Computing Similarity Transformations from Only Image Correspondences

Chris Sweeney[1] Laurent Kneip[2] Tobias Höllerer[1] Matthew Turk[1]

[1]University of California Santa Barbara
{cmsweeney, holl, mturk}@cs.ucsb.edu

[2]Research School of Engineering
Australian National University
laurent.kneip@anu.edu.au

Abstract

We propose a novel solution for computing the relative pose between two generalized cameras that includes reconciling the internal scale of the generalized cameras. This approach can be used to compute a similarity transformation between two coordinate systems, making it useful for loop closure in visual odometry and registering multiple structure from motion reconstructions together. In contrast to alternative similarity transformation methods, our approach uses 2D-2D image correspondences thus is not subject to the depth uncertainty that often arises with 3D points. We utilize a known vertical direction (which may be easily obtained from IMU data or vertical vanishing point detection) of the generalized cameras to solve the generalized relative pose and scale problem as an efficient Quadratic Eigenvalue Problem. To our knowledge, this is the first method for computing similarity transformations that does not require any 3D information. Our experiments on synthetic and real data demonstrate that this leads to improved performance compared to methods that use 3D-3D or 2D-3D correspondences, especially as the depth of the scene increases.

1. Introduction

Computing the relative pose between two cameras is one of the most fundamental problems in multi-view geometry. A generalization of this problem is to compute the relative pose between two sets of multiple cameras. Each set of multiple cameras may be described by the generalized camera model which allows a set of image rays that do not necessarily have the same ray origin to be represented in a uniform expression. Generalized cameras are extremely useful for many practical applications such as omni-directional camera systems and vehicle-mounted multi-camera systems. Solutions exist for computing relative pose between generalized cameras [9, 15, 21]; however, these methods require that the internal scale of the multi-camera system (*i.e.*, the distance between all camera centers within the multi-camera system) is known. This limits the use of generalized cameras to cases where scale calibration can be easily captured. In this paper, we provide a further generalization of the relative pose problem and remove the requirement of known scale to solve a new problem: the generalized relative pose and scale problem.

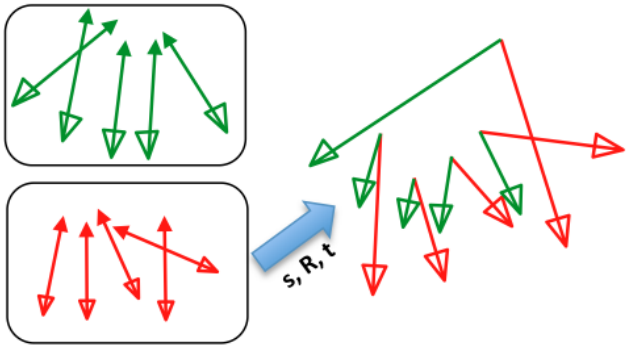

Figure 1. We present a method to solve the generalized relative pose and scale problem. We first align the generalized cameras to a common vertical direction then use image rays obtained from 5 2D-2D correspondences to solve for the remaining degrees of freedom. Solving this problem is equivalent to computing a similarity transformation

Reconciling the relative pose between two generalized cameras as well as the unknown scale is equivalent to recovering a 7 degrees-of-freedom (d.o.f.) similarity transformation. This allows for a much broader use of generalized cameras. In particular, similarity transformations can be used for loop closure in SLAM (where scale drift occurs) and for merging multiple Structure-from-Motion (SfM) reconstructions when the scale between the reconstructions is unknown. This problem arises frequently because scale cannot be explicitly recovered from images alone without metric calibration, so developing accurate, efficient, and robust methods to solve this problem is of great importance.

Using generalized cameras to compute similarity transformations was recently introduced with the generalized

pose-and-scale problem [24, 28] which computes similarity transformations from 4 or more 2D-3D correspondences. These methods, however, are subject to the quality of the estimated 3D points. In SfM, it is common for 3D points to have a high uncertainty especially as the depth of the 3D point relative to the cameras that observe it increases. The solution proposed in this paper solves the generalized relative pose and scale problem from 5 2D-2D correspondences, eliminating the dependence on potentially uncertain 3D points. We solve this problem in two steps. First, we align the vertical directions of the generalized cameras and describe a robust method for performing this alignment when IMU data is not available (*c.f*. Section 4.1). Then we utilize the knowledge of the vertical direction to formulate the generalized relative pose and scale problem as a Quadratic Eigenvalue Problem which is simple to construct and efficient to solve. We demonstrate that our method has comparable or better accuracy to the state-of-the-art methods through experiments with synthetic and real data. We have provided an efficient C++ implementation of our method as publicly available open-source software incorporated into the Theia structure from motion library[22] [1].

