2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Approximating the Polynomial System for Effective Relative Pose Estimation

Deshun Hu[1,*]

Abstract— Finding relative pose for cameras is of vital importance in computer vision and robotics. We investigate the problem of relative motion estimation between successive frames from a minimal number of correspondences. Existing approximated methods use a first-order approximation to relative pose in order to simplify the problem and produce an estimate quickly. Our solution uses Cayley parameterization to represent rotation and simplifies the high-degree polynomials only at the very end of the formulation, resulting in more accurate models. Furthermore, our method can be more effective if the camera rotates mainly around one coordinate axis. By treating the main rotation component as the hidden variable in the solution, we can retain more high-degree terms for the main rotation part, considerably widening the effective approximation range. Our experiments show that our method is more accurate than existing approximated solver, and that it is still effective for relatively large motions. Besides, our method produces far fewer solutions than essential matrix parameterized solvers.

Index Terms— Rotation Approximation, Minimal Solver, Relative Pose Estimation, SLAM

I. INTRODUCTION

Computing the motion of cameras from the observed movements of points in successive frames has attracted significant interests due to its wide applications in visual odometry (VO), structure from motion (SFM) and simultaneous localization and mapping (SLAM). Relative pose for calibrated cameras is usually characterized by the essential matrix and epipolar geometry [1], [2]. Fig. 1 shows the problem of relative pose estimation. Due to scale ambiguity of translation, the essential matrix has only five degrees of freedom, hence five point correspondences are sufficient to identify the relative pose [3]. The so-called minimal solvers use fewest point correspondences possible to efficiently determine the relative pose and are usually used in a hypothesize and test framework (RANSAC) [4] to gain robustness against outliers.

Minimal relative pose solvers can be utilized to bootstrap the SFM and SLAM, where we can assume that the rotations between consecutive frames are small. The small rotation assumption makes the approximated motion model an appropriate choice [6]–[8]. Existing approximated solvers derive their solutions to the relative pose problem by applying the first-order approximation to relative pose, which simplifies the problem and leads to much simpler equations. First-order approximation methods are computationally very efficient, but their accuracy is inadequate as the approximation is only valid for very small rotations. As a general rule, these methods [6]–[8] should only be used when the amount of rotation is less than 4°.

(a) Relative pose problem. (b) Point correspondences between two images.

Fig. 1. Relative pose problem from point correspondences. Blue triangles represent cameras and stars are the landmarks in 3D space. Minimal relative pose solver estimates the relative transform of cameras from 5 point correspondences. Two images in (b) are from KITTI dataset [5].

In this work, we derive a novel approximated minimal solver to the relative pose problem for cameras, which is applicable for significantly larger motions and has higher accuracy. Rather than utilizing first-order approximation for rotation, we proceed with Cayley rotation parameterization and then simplify the polynomials at the end of the derivation, producing better approximations. Furthermore, as seen in Fig. 2, cameras on ground robots usually rotate mostly around the perpendicular axis. We propose using the rotation component in the perpendicular axis as the algorithm's hidden variable [9], and keeping more high-degree terms for this major rotation part. Our method can handle significantly larger rotations than [6] while being computationally efficient enough for practical use. Furthermore, our method generates fewer solutions than essential matrix parameterized solvers [1], [2].

The main technical contributions of our work are:

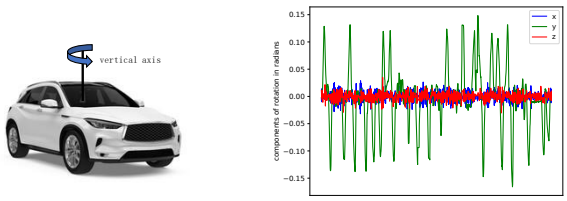
1) We propose only approximating polynomial system at the end of the derivation to obtain a more accurate model;
2) We use the hidden variable method to keep high-degree terms in the primary rotation component;
3) We validate the effectiveness of the proposed approach for small and large motions on both synthetic and real datasets.

II. RELATED WORK

The relative pose problem is discussed extensively in photogrammetry and computer vision. In [1], a 5-

[1]Deshun Hu is with Department of Communication Engineering, Harbin Institute of Technology, Harbin, China hudeshun1993@gmail.com
*Corresponding author, E-mail: hudeshun1993@gmail.com

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9811709



(a) Vehicles on ground plane rotate primarily around the vertical axis.

(b) Components of the relative rotation between consecutive images in downsampled KITTI sequence 00.

