An Efficient Minimal Solution for Multi-Camera Motion

Jonathan Ventura
Department of Computer Science
University of Colorado Colorado Springs
jventura@uccs.edu

Clemens Arth Vincent Lepetit
Institute for Computer Graphics and Vision
Graz University of Technology
{arth,lepetit}@icg.tugraz.at

Abstract

We propose an efficient method for estimating the motion of a multi-camera rig from a minimal set of feature correspondences. Existing methods for solving the multi-camera relative pose problem require extra correspondences, are slow to compute, and/or produce a multitude of solutions. Our solution uses a first-order approximation to relative pose in order to simplify the problem and produce an accurate estimate quickly. The solver is applicable to sequential multi-camera motion estimation and is fast enough for real-time implementation in a random sampling framework. Our experiments show that our approach is both stable and efficient on challenging test sequences.

1. Introduction

A fundamental problem in computer vision is computing the motion of a camera, or multi-camera rig, from the observed movement of points in successive images. The camera motion between two image captures is called the relative pose. While the single-camera relative pose problem has been extensively investigated in prior work, the multi-camera case has been given attention only recently. However, multi-camera motion estimation is a critical technology for many emerging applications such as self-driving cars with vehicle-mounted cameras and aerial vehicles such as quadcopters.

Existing solutions to the multi-camera relative pose problem [30, 15, 20] have deficiencies such as slow computation, producing a huge number of solutions, or using extra correspondences above the minimal six. In order to deal with feature mismatches, any relative pose solution needs to be applied repeatedly in a random sampling loop such as RANSAC [8]. This motivates the need for a solution that is fast and uses the minimal number of correspondences in order to reduce the number of samples required.

In this work, we derive a novel solution to the generalized camera relative pose problem which is faster than existing solutions while maintaining similar accuracy. The insight in our work comes from applying a first-order approximation to relative pose, which simplifies the problem and leads to an efficient solution. Our method is intended to be applied to successive video frames from a moving multi-camera rig. This allows us to assume small camera motion between frames, which makes the approximated motion model an appropriate choice.

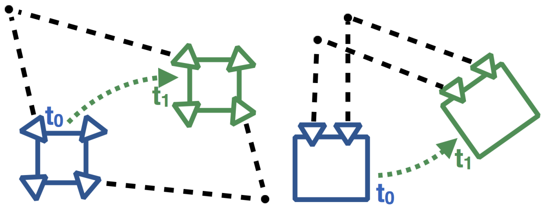

Figure 1. Illustration of moving multi-camera rigs. The black points and dashed lines indicate corresponding observations of a feature in the two capture instances. *Left:* A general multi-camera configuration with four cameras. *Right:* An axial camera configuration with two cameras (a stereo camera).

Applying the approximated motion model to the problem leads to a simplified system of polynomials which can be solved in a specialized and efficient way. Because our solution is both computationally efficient and accurate, it would be very useful for motion estimation on resource-constrained platforms such as unmanned aerial vehicles, small drones, mobile robots, and handheld devices.

2. Related Work

Stewénius et al. introduced a minimal solution for the generalized epipolar constraint [30]; however, this method produces up to 64 solutions and is too computationally complex to be considered for a real-time implementation. The linear solution of Li et al. [20] uses seventeen points and, while fast, it is sensitive to noise and requires a huge number of RANSAC iterations because of the sample size.

Specialized solutions to the generalized relative pose problem have been investigated previously, such as using point triangulation [4] or initial motion estimation from a single camera [3, 12]. Other authors have investigated the application of special-case motion models to the generalized epipolar constraint [17, 18] or assumptions on the camera overlap [4, 7]. Kim et al. [13] approximate the geometry of the multi-camera rig by treating it as a spherical camera with a common optical center. This leads to a simplified solution but loses the advantages of using multiple cameras together for relative pose estimation, i.e. a correctly scaled translation magnitude and no instability with small translation.

Kneip and Li introduced an iterative solution for generalized camera relative pose based on eigenvalue minimization [15]. They require seven or more correspondences in order to ensure convergence to a single solution. This method is much faster than previous general-case methods, but does require extra correspondences and is also susceptible to falling into local minima.

The first-order approximation to relative pose was previously applied to the single-camera relative pose problem to derive an efficient solver for either the translation [29] or rotation [33] independently. Clipp et al. applied a first-order approximation to stereo camera motion estimation assuming a single triangulated point [4]. The first-order rotation matrix assumes that the rotation is small between images, but does not enforce any other motion constraints, such as vehicle motion [17], planar motion [27] or rotation about a single axis [18], as previous solutions have done. Using the first-order approximation to relative pose, as we do in this paper, allows us to achieve an efficient solver for general multi-camera motion estimation.

Mathematical Notation We use a bold lower-case letter ($\mathbf{a}$) to represent a column vector, a sans-serif upper-case letter (A) to represent a matrix, and an italic lower-case letter (a) for a scalar. We use $[\mathbf{a}]_\times$ to denote the skew-symmetric matrix such that $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$ for all $\mathbf{b}$. We use a subscript to give the dimensions of a matrix when necessary, e.g. $\mathsf{A}_{3\times 3}$ for a 3×3 matrix.