The rest of the paper is as follows: Section 2 provides an overview of related work. The generalized relative pose and scale problem is then introduced in Section 3. Our approach is described in detail in Section 4 along with a description of several techniques for estimating the vertical direction. 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 has been much interest in developing minimal pose solvers in computer vision [5, 10, 11, 12, 16, 23]. Most works have been focused on single perspective camera setups, though there has recently been an increased interested in developing methods for generalized cameras [14, 15, 9, 17, 21, 23]. We build on previous work for generalized cameras as well as work that estimates similarity transformations for SfM and SLAM loop closure.

Grossberg and Nayar first introduced the generalized camera model [6] which has since become the standard representation for multi-camera setups [7, 18], particularly for multi-camera rigs on moving vehicles [13, 23]. Generalized cameras can produce highly stable motion estimates because of their potentially large visual coverage. Stewénius *et al*. solved the problem of determining the relative pose between generalized cameras using 6 correspondences. The authors employ the Gröbner basis techniqe to compute up to 64 solutions. However, their method is very slow and the authors advise that it is not suitable for real-time use. Li *et al*. [15] provide an efficient linear approach to the generalized relative pose problem but it requires 17 correspondences, making it insuitable for use in a RANSAC scheme in low-inlier scenarios. Níster and Stewénius solve the absolute pose problem for generalized cameras from 3 correspondences [17] by solving for the roots of an octic polynomial. All of these methods are limited because they assume that the internal scale between the two generalized cameras is known. This is a suitable assumption if you are computing the relative pose between known cameras or if a metric calibration is available. However, there are many cases where this calibration is difficult or impossible to accurately obtain and so the scale ambiguity must be estimated.

Ventura *et al*. [28] presented the first minimal solution to the generalized absolute pose and scale problem. This method uses 4 2D-3D correspondences and employ the Gröbner basis technique to estimate rotation, translation, and scale to localize a generalized camera efficiently. Sweeney *et al*. [24] extended this method to a globally optimal non-minimal solver that has significantly increased accuracy, however, it is much slower than the work of Ventura *et al*. The accuracy of these methods degrades as the depth of the scene increases because of the reliance on 3D points. Further, using these methods to repeatedly merge many reconstructions will give different results depending on the order in which the reconstructions are merged. This is because the 3D points are given greater importance in the localization.

In contrast, our method utilizes 2D-2D correspondences and thus avoids relying on 3D points whose uncertainty depends directly on the depth from the observing cameras. To our knowledge, no previous work has been presented that computes the generalized relative pose and scale. The proposed algorithm is especially useful in applications like loop closure in visual odometry, SLAM, and SfM. Most strategies for loop closure involve computing the absolute orientation to align known scene landmarks, or they utilize PnP algorithms repeatedly to localize individual cameras [2, 3, 4, 8, 25, 29]. Iterative Closest Point (ICP) [1, 30] methods may also be used to align two 3D point clouds, though are often slow to converge and depend heavily on initialization. Our proposed algorithm is a direct method that will return an estimate for a full 7 d.o.f. similarity transformation from just 5 correspondences and is effectively a drop-in replacement for the aforementioned loop closure methods.

3. The Generalized Relative Pose and Scale Problem

The generalized relative pose and scale problem is a direct generalization of the generalized relative pose problem. The generalized relative pose problem uses ray cor-

[1] The Theia library is located at: http://cs.ucsb.edu/~cmsweeney/theia/

respondences to compute the rotation and translation that will transform one set of rays so that they intersect with the second set of rays. Let f_i and f'_i be corresponding unit vectors that intersect in 3D space with ray origins o_i and o'_i. These rays can be represented in Plücker coordinates [19] such that:

$$l_i = \begin{pmatrix} f_i \\ o_i \times f_i \end{pmatrix} \text{ and } l'_i = \begin{pmatrix} f'_i \\ o'_i \times f'_i \end{pmatrix}. \quad (1)$$

The generalized epipolar constraint [18] that describes the intersection of two Plücker coordinates may then be written as:

$$(f_i \times Rf'_i)^\top t + f_i^\top([o_i]_\times R - R[o'_i]_\times)f'_i = 0, \quad (2)$$

where R and t are the rotation and translation that transform f'_i and o'_i such that the ray correspondences intersect in 3D space. This problem has been solved previously with minimal [21], linear [15], and nonlinear approaches [9]. However, these methods assume that the scale between the two generalized cameras has been reconciled yet in many cases the scale is not available or may be inherently ambiguous without metric calibration (*e.g.*, in SfM reconstructions). Thus, we are interested in additionally solving for the unknown scale transformation between the two generalized camera.