Fig. 2. (a) shows a vehicle on the ground plane. (b) shows the $x/y/z$ components of the angle-axis form of the relative rotation between consecutive images in downsampled KITTI sequence 00 [5]. We downsample the sequence by taking every third image. The x-axis of (b) represents the images sorted by time. The relative rotations only have large components along the y axis, as can be observed.

point method based on Gauss-Jordan elimination and hidden variable method is introduced, extracting the roots of a tenth-degree polynomial via Sturm sequences. Solving polynomial equations with Groebner bases is fairly common [10], and its use in relative pose has been proposed [2], [11]. Methods based on the hidden variable resultant and polynomial eigenvalue method are proposed in [9], [12], [13], but they are less efficient. Estimating the rotation and translation separately is introduced in [14], and is later incorporated in an efficient eigenvalue minimization framework to provide more stable solutions [15]. Using direct linear transform (DLT) [3] on the epipolar constraints is another popular and efficient option. The DLT method does, however, necessitate at least 8 point correspondences, and the solutions are not on the essential manifold. For essential matrix parameterized solvers, the solutions must be decomposed and checked for cheirality [3].

There are many solvers with motion priors, such as known gravity direction and known rotation angle [16], [17]. When the gravity direction is known [16], rotation has only one degree of freedom, and the relative pose problem can be greatly simplified. Solvers for known rotation angle are substantially more complex than 5-point approaches [17], even though the degrees of freedom are decreased by 1. Recently, non-minimal relative pose problem is investigated [15], [18], showing more accurate results than 5-point methods. However, in practice, non-minimal solvers require more point correspondences than the minimal solvers, and are usually computational much more complex. In [19], the authors there propose an efficient approach for globally optimal estimation of relative pose problem with gravity prior, by applying the eigenvalue minimization framework in [15].

With the assumption of small rotation, several algorithms have been proposed with the first-order approximation of rotation matrix [6]–[8]. The first-order rotation approximation assumes that the rotation is small between consecutive images, but does not enforce any other motion constraints, such as Ackerman motion, planar motion or rotation about a known axis. Based on linearization and truncation of high-degree terms, authors in [6] obtain smaller polynomial systems and faster solvers for relative pose problem. An extension to multi-camera motion estimation is proposed in [7], which produces accurate results and has very small computational complexity. In [8], the authors consider gravity prior and small rotation approximation together and propose a simple and effective method using only linear algebra operations. These methods are very fast and efficient, and while accurate for very small rotations, their performance drops dramatically even for only slightly faster motions.

III. PRELIMINARIES

In this section, we first describe the notations used in this paper. Then we briefly recall some notions from multiple view geometry [3], e.g. the epipolar constraint which is needed to follow the derivation of our minimal solver. Our derivation for this section mainly follows [17], [20].

A. Notation

We denote matrices with upper case boldface letters and vectors with lower case boldface letters, e.g. $\mathbf{A}$ and $\mathbf{x}$. The transpose of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}^\top$ and the determinant is $\det(\mathbf{A})$. And $\text{trace}(\mathbf{A})$ is the trace of matrix $\mathbf{A}$. $A_{i,j}$ stands for the entry of $\mathbf{A}$ indexed by row i and column j, and x_i denotes the ith element of $\mathbf{x}$. For two 3D vectors $\mathbf{a}$ and $\mathbf{b}$, the cross product is $\mathbf{a} \times \mathbf{b}$ and we denote by $\lfloor \mathbf{a} \rfloor_\times$ the matrix form for the cross product with $\mathbf{a}$. $\mathbf{0}$ is the zero matrix whose dimension can be inferred from the context. Similarly the identity matrix is written as $\mathbf{I}$. Terms and monomials [21] are used interchangeably in this work.

B. Epipolar geometry

We represent two given calibrated cameras as 3×4 matrices of the form

$$\mathbf{P}_1 = \left[\mathbf{R}_1, \mathbf{t}_1\right], \quad \mathbf{P}_2 = \left[\mathbf{R}_2, \mathbf{t}_2\right], \tag{1}$$

where $\mathbf{R}_1, \mathbf{R}_2$ are two rotation matrices and $\mathbf{t}_1, \mathbf{t}_2$ are two translation vectors. Let $\mathbf{q}$ be a homogeneous 3D point in 3-space, $\mathbf{p}_1$ and $\mathbf{p}_2$ be its homogeneous images, that is

$$\mathbf{p}_1 \sim \mathbf{P}_1\mathbf{q}, \quad \mathbf{p}_2 \sim \mathbf{P}_2\mathbf{q}, \tag{2}$$

where $\sim$ means an equality up to scale.