3. Problem Description

The multi-camera case is illustrated in Figure 1. The geometry of image observations from a moving multi-camera rig can be expressed in a manner similar to the single-view case. A multi-camera rig can be represented as a generalized camera, where image observations are represented by 3D rays, not necessarily emanating from the same camera center. The 3D rays are parameterized as six-dimensional vectors in Plücker coordinates. For a generalized camera model, the epipolar constraint is replaced with the generalized epipolar constraint [24]:

$$\mathbf{v}_i^\top \begin{pmatrix} -[\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \\ \mathsf{R} & \mathsf{0}_{3\times 3} \end{pmatrix} \mathbf{u}_i = 0 \tag{1}$$

Here, $\mathbf{u}_i$ and $\mathbf{v}_i$ are corresponding rays in the previous and current frames, respectively, R is the rotation between frames and $\mathbf{t}$ is the translation. In this case, six observations are required for a minimal solution.

Here we apply the first-order approximation to the rotation matrix R, parameterized by a 3-vector $\mathbf{r} = [x\ y\ z]^\top$:

$$\mathsf{R} \approx \mathsf{I}_{3\times 3} + [\mathbf{r}]_\times \tag{2}$$

The approximated generalized epipolar constraint can now be re-arranged to isolate the rotation and translation parameters, as Kneip and Li did for the un-approximated case [15]. After stacking all six correspondences, we arrive at an equation system

$$\mathsf{M}(\mathbf{r}) \begin{bmatrix} \mathbf{t} \\ 1 \end{bmatrix} = \mathbf{0} \tag{3}$$

where $\mathsf{M}(\mathbf{r})$ is a 6×4 matrix of linear expressions in x, y, z.

Since $\mathsf{M}(\mathbf{r})$ has a null vector, it must be of rank at most three. Hence, all the 4×4 sub-determinants of $\mathsf{M}(\mathbf{r})$ must equal zero. This gives $\binom{6}{4} = 15$ equations which only involve the rotation parameters. These fifteen equations can be written in matrix form by separating the coefficients into a 15×35 matrix A and the terms into a vector of monomials $\mathbf{m}$:

$$\mathsf{A}\mathbf{m} = \mathbf{0}. \tag{4}$$

3.1. Solution Procedure

In the following we derive two different solutions to the system of equations described in Equation 4.

3.1.1 Solution using Gröbner basis method

Kukelova et al. [16] described a general, automatic procedure for producing a solver for a system of multivariate polynomial equations using the Gröbner basis method[1]. This procedure iteratively adds polynomials to the system until it can be reduced to a Gröbner basis.

In our particular case, however, we found that no extra polynomials need to be added. Using Macaulay2 [11], we determined that the system has at most twenty solutions. By ordering the monomials in $\mathbf{m}$ using graded reverse lexicographic ordering and running Gaussian elimination on A, we immediately arrive at a Gröbner basis for the ideal I generated by the fifteen polynomial equations, since this leaves only twenty monomials that are not divisible by any of the leading monomials in the equations. These monomials form a basis for the quotient ring $\mathbb{C}[x, y, z]/I$ and are the same basis monomials reported by Macaulay2. Similar reasoning was used by Stewénius et al. [28].

[1] See the textbooks of Cox, Little and O'Shea [6, 5] for an introduction to Gröbner basis methods and the work of Stewénius [26] and Byröd et al. [2] for an explanation of the use of Gröbner basis methods for minimal problems in computer vision.

Once we have the Gröbner basis, we can extract a 20×20 "action matrix" A' for variable z, so that

$$\mathsf{A}'\mathbf{v} = z\mathbf{v} \tag{5}$$

where $\mathbf{v}$ contains the twenty basis monomials from $\mathbf{m}$, the last terms of which being $x, y, z, 1$.

Now we can use eigenvalue decomposition to find the twenty eigenvalues and eigenvectors of the action matrix. The eigenvalues will be solutions for z and corresponding solutions for x and y are given in the eigenvectors. This procedure is similar to solutions to the single-camera relative pose problem for both the un-approximated [28] and approximated case [29, 33].

3.1.2 Solution by reduction to single polynomial

We note here that if we "hide" variable z, the expression $\mathsf{A}\mathbf{m} = \mathbf{0}$ can be re-written as

$$\mathsf{C}(z)\mathbf{m}' = \mathbf{0} \tag{6}$$

where $\mathsf{C}(z)$ is a 15×15 matrix of polynomials in z and $\mathbf{m}'$ contains monomials in x and y. Now we can use the fact that

$$\det(\mathsf{C}(z)) = 0 \tag{7}$$

to arrive at a single twentieth-degree polynomial in z.

This is called the hidden variable resultant method, which has been applied previously to minimal problems in computer vision, including the single-camera relative pose problem [19]. However, computing the 15×15 determinant is slow in practice, and thus the hidden variable resultant method does not lead to a fast solver here.

An alternative approach that works in general is to transfer the eigenvalue problem given in Equation 5 to a polynomial problem whose roots are the eigenvalues of the original problem (i.e. find the characteristic polynomial of A'). Bujnak et al. [1] considered several different general methods for this procedure.

In our particular case, however, we can directly derive a single polynomial from the original system of equations given in Equation 4. Because we derive the solution analytically, rather than numerically as proposed by Bujnak et al. [1], the derivation of the polynomial is closed-form and thus leads to a faster solver.