To solve the generalized relative pose and scale problem we must additionally recover the unknown scale s that stretches the ray origins o'_i. Thus, the generalized epipolar constraint becomes:

$$(f_i \times Rf'_i)^\top t + f_i^\top([o_i]_\times R - Rs[o'_i]_\times)f'_i = 0 \quad (3)$$

$$(f_i \times Rf'_i)^\top t - sf_i^\top R[o'_i]_\times f'_i + f_i^\top[o_i]_\times Rf'_i = 0. \quad (4)$$

Inspired by [9] and [23], this equation may be rewritten as:

$$m_i^\top \cdot \tilde{t} = 0, \text{where} \quad (5)$$

$$m_i = \begin{pmatrix} f_i \times Rf'_i \\ -f_i^\top R[o'_i]_\times f'_i \\ f_i^\top[o_i]_\times Rf'_i) \end{pmatrix} \text{ and } \tilde{t} = \begin{pmatrix} t \\ s \\ 1 \end{pmatrix}. \quad (6)$$

The generalized relative pose and scale problem has 7 d.o.f. and thus requires 7 correspondences in the minimal case. We may stack the constraints from each correspondence such that

$$M^\top \tilde{t} = (m_1 \quad \dots \quad m_7)^\top \tilde{t} = 0. \quad (7)$$

Notice that the matrix M is a function of only the unknown rotation R and known parameters f_i and o_i. Let us consider the quaternion rotation parameterization $q = (x,\ y,\ z,\ \alpha)^\top$ such that the rotation matrix

$$R = 2(vv^\top + \alpha[v]_\times) + (\alpha^2 - 1)I, \quad (8)$$

where $v = (x,\ y,\ z)^\top$ and $[v]_x$ is the skew-symmetric cross product matrix of v. Thus, M is quadratic in the quaternion parameters and the generalized epipolar constraint of Eq. (7) is a 4-parameter Quadratic Eigenvalue Problem (QEP). No methods currently exist to directly solve a 4-parameter QEP and it should be noted that a non-iterative solution to Multiparameter Eigenvalue Problems with more than two parameters is an open problem in mathematics. However, an iterative optimization similar to [9] may be used to minimize the smallest eigenvalue of M and determine the unknowns if a good initialization is available. Indeed, solving the generalized relative pose and scale problem directly is quite difficult as there are 140 solutions in the minimal case, and a closed form solution would likely be very unstable.

4. Solution Method

To compute a solution to the generalized relative pose and scale problem we use a slight relaxation of the original problem so that we are left with a 1-parameter QEP that can be efficiently solved with only 5 correspondences. Rather than attempt to directly compute the full 7 d.o.f. similarity transformation, we solve the problem in two steps. First, we align the vertical direction of the generalized cameras. This removes 2 d.o.f. from the rotation, leaving only a single unknown d.o.f. in the rotation. It is important to note that aligning the vertical direction (and rotations in general) is independent of the scale and translation. Next, once the vertical direction is known, our 4-parameter QEP of Eq. (7) becomes a 1-parameter QEP and we can directly solve for the single remaining unknown rotation d.o.f. as well as the translation and scale.

In this section we will first discuss how to align the vertical direction even when IMU data is not available before providing a detailed explanation of how to solve for the generalized relative pose and scale from our simplified 1-parameter QEP.

4.1. Determining a Vertical Direction

The vertical direction of a camera provides knowledge of the gravity vector or the "up" direction of the camera relative to a known environment. Often, this direction may be obtained from IMU or accelerometer data that is increasingly provided on cameras and smartphones. These sensor measurements typically have an accuracy within 0.5 degrees. However, in cases where IMU data is not available the vertical direction may still be obtained with computer vision techniques. One common technique is to detect vertical vanishing points in each image and align this vanishing point to the "up" vector $(0,\ 1,\ 0)^\top$. This method has been proven to be efficient and accurate when used in the context of SfM [20].

Detecting and aligning vertical vanishing points is well-suited as a repeated operation on single images. However,

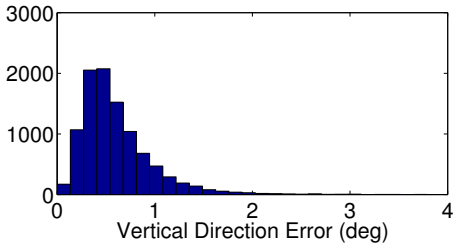

Figure 2. We measured the error in our vertical direction alignment method over 10,000 trials on our real data experiments. The error is quite small in all cases, resulting in a good initialization to our QEP solution.

using this method in the context of generalized cameras would be suboptimal because it ignores the fact that we have explicit knowledge of the relative poses between each individual camera in our generalized camera (*e.g.*, in a calibrated multi-camera rig or in a posed SLAM sequence). We would instead like to utilize this relative pose information to align the vertical direction for *all cameras* simultaneously with a single rotation.

