gDLS: A Scalable Solution to the Generalized Pose and Scale Problem

Chris Sweeney, Victor Fragoso, Tobias Höllerer, and Matthew Turk

University of California, Santa Barbara, USA
{cmsweeney,vfragoso,holl,mturk}@cs.ucsb.edu

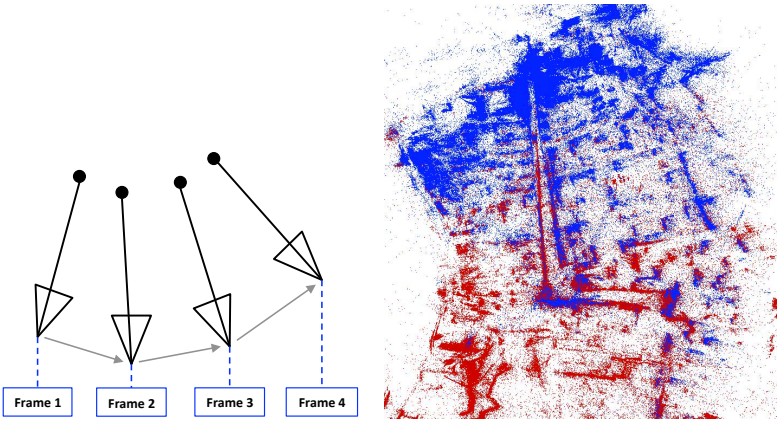
Abstract. In this work, we present a scalable least-squares solution for computing a seven degree-of-freedom similarity transform. Our method utilizes the generalized camera model to compute relative rotation, translation, and scale from four or more 2D-3D correspondences. In particular, structure and motion estimations from monocular cameras lack scale without specific calibration. As such, our methods have applications in loop closure in visual odometry and registering multiple structure from motion reconstructions where scale must be recovered. We formulate the generalized pose and scale problem as a minimization of a least squares cost function and solve this minimization without iterations or initialization. Additionally, we obtain all minima of the cost function. The order of the polynomial system that we solve is independent of the number of points, allowing our overall approach to scale favorably. We evaluate our method experimentally on synthetic and real datasets and demonstrate that our methods produce higher accuracy similarity transform solutions than existing methods.

1 Introduction

The problem of determining camera position and orientation given a set of correspondences between image observations and known 3D points is a fundamental problem in computer vision. This set of problems has a wide range of applications in computer vision, including camera calibration, object tracking, simultaneous localization and mapping (SLAM), structure-from-motion (SfM), and augmented reality. In the case of a calibrated camera, several minimal methods exist to determine the camera pose from three correspondences (P3P) [9, 13]. The Perspective-n-Point (PnP) problem determines the pose for a single calibrated camera from n 2D-3D observations [10, 16]. These methods, however, are all designed for localization of a single perspective camera and there are few methods that are able to jointly utilize information from many cameras simultaneously. As illustrated in Figure 1a, multiple cameras (or multiple images from a single moving camera) can be described with the generalized camera model [20]. However, the internal scale of each generalized camera is not guaranteed to be consistent, so the relative scale between the generalized cameras must be recovered in addition to the rotation and translation.

In this paper, we propose a new solution for the seven degree-of-freedom (d.o.f.) generalized pose-and-scale problem for multiple cameras. The generalized

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 16–31, 2014.



(a) SLAM motion.

(b) Two SfM reconstructions of Dubrovnik [17] (one red, one blue) are accurately aligned with our method.

Fig. 1. (a) The generalized camera model represents a set of image rays that do not need to share a common origin. This allows for multiple cameras, or multiple images from a single camera in motion to be modeled with the generalized camera model. We use the generalized camera model to solve the generalized pose-and-scale problem. (b) Our scalable method can be used to accurately align two SfM reconstructions containing millions of points.

pose-and-scale problem is equivalent to the problem of estimating a similarity transform between two coordinate systems. Our method is a generalization of the PnP problem to multiple cameras which are represented by a generalized camera; we recover the position and orientation as well as the internal scale of the generalized camera with respect to known 3D points.

The generalized pose-and-scale problem is one that frequently arises in SLAM and SfM. It is impossible to recover scale from images alone. Scale may be recovered in special cases, for instance, when observing an object with known metric dimensions or with the aid of sensor measurements, but these cases are not common. As a result, the relative scale must be reconciled, for example, during loop closure for SLAM or when registering multiple reconstructions from SfM (see Figure 1b). The generalized pose-and-scale problem aims to compute a similarity transform that will align two coordinate systems from 2D-3D correspondences.

The solution proposed in this paper solves the generalized pose-and-scale problem, estimating rotation, translation, and scale directly given n 2D-3D observations. Our approach is $O(n)$ in the number of observations, making it useful for real-time applications, and does not require initialization. Additionally, we solve for all minima of our least squares cost function simultaneously instead of

a single local minimum. Experiments on synthetic and real data show that our method is more accurate and scalable than other alignment methods.

The rest of the paper is as follows: Section 2 provides an overview of related work. We present the generalized pose-and-scale problem in Section 3, and our approach is described in detail in Section 4. We describe synthetic and real data experiments with comparisons to alternative approaches in Section 5, before providing concluding remarks in Section 6.

2 Related Work

There is much recent work on solving for the camera pose of calibrated cameras [2,9,13–15]. However, these methods only handle the case of solving for the pose of a single calibrated camera.

