Approximated Relative Pose Solvers for Efficient Camera Motion Estimation

Jonathan Ventura[1,2], Clemens Arth[2], Vincent Lepetit[2]

[1]Department of Computer Science, University of Colorado Colorado Springs, USA
[2] Institute for Computer Graphics and Vision, Graz University of Technology, Austria

Abstract. We propose simple and efficient methods for estimating the camera motion between two images when this motion is small. While current solutions are still either slow, or unstable in case of small translation, we show how to considerably speed up a recent stable but slow method. The reasons for this speed-up are twofold. First, by approximating the rotation matrix to first order, we obtain a smaller polynomial system to be solved. Second, because of the small rotation assumption, we can use linearization and truncation of higher-order terms to quickly obtain a single solution. Our experiments show that our approach is both stable and fast on challenging test sequences from vehicle-mounted cameras.

Keywords: camera pose, relative pose, relative orientation, five-pt algorithm, essential matrix, Gröbner basis

1 Introduction

Estimating the rotational and translational movement of a calibrated camera between two images is a fundamental problem in computer vision, with many applications such as visual odometry, structure-from-motion, and simultaneous localization and mapping (SLAM). While many solutions to the relative pose problem have been proposed in the past [3, 4, 9, 12, 13, 15, 17–20, 22, 23, 25], existing solutions have still three critical issues. The first issue is numerical instability when the translation is small with respect to the scene depth. A second issue is multiplicity of solutions; even in the minimal case of five point correspondences, up to ten camera pose solutions are possible, and additional processing is required to select the correct solution. Third, while recent work has shown that the rotation can be estimated independently of the translation [13], thus avoiding the instability with small translation, this method is much slower than the state-of-the-art.

In this work, we introduce novel solution procedures for solving for the rotation independent of the translation. These solvers are as fast or faster than the state-of-the-art, and as accurate for small rotations. Our key observation is that, in many practical cases, the amount of rotation is small, and we can safely approximate the rotation matrix to the first-order. Using this approximation, we obtain a system of ten polynomial equations involving only the rotation parameters, which is a smaller system than the thirty equations required for the general case [13]. Once the rotation estimate is obtained, the translation can easily be estimated as well.

Our evaluations on synthetic and real datasets show that our solutions are as fast or faster than existing methods, without any significant loss in accuracy for small motions. The simplicity of our solvers are such that they are suitable for implementation in embedded hardware for ground vehicles or low-powered micro-aerial vehicles, or for application with high-speed cameras.

The remainder of the paper is structured as follows. We first review related work in Section 2. We then formalize the approximated relative pose problem in Section 3, and describe various solution procedures in Section 4. In Section 5, we compare our methods to state-of-the-art algorithms applied to visual odometry on two public datasets.

2 Related Work

The relative pose problem, the computation of the camera motion between two images from point correspondences, is an essential problem encountered in photogrammetry and computer vision with an eventful history. The first solution to the five-point problem was proposed by Kruppa [14], who proved that the problem has at most eleven solutions. Demazure [3], Faugeras [4], Maybank [20] and others improved upon this approach later. They showed that this problem has at most ten solutions in general, including complex ones, being the roots of a tenth-degree polynomial.

While Kruppa proposed an algorithm with little practical applicability, solutions in the context of modern Computer Vision based on eight and seven points were proposed by Longuet-Higgins [19] and Maybank [20]. A six-point solution was introduced by Philip [23], extracting the roots of a thirteen-degree polynomial. Nistér improved on this approach, solving a tenth-degree polynomial corresponding to the exact problem difficulty [22]. His solution is based on QR-factorization, Gauss-Jordan elimination on a 5×9 matrix and Sturm sequences. From there on, algorithms based on five point correspondences, the minimal number of required correspondences, raised special interest for their application in a hypothesize-and-test framework [5].

A first solver based on Gröbner bases [2] was given by Stewénius *et al.* [24]. Alternative formulations also based on Gröbner bases were proposed by Kukelova *et al.* [15] and Kalantari *et al.* [9]. Li *et al.* proposed relatively simpler solutions for the five-point and six-points problem based on the hidden variable resultant [17, 16]; however, this is less efficient.

More recently, Lim *et al.* proposed estimating the rotation and translation separately, relying on a special feature correspondence distribution [18]. Kneip *et al.* later proposed considering rotation and translation separately in the general case [13]. Their algorithm still exhibits instability in cases of negligible translational motion, which need to be explicitly detected. They parameterized the rotation with a 3×3 matrix, which requires adding twenty additional constraints which enforce the matrix to be orthogonal and have unit determinant. They then used the method of Gröbner bases to find up to twenty solutions for the rotation from a matrix of size 66×197 [13]. The authors also mention that they tried the Cayley rotation parameterization [1], which has the minimal three parameters, but leads to a larger and slower solution procedure. Kneip *et al.* later proposed an iterative solution which is more stable, but in practice requires using more than the minimal five correspondences in order to avoid local minima [12].