We may assume that the vertical direction is $v = (0,\ 1,\ 0)^\top$ without loss of generality, and that we are attempting to rotate the generalized cameras so that the vertical directions are aligned. A straightforward procedure to align the vertical direction of a generalized camera is to first determine the vertical direction v_i of each camera within the generalized camera then compute a rotation R such that $d(Rv_i, v)$ is minimized over all cameras where $d(x, y)$ is the angular distance between two unit-norm vectors x and y. This formulation is most useful if the generalized camera is perfectly calibrated. In many cases, however, there is noise in the computed vertical direction. To increase robustness to noise we propose to instead compute R using only a subset of n cameras in a RANSAC-like procedure. We compute the alignment through many random trials and choose R such that the highest number of cameras have an error $d(Rv_i, v) < \tau$. We demonstrate the error in the RANSAC vertical alignment technique (using $n = 5, \tau = 3$ degrees and ground plane detection to determine the vertical direction v_i for each camera) in 10,000 trials in Figure 2. The dataset from Section 5.6 was used for this experiment, demonstrating that this method works well in practice.

4.2. A Quadratic Eigenvalue Problem Solution

Recall our quaternion rotation parameterization of Eq. 8. Now that the vertical directions of the two generalized cameras have been aligned, we have removed 2 d.o.f. from the unknown rotation and are left with solving one remaining unknown d.o.f. in the rotation. If we consider the rotation as an angle-axis rotation, it is clear to see that the vertical direction may serve as the axis and we must solve for the unknown rotation angle about this axis. In the quaternion parameterization, this means that $v = (0,\ 1,\ 0)^\top$ and we are left with solving for the unknown parameter α which is related to the rotation angle about the axis v [23].

Let us now consider this in the context of the generalized relative pose and scale problem. The intractable 4-parameter QEP from Eq. 7 has now been reduced to a single unknown parameter α in matrix M:

$$(\alpha^2 A + \alpha B + C) \cdot \tilde{t} = 0, \tag{9}$$

where A, B, and C are 5×5 matrices formed from matrix M in Eq. (7). Note that after the vertical directions have been aligned, the minimal solution to this problem only requires 5 correspondences instead of 7. We now have a standard 1-parameter QEP which has been thoroughly examined in linear algebra [26]. To solve this QEP, we first convert it to a Generalized Eigenvalue Problem of the form:

$$\begin{bmatrix} B & C \\ -I & 0 \end{bmatrix} z = s \begin{bmatrix} -A & 0 \\ 0 & -I \end{bmatrix} z, \tag{10}$$

where $z = \begin{bmatrix} \alpha\tilde{t}^\top & \tilde{t}^\top \end{bmatrix}^\top$ is the eigenvector and s is the eigenvalue. This can be converted to a standard eigenvalue problem by inverting the right-hand matrix of Eq. (10). The inverse is particularly simple and efficient in this case:

$$\begin{bmatrix} -A & 0 \\ 0 & -I \end{bmatrix}^{-1} = \begin{bmatrix} -A^{-1} & 0 \\ 0 & -I \end{bmatrix}.$$

The Generalized Eigenvalue Problem of Eq. (10) may now be reduced to a standard eigenvalue problem,

$$\begin{bmatrix} -A^{-1}B & -A^{-1}C \\ I & 0 \end{bmatrix} z = sz,$$

which can be solved with standard methods. The solution to this produces 10 candidate solutions where the eigenvalues correspond to α and the translation and scale may be extracted from the eigenvector. We may eliminate some of the candidate solutions by only considering real eigenvalues and the eigenvectors where the first 5 entries are equal to the last 5 entries scaled by α to ensure our solution is consistent with the construction of vector z.

4.3. A Closed Form Solution

An alternative method for solving Eq. (9) arises by examining the determinant. Note that M from Eq. (7) will be rank-deficient in non-degenerate cases, so it must hold that:

$$det(\alpha^2 A + \alpha B + C) = 0. \tag{11}$$

This leads to a degree 10 univariate polynomial in α such that the roots correspond to valid solutions to α. Further, it can be shown that this polynomial is always divisible by $\alpha^2 + 1$, leading to at most 8 real solutions. This result also

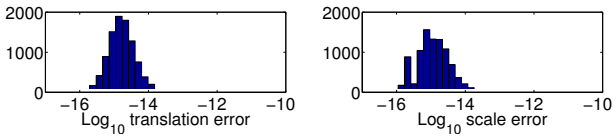

Figure 3. We measured the numerical stability of our algorithm with zero pixel noise and a perfect known axis of rotation. The translation and scale errors are very small, and the rotation error cannot be displayed because it was within the machine precision.

means our QEP method will have at most 8 real solutions since the roots of this polynomial correspond to the eigenvalues of our QEP. However, in practice this polynomial is ill-conditioned and solutions are very unstable. The significant loss in numerical precision and accuracy is not worth the 10-20% speed increase so we only consider the QEP method for the remainder of the paper.