We solve a problem in the family of Non-Perspective-n-Point (NPnP) problems. Minimal solutions to the NP3P problem were proposed by Nís ter and Stéwenius [19] and Chen and Chang [3] for generalized cameras. Nís ter and Stéwenius reduce the NP3P problem to the solution of an octic polynomial which can be solved efficiently with root finding schemes. Chen and Chang propose an iterative solution to the NPnP problem, deriving a solution to the NP3P problem as a special case. Other iterative solutions have been proposed [21, 23], though they are computationally expensive and depend heavily on a good initialization. Ess *et al.* [8] presented a non-iterative method, however, the complexity is at least quadratic in the number of points. Lepetit *et al.* [16] propose the EPnP algorithm that is $O(n)$ by representing the 3D points as a sum of 4 virtual control points, reducing the problem to estimating the positions of the control points. This method, however, is only applicable for a single perspective camera. The gPnP algorithm of Kneip *et al.* [12] is an extension of the EPnP algorithm [16] to generalized cameras. By utilizing Gröbner basis computations, they solve the NPnP problem in $O(n)$. However, the gPnP algorithm does not estimate scale. Ventura *et al.* [26] presented the first minimal solution to the similarity transformation problem. They use the generalized camera model and employ Gröbner basis computations to efficiently solve for scale, rotation, and translation using 4 2D-3D correspondences. They call this minimal problem the gP+s or NP4P+s problem. We are solving for scale, rotation, and translation using $n \geq 4$ points and thus call our problem the gPnP+s, or NPnP+s problem.

Our work is closely related to the Direct Least Squares (DLS) PnP algorithm of Hesch and Roumeliotis [10]. In this work, a modified least squares cost function is used to first solve for the rotation estimation as the solution to a set of third-order polynomials, then solve for translation with back-substitution. We derive a generalization of the DLS PnP algorithm that allows for multiple cameras while additionally recovering scale. As such, we call our algorithm gDLS. The solution proposed in this paper requires solving a polynomial system of exactly the same scale as DLS PnP, despite the additional complexity of our problem. This allows our solution to remain $O(n)$ in the number of points.

The proposed algorithm is especially useful in applications like loop closure in visual odometry, SLAM, and SfM. Various strategies for loop closure ex-

ist [5–7, 11, 24, 27], most of which involve either computing the absolute orientation to align landmarks, or PnP algorithms used to localize each camera individually. Iterative Closest Point (ICP) [1, 28] methods may also be used to align two 3D point clouds, though they are extremely sensitive to initialization and are often slow to converge. In contrast, our method uses 2D-3D correspondences. Additionally, these methods do not directly estimate relative scale, but rather estimate it as a preprocessing step by, for instance, using the distance between cameras or the depth of 3D points to estimate relative scale [4, 27]. Estimating similarity transformations from 2D-2D correspondences is currently an open problem in computer vision. Our proposed algorithm is a replacement for the aforementioned loop-closing algorithms and can estimate the similarity transformation directly and efficiently even for a large number of correspondences.

3 Problem Statement

Our aim is to solve the generalized pose-and-scale problem given n 2D-3D correspondences. That is, we would like to determine the pose and internal scale of a generalized camera with respect to n known 3D points. This is equivalent to aligning the two coordinate systems that define the generalized camera and the 3D points. The generalized camera model jointly considers image observations from multiple cameras such that each observation i has a ray origin at point q_i and a unit direction $\bar{r_i}$. The corresponding world point is denoted r_i. We want to find a rotation R, translation t and scale s such that the image rays coincide with the world points:

$$sq_i + \alpha_i \bar{r_i} = Rr_i + t, \qquad i = 1, \ldots, n \tag{1}$$

where α_i is a scalar which stretches the image ray such that it meets the world point r_i such that $\alpha_i = ||Rr_i + t - sq_i||$. We use the Cayley-Gibbs-Rodriguez parameterization of the rotation matrix such that R can be formed with just three unknowns. When considering all n correspondences, there exists $7 + n$ unknown variables (3 for rotation, 3 for translation, 1 for scale, and 1 unknown depth per observation). The gPnP+s problem can be formulated from Eq. (1) as a non-linear least-squares minimization such that the sum of squared measurement errors is minimized. Thus, we aim to minimize the cost function:

$$C(R, t, s) = \sum_{i=1}^{n} ||\bar{r_i} - \frac{1}{\alpha_i}(Rr_i + t - sq_i)||^2. \tag{2}$$

This non-linear least squares problem can be solved with iterative methods such as Gauss-Newton, however, these techniques are sensitive to initialization and only converge to a single local minimum. In Section 4, we describe our method for directly solving for all minima of a slightly modified cost function without the need for initialization. We call our method the generalized Direct Least Squares (gDLS) solution, as it is a generalization of the DLS algorithm presented in [10].

4 Solution Method

The geometric constraint equation of Eq. (1) leads to a non-linear system of equations that can be minimized by a least squares solver that minimizes Eq. (2). We would instead like to rewrite this system of equations in terms of fewer unknowns. Specifically, we can rewrite this equation solely in terms of the unknown rotation, R. When we relax the constraint that $\alpha_i = ||Rr_i + t - sq_i||$ and treat each α_i as a free variable, α_i, s, and t appear linearly and can be easily reduced from Eq. (2). Note that this relaxation is reasonable since solving the optimality conditions results in $\alpha_i^* = z_i^\top (Rr_i + t - sq_i)$ where z_i is $\bar{r_i}$ corrupted by measurement noise.