The work most closely related to ours is that of Stewenius et al. [25], who also use a first-order approximated rotation matrix. They briefly mention that it is possible to solve for the rotation independently of the translation, but then proceed to describe a method for instead computing the translation. This leads to a solver which is faster than Nistér's solver [22]. However, in this work we choose to solve for the rotation because it leads to an even faster solver based on linearization and truncation of higher-order terms.

3 Problem Statement

For the sake of completeness, we briefly describe here the method of Kneip *et al.* [13], which is our starting point. Let's consider two images of a rigid scene. The essential matrix E relates corresponding image locations u_i and v_i expressed in homogeneous coordinates, in the first and second images, respectively, by

$$v_i^T E u_i = 0 \,, \tag{1}$$

with $E = [t]_\times R$, where R and t are the rotation matrix and translation vector of the rigid motion of the camera between the two images. This can be re-arranged to isolate the rotation and translation parameters:

$$(u_i^T R^T [v_i]_\times) \cdot t = 0 \,. \tag{2}$$

Five point correspondences between the two images give five equations of the form of Eq. (2), which can be arrange into an equation system:

$$A(r) \cdot t = 0 \,, \tag{3}$$

where $A(r)$ is a 5×3 matrix and r are the parameters of the rotation. Since $A(r)$ has a null vector, it must be of at most rank two. Hence, all the 3×3 sub-determinants $|A_{ijk}|$ of $A(r)$ must be zero. This gives $\binom{5}{3} = 10$ equations which only involve the rotation parameters:

$$|A_{ijk}| = 0 \;\; \forall (i,j,k) \in \mathcal{S} \,, \tag{4}$$

with $\mathcal{S} = \{(1,2,3), (1,2,4), (1,2,5), (1,3,4), (1,3,5), (1,4,5), (2,3,4), (2,3,5), (2,4,5), (3,4,5)\}$.

Kneip *et al.* [13] solve these equations in the general case, which requires twenty additional constraints on the rotation matrix and results in a slow solver. We show below that under the assumption of a small motion, solving these equations becomes very simple.

4 Approach

In our approach, we first introduce an approximation of the rotation matrix, and use it to rewrite the system of equations (4). We then show how this system can be simplified, and very easily solved.

We assume that the motion between the two images is small. This allows us to replace the rotation matrix R by its first-order expansion:

$$\hat{R}(r) = I + [r]_\times \ , \tag{5}$$

where $r = [r_1, r_2, r_3]^T$ is a three-vector. The corresponding exact rotation matrix can be retrieved as $R(r) = \exp_{SO(3)}(r)$. This gives us a simple parameterization for the rotation matrix. Plugging it into the ten equations of (4) leads us to a system of ten cubic polynomials in twenty monomials:

$$M_{10\times 20}\, x = 0 \ , \tag{6}$$

with

$$x = \begin{bmatrix} r_1^3, r_1^2\, r_2, r_1\, r_2^2, r_2^3, r_1^2\, r_3, r_1\, r_2\, r_3, r_2^2\, r_3, r_1\, r_3^2, r_2\, r_3^2, r_3^3, \\ r_1^2, r_1\, r_2, r_2^2, r_1\, r_3, r_2\, r_3, r_3^2, r_1, r_2, r_3, 1 \end{bmatrix}^T \ . \tag{7}$$

The following sections give three different solution procedures for solving this system of equations.

4.1 Reduction to a Single Polynomial

The system of equations given by Equation 6 has the same form as found in the five-point algorithm of Nistér [22], and thus can be solved in the same manner, namely, reduction to a single tenth-degree polynomial in r_3. The root-finding procedure leads to ten solutions for r_3; corresponding solutions for r_1 and r_2 are found by back-substitution.

Sturm sequences are used to bracket the roots, which are then quickly located exactly through bisectioning. In our case, we speed up the root-finding procedure by restricting our search to reasonable bounds on the rotation magnitude. In practice we assume that solutions for r_3 should lie within -15 and 15 degrees.

4.2 Neglecting the Cubic Terms and Solving by Linearization

Our assumption of a small motion implies that the components of the r vector are small. It follows that the higher-order terms in the equations (6) are negligible in comparison to lower order terms. Neglecting the cubic terms in Eq. (6) reduces the system to only ten quadratic equations in ten monomials, where

$$N_{10\times 10}\, y = 0 \ , \tag{8}$$

and

$$y = \left[r_1^2, r_1\, r_2, r_2^2, r_1\, r_3, r_2\, r_3, r_3^2, r_1, r_2, r_3, 1 \right]^T \ . \tag{9}$$

These monomials are the second half of the x vector defined in Eq. (7).

Eq. (8) is a polynomial system in the components of r. However, we noticed that in practice we can solve it by linearization [10], that is, we solve it as if it was a linear system. Since the last component of y is 1, we use the Cholesky decomposition method to find the other components, as it is the fastest method.