5. Experiments

5.1. Numerical stability

We tested the numerical stability of our QEP method 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 cube $[-1,1]\times[-1,1]\times[4,6]$ and ray directions were computed as unit vectors from camera origins to 3D points. Correspondences were computed from image rays that observed the same 3D points. An identity similarity transformation was used (*i.e.*, $R = I$, $t = 0$, $s = 1$). For each trial, we computed solutions using the minimal 5 correspondences. We calculated the angular rotation error, the translation error, and the scale error for each trial, and plot the results in Figure 3. The errors are very stable, with 99% of all errors less than 10^{-12}.

5.2. Image noise experiment

We performed experiments on synthetic data to determine the effect of image noise on our algorithm. We compared our algorithm to three alternative algorithms: the gDLS algorithm [24], the gP+s algorithm [28], and the Absolute Orientation algorithm [27].

For our synthetic setup we generated two generalized cameras that each consist of 5 cameras randomly placed in the $2\times2\times2$ cube centered at the origin. 3D points were then randomly generated with a mean distance of 5 units from the origin, and correspondences were established as rays that observed the same 3D points such that each camera observes a single 3D point. We then applied a similarity transformation with a random rotation, a translation in a random direction with a random baseline in the range of $[0.1,\ 100]$, and a random scale in the range of $[0.1,\ 100]$ to the second generalized camera. Image noise is added to the second generalized camera and the similarity transformation is estimated. We report the angular rotation error, absolute translation error, and the normalized scale error $|s-\hat{s}|/s$.

For all synthetic experiments we used the ground truth vertical direction and added 0.5 degrees of gaussian noise to simulatel the real accuracy of vertical direction estimation for our algorithm (*c.f.* Figure 2). For the Absolute Orientation algorithm, we created 3D-3D matches by triangulating 3D points in the second generalized camera from the noisy image rays and used these 3D points to establish correspondences. Additionally, we used 5 correspondences for each algorithm for a fair comparison.

Using the setup described, we ran 1000 trials testing the accuracy of each algorithm as increasing levels of image pixel noise were added (Figure 4 top). Scenes were randomly generated for each trial, and all algorithms used the same scene configuration for a given trial. Our algorithm performed best at estimating the rotation and translation of the similarity transformation but is less accurate than the gDLS and Absolute Orientation algorithms for estimating scale. It should noted that the scale errors are very small for all algorithms. Our algorithm is robust to image noise because ray intersection in 3D space is a very tight constraint that is independent of the depth of the 3D point.

5.3. Scene depth experiment

In SLAM and SfM it is common to have 3D points with large and varying scene depth. It is especially important in the case of urban and large-scale SfM to be robust to large scene depths when computing a similarity transformation to align models. To examine our algorithm's robustness to scene depth, we ran an experiment using the same setup as above while increasing the mean scene depth from 5 units to 200 units. We used an image noise of 1 pixel for all depth levels and executed 1000 trials at each depth level. The results of our experiment are shown in the bottom row of Figure 4. It is clear to see that our algorithm is least affected by scene depth. The Absolute Orientation and gP+s algorithms completely degrade as the scene depth increases. The gDLS algorithm has comparable depth robustness to our algorithm in terms of the rotation and translation but is not as accurate at computing scale.

Conceptually, our algorithm has an advantage over gDLS [24], gP+s [28], and the Absolute Orientation algorithm [27] because it does not use 3D points and thus is not subject to uncertainty in the 3D position. It is well known that the uncertainty of a triangulated 3D point increases as the depth of the point relative to the baseline of the cameras observing it increases. Therefore, our algorithm should produce more accurate similarity transformations as the scene depth increase. Indeed, the results of this experiment support this notion.