We begin by rewriting our system of equations from Eq. (1) in matrix-vector form:

$$\underbrace{\begin{bmatrix} \bar{r_1} & & & q_1 & -I \\ & \ddots & & \vdots & \vdots \\ & & \bar{r_n} & q_n & -I \end{bmatrix}}_{\text{A}} \underbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ s \\ t \end{bmatrix}}_{\text{x}} = \underbrace{\begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}}_{\text{W}} \underbrace{\begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}}_{\text{b}} \tag{3}$$

$$\Leftrightarrow Ax = Wb, \tag{4}$$

where A and b consist of known and observed values, x is the vector of unknown variables we will eliminate from the system of equations, and W is the block-diagonal matrix of the unknown rotation matrix. From Eq. (3), we can create a simple expression for x:

$$x = (A^\top A)^{-1} A^\top W b = \begin{bmatrix} U \\ S \\ V \end{bmatrix} Wb. \tag{5}$$

We have partitioned $(A^\top A)^{-1} A^\top$ into constant matrices U, S, and V such that the depth, scale, and translation parameters are functions of U, S, and V respectively. Matrices U, S, and V can be efficiently computed in closed form by exploiting the sparse structure of the block matrices (see Appendix A for the full derivation). Note that α_i, s, and t may now be written concisely as linear functions of the rotation:

$$\alpha_i = u_i^\top Wb \tag{6}$$

$$s = SWb \tag{7}$$

$$t = VWb, \tag{8}$$

where $u_i^\top$ is the i-th row of U. Through substitution, the geometric constraint equation (1) can be rewritten as:

$$\underbrace{SWb}_{s} q_i + \underbrace{u_i^\top Wb}_{\alpha_i} \bar{r_i} = Rr_i + \underbrace{VWb}_{t}. \tag{9}$$

This new constraint is quadratic in the three unknown rotation variables given by the Cayley-Gibbs-Rodriguez representation.

4.1 A New Least Squares Cost Function

The new geometric constraint equation (9) assumes noise-free observations. We assume that each observation is noisy, with a zero mean noise η_i. We can denote our noisy observations as $\bar{z}_i = \bar{r}_i + \eta_i$. We can rewrite our measurement constraint in terms of our noisy observation:

$$SWbq_i + u_i^\top Wb(\bar{z}_i - \eta_i) = Rr_i + VWb \tag{10}$$

$$\Rightarrow \eta_i' = SWbq_i + u_i^\top Wb\bar{z}_i - Rr_i - VWB, \tag{11}$$

where η_i' is a zero-mean noise term that is a function of η_i (but whose covariance depends on the system parameters, as noted by Hesch and Roumeliotis [10]). We evaluate u_i, S, and V at $\bar{r}_i = \bar{z}_i$ without loss of generality. Observe that u_i can be eliminated from Eq. 11 by noting that:

$$UWb = \begin{bmatrix} \bar{z_i}^\top & & \\ & \ddots & \\ & & \bar{z_n}^\top \end{bmatrix} Wb - \begin{bmatrix} \bar{z_1}^\top q_1 \\ \vdots \\ \bar{z_n}^\top 1_n \end{bmatrix} SWb + \begin{bmatrix} \bar{z_1}^\top \\ \vdots \\ \bar{z_n}^\top \end{bmatrix} VWb \tag{12}$$

$$\Rightarrow u_i^\top Wb = \bar{z_i}^\top Rr_i - \bar{z_i}^\top q_i SWbq_i + \bar{z_i}^\top VWb. \tag{13}$$

Through substitution, Eq. (11) can be refactored such that:

$$\eta_i' = (\bar{z}_i\bar{z}_i^\top - I_3)(Rr_i - SWbq_i + VWb). \tag{14}$$

Eq. (14) allows the gPnP+s problem to be formulated as an unconstrained least-squares minimization in 3 unknown rotation parameters. We formulate our new least squares cost function, C', as the sum of the squared constraint errors from Eq. (14):

$$C'(R) = \sum_{i=1}^{n} ||(\bar{z}_i\bar{z}_i^\top - I_3)(Rr_i - SWbq_i + VWb)||^2 \tag{15}$$

$$= \sum_{i=1}^{n} \eta_i'^\top \eta_i'. \tag{16}$$

Thus, we have reduced the number of unknowns in our system from $7 + n$ to 3. This is an important part of our formulation, as it allows the size of the system we solve to be independent of the number of observations and thus scalable.

4.2 Macaulay Matrix Solution

We have reduced our original geometric constraint of Eq. (1) to a least-squares minimization as a function of only the three unknown rotation parameters in

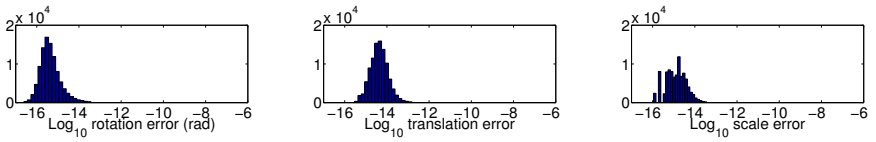


Fig. 2. Histograms of numerical errors in the computed similarity transforms based on 10^5 random trials with the minimal 4 correspondences

Eq. (15). That is, we wish to find unknown rotation parameters v_1, v_2, v_3 such that C' is minimized. This polynomial is of the same form and order as the DLS PnP algorithm [10], but with different coefficients. Thus, we can solve our least squares system of Eq. (15) with the same technique without modification. For a full explanation, see [10]. We will briefly summarize the technique in this section. The cost function of Eq. (15) may be minimized with the Gröbner basis technique; however, we found that the technique of [10] produced more accurate results at nearly the same efficiency.

