteria needs to be modified, for which we follow the method proposed in [4]. We keep track of *two* inlier ratios; one for point-point correspondence ϵ_p and one four point-ray correspondences ϵ_r. The number of samples used as a stopping criterion becomes

$$k = \log(1-\eta) / \log\left(1 - \epsilon_p \epsilon_r^2\right) \tag{15}$$

where η is the confidence that we pick one outlier-free sample.

Our known-scale solver, g1P2R, outperforms all other minimal methods in 6 occasions (cf. Fig. 6). However, it is the least accurate for two trajectories. Our unknown-scale solver performs very well against the tested methods, outperforming even gDLS in two sequences and always outperforming gP+s.

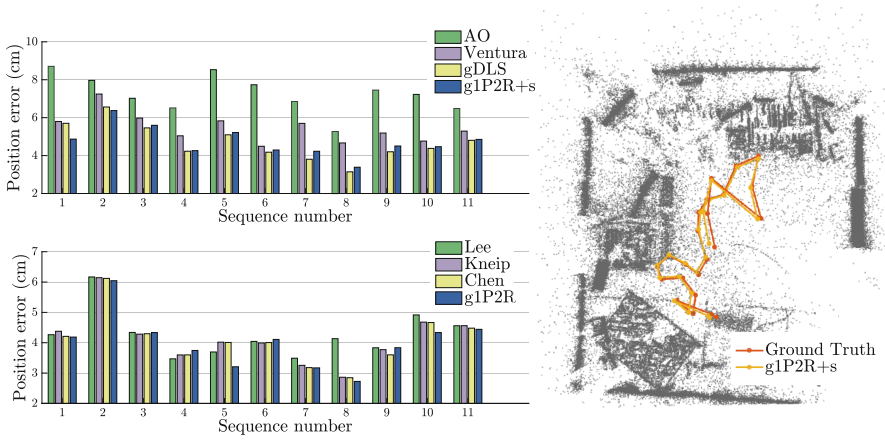


Fig. 6. *Left:* Comparison of position error (in centimeters) for the unknown-scale (*top-left*) and known-scale (*bottom-left*) algorithms. *Right:* Top-down view of the SfM used to register each sequence against. Shown in *orange* is the ground-truth positions given by the ART-2 tracker, and in *yellow* our solution. (Color figure online)

6 Conclusion

In this paper, we have considered the generalized PnP problem for a moving camera. We have derived closed-form solutions based on one point-point and two point-ray correspondences for both the known and unknown-scale cases. The resulting minimal solvers are extremely efficient, resulting in run-times that are orders of magnitude faster than current state-of-the-art methods that purely rely

on point-ray correspondences. At the same time, our solvers achieve a similar or even better pose accuracy. Our formulation vastly simplifies the pose estimation problem, and our results show that—contrary to what one might expect—this does not come at the price of reduced accuracy.

Acknowledgements. This research was funded by Google's Tango.

References

1. Arth, C., Klopschitz, M., Reitmayr, G., Schmalstieg, D.: Real-time self-localization from panoramic images on mobile devices. In: International Symposium on Mixed and Augmented Reality (ISMAR) (2011)
2. Bujnak, M., Kukelova, Z., Pajdla, T.: A General solution to the P4P problem for camera with unknown focal length. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
3. Chen, C.S., Chang, W.Y.: On pose recovery for generalized visual sensors. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **26**(7), 848–861 (2004)
4. Clipp, B., Zach, C., Frahm, J.M., Pollefeys, M.: A new minimal solution to the relative pose of a calibrated stereo camera with small field of view overlap. In: International Conference on Computer Vision (2009)
5. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **29**(6), 1052–1067 (2007)
6. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part II. LNCS, vol. 8690, pp. 834–849. Springer, Heidelberg (2014)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM (CACM) **24**(6), 381–395 (1981)
8. Haralick, R., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. Int. J. Comput. Vis. (IJCV) **13**(3), 331–356 (1994)
9. Hesch, J.A., Kottas, D.G., Bowman, S.L., Roumeliotis, S.I.: Camera-IMU-based localization: observability analysis and consistency improvement. Int. J. Robot. Res. (IJRR) **33**(1), 182–201 (2014)
10. Josephson, K., Byröd, M.: Pose estimation with radial distortion and unknown focal length. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
11. Kneip, L., Furgale, P., Siegwart, R.: Using multi-camera systems in robotics: efficient solutions to the NPnP problem. In: International Conference on Robotics and Automation (ICRA), pp. 3770–3776. IEEE (2013)
12. Kneip, L., Li, H., Seo, Y.: UPnP: an optimal $O(n)$ solution to the absolute pose problem with universal applicability. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 127–142. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10590-1_9
13. 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: Conference on Computer Vision and Pattern Recognition (CVPR) (2011)