This method of solving is much faster than reduction to a single polynomial. It also produces a single solution instead of up to ten. However, this increased speed comes at the cost of decreased robustness to larger rotations, and greater sensitivity to noise.

4.3 Six-Point Least Squares Solution

If we build Eq. 4 using six point correspondences instead of five, we obtain $\binom{6}{3} = 20$ equations. This gives a system of twenty cubic equations in twenty monomials.

$$M_{20\times 20}\, x = 0\,, \tag{10}$$

This system can also be solved as a linearized least-squares problem, this time without having to remove higher-order terms as we did for our five-point solution.

The speed of this method is roughly on par with the solution by reduction to a single polynomial, but it produces a single solution instead of ten. Because it avoids truncating the higher-order terms, it has better robustness to noise and larger rotations than the linearized five-point solver. However, it introduces a degeneracy when viewing a plane, because, in this case, the sixth correspondence is linearly dependent on the first five, meaning that the system is rank-deficient.

5 Evaluation

In the following we evaluate our approach in detail concerning different aspects. We demonstrate the accuracy of our methods for visual odometry on vehicle-mounted camera image sequences, and show its performance in terms of run-time.

We refer below to our novel algorithms using the following abbreviations: **Poly. 5pt.** is the solution by reduction to a single polynomial (Section 4.1), **Lin. 5pt.** is the truncated, linearized solution (Section 4.2), and **Lin. 6pt.** is the linearized solution using six points (Section 4.3). We compare our methods against the following existing methods: **5 pt. (Nistér)** and **5 pt. (Stewénius)** refer to the essential matrix solvers of [22] and [25] respectively, **5 pt. (Kneip 2012)** is the direct rotation-only solution proposed by Kneip et al. [13], and **10 pt. (Kneip 2013)** is the iterative rotation-only method proposed by Kneip et al. [12]. The iterative method does not require ten correspondences, but this is the number recommended by Kneip et al. to avoid local minima. We use the reference implementations of **5 pt. (Stewénius)**, **5 pt. (Kneip 2012)**, **10 pt. (Kneip 2013)** from the *OpenGV* library [11], and we use the hand-optimized implementation of **5 pt. (Nistér)** provided by Richard Hartley[1].

[1] http://users.cecs.anu.edu.au/ hartley/Software/5pt-6pt-Li-Hartley.zip

Table 1. Average computation time for various solvers, in microseconds.

Method	Time (μs)
Poly. 5pt.	7.43
Lin. 5pt.	3.51
Lin. 6pt.	10.71
5 pt. (Nistér)	6.32
5 pt. (Stewénius)	61.50
5 pt. (Kneip 2012)	475.09
10 pt. (Kneip 2013)	100.74

5.1 Solver Computation Time

The average computation time for each solver is given in Table 1. The times were recorded using a *Apple Macbook Pro Late 2011* with a *2.5 GHz* i7 CPU. Each algorithm was run 10,000 times using randomly generated input data.

The fastest existing method is **5 pt. (Nistér)** which requires about 6 μs. Our **Poly. 5pt.** solver is only slightly slower, requiring about 7.5 μs. We believe the speed could be improved by using hand-optimization to build the constraint matrix, as was done for the **5 pt. (Nistér)** implementation.

Lin. 5pt., the truncated linearized five-point solver, is almost twice as fast as **5 pt. (Nistér)**. However, this speed comes at the cost of lower accuracy with greater rotation magnitudes, as will be seen in the following.

Lin. 6pt. is a bit slower than **5 pt. (Nistér)**, although again the speed could be improved by hand-optimizing the code to build the constraint matrix.

5.2 Accuracy with Increasing Rotation

We evaluated our solvers with respect to increasing amounts of rotation. We ran 10,000 trials on synthetic data to assess the angular error, given translational motion in either x, y or z direction and varying the overall rotation magnitude between 0 and 10 degrees.

The results are shown in Figure 1. As expected, on average the **Lin. 5pt.** gives a higher angular error for increasing amounts of rotation than the **Lin. 6pt.** and the **Poly. 5pt.** solver, which give similar results. The median angular error is significantly lower for all solvers. This suggests that the mean is affected by outliers within all the results acquired. There is no significant difference comparing situations of translational motion along either the x, y or z axis. This indicates that our solvers don't exhibit any superior or inferior behavior for certain motion patterns (*i.e.* forward or sideward motion).