We employ the Macaulay matrix [18] to determine the solution to our polynomial system. This matrix is formed from the partial derivatives of our cost function C' with respect to the three unknown rotation parameters. These three equations each equal zero when C' is minimal, so the roots of these polynomials produce the solution to our system. We consider one additional linear equation of the form $F_0 = u_0 + u_1v_1 + u_2v_2 + u_3v_3$ for random coefficients $u_0, \ldots, u_3$. This equation is generally non-zero at the roots of our polynomial system. The Macaulay matrix, M, formed from the three partial derivatives and the additional linear equation forms an extended polynomial system as a 120×120 matrix that contains coefficients to 120 monomials of the polynomial system.

Using the Schur complement trick, the Macaulay resultant matrix can be reduced to a 27×27 matrix whose eigenvectors correspond to the monomials of our cost function. The unknown rotation variables appear in these monomials, and can be directly extracted from the eigenvectors. This leads to 27 real and imaginary critical points, though the number of solutions can be reduced by considering only real solutions that place points in front of cameras. In practice, when using $n \geq 6$ points there exists only one valid minimum. After obtaining all minima, we evaluate the cost function Eq. (15) to determine the best orientation and compute the corresponding scale and translation through back substitution.

5 Experiments

5.1 Numerical Stability

We tested the numerical stability of our solution over 10^5 random trials. We generated random camera configurations that placed cameras (*i.e.*, ray origins) in the cube $[-1, 1] \times [-1, 1] \times [-1, 1]$ around the origin. 3D points were randomly placed in the volume $[-1, 1] \times [-1, 1] \times [2, 4]$. Ray directions were computed as unit vectors from camera origins to 3D points. An identity similarity transformation

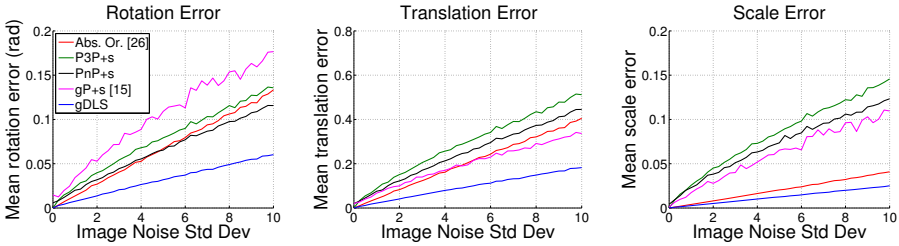


Fig. 3. We compared similarity transform algorithms with increasing levels of image noise to measure the pose error performance: the absolute orientation algorithm of [25], P3P+s, PnP+s, gP+s [26], and our algorithm, gDLS. Each algorithm was run with the same camera and point configuration for 1000 trials per noise level. Our algorithm has mean better rotation, translation, and scale errors for all levels of image noise.

was used (*i.e.*, $R = I$, $t = 0$, $s = 1$). For each trial, we computed solutions using the minimal 4 correspondences. We calculated the angular rotation error, the translation error, and the scale error for each trial, and plot the results in Figure 2. The errors are very stable, with 98% of all errors less than 10^{-12}.

5.2 Simulations with Noisy Synthetic Data

We performed two experiments with synthetic data to analyze the performance of our algorithm as the amount of image noise increases and as the number of correspondences increases. For both experiments we use a focal length of 800 and $[640, 480]$ resolution. Two cameras are placed randomly in the cube $[-1, 1] \times [-1, 1] \times [-1, 1]$ around the origin with three 3D points randomly placed in the volume $[-1, 1] \times [-1, 1] \times [2, 4]$. Both cameras observe each 3D point, so there are six total 2D-3D observations. Using the known 2D-3D correspondences, we apply a similarity transformation with a random rotation in the range of $[-30, 30]$ degrees about each of the x, y, and z axes, a random translation with a distance between 0.5 and 10, and a random scale change between 0.1 and 10. We measure the performance of the following similarity transform algorithms:

- **Absolute Orientation:** The absolute orientation method of Umeyama [25] is used to align the known 3D points to 3D points triangulated from 2D correspondences. This algorithm is only an alignment method and does not utilize any 2D correspondences.
- **P3P+s:** The P3P algorithm of Kneip *et al.* [13] is used to localize the first camera and the corresponding rotation and translation is used for the similarity transformation. The scale is then estimated from the median estimate from triangulated point matches. This process is repeated for all cameras, and the camera localization and scale estimation that yields the largest number of inliers is used as the similarity transformation.
- **PnP+s:** The similarity transformation is computed the same way as P3P+s, but the DLS PnP algorithm of Hesch and Roumeliotis [10] is used to localize