We use a special monomial ordering to reduce the problem to a single polynomial in z. Our procedure was inspired by Nistér's solution to the single-camera relative pose problem [22].

The particular monomial ordering we use is as follows:

$$\begin{aligned}\mathbf{m}'' = [&x^4, x^3y, x^2y^2, xy^3, y^4,\\ &x^3z, x^3, x^2yz, x^2y, xy^2z, xy^2, y^3z, y^3, z^4, z^3,\\ &x^2z^2, x^2z, x^2, xyz^2, xyz, xy, y^2z^2, y^2z, y^2,\\ &xz^3, xz^2, xz, x, yz^3, yz^2, yz, y, z^2, z, 1]^\top\end{aligned}$$

We then arrive at a matrix equation $\mathsf{G}\mathbf{m}'' = \mathbf{0}$ where G is matrix A after re-ordering and Gaussian elimination. This gives fifteen equations denoted as $\langle g_1 \rangle \ldots \langle g_{15} \rangle$.

We remove higher-order terms in x and y from these equations by elimination:

$$\langle b_1 \rangle \equiv \langle g_6 \rangle - z\langle g_7 \rangle \tag{8}$$
$$\langle b_2 \rangle \equiv \langle g_8 \rangle - z\langle g_9 \rangle \tag{9}$$
$$\langle b_3 \rangle \equiv \langle g_{10} \rangle - z\langle g_{11} \rangle \tag{10}$$
$$\langle b_4 \rangle \equiv \langle g_{12} \rangle - z\langle g_{13} \rangle \tag{11}$$
$$\langle b_5 \rangle \equiv \langle g_{14} \rangle \tag{12}$$
$$\langle b_6 \rangle \equiv \langle g_{15} \rangle \tag{13}$$

and thus reduce the system to six equations $\langle b_1 \rangle, \ldots, \langle b_6 \rangle$ which can be re-written as a matrix equation:

$$\mathsf{B}(z)\begin{bmatrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{bmatrix} = \mathbf{0} \tag{14}$$

where $\mathsf{B}(z)$ is a 6×6 matrix of polynomials in z. Now, as with the hidden variable resultant method before, we can find a single twentieth-degree polynomial in z:

$$\langle n \rangle \equiv \det(\mathsf{B}(z)) = 0. \tag{15}$$

Because this involves a smaller 6×6 determinant, it is much less costly to compute this determinant than the 15×15 determinant derived earlier. Once solutions for z are found by root-finding, the corresponding solutions for x and y are found by substituting the solution for z into B and computing the null vector.

To find the roots of z, we could find eigenvalues of the companion matrix of $\langle n \rangle$. However, using the eigenvalue decomposition would deprive this solution procedure of any computational advantage over the Gröbner basis method described earlier. Instead, we use numerical root-finding using Sturm sequences to bracket the roots. This root-finding procedure is much faster than eigenvalue decomposition of the companion matrix. Because our solver is intended for small motions, we bound the roots to -15 to 15 degrees.

3.2. Solution for Axial Camera Configuration

As illustrated in Figure 1, the case where all camera optical centers lie on a line is called an axial camera configuration [31]. If the camera baseline is in the X direction, this makes the fourth Plücker coordinate zero for all rays.

By assuming that the fourth coordinate of all rays is zero, we can simplify computation of the A matrix. However, the remaining sections of either solution procedure remains the same, meaning that we can still find a Gröbner basis or reduce to a single twentieth-degree polynomial in the same manner.

This means that the axial camera configuration is not a special degenerate case for our method, and we do not need to derive a new formulation of the generalized epipolar constraint for the axial camera case, as has been previously investigated [32]. Instead, we can simply replace the code for computation of the A matrix and keep all other parts of the code the same.

3.3. Translation Observability

It is important to differentiate intra-camera observations, which arise from movement of points in the same camera, from inter-camera observations, which arise from correspondence between different cameras.

It is well-known [17, 15] that with only intra-camera correspondences and zero-magnitude rotation, the translation of the multi-camera rig becomes unobservable. This is a point of degeneracy and thus, in the case of near-zero rotations and only intra-camera correspondences, the translation estimation will be unstable.

However, in our evaluation (Section 4), we did not notice any instability in the rotation estimation produced by our solvers in cases of near-zero translation, while the method of Kneip and Li [15] did exhibit higher error with small rotations. The stability of our method may be due to the first-order approximation, which reduces the amount of computation and thus reduces the chance for propagation of numerical instability.

4. Evaluation

We compared our solvers with state-of-the-art methods in terms of both computation time and accuracy on synthetic and real datasets. We refer to the solutions tested with the following abbreviations:

- **GB**: Our solution to the generalized camera relative pose problem using a Gröbner basis.
- **Poly**: Our solution to the generalized camera relative pose problem using reduction to a single polynomial and root-finding using Sturm sequences.
- **Kneip**: The solution of Kneip and Li to the generalized camera relative pose problem using eigenvalue minimization [15].
- **Stewénius**: The solution of Stewénius et al. to the generalized camera relative pose problem using a Gröbner basis [30].
- **Linear 17 pt.**: The solution of Li et al. to the generalized camera relative pose problem using linear least squares [20].