5.3 Image Sequences from Vehicle-mounted Camera

We evaluated our approach on the KITTI visual odometry dataset [7]. It contains 11 sequences captured by a camera mounted on car driving around the streets of

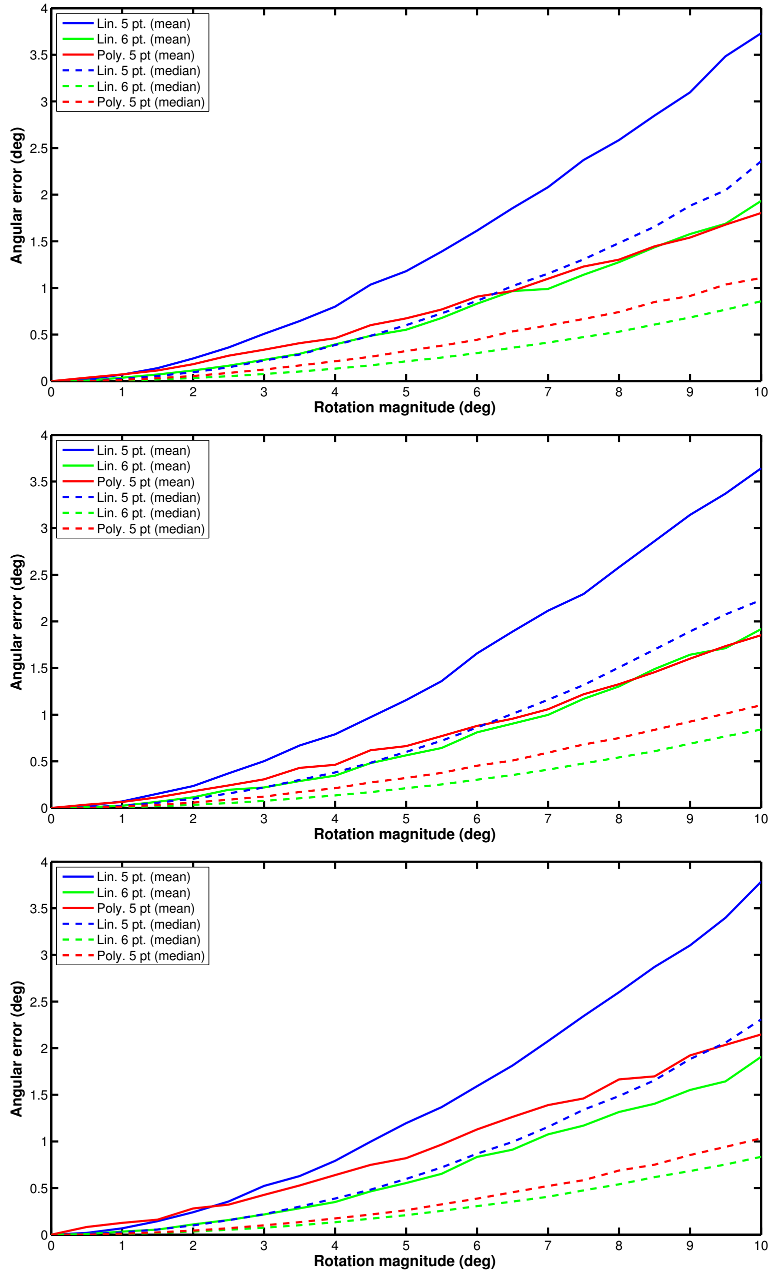

Fig. 1. Mean and Median accuracy for translational motion in x (*top*), y (*middle*) and z (*bottom*) direction respectively for increasing magnitudes of rotation about a random axis.

Karlsruhe, Germany. Ground truth trajectories were obtained using a combined GPS/IMU inertial navigation system. Stereo sequences are available, but we used the images from only one camera.

For each test sequence, feature tracks are obtained using the method of Geiger et al. [8]. To estimate the relative pose between successive frames, we used each solver in a *Preemptive RANSAC* [21] loop for robust estimation. In *Preemptive RANSAC*, a fixed number of hypotheses N is sampled, and then all correspondences are evaluated in blocks of size B. After each block is processed, the number of hypotheses is reduced by keeping only the best ones; this is repeated until all correspondences have been tested. This method is typically used in visual odometry applications, because it guarantees a fixed computation time for the robust estimation step and is thus ideal for embedded implementation [6]. For all methods we used $N = 200$ hypotheses and a block size of $B = 10$. Because the translation magnitude cannot be recovered directly from monocular motion, we scaled each resulting translation estimate to match the ground truth translation magnitude. This allows us to fairly compare all methods without having to choose between various visual odometry or SLAM multi-frame integration approaches.

In Figure 2, we plot the accuracy of the estimates against the time it takes to compute them, averaged over all the frames of the sequences from the KITTI dataset. We use the rotational error in degrees as our accuracy measure.

We considered the mean evaluation times both for single solver estimates and for the complete Preemptive RANSAC loop. **Lin. 5pt.** is the fastest method, but the accuracy averaged over all frames is slightly worse than **5 pt. (Nistér)**. The **Poly. 5pt.** and **Lin. 6pt.** methods are slightly slower than **5 pt. (Nistér)**, but have better average accuracy. These solutions all have an average RANSAC loop time of under ten milliseconds. While **5 pt. (Stewénius)** and **10 pt. (Kneip 2013)** have the best accuracy, their computational time is much higher. Finally, **5 pt. (Kneip 2012)** has worse average accuracy than our **Poly. 5pt.** solution and is also much slower.