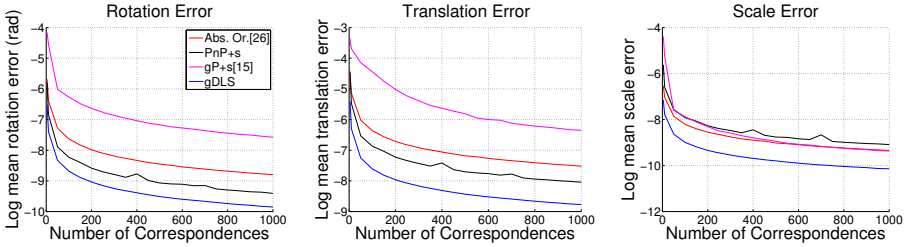


Fig. 4. We measured the accuracy of similarity transformation estimations as the number of correspondences increased. The mean of the log rotation, translation, and scale errors are plotted from 1000 trials at each level of correspondences used. A Gaussian image noise of 0.5 pixels was used for all trials. We did not use P3P+s in this experiment because P3P only uses 3 correspondences. Our algorithm has better accuracy for all number of correspondences used and a runtime complexity of $O(n)$, making it ideal for use at scale.

each camera instead of P3P[1]. PnP+s uses $n \geq 3$ 2D-3D correspondences, whereas P3P+s can only use 3.

- **gP+s:** The minimal solver of Ventura *et al.* [26] is used with 2D-3D correspondences from all cameras. While the algorithm is intended for the minimal case of $n = 4$ correspondences, it can compute an overdetermined solution for $n \geq 4$ correspondences.
- **gDLS:** The algorithm presented in this paper, which uses $n \geq 4$ 2D-3D correspondences from all cameras.

After running each algorithm on the same camera and point configuration, we calculate the rotation, translation, and scale errors with respect to the known similarity transformation.

Image Noise Experiment: For our first experiment, we evaluated the similarity transformation algorithms under increased levels of image noise. Using the configuration described above, we increased the image noise from 0 to 10 pixels standard deviation, and ran 1000 trials at each level. Our algorithm outperforms each of the other similarity transformation algorithms for all levels of image noise, as shown in Figure 3. The fact that our algorithm returns all minima of our modified cost function is advantageous under high levels of noise as we are not susceptible to getting stuck in a bad local minimum. This allows our algorithm to be very robust to image noise as compared to other algorithms.

Scalability Experiment: For the second experiment, we evaluate the rotation, translation, and scale error as the number of 2D-3D correspondences increases. We use the same camera configuration described above, but vary the number of 3D points used to compute the similarity transformation from 4 to 1000. Each

[1] We found that DLS [10] performed comparably to alternative algorithms such as OPnP [29] in the context of PnP+s.

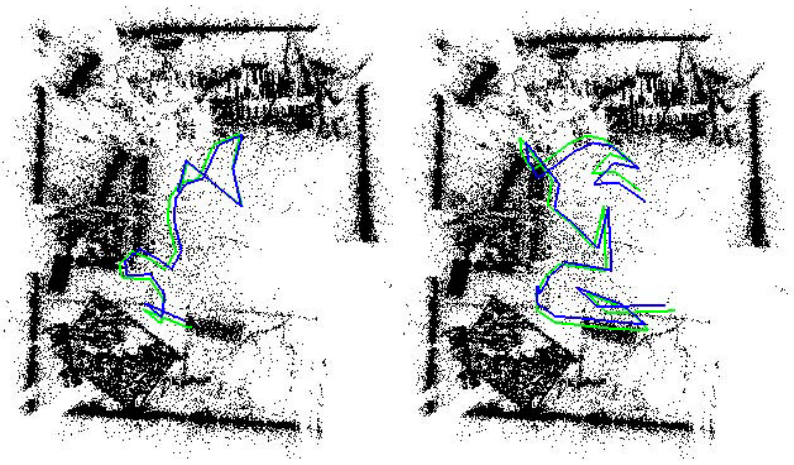


Fig. 5. In our real data experiments we compute the similarity transformation that aligns cameras from a SLAM system (blue) to a preexisting SfM reconstruction using 2D-3D correspondences. The ground truth positions (green) were recorded with a high-accuracy ART-2 tracker.

3D point is observed by both cameras. We ran 1000 trials for each number of correspondences used with a Gaussian noise level of 0.5 pixels standard deviation for all trials. We did not use the P3P+s algorithm for this experiment since P3P is a minimal solver and cannot utilize the additional correspondences. Although gP+s is a minimal solver, it can utilize all n correspondences in an overdetermined solution. The accuracy of each similarity transformation algorithm as the number of correspondences increases is shown in Figure 4. Our algorithm performs very well as the number of correspondences increases, and is more accurate than alternative algorithms for all numbers of correspondences tested. Further, our algorithm is $O(n)$ so the performance cost of using additional correspondences is favorable compared to the alternative algorithms (see Section 5.4 for a full runtime analysis).

5.3 SLAM Registration with Real Images

We tested our solver for registration of a SLAM reconstruction with respect to an existing SfM reconstruction using a dataset from [26]. This dataset consists of an indoor reconstruction with precise 3D and camera position data obtained with an ART-2 optical tracker. Several image sequences in this environment were run through a real-time keyframe-based SLAM system to obtain a local tracking sequence that can be registered to the ground-truth environment via a similarity transform (see Figure 5). SIFT keypoints were used to establish 2D-3D correspondences using approximate nearest-neighbor techniques and a ratio test. We compare our method to several other techniques for registering SLAM maps to a global point cloud. We compare our algorithm to the absolute

Table 1. Average position error in centimeters for aligning a SLAM sequence to a pre-existing SfM reconstruction. An ART-2 tracker was used to provide highly accurate ground truth measurements for error analysis. Camera positions were computed using the respective similarity transformations and the mean camera position error of each sequence is listed below. Both the minimal version of our solver, gDLS4, and the nonminimal gDLS10 (both shown in bold below) outperform the alternative methods.