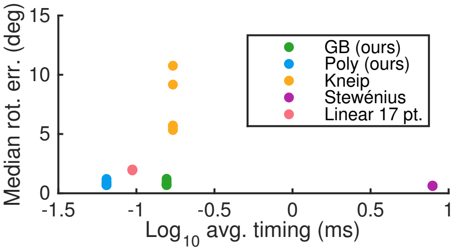

Figure 2. Comparison of computation time and rotational accuracy using synthetic data. We ran multiple tests, varying the amount of rotation between 0 and 5 degrees. Each dot represents one rotation setting. Gaussian noise was added to the observations with one pixel standard deviation. Our **Poly** algorithm is the fastest. Both our **GB** and **Poly** algorithms are competitive with state of the art in accuracy for small rotations.

All code was implemented in C++ and tested on a 2.6 GHz Intel Core i5 with 16 GB RAM. The MATLAB symbolic toolbox was used to produce optimized C++ code for our solvers. We also used templated C++ code for math operations and for polynomial manipulation. Using templated C++ allows the compiler to better optimize the code and avoids dynamic memory allocation. The implementations of **Kneip**, **Stewénius** and **Linear 17 pt.** were provided in the OpenGV library [14].

4.1. Timings

We tested each solver on 10000 randomly generated problems to compute the average computation times shown in Table 1. **Poly** is the fastest, while the **GB** method is slightly faster than **Kneip**. **Stewénius** is by far the slowest method.

Method	Time (μs)
GB (ours)	156
Poly (ours)	64
Kneip	171
Stewénius	7891
Linear 17 pt.	94

Table 1. Average computation time for various solvers.

Modifying our **GB** and **Poly** methods for the axial camera case leads to an improvement of about five microseconds for both methods.

4.2. Synthetic Data

To test the accuracy of our solutions in comparison to state-of-the-art solvers, we evaluated them on randomly generated problems. We tested a range of camera rotation magnitudes in order to examine the performance of our methods with increasing amounts of rotation. Because of the first-order approximation, our solvers will not be as accurate with higher amounts of camera rotation. We tested

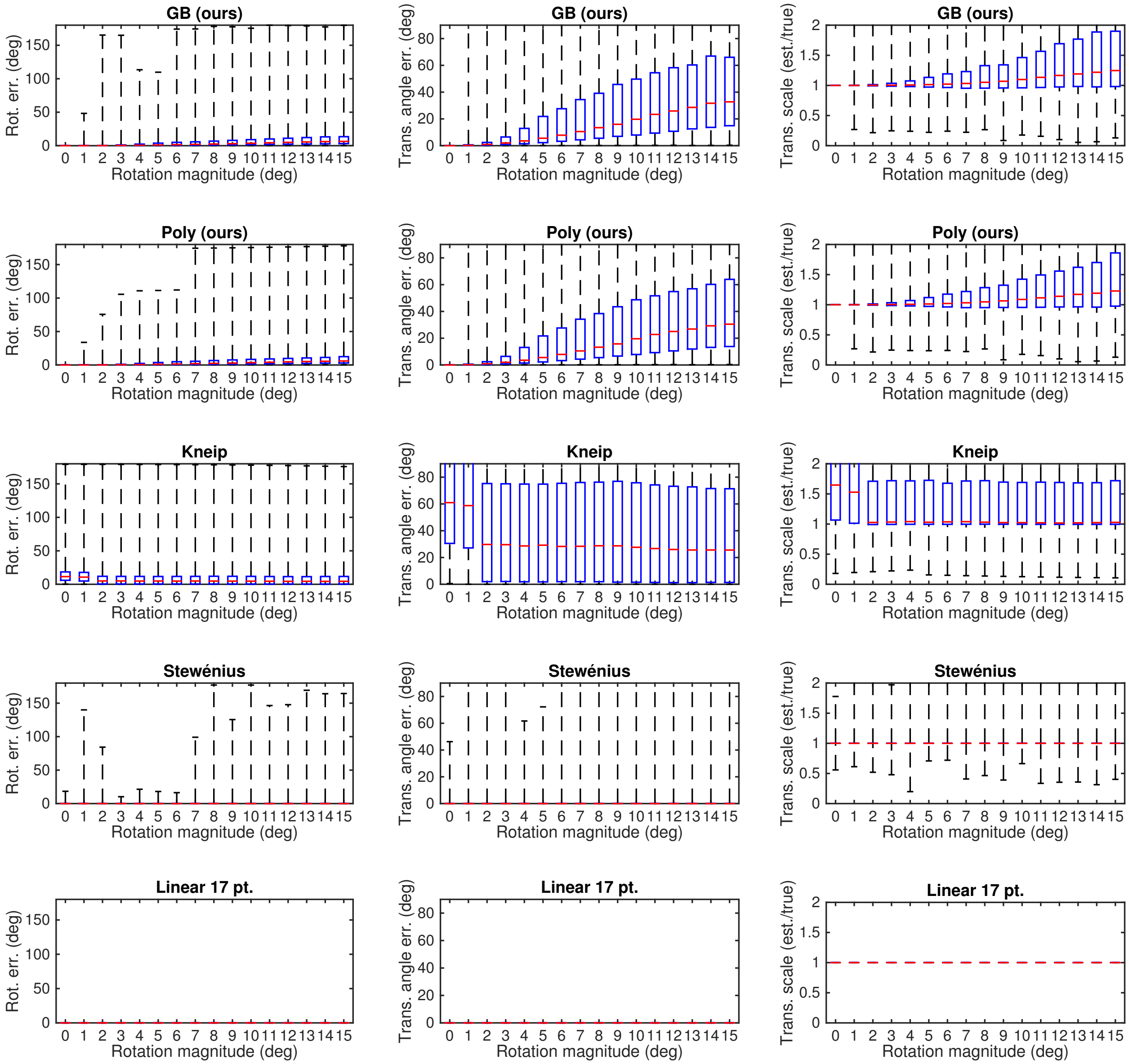

Figure 3. Comparison of our solvers with previously proposed algorithms tested on synthetic data using random generalized camera configurations. No noise was added to the observations.