5.4 Accuracy and Increasing Amounts of Rotation

We also assess the accuracy of the individual solutions with respect to increasing amounts of rotation as observed in the KITTI sequences. The results are shown in Figure 3.

All methods show a trend towards higher error with greater rotation magnitudes. This is likely due to the increasing inaccuracy of feature matching with greater intra-image motion. Our **Poly. 5pt.** and **Lin. 6pt.** solvers exhibit almost the same accuracy as the state-of-the-art methods across the range of rotations. **Lin. 5pt.** has the worst accuracy with increasing rotation, because of the truncated terms, and is only on par with the other solvers up to one degree of rotation. **10 pt. (Kneip 2013)** and **5 pt. (Kneip 2012)** also show slightly higher for the largest rotation range (four to five degrees).

Few sequential image pairs have rotation above five degrees in the sequences, indicating that this about the maximum expected for a car-mounted camera

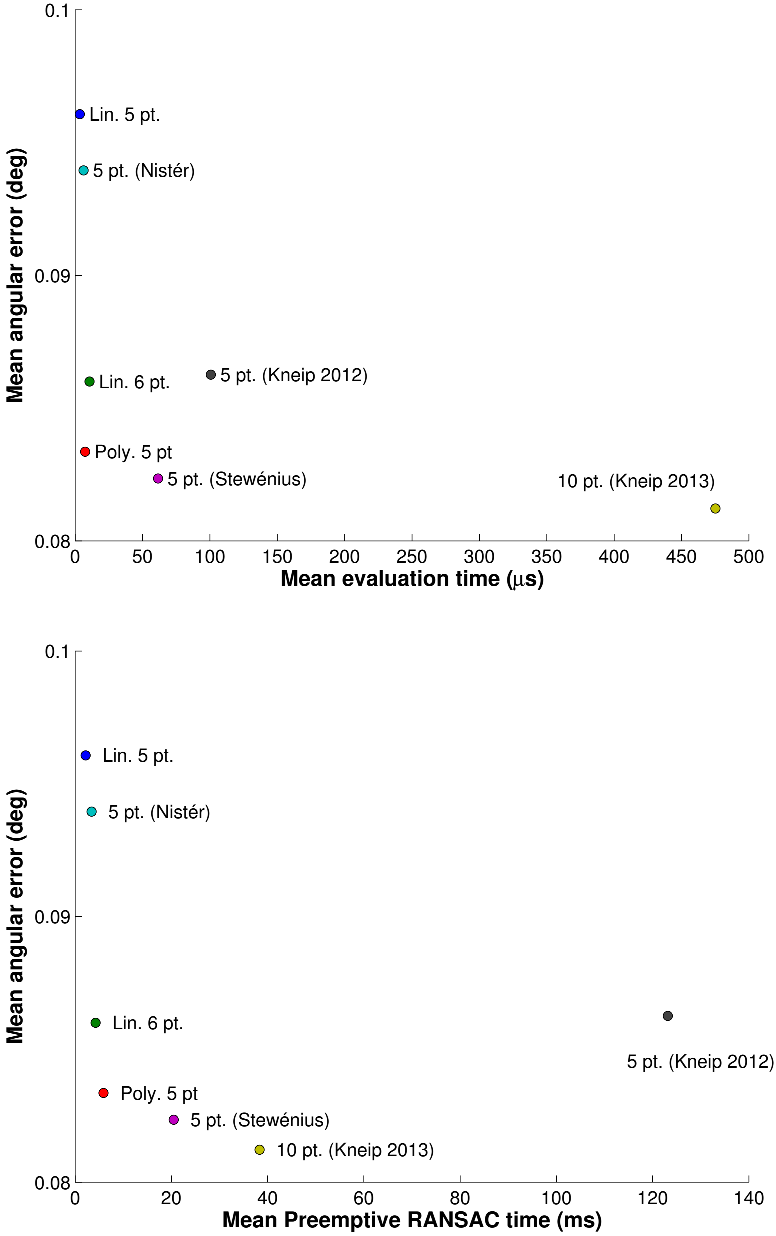

Fig. 2. Average rotational accuracy versus computation time, evaluated using the sequences from the KITTI dataset. Top: for a single evaluation and Bottom: for Preemptive RANSAC. Our **Lin. 5pt.** solver is faster than previous methods. Our **Poly. 5pt.** and **Lin. 6pt.** solvers are slightly slower than **5 pt. (Nistér)**, but more accurate.