14. Kroeger, T., Van Gool, L.: Video registration to SfM models. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 1–16. Springer, Heidelberg (2014)
15. Kukelova, Z., Heller, J., Fitzgibbon, A.: Efficient intersection of three quadrics and applications to computer vision. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
16. Lee, G., Fraundorfer, F., Pollefeys, M.: Motion estimation for self-driving cars with a generalized camera. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
17. Hee Lee, G., Li, B., Pollefeys, M., Fraundorfer, F.: Minimal solutions for pose estimation of a multi-camera system. In: Inaba, M., Corke, P. (eds.) Robotics Research. STAR, vol. 114, pp. 521–538. Springer, Heidelberg (2016). doi:10.1007/978-3-319-28872-7_30
18. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3d point clouds. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 15–29. Springer, Heidelberg (2012)
19. Lindstrom, P.: Triangulation made easy. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2010)
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) **60**(2), 91–110 (2004)
21. Lynen, S., Sattler, T., Bosse, M., Hesch, J., Pollefeys, M., Siegwart, R.: Get out of my lab: large-scale, real-time visual-inertial localization. In: Robotics Science and Systems (RSS) (2015)
22. Middelberg, S., Sattler, T., Untzelmann, O., Kobbelt, L.: Scalable 6-DOF localization on mobile devices. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part II. LNCS, vol. 8690, pp. 268–283. Springer, Heidelberg (2014)
23. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2004)
24. Nistér, D., Stewénius, H.: A minimal solution to the generalised 3-point pose problem. J. Math. Imaging Vis. **27**(1), 67–79 (2007)
25. Pless, R.: Using many cameras as one. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2003)
26. Sattler, T., Havlena, M., Radenovic, F., Schindler, K., Pollefeys, M.: Hyperpoints and fine vocabularies for large-scale location recognition. In: International Conference on Computer Vision (ICCV) (2015)
27. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: SIGGRAPH (2006)
28. Stewénius, H., Nistér, D., Oskarsson, M., Åström, K.: Solutions to minimal generalized relative pose problems. In: Workshop on Omnidirectional Vision, Beijing, China (2005)
29. Svärm, L., Enqvist, O., Oskarsson, M., Kahl, F.: Accurate localization and pose estimation for large 3d models. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
30. Sweeney, C., Fragoso, V., Höllerer, T., Turk, M.: gDLS: a scalable solution to the generalized pose and scale problem. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part IV. LNCS, vol. 8692, pp. 16–31. Springer, Heidelberg (2014)
31. Sweeney, C., Fragoso, V., Höllerer, T., Turk, M.: Large scale SfM with the distributed camera model (2016). arXiv:1607.03949

32. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **13**(4), 376–380 (1991)
33. Ventura, J., Arth, C., Reitmayr, G., Schmalstieg, D.: A minimal solution to the generalized pose-and-scale problem. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
34. Xu, S., Honegger, D., Pollefeys, M., Heng, L.: Real-time 3d navigation for autonomous vision-guided MAVs. In: Intelligent Robots and Systems (IROS) (2015)
35. Zeisl, B., Sattler, T., Pollefeys, M.: Camera pose voting for large-scale image-based localization. In: International Conference on Computer Vision (ICCV) (2015)