With this formulation, the epipolar geometry [3], [17] can be expressed as

$$\mathbf{p}_2^\top(\mathbf{R}\lfloor \mathbf{t}_1 \rfloor_\times - \lfloor \mathbf{t}_2 \rfloor_\times \mathbf{R})\mathbf{p}_1 = 0, \tag{3}$$

where $\mathbf{R} = \mathbf{R}_2^\top \mathbf{R}_1$ is the relative rotation between two frames. A key oberservation is that (3) is homogeneous in $\mathbf{R}$. We can use a scaled rotation matrix, i.e. the scaled Cayley parameterization, instead.

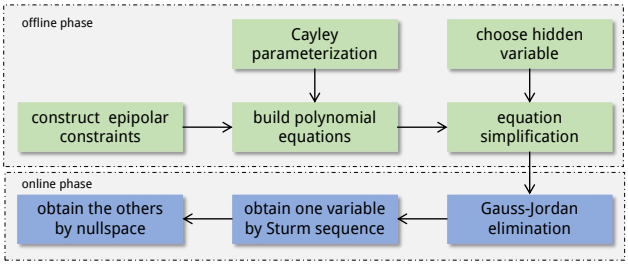


Fig. 3. The process of our algorithm. During the offline phase, we use symbolic computation to build and simplify the equations. Online, we substitute variables and perform Gauss-Jordan elimination. The solution to the hidden variable is found via the Sturm sequence method, while solutions to other variables are calculated by the null space.

C. Rotation representation

When relative rotation is small, we can apply first-order approximation to the rotation matrix $\mathbf{R}$, parameterized with angle-axis vector $\mathbf{r}$

$$\mathbf{R} \approx \mathbf{I} + \lfloor \mathbf{r} \rfloor_\times. \tag{4}$$

This parameterization is used in [6]–[8] for small motions. As the first-order approximation is not accurate for large motions, we use Cayley parameterization for rotation in our derivation. Rotation $\mathbf{R}$ using Cayley parameterization can be written as

$$\mathbf{R} = \frac{1}{1+x^2+y^2+z^2} \cdot \begin{bmatrix} 1+x^2-y^2-z^2 & 2xy-2z & 2xz+2y \\ 2xy+2z & 1-x^2+y^2-z^2 & 2yz-2x \\ 2xz-2y & 2yz+2x & 1-x^2-y^2+z^2 \end{bmatrix}, \tag{5}$$

where $[1, x, y, z]^\top$ is a homogeneous quaternion vector.

Since epipolar constraint is homogeneous in $\mathbf{R}$, we can ignore the common coefficient in (5), and rewrite the scaled rotation matrix as

$$\mathbf{R} = \begin{bmatrix} 1+x^2-y^2-z^2 & 2xy-2z & 2xz+2y \\ 2xy+2z & 1-x^2+y^2-z^2 & 2yz-2x \\ 2xz-2y & 2yz+2x & 1-x^2-y^2+z^2 \end{bmatrix}. \tag{6}$$

This formulation for $\mathbf{R}$ is used for our derivation in the sequel.

IV. SOLUTION PROCEDURE

In this section, we derive the solution to the relative pose problem with 5 point correspondences. We isolate the rotation and translation parameters, and solve for the rotation parameters using the hidden variable trick [9]. Fig. 3 shows the flow chart of our algorithm.

A. Constructing equations

The initial data consists of five-point correspondences given as bearing vectors $\mathbf{p}_{i1} \leftrightarrow \mathbf{p}_{i2}$, $i = 1, \cdots, 5$. We build epipolar equations from the point correspondences using the similar procedure in [17]. From [17], for every distinct three indices $i, j, k \in 1, \cdots, 5$, the epipolar constraint (3) yields

$$\mathbf{F}_{ijk} \begin{bmatrix} \lambda_i \\ \mu_i \end{bmatrix} = \mathbf{0}, \tag{7}$$

where λ_i and μ_i are some scalars, and $\mathbf{F}_{ijk}$ is defined as

$$\mathbf{F}_{ijk} = \begin{bmatrix} \mathbf{p}_{j2}^\top \mathbf{R} \mathbf{p}_{ij1} & \mathbf{p}_{ij2}^\top \mathbf{R} \mathbf{p}_{j1} \\ \mathbf{p}_{k2}^\top \mathbf{R} \mathbf{p}_{ik1} & \mathbf{p}_{ik2}^\top \mathbf{R} \mathbf{p}_{k1} \end{bmatrix}, \tag{8}$$