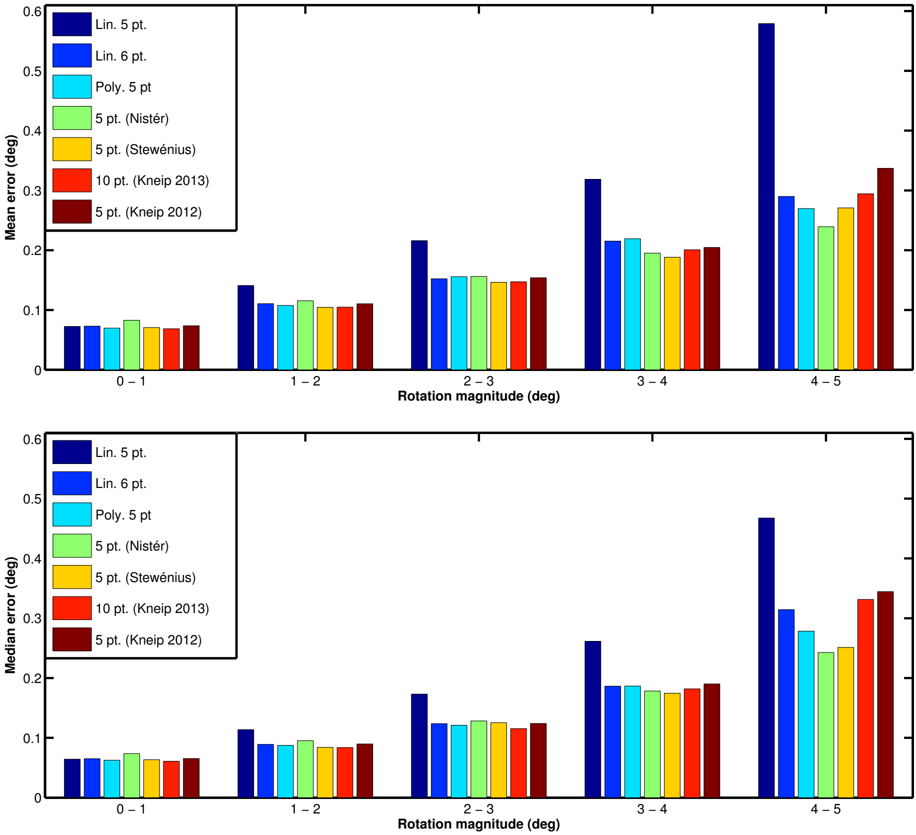

Fig. 3. Mean (top) and median (bottom) rotational accuracy for the KITTI sequences, aggregated by rotation magnitude. The error increases with the amount of rotation for all the solvers. Our **Lin. 5pt.** algorithm is more sensitive than the other ones, however after the final optimization step, it provides an accuracy similar to the other solvers up to four degrees, which seems enough in practice as shown in Figures 4 and 5.

driving at city-street speeds. Furthermore, the camera was operated at 10 Hz, and with a faster camera rate, the maximum rotation observed would be even lower. This indicates that our solvers for approximated rotation are a reasonable choice for such an application.

The complete trajectories as estimated by each method on several sequences from KITTI are depicted in Figures 4 and 5. No non-linear optimization, bundle adjustment, or loop closure was applied to the trajectories; the trajectories were simply produced by integrating frame-to-frame motion estimates. Our **Poly. 5pt.** and **Lin. 6pt.** solvers give comparable results to the state-of-the-art. The **Lin. 5pt.** solver shows more severe errors accumulating over time. These