the zero-noise case on synthetic data in order to evaluate performance of each method with ideal measurements, as is standard practice when evaluating minimal solver algorithms [22, 28]. We also tested the algorithms with noise added to the observations (discussed below), and on real image sequences (Section 4.3).

While details of our testing procedure will be explained in the following, Figure 2 shows a summary comparison all methods. Here, the ground truth rotation ranged in magnitude from zero to five degrees and Gaussian noise of one pixel standard deviation added to the observations. These conditions were chosen to reflect real-world conditions. The plot shows that **Poly** method is the fastest of all methods while also being close in accuracy to **Stewénius** under these conditions.

In our tests, random problems were generated in the following manner. For each problem, a random rotation of the specified rotation magnitude is generated, as well as a random unit translation. Camera centers are randomly generated in the box $[-1\ 1] \times [-1\ 1] \times [-1\ 1]$. For each camera center, a 3D point is randomly selected on the unit sphere, and then randomly scaled by a scale factor in the range $[4\ 8]$. The ray between the camera center and the point is then added as an observation ray at the first time frame, and the

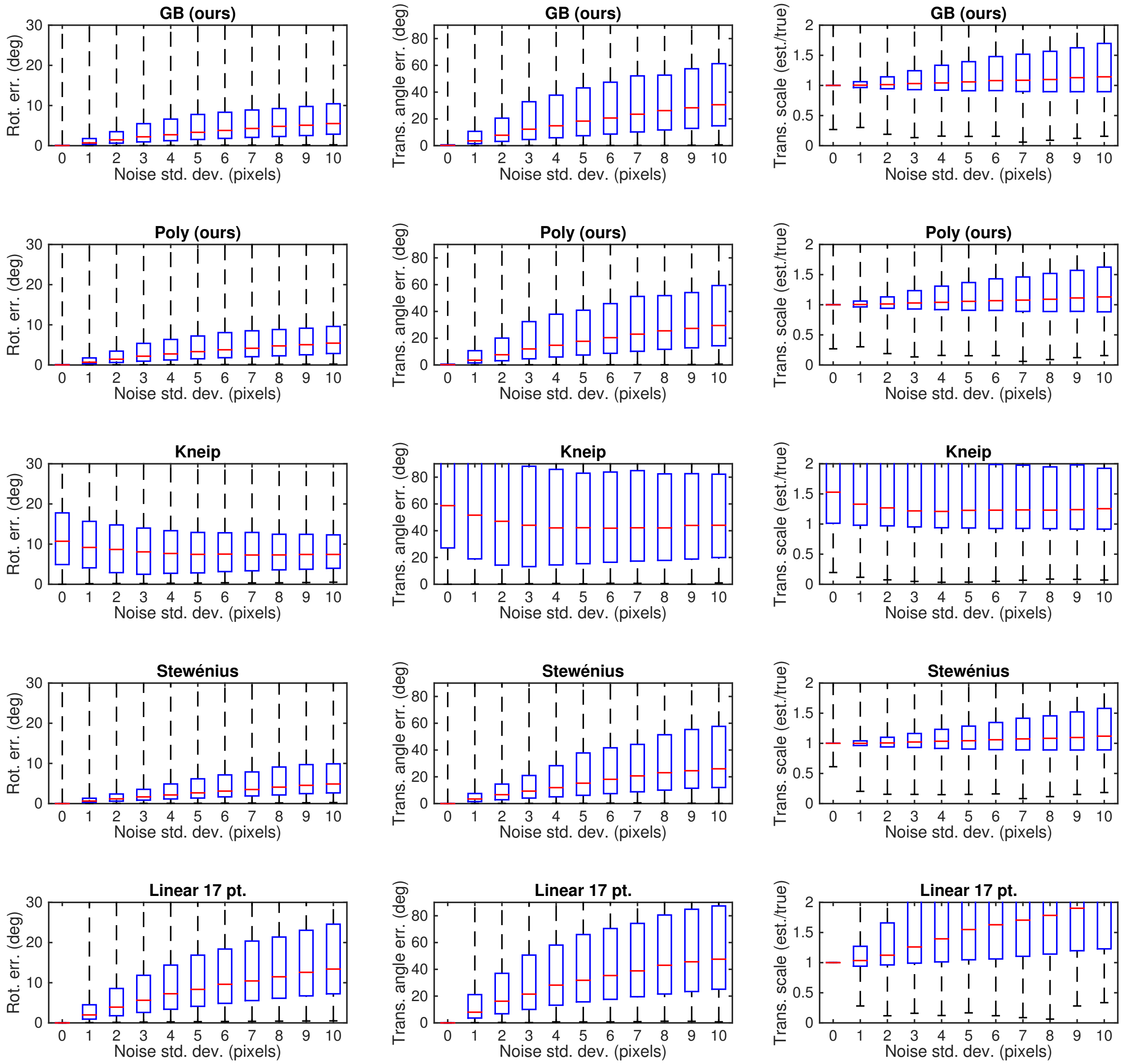

Figure 4. Comparison of our solvers with previously proposed algorithms tested on synthetic data using random generalized camera configurations. Increasing amounts of noise was added to the observations. One pixel corresponds to about 0.1 degrees.

ray between the camera center and the transformed point (using the generated rotation and translation) is added as an observation ray at the second time frame.