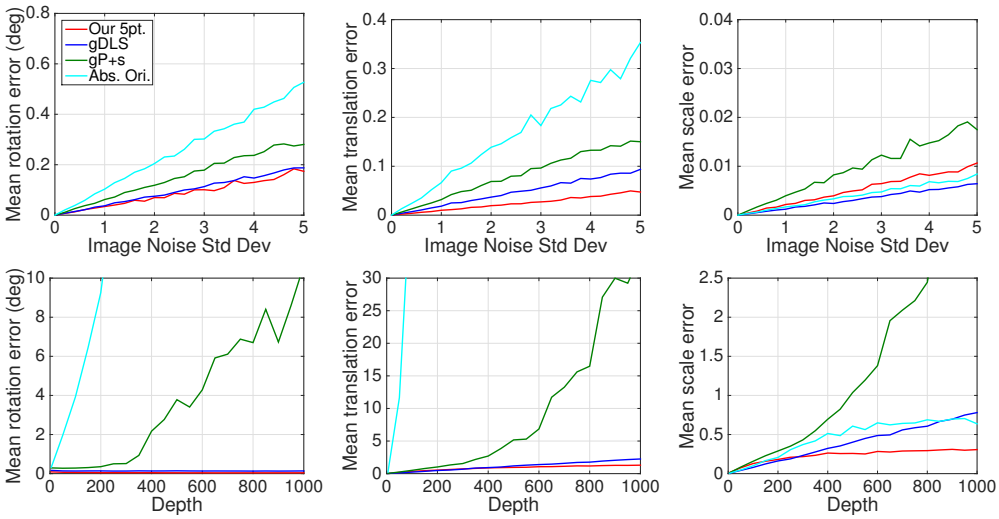

Figure 4. We measured the error in the computed similarity transformation as the amount pixel noise increased and plot the mean rotation, translation and scale error. All cameras were randomly generated within a $2 \times 2 \times 2$ cube centered at the origin. **Top row:** we generated random 3d points with an average depth of 5 units away from the orign. Our algorithm is the most accurate at computing the rotation and translation but is not as accurate at computing scale, however, the scale errors are very small for all algorithms. **Bottom row:** we kept the image noise at 1.0 pixels standard deviation while increasing the average depth of hte 3D points used to establish correspondences. Our algorithm is least affected by the change in scene depth meaning that it is robust to uncertainty in 3D point positions.

5.4. IMU noise experiment

We performed experiments on synthetic data to determine how the accuracy of the estimated vertical direction affects our algorithm. To simulate noise in the estimated vertical direction we added gaussian noise to a synthetic IMU ranging from 0 to 1 degree of standard deviation.

Using the same scene setup as the image noise experiment, we ran 1000 trials testing the similarity transformation accuracy as increasing levels of IMU noise were added (Figure 5). Standard mobile devices have less than 0.5 degree of IMU noise with high quality sensors often having less than 0.01 degrees of noise. Our algorithm demonstrates good accuracy in the presence of IMU noise within this range, verifying its robustness to potentially inaccurate vertical direction estimations.

5.5. Time Complexity

A major benefit of our method is that the QEP solution is simple to construct and very efficient. The most costly operations involved in our method are inversion of a 5×5 matrix and computing the eigenvectors and eigenvalues of a 10×10 matrix. Both of these operations are highly efficient on small matrices in standard linear algebra packages. Over 10,000 trials our algorithm ran with a mean execution time of 44μs. In comparison, the gDLS [24] method had a mean execution time of 606μs and the gP+s [28] method had a mean execution time of 118 μs. All timing experiments were run on a 2011 Macbook Pro with a 2GHz Intel Core i7 processor. While the Absolute Orientation algorithm is more efficient at 3μs, it is not as accurate or as robust to image noise and depth variance as our algorithm (*c.f.* Figure 4). Our algorithm has comparable accuracy to gDLS in the presence of image noise and is more robust to depth variance, yet it has a speedup of over $10\times$. This makes our algorithm more desirable for real-time use in a RANSAC scheme because of speed gains that will be realized.

5.6. Real-data experiments

Our method's robustness to 3D point and depth variance makes it well-suited for real-world applications. We tested the performance of our solver using the SLAM dataset from [28] that has highly accurate ground truth poses obtained with an ART-2 optical tracker for measuring the error of our similarity transformation registration method. Example images from this dataset are provided in Figure 6. For our experiment, we created an SfM reconstruction (using the ground truth poses) from one image sequence to use as our reference image sequence and point cloud. We then run 12 image sequences through a keyframe-based SLAM system to obtain a local tracking sequence that can be registered with respect to the reference sequence with a similarity transformation (see Figure 7). We then compute a similarity transformation in the following manner:

Our 5 pt.: 2D-2D feature correspondences are established between the reference and query image sequences using an approximate nearest neighbor search (ANN), and the vertical directions are aligned using ground plane detection and computing the normal. These correspondences are then

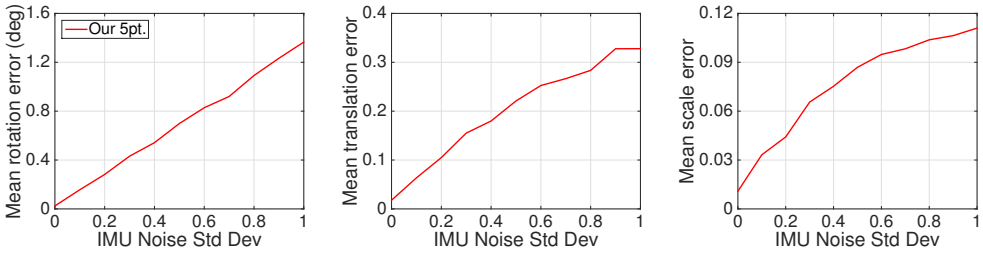

Figure 5. Using the same scene configuration as the image noise experiment, we measured the similarity transformation error as noise was added to the synthetic IMU to perturb the vertical direction. We only show our algorithm since it is the only one that depends on knowledge of the vertical direction. We used 1 pixel of image noise for all experiments. For levels of IMU noise expected on mobile devices (less than 0.5 degrees) our algorithm still maintains good accuracy, demonstrating robustness to noise in the vertical direction estimation.