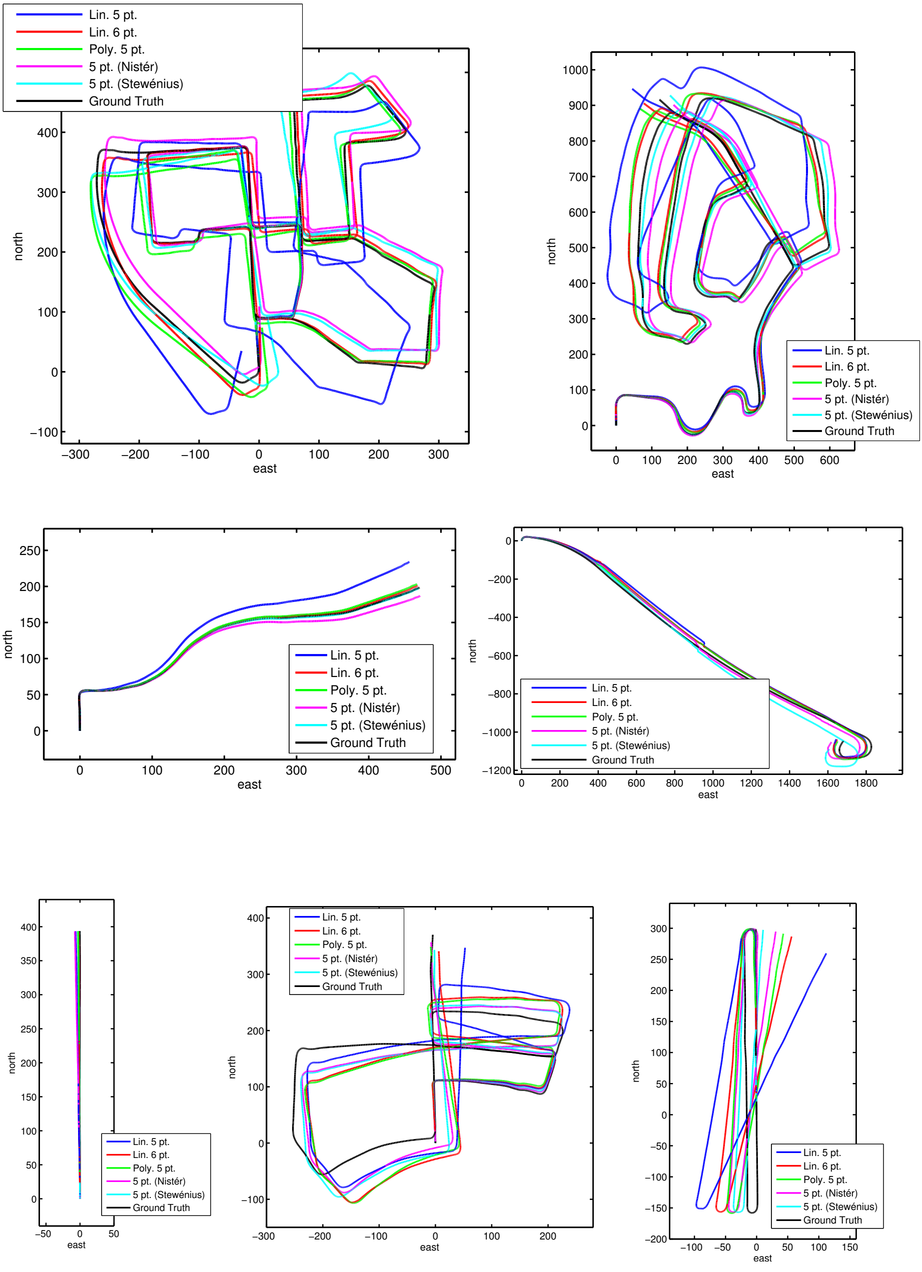

Fig. 4. Estimated trajectories for KITTI sequences with several solvers. Our solutions provide results similar to the existing solvers, while being as fast or faster to compute.

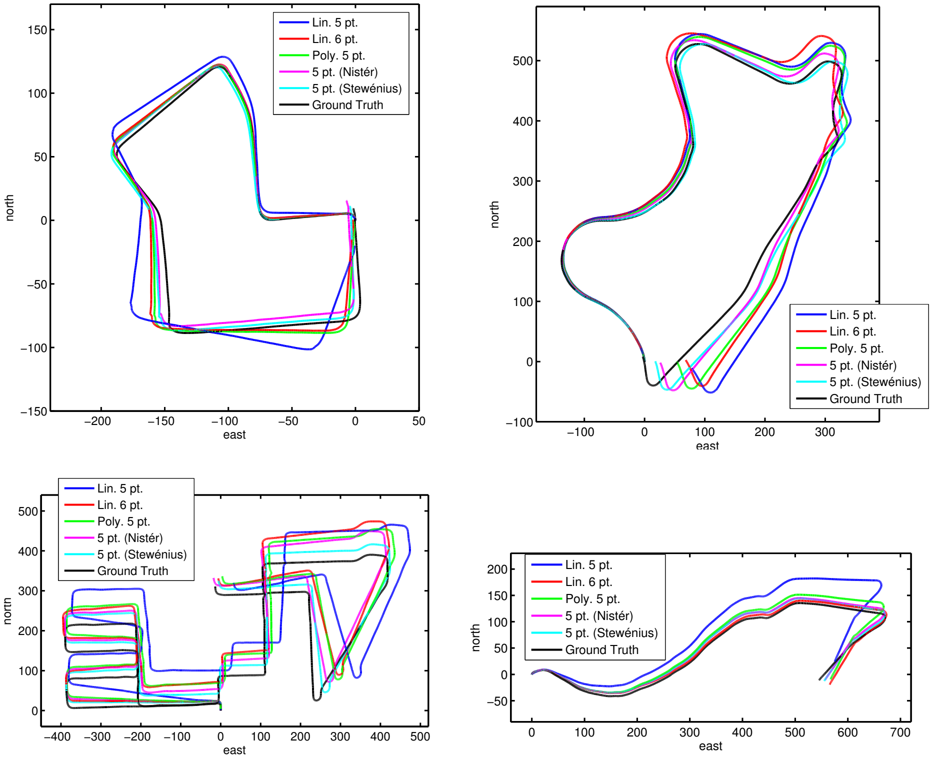

Fig. 5. Estimated trajectories for KITTI sequences with several solvers (cont.). Our solutions provide results similar to the existing solvers, while being as fast or faster to compute.

come from the inaccurate estimates at image pairs observing larger rotations, while the majority of image pairs have correctly estimated relative pose.