This setup tests multiple random multi-camera configurations with reasonable depth from the cameras to the observed points. For each algorithm tested we used the minimal number of correspondences allowed: six for **Stewénius** and our methods **GB** and **Poly**, seven for **Kneip**, and seventeen for **Linear 17 pt.**. We use an extra correspondence to disambiguate in the case of multiple solutions.

For each rotation magnitude in the range $[0\ 15]$ degrees, we generated 1000 random problems in the described manner. Figure 3 plots the range of errors produced by each solver in terms of rotation angular error, translation angular error, and translation scale ratio (estimated / true). **Stewénius** provides the lowest error at all rotation magnitudes, although the method is significantly slower than all others. **Linear 17 pt.** also produces very low error, but it is much more unstable with noise as will be seen in the following. Our **GB** and **Poly** methods produce comparable accuracy to **Stewénius** and **Linear 17 pt.** with small rotation magnitudes.

Kneip produced a high median error in all measures and all rotation magnitudes. We suspect that this is due to the

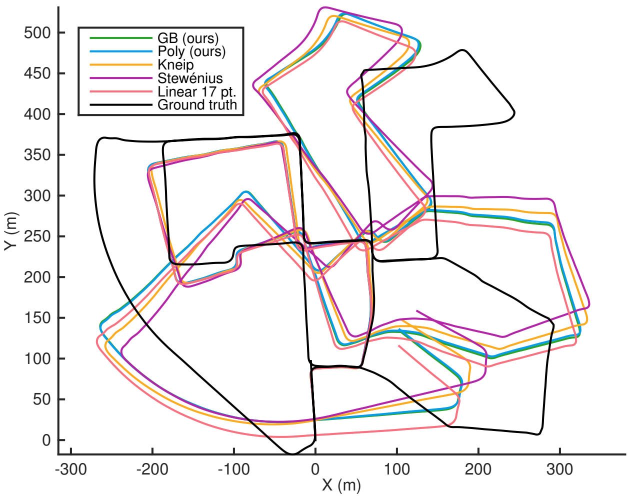

Figure 5. Estimated paths on KITTI odometry sequence 00 using various methods compared to ground truth.

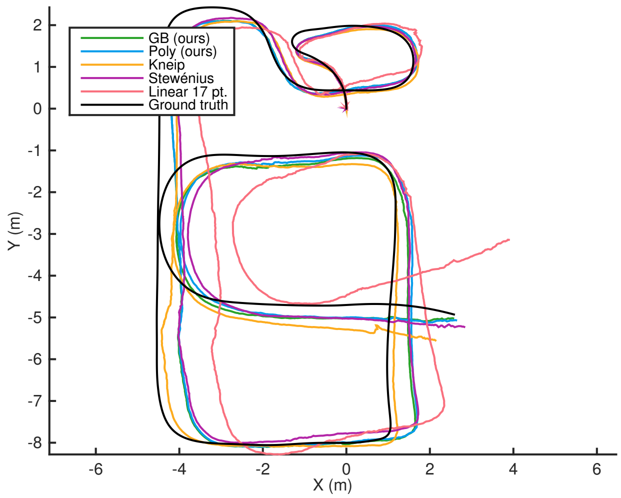

Figure 6. Estimated paths on New Tsukuba 'fluorescent' sequence using various methods compared to ground truth.

iterative optimization falling into local minima. This problem can be alleviated somewhat by using more correspondences. However, our tests on real image sequences (see following section) shows that the **Kneip** method works well in practice in a random sampling loop, suggesting that random sampling can overcome the instability of this method.

In a second test, we again generated 1000 random problems, but this time varied the amount of noise added to the observations while keeping the rotation magnitude constant at one degree. We added Gaussian noise with a standard deviation between zero and ten pixels. We used a focal length of 600 pixels, so that one pixel corresponds to about 0.1 degrees. The results are plotted in Figure 4. As can be seen, the accuracy of our methods (**GB** and **Poly**) is very similar to **Stewénius** under these conditions, which resemble real-world image observations. However, with noise added, **Linear 17 pt.** produces much worse results than our methods.

We also tested our methods on synthetic data using an axial camera configuration. We compared our solvers with and without the modification for an axial camera configuration. We did not find any difference in accuracy between the two versions, and so conclude that the only effect of the modification is a slight speedup.

4.3. Real Image Sequences

We tested our methods in comparison to the state-of-the-art on two real image sequence datasets: the KITTI benchmark dataset [9] and the New Tsukuba stereo dataset [21]. Both datasets use a stereo camera configuration with left and right cameras. Each sequence provides left and right images at each capture point as well as ground truth position and orientation data. Because both datasets use a stereo camera configuration, we used the axial camera version of our solvers in the tests.