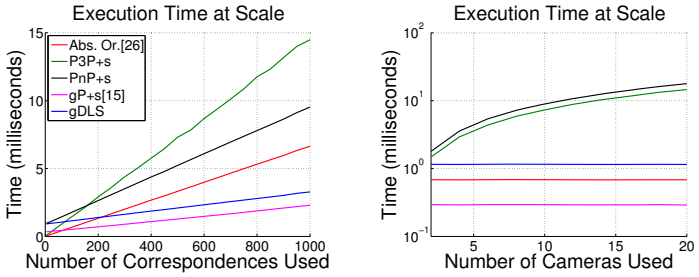
Sequence	# Images	Abs. Ori. [25]	P3P+s	PnP+s	gP+s [26]	**gDLS4**	**gDLS10**
office1	9	6.37	6.14	4.38	6.12	**3.97**	**3.04**
office2	9	8.09	7.81	6.90	9.32	**5.89**	**5.80**
office3	33	8.29	9.31	8.89	6.78	**6.08**	**4.69**
office4	9	4.76	4.48	3.98	4.00	**3.81**	**3.35**
office5	15	3.63	3.42	3.39	4.75	**3.39**	**3.09**
office6	24	5.15	5.23	5.01	5.91	**4.51**	**4.45**
office7	9	6.33	7.08	7.16	7.07	**4.65**	**3.21**
office8	11	4.72	4.85	3.62	4.59	**2.85**	**2.45**
office9	7	8.41	8.44	4.08	6.65	**3.19**	**2.33**
office10	23	5.88	6.60	5.73	5.88	**4.94**	**4.87**
office11	58	5.19	4.85	4.80	6.74	**4.77**	**4.65**
office12	67	5.53	5.20	4.97	4.86	**4.81**	**4.75**

orientation algorithm [25], P3P+s, PnP+s (using 10 correspondences), and gP+s [26] as described in Section 5.2. All algorithms are used in a PROSAC loop except for the absolute orientation algorithm which is used in a RANSAC loop. The absolute orientation algorithm does not use feature matches (it only aligns 3D point clouds) and thus cannot utilize matching scores in a PROSAC loop. We compare these algorithms to two versions of our solver: one using the minimal 4 correspondences (gDLS4) inside a PROSAC loop, and an over-determined solution using 10 correspondences (gDLS10) inside a PROSAC loop. No refinement is performed after RANSAC/PROSAC for any of the algorithms.

We compute the average position error of all keyframes with respect to the ground truth data. The position errors, reported in centimeters, are shown in Table 1. Both the minimal and non-minimal versions of our solver give higher accuracy results for every image sequence tested compared to alternative algorithms. By using the generalized camera model, we are able to exploit 2D-3D constraints from multiple cameras at the same time as opposed to considering only one camera (such as P3P+s and PnP+s). This allows the similarity transformation to be optimized for all cameras and observations simultaneously, leading to high-accuracy results.

5.4 Runtime Analysis

In this section we present a runtime analysis of each similarity transformation algorithm when registering n points and m cameras. The absolute orientation algorithm requires aligning sets of 3D points, making it $O(n)$. In theory, the covariance matrix used to compute the least-squares solution has a lower bound



(a) Increase number of points, constant number of cameras

(b) Constant number of points, increasing number of cameras

Fig. 6. (a) We plot the mean execution time while increasing the number of 2D-3D correspondences used and keeping the number of cameras constant at 2. Our gDLS method is slightly slower than gP+s [26] though our method is much more accurate at scale (Figure 4). (b) The number of cameras was increased while using the 100 2D-3D correspondences (each point is seen in every camera). The runtimes of the absolute orientation method [25], gP+s [26], and our gDLS method are independent of the number of cameras.

of $O(n)$, however, in practice it is often slower. The P3P+s algorithm relies on the extremely efficient P3P algorithm (which can be considered to run in constant time), however, in order to recover scale it must triangulate points across all cameras, leading to $O(n)$ complexity. Further, P3P+s computes the similarity transformation by localizing each camera and estimating scale. Thus, for m cameras the expected runtime is $O(mn)$. The PnP+s algorithm operates the same way as P3P, though the PnP algorithm is at best $O(n)$, resulting in an overall runtime of $O(mn^2)$. In practice, the PnP+s algorithm outperforms the P3P+s algorithm as the number of points increases as shown in Figure 6. This is because the P3P algorithm returns 4 solutions, while the DLS PnP algorithm returns only 1 solution in most cases. All n points must be triangulated for each solution, leading to an increased runtime for P3P+s. The gP+s algorithm requires computing the null space of a matrix that is of size $2n$, which is $O(n)$ in theory though efficient in practice even for large n. The gP+s algorithm is roughly $2-8\times$ faster than our algorithm, however, as shown in Sections 5.2 and 5.3 it is less accurate than our algorithm.