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. Our method is has comparable or better accuracy than the state-of-the-art method, gDLS, but does not require any 3D points.

Sequence	# Images	Abs. Ori. [27]	gP+s[28]	gDLS [24]	Our 5 pt.
office1	9	6.37	6.12	**3.97**	4.30
office2	9	8.09	9.32	5.89	**4.17**
office3	33	8.29	6.78	6.08	**5.10**
office4	9	4.76	4.00	3.81	**2.61**
office5	15	3.63	4.75	**3.39**	3.41
office6	24	5.15	5.91	**4.51**	4.81
office7	9	6.33	7.07	4.65	**4.06**
office8	11	4.72	4.59	**2.85**	3.12
office9	7	8.41	6.65	3.19	**2.62**
office10	23	5.88	5.88	4.94	**3.55**
office11	58	5.19	6.74	**4.77**	5.03
office12	67	5.53	4.86	4.81	**4.12**

used in a RANSAC loop with the 5 pt. method described in this paper to determine a similarity transformation.

gDLS: We obtain 2D-3D correspondences with an ANN search between the 3D points in the point cloud generated by the reference sequence and the 2D image features in the query sequences. These correspondences are then used in a RANSAC loop using the minimal number of 4 correspondences with the gDLS algorithm of Sweeney *et al*. [24].

gP+s: We obtain 2D-3D correspondences in the same way as the gDLS method and use these correspondences in a RANSAC loop with the algorithm of Ventura *et al*. [28] to estimate the similarity transformation. This method requires 4 correspondences in the minimal case.

Absolute Orientation: The absolute orientation method of Umeyama [25] is used to align the 3D points from the reference point cloud to 3D points triangulated from 2D correspondences in the query point cloud. Correspondences are determined from an ANN search of the mean descriptor of the triangulated point and the 3D points in the reference point cloud. We use 4 correspondences for this method.

After applying the computed similarity transformation directly from RANSAC (*i.e*., no refinement is performed), we compute the average position error of all keyframes with respect to the ground truth data. We report the mean position error of all keyframes in the image sequence (in centimeters) over 1000 trials in Table 1. Our method performs better than all other methods in most of the scenes. The globally optimal gDLS algorithm [24] is the only method that is competitive with our algorithm. We expect that our algorithm will perform even better for large-scale SfM applications. However, acquiring ground truth datasets for large-scale SfM is difficult and we leave the incorporation and evaluation of our algorithm into a large-scale hierarchical SfM pipeline for future work.

6. Conclusion

We have presented a new problem called the generalized relative pose and scale problem and to our knowledge provide the first solution to this problem. The generalized relative pose and scale problem is equivalent to estimating a 7 d.o.f. similarity transformation and so this work is useful for loop closure in visual odometry and merging SfM

Figure 6. Example images from our real data experiments. The images created a SLAM sequence that was then aligned to a reference sequence with our method to estimate a similarity transformation.

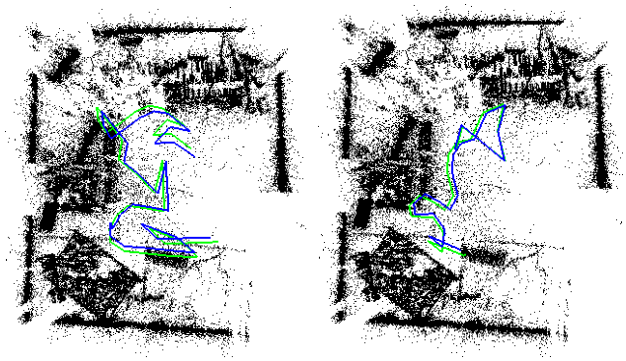

Figure 7. We compare our method with several alternative methods for computing similarity transformation using a dataset comprised of SLAM sequences that contain highly accurate ground truth poses. Each method is used to align 12 image sequences and the camera position errors are reported in Table 1. Green represents the ground truth SLAM sequence and blue SLAM sequence after applying the similarity transformation with our method in a RANSAC scheme.

reconstructions. We showed that the standard generalized relative pose and scale problem leads to an intractable 4-parameter QEP and instead provide a two step solution to the problem where we first align the vertical directions of all cameras then reduce the problem to a 1-parameter QEP that can be solved with standard linear algebra. Our method is simple, efficient, and robust to image noise and scene depth. We show on synthetic and real data experiments that our method has comparable or better performance to alternative algorithms. We have published a C++ implementation of our algorithm as open source software for fellow researchers to utilize. In future work, we plan to remove the necessity for vertical alignment to allow additional flexibility to our algorithm, and would like to incorporate this method into a large scale multi-camera SfM pipeline where the scale of reconstructions may be ambiguous.