Feature tracks were generated using the quad-matching method of Geiger et al. [10]. This method applies corner detection and performs a circular matching of features across the stereo pairs at consecutive capture points. Matches which are not consistent across all views or which stray too far from the stereo epipolar lines are removed. Each quad-match produces two intra-image correspondences and two inter-image correspondences. Because of the known degeneracy with only intra-image correspondences and near-zero rotation, we used all four correspondences in our tests.

For each sequence, we ran each relative pose solver in a Preemptive RANSAC [23] loop on each consecutive pair of stereo frames. Preemptive RANSAC is an alternative to RANSAC [8] which is commonly used in visual odometry applications and on embedded platforms because it produces an essentially fixed computation time per frame [25]. Preemptive RANSAC first generates a fixed number of random samples to compute hypotheses and then tests blocks of observations while reducing the number of hypotheses under consideration after each block tested. We used 200 samples and a block size of 100 observations. As a scoring function to determine inliers, we used the re-projection error of the point as triangulated using the stereo baseline. In all tests we used an inlier threshold of two pixels. The output of Preemptive RANSAC is accepted if at least 40% of matches are classified as inliers. The percentage of successful frames, reported in the following results, is the percentage of frames in the sequence for which a particular solver produced an acceptable solution.

We did not apply any non-linear refinement, multi-frame bundle adjustment, or loop closure. Thus we tested purely the frame-to-frame visual odometry component in order to fairly compare the methods and not test the effects of re-

Method	**GB** (ours)	**Poly** (ours)	**Kneip**	**Stewénius**	**Linear 17 pt.**
Rotational error (deg)	0.07	0.07	0.07	0.08	0.10
Translational error (deg)	0.95	0.95	1.01	1.11	1.25
Translation scale (estimate/true)	1.00	1.00	1.00	1.00	1.00
Succesful frames (%)	99.98%	99.98%	99.98%	99.98%	99.98%

Table 2. Accuracy results on KITTI odometry sequence 00. The median for each error measure is given.

Method	**GB** (ours)	**Poly** (ours)	**Kneip**	**Stewénius**	**Linear 17 pt.**
Rotational error (deg)	0.09	0.09	0.10	0.10	0.19
Translational error (deg)	6.22	6.25	7.50	7.19	13.04
Translation scale (estimate/true)	0.99	0.99	0.99	0.99	1.02
Successful frames (%)	99.33%	99.33%	99.28%	99.06%	97.33%

Table 3. Accuracy results on New Tsukuba 'fluorescent' sequence. The median for each error measure is given.

finement methods. The relative pose from each pair is accumulated in order to produce the continuous trajectory of the camera rig over time.

KITTI Results from testing on KITTI sequence 00 are shown in Table 2. This sequence has 4,541 stereo frames captured from a moving vehicle with a GPS/IMU unit for ground truth estimation. The video was recorded at 10 Hz and exhibits frame-to-frame rotation of about five degrees at maximum.

Our methods **GB** and **Poly** exhibit almost exactly the same accuracy in terms of median error, and match or outperform all other methods on all measures. The estimated trajectories for KITTI sequence 00 are plotted with ground truth in Figure 5. The trajectories for all methods are roughly consistent. Because we are directly aggregating frame-to-frame relative pose measurements without any post-refinement, there is significant accumulation of drift.

New Tsukuba Results from testing on the New Tsukuba 'fluorescent' sequence are shown in Table 3 and estimated trajectories are plotted in Figure 6. The New Tsukuba sequences were generated using photorealistic rendering of a highly detailed, 3D office scene model. While the KITTI dataset exhibits primarily forward and turning motion, the New Tsukuba sequence has more significant horizontal and vertical movement in addition to forward and turning motion.

Again, our methods match or outperform all other methods on all accuracy measures. The error in translation direction is significantly higher for all methods on this sequence in comparison to KITTI. We believe this is due to two reasons: the translational movement between successive frames is much smaller for this sequence, and the image resolution is also much smaller (640×480 versus 1241×376 for KITTI). It is well-known that translation estimation becomes unstable when the translation magnitude is small and thus induces little feature movement. This problem is compounded by the low image resolution of the dataset.

5. Discussion

Our evaluations show that our methods outperform state-of-the-art methods in accuracy on real-world image sequences while being faster to compute. Our **GB** method shows a speedup of about 50× over the method of **Stewénius**, while returning fewer solutions, and has about the same computation time as **Kneip** with better stability. Our **Poly** method shows a speedup of 123× over **Stewénius** and about 2.7× over **Kneip** while having the same accuracy and stability as our **GB** method.

Although in our synthetic tests, we tested rotation magnitudes of up to fifteen degrees, the practical applications we envision – such as car or aerial vehicle motion with at least a 10 Hz frame rate – typically will not exhibit frame-to-frame rotation beyond five degrees.

6. Conclusions and Future Work

By using a first-order approximation to relative pose, we achieved efficient and direct solutions to multi-camera relative pose which do not require iterative optimization and are almost completely closed form. Our novel solutions are as fast or faster than the state-of-the-art while providing better stability and comparable accuracy for motion estimation[2].

Future work includes real-time implementation for online ego-motion estimation in a mobile embedded system and application of our solution procedure to other related geometric problems.