As described in Section 4.1, our algorithm is $O(n)$ in the number of points, making the runtime favorable as the number of points increases. Additionally, our algorithm is independent of the number of cameras, as shown in Figure 6b. In our experiments conducted on a 2.26 GHz Quad-Core Mac Pro, we observed a mean runtime of roughly 0.842 milliseconds over 10^5 trials when using 4 points. When using 10 points, the mean runtime increased to roughly 0.863 milliseconds. The timing results as the number of correspondences and cameras increases is shown in Figure 6. This efficiency allows our method to be used in real-time within a RANSAC loop. Additionally, the fact that our method can be used with only

4 correspondences allows for the theoretical convergence rate of RANSAC to remain low compared to algorithms that require more correspondences.

6 Conclusion

In this work, we proposed a new solution to the pose-and-scale problem for generalized camera models with n 2D-3D correspondences. This problem is equivalent to computing a similarity transformation. Our method, gDLS, is flexible, accurate, and efficient. It can handle the minimal case of $n = 4$, as well as the overdetermined case of $n > 4$. We formulate a least squares cost function that can be solved efficiently as a system of third degree polynomials, resulting in a system that is $O(n)$ in the number of correspondences. This makes it applicable for real-time frameworks and is useful, for example, for loop closure with SLAM and visual odometry. We have evaluated our method on synthetic data to show the numerical stability, accuracy under image noise, and scalability of our method. We validated our method with experiments using real data which shows that our method is more accurate than other methods when computing the similarity transformation for registering reconstructions. Our gDLS algorithm has been made publicly available as part of the open source library Theia[2] [22].

Experiments have shown our method to be extremely accurate and efficient even at scale. Our method can be used on thousands of correspondences when the ground truth correspondences are known, or as a refinement step on inliers from a minimal estimation. However, as with all non-minimal pose solvers, it is difficult to make use of a large number of correspondences because of the likelihood of false features matches when using many correspondences. For future work, we plan to explore ways to increase robustness to false correspondences by incorporating feature distances into the similarity transform estimation process so that our method can be more readily used with thousands of correspondences.

Acknowledgments. The authors would like to thank Jonathan Ventura for his insights and for providing the SLAM datasets used in our real data experiments. This work was supported in part by UC MEXUS-CONACYT (Fellowships 212913), NSF Grant IIS-1219261, NSF Graduate Research Fellowship Grant DGE-1144085, NSF CAREER Grant IIS-0747520, and ONR Grant N00014-09-1-113.

Appendix

A Computing Matrices U, S, and V for Depth, Scale, and Translation

The constant matrices U, S, and V are used to recover depth, scale, and translation from the solution for the rotation matrix. These matrices are constructed

[2] `http://cs.ucsb.edu/~cmsweeney/theia`

as a function of known measurements from the matrix $(A^\top A)^{-1}A^\top$. Using the expression for A from Eq. (3), we have:

$$A^\top A = \left[\begin{array}{ccc|cc} 1 & & & \bar{r_1}^\top q_1 & -\bar{r_1}^\top \\ & \ddots & & \vdots & \vdots \\ & & 1 & \bar{r_n}^\top q_n & -\bar{r_n}^\top \\ \hline \multicolumn{3}{c|}{q_1^\top \bar{r_1} \ \dots \ q_n^\top \bar{r_n}} & \sum_{i=1}^n q_i^\top q_i & \sum_{i=1}^n -q_i^\top \\ \multicolumn{3}{c|}{-\bar{r_1} \ \dots \ -\bar{r_n}} & \sum_{i=1}^n -q_i & nI \end{array}\right] = \left[\begin{array}{c|c} \mathcal{A} & \mathcal{B} \\ \hline \mathcal{B}^\top & \mathcal{D} \end{array}\right], \tag{17}$$

where solid lines represent the block-matrix boundaries. Through block matrix inversion, we can conveniently solve for the inverse:

$$\begin{aligned} (A^\top A)^{-1} &= \left[\begin{array}{c|c} \mathcal{E} & \mathcal{F} \\ \hline \mathcal{G} & \mathcal{H} \end{array}\right] \\ \mathcal{E} &= I + \mathcal{B}\mathcal{H}\mathcal{B}^\top \\ \mathcal{F} &= -\mathcal{B}\mathcal{H} \\ \mathcal{G} &= -\mathcal{H}\mathcal{B}^\top \\ \mathcal{H} &= \left(\begin{bmatrix} \sum_{i=1}^n q_i^\top q_i & \sum_{i=1}^n -q_i^\top \\ \sum_{i=1}^n -q_i & nI \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^n q_i^\top \bar{r_i}\bar{r_i}^\top q_i & \sum_{i=1}^n -q_i^\top \bar{r_i}\bar{r_i}^\top \\ \sum_{i=1}^n -\bar{r_i}\bar{r_i}^\top q_i & \sum_{i=1}^n \bar{r_i}\bar{r_i}^\top \end{bmatrix} \right)^{-1}. \end{aligned} \tag{18}$$