7. Acknowledgements

This work was supported in part by NSF Grant IIS-1219261 and NSF Graduate Research Fellowship Grant DGE-1144085. The work has furthermore received support from ARC grants DP120103896 and DP130104567.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(2):239–256, 1992. 2

[2] J. Courchay, A. Dalalyan, R. Keriven, and P. Sturm. Exploiting loops in the graph of trifocal tensors for calibrating a network of cameras. In *European Conference on Computer Vision*, pages 85–99. Springer, 2010. 2

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007. 2

[4] E. Eade and T. Drummond. Unified loop closing and recovery for real time monocular slam. In *Proc. British Machine Vision Conference*, volume 13, page 136. Citeseer, 2008. 2

[5] F. Fraundorfer, P. Tanskanen, and M. Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *Proc. of the European Conference on Computer Vision*, pages 269–282. Springer, 2010. 2

[6] M. D. Grossberg and S. K. Nayar. A general imaging model and a method for finding its parameters. In *Proc. of IEEE Intn'l. Conf. on Computer Vision*, 2001. 2

[7] J.-H. Kim, H. Li, and R. Hartley. Motion estimation for nonoverlapping multicamera rigs: Linear algebraic and l geometric solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1044–1059, 2010. 2

[8] M. Klopschitz, C. Zach, A. Irschara, and D. Schmalstieg. Generalized detection and merging of loop closures for video sequences. In *Proc. 3D Data Processing, Visualization, and Transmission*, 2008. 2

[9] L. Kneip and H. Li. Efficient computation of relative pose for multi-camera systems. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2014. 1, 2, 3

[10] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2969–2976. IEEE, 2011. 2

[11] Z. Kukelova, M. Bujnak, and T. Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *Proc. of Asian Conference on Computer Vision*, pages 216–229. Springer, 2011. 2

[12] Z. Kukelova, M. Bujnak, and T. Pajdla. Polynomial eigenvalue solutions to minimal problems in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1381–1393, 2012. 2

[13] G. H. Lee, F. Fraundorfer, M. Pollefeys, P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Muhlfellner, S. Wonneberger, et al. Motion Estimation for Self-Driving Cars With a Generalized Camera. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2013. 2

[14] G. H. Lee, M. Pollefeys, and F. Fraundorfer. Relative pose estimation for a multi-camera system with known vertical direction. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2014. 2

[15] H. Li, R. Hartley, and J.-h. Kim. A linear approach to motion estimation using generalized camera models. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1, 2, 3

[16] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. 2

[17] D. Nistér and H. Stewénius. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79, 2007. 2

[18] R. Pless. Using many cameras as one. In *Proc. IEEE Conference on Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–587. IEEE, 2003. 2, 3

[19] J. Plùcker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, 155:725–791, 1865. 3

[20] S. N. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In *Trends and Topics in Computer Vision*, pages 267–281. Springer, 2012. 3

[21] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision*, 2005. 1, 2, 3

[22] C. Sweeney. *Theia Multiview Geometry Library: Tutorial & Reference*. University of California, Santa Barbara. http://cs.ucsb.edu/~cmsweeney/theia. 2

[23] C. Sweeney, J. Flynn, and M. Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *Proc. of the International Conference on 3D Vision*. IEEE, 2014. 2, 3, 4

[24] C. Sweeney, V. Fragoso, T. Hollerer, and M. Turk. gdls: A scalable solution to the generalized pose and scale problem. In *European Conference on Computer Vision*, volume 8692, pages 16–31. Springer, 2014. 2, 5, 6, 7

[25] S. Thrun and M. Montemerlo. 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. 2

[26] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM review*, 43(2):235–286, 2001. 4

[27] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. 5, 7

[28] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg. A minimal solution to the generalized pose-and-scale problem. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014. 2, 5, 6, 7

[29] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. An image-to-map loop closing method for monocular slam. In *Proc. International Conference on Intelligent Robots and Systems*, pages 2053–2059. IEEE, 2008. 2

[30] J. Yang, H. Li, and Y. Jia. Go-icp: Solving 3d registration efciently and globally optimally. In *Proc. The International Conference on Computer Vision*. IEEE, 2013. 2