Acknowledgments

This work was partially supported by NSF IIS-1464420 and the Christian Doppler Laboratory on Handheld AR.

[2] Source code is available at `http://jventura.net/code`.

References

[1] M. Bujnak, Z. Kukelova, and T. Pajdla. Making Minimal Solvers Fast. In *Conference on Computer Vision and Pattern Recognition*, pages 1506–1513, 2012. 4323

[2] M. Byröd, K. Josephson, and K. Åström. Improving Numerical Accuracy of Gröbner Basis Polynomial Equation Solvers. In *International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 4322

[3] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley. Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems. In *2008 IEEE Workshop on Applications of Computer Vision (WACV)*, 2008. 4322

[4] B. Clipp, C. Zach, J.-M. Frahm, and M. Pollefeys. 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*, pages 1725–1732, 2009. 4321, 4322

[5] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer New York, 2007. 4322

[6] D. A. Cox, J. Little, and D. O'Shea. Using Algebraic Geometry, 2005. 4322

[7] E. Dunn, B. Clipp, and J.-M. Frahm. A Geometric Solver for Calibrated Stereo Egomotion. In *International Conference on Computer Vision*, pages 1187–1194, 2011. 4322

[8] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, June 1981. 4321, 4327

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision Meets Robotics: the KITTI Dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, September 2013. 4327

[10] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3D Reconstruction in Real-Time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011. 4327

[11] D. R. Grayson and M. E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at `http://www.math.uiuc.edu/Macaulay2/`. 4322

[12] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart. Real-Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View. In *Conference on Computer Vision and Pattern Recognition*, pages 1529–1536, 2012. 4322

[13] J. S. Kim, M. Hwangbo, and T. Kanade. Spherical Approximation for Multiple Cameras in Motion Estimation: Its Applicability and Advantages. *Computer Vision and Image Understanding*, 2010. 4322

[14] L. Kneip and P. Furgale. OpenGV: A Unified and Generalized Approach to Real-Time Calibrated Geometric Vision. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2004. 4324

[15] L. Kneip and H. Li. Efficient Computation of Relative Pose for Multi-Camera Systems. In *Conference on Computer Vision and Pattern Recognition*, pages 446–453, 2014. 4321, 4322, 4324

[16] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic Generator of Minimal Problem Solvers. In *European Conference on Computer Vision*, pages 302–315. Springer Berlin Heidelberg, 2008. 4322

[17] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Motion Estimation for Self-Driving Cars with a Generalized Camera. In *Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013. 4322, 4324

[18] G. H. Lee, M. Pollefeys, and F. Fraundorfer. Relative Pose Estimation for a Multi-Camera System with Known Vertical Direction. In *Conference on Computer Vision and Pattern Recognition*, 2014. 4322

[19] H. Li and R. Hartley. Five-Point Motion Estimation Made Easy. In *Proceedings of the International Conference on Pattern Recognition*, pages 630–633, 2006. 4323

[20] H. Li, R. Hartley, and J.-H. Kim. A Linear Approach to Motion Estimation Using Generalized Camera Models. In *Conference on Computer Vision and Pattern Recognition*, 2008. 4321, 4324

[21] S. Martull, M. P. Martorell, and K. Fukui. Realistic CG Stereo Image Dataset with Ground Truth Disparity Maps. In *ICPR 2012 workshop: TrakMark2012*, 2012. 4327

[22] 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. 4323, 4325

[23] D. Nistér. Preemptive RANSAC for Live Structure and Motion Estimation. *Mach. Vision Appl.*, 16(5):321–329, December 2005. 4327

[24] R. Pless. Using Many Cameras as One. In *Conference on Computer Vision and Pattern Recognition*, 2003. 4322

[25] D. Scaramuzza and F. Fraundorfer. Visual Odometry [tutorial]. *Robotics & Automation Magazine, IEEE*, 18(4):80–92, 2011. 4327

[26] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund Institute of Technology, 2005. 4322

[27] H. Stewénius and K. Åström. Structure and Motion Problems for Multiple Rigidly Moving Cameras. In *European Conference on Computer Vision*, 2004. 4322

[28] H. Stewenius, C. Engels, and D. Nistér. Recent Developments on Direct Relative Orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, June 2006. 4322, 4323, 4325

[29] H. Stewénius, C. Engels, and D. Nistér. An Efficient Minimal Solution for Infinitesimal Camera Motion. In *Conference on Computer Vision and Pattern Recognition*, 2007. 4322, 4323

[30] H. Stewenius, M. Oskarsson, K. Åström, and D. Nistér. Solutions to Minimal Generalized Relative Pose Problems. In *OMNIVIS 2005*, 2005. 4321, 4324

[31] P. Sturm. Multi-View Geometry for General Camera Models. In *Conference on Computer Vision and Pattern Recognition*, pages 206–212, 2005. 4323

[32] F. Vasconcelos and J. Barreto. Towards a Minimal Solution for the Relative Pose Between Axial Cameras. In *British Machine Vision Conference*, pages 1241–12411, 2013. 4324

[33] J. Ventura, C. Arth, and V. Lepetit. Approximated Relative Pose Solvers for Efficient Camera Motion Estimation. In *European Conference on Computer Vision*, 2014. 4322, 4323