Finally, we can compute U, S, and V from Eq. (18). Many of the terms can be simplified because of multiplications involving $\bar{r_i}^\top \bar{r_i} = 1$. This leaves us with a greatly simplified expression for U, S, and V:

$$\begin{aligned} \left[\begin{array}{c} U \\ \hline S \\ \hline V \end{array}\right] &= (A^\top A)^{-1} A^\top \\ U &= \begin{bmatrix} \bar{r_i}^\top & & \\ & \ddots & \\ & & \bar{r_n}^\top \end{bmatrix} + \mathcal{B} \begin{bmatrix} S \\ V \end{bmatrix} \\ \left[\begin{array}{c} S \\ \hline V \end{array}\right] &= -\mathcal{H}\mathcal{B}^\top \begin{bmatrix} \bar{r_i}^\top & & \\ & \ddots & \\ & & \bar{r_n}^\top \end{bmatrix} + \mathcal{H} \begin{bmatrix} q_1^\top & \dots & q_n^\top \\ -I & \dots & -I \end{bmatrix} \\ &= \mathcal{H} \begin{bmatrix} q_1^\top - q_1^\top \bar{r_1}\bar{r_1}^\top & \dots & q_n^\top - q_n^\top \bar{r_n}\bar{r_n}^\top \\ \bar{r_1}\bar{r_1}^\top - I & \dots & \bar{r_n}\bar{r_n}^\top - I \end{bmatrix}. \end{aligned} \tag{19}$$

References

1. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 4(2), 239–256 (1992)
2. Bujnak, M., Kukelova, Z., Pajdla, T.: A general solution to the p4p problem for camera with unknown focal length. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
3. Chen, C.S., Chang, W.Y.: On pose recovery for generalized visual sensors. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(7), 848–861 (2004)
4. Clemente, L.A., Davison, A.J., Reid, I.D., Neira, J., Tardós, J.D.: Mapping large loops with a single hand-held camera. In: Robotics: Science and Systems (2007)
5. Courchay, J., Dalalyan, A., Keriven, R., Sturm, P.: Exploiting loops in the graph of trifocal tensors for calibrating a network of cameras. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 85–99. Springer, Heidelberg (2010)
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 1052–1067 (2007)
7. Eade, E., Drummond, T.: Unified loop closing and recovery for real time monocular slam. In: Proc. British Machine Vision Conference, vol. 13, p. 136. Citeseer (2008)
8. Ess, A., Neubeck, A., Van Gool, L.J.: Generalised linear pose estimation. In: Proc. British Machine Vision Conference, pp. 1–10 (2007)
9. Fischler, M.A., Bolles, R.C.: 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)
10. Hesch, J., Roumeliotis, S.: A direct least-squares (dls) solution for pnp. In: Proc. of the International Conference on Computer Vision. IEEE (2011)
11. Klopschitz, M., Zach, C., Irschara, A., Schmalstieg, D.: Generalized detection and merging of loop closures for video sequences. In: Proc. 3D Data Processing, Visualization, and Transmission (2008)
12. Kneip, L., Furgale, P., Siegwart, R.: Using multi-camera systems in robotics: Efficient solutions to the npnp problem. In: Proc. International Conference on Robotics and Automation, pp. 3770–3776. IEEE (2013)
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: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2969–2976. IEEE (2011)
14. 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)
15. Kukelova, Z., Bujnak, M., Pajdla, T.: Polynomial eigenvalue solutions to minimal problems in computer vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(7), 1381–1393 (2012)
16. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the pnp problem. International Journal of Computer Vision 81(2), 155–166 (2009)
17. Li, Y., Snavely, N., Huttenlocher, D.: Location recognition using prioritized feature matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 791–804. Springer, Heidelberg (2010)
18. Macaulay, F.: Some formulae in elimination. In: Proc. London Mathematical Society, vol. 1(1), pp. 3–27 (1902)

19. Nistér, D., Stewénius, H.: A minimal solution to the generalised 3-point pose problem. Journal of Mathematical Imaging and Vision 27(1), 67–79 (2007)
20. Pless, R.: Using many cameras as one. In: Proc. IEEE Conference on Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II–587. IEEE (2003)
21. Schweighofer, G., Pinz, A.: Globally optimal o(n) solution to the pnp problem for general camera models. In: Proc. British Machine Vision Conference, pp. 1–10 (2008)
22. Sweeney, C.: Theia Multiview Geometry Library: Tutorial & Reference. University of California, Santa Barbara, `http://cs.ucsb.edu/~cmsweeney/theia`
23. Tariq, S., Dellaert, F.: A multi-camera 6-dof pose tracker. In: Proc. International Symposium on Mixed and Augmented Reality, pp. 296–297. IEEE (2004)
24. Thrun, S., Montemerlo, M.: The graph slam algorithm with applications to large-scale mapping of urban structures. The International Journal of Robotics Research 25(5-6), 403–429 (2006)
25. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(4), 376–380 (1991)
26. Ventura, J., Arth, C., Reitmayr, G., Schmalstieg, D.: A minimal solution to the generalized pose-and-scale problem. Accepted to: IEEE Conference on Computer Vision and Pattern Recognition (2014)
27. Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., Tardós, J.: An image-to-map loop closing method for monocular slam. In: Proc. International Conference on Intelligent Robots and Systems, pp. 2053–2059. IEEE (2008)
28. Yang, J., Li, H., Jia, Y.: Go-icp: Solving 3d registration efficiently and globally optimally. In: Proc. The International Conference on Computer Vision. IEEE (2013)
29. Zheng, Y., Kuang, Y., Sugimoto, S., Astrom, K., Okutomi, M.: Revisiting the pnp problem: A fast, general and optimal solution. In: Proc. of the International Conference on Computer Vision. IEEE (December 2013)