6 Conclusions

In this work we presented several novel solutions to the five-point relative pose problem. By applying an approximated rotation representation, we produce a system of equations involving only the rotation terms that can be solved by finding the roots of a single univariate polynomial. We also explored two alternate solution procedures involving truncation of higher-order terms and linearization. Our methods are as fast or faster than the state-of-the-art, while exhibiting similar accuracy for small rotation magnitudes.

Evaluation on image sequences a from vehicle-mounted camera show that our solvers are very competitive and suitable for such an application. Although the solvers are, to varying degrees, limited in the amount of rotation they can handle, they are also generally as fast or faster than the state-of-the-art. This implies that they would permit a higher frame-rate camera to be used, since the processing time per-frame is reduced. This in turn reduces the amount of rotation which is expected to be observed, and so makes the approach viable. Manual optimization of the solvers would improve their speed even further. Also, the simplicity of the solution procedures is such that we believe they could be easily implemented on embedded hardware.

The approaches used in this work – approximating the rotation matrix, discarding polynomial terms of higher order, and resolution by linearization – are most certainly not limited to the relative pose problem. The success of this approach encourages us to consider other minimal problems found in the literature in the future, for which we could possibly create considerably improved solvers. Examples where this approach might also be applicable are the eight-point radial distortion problem, the six-point calibrated radial distortion problem or, more prominently, the four-point absolute pose problem with unknown focal length.

References

1. Cayley, A.: About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic, vol. 32. Reine Angewandte Mathematik (1846)
2. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties and Algorithms. Springer (2006)
3. Demazure, M.: Sur deux problèmes de reconstruction. Tech. Rep. 882, Rocquencourt, France: INRIA (1988)
4. Faugeras, O.D., Maybank, S.: Motion from Point Matches: Multiplicity of Solutions. International Journal of Computer Vision 4(3), 225–246 (1990)
5. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24(6) (1981)
6. Fraundorfer, F., Scaramuzza, D.: Visual odometry: Part ii - matching, robustness, and applications. IEEE Robotics and Automation Magazine 19(2) (2012)
7. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets Robotics: The KITTI Dataset. International Journal of Robotics Research (2013)
8. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: Dense 3d reconstruction in real-time. In: IEEE Intelligent Vehicles Symposium (IV) (2011)
9. Kalantari, M., Jung, F., Guedon, J.P., Paparoditis, N.: The Five Points Pose Problem: A New and Accurate Solution Adapted to any Geometric Configuration. Advances in Image and Video Technology pp. 215–226 (2009)
10. Kipnis, A., Shamir, A.: Advances in Cryptology, chap. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization, pp. 19–30. Springer Berlin / Heidelberg (1999)
11. Kneip, L., Furgale, P.: OpenGV: A Unified and Generalized Approach to Real-Time Calibrated Geometric Vision. In: Proceedings of the IEEE International Conference on Robotics and Automation (2014)
12. Kneip, L., Lynen, S.: Direct optimization of frame-to-frame rotation. In: Proceedings of the IEEE International Conference on Computer Vision (December 2013)
13. Kneip, L., Siegwart, R., Pollefeys, M.: Finding the Exact Rotation between two Images Independently of the Translation. In: Proceedings of the European Conference on Computer Vision (2012)

14. Kruppa, E.: Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. Sitzungsberichte der Mathematisch Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften 122, 1939–1948 (1913)
15. Kukelova, Z., Bujnak, M., Pajdla, T.: Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence (2012)
16. Li, H.: A Simple Solution to the Six-Point Two-View Focal Length Problem. In: Proceedings of the European Conference on Computer Vision. pp. 200–213 (2006)
17. Li, H., Hartley, R.: Five-Point Motion Estimation Made Easy. In: Proceedings of the International Conference on Pattern Recognition. pp. 630–633. Ieee (2006)
18. Lim, J., Barnes, N., Li, H.: Estimating Relative Camera Motion from the Antipodal-Epipolar Constraint. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(10), 1907–1914 (2010)
19. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. Nature 293(5828), 133–135 (09 1981)
20. Maybank, S.: Theory of Reconstruction from Image Motion. Springer-Verlag, New York, Inc., Secaucus, NJ, USA (1992)
21. Nistér, D.: Preemptive RANSAC for Live Structure and Motion Estimation. Machine Vision and Applications 16(5), 321–329 (2005)
22. Nistér, D.: An Efficient Solution to the Five-Point Relative Pose Problem. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2003)
23. Philip, J.: A Non-Iterative Algorithm for Determining All Essential Matrices Corresponding to Five Point Pairs. The Photogrammetric Record 15(88), 589–599 (1996)
24. Stewénius, H., Engels, C., Nistér, D.: Recent Developments on Direct Relative Orientation. ISPRS Journal of Photogrammetry and Remote Sensing 60, 284–294 (Jun 2006)
25. Stewenius, H., Engels, C., Nistér, D.: An Efficient Minimal Solution for Infinitesimal Camera Motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007)