where $\mathbf{p}_{ij1}$ is defined as $\mathbf{p}_{i1} \times \mathbf{p}_{j1}$ and similarly for $\mathbf{p}_{ij2}$. Since $\mathbf{F}_{ijk}$ has a null vector, the determinant of $\mathbf{F}_{ijk}$ must equal zero, i.e. $\det(\mathbf{F}_{ijk}) = 0$. Furthermore, from [17], the following identity holds

$$\det(\mathbf{F}_{ijk}) = \det(\mathbf{F}_{jki}). \tag{9}$$

With different indices i, j and k, one can obtain several equations $\det(\mathbf{F}_{ijk}) = 0$, of which only 10 equations are actually distinct due to identity (9). The 10 equations, denoted as f_i, $i = 1, \cdots, 10$, only involve the rotation. When the scaled Cayley parameterization (6) is substituted into the rotation in $\mathbf{F}_{ijk}$, it is clear that f_i, $i = 1, \cdots, 10$, are fourth-degree polynomials in Cayley rotation parameter $[x, y, z]^\top$. There are 35 monomials in three variables with total degree at most 4 [7].

B. Simplifying equations

We now develop the approximation scheme for solving the ten fourth-degree equations with 35 monomials in three variables. Without loss of generality, we assume that the camera rotates mainly around the y axis[1] and we want to preserve more high-degree terms in y to improve the model accuracy. We propose to keep all monomials with total degree at most 3 and remove fourth-degree monomials whose degrees in y are low in the equations. To be specific, we delete fourth-degree terms whose degrees in y are at most 2, e.g. x^3y and xy^2z. Monomials with total degree at most 3 are referred to as low-degree monomials.

We will now go into the reasons for the simplication. Since the camera rotates mainly around y axis, the y component in Cayley parameter is much larger than other components. Fourth-degree monomials in x and z only, e.g. x^3z and z^4, are negligible and hence are removed in the polynomials for simplicity. Fourth-degree terms with degree in y at most 2 are deleted as well, as they are less significant than other fourth-degree monomials, e.g. y^4 and y^3z. In other words, we retain all low-degree terms and fourth-degree terms whose degrees in y are at least 3. The kept fourth-degree monomials are xy^3, y^4 and y^3z. Together with 20 low-degree monomials, we have 23 retained monomials in total.

Compared to the approximated solver [6], which uses first-order approximation and whose resulting polynomials contain only third-degree or lower terms, our

[1]Despite assuming a main rotation axis, our method still outperforms [6] for general rotations, as demonstrated by the experiments.

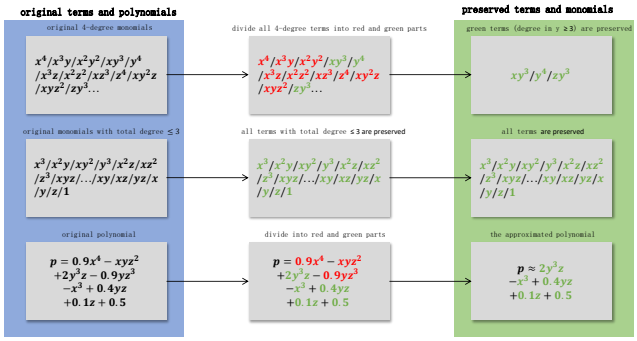


Fig. 4. Illustration for the approximation scheme. For fourth-degree terms, only those with degree in y at least 3 are retained, while terms with total degree at most 3 are all kept. An example polynomial is shown at the bottom of the figure. The details can be seen by zooming in.

model retains more high-degree terms, especially in the main rotation component y. And, rather than using the first-order approximation throughout the derivation [6]–[8], our method approximates the polynomials only at the very end. Fig. 4 illustrates our procedure for the approximation.

C. Solving equations

We can solve the equations with remaining monomials by reducing the problem to a single polynomial in y [1]. By treating y as a hidden variable, all remaining terms have total degrees at most three when expressed in x and z only.

The approximation scheme results in 10 equations with 23 monomials in three variables. We order the 23 monomials as

$$\begin{aligned}\mathbf{m} = [&x^3, z^3, x^2z, xz^2, x^2y, x^2, z^2y,\\ &z^2, xzy, xz, xy^3, xy^2, xy, x,\\ &zy^3, zy^2, zy, z, y^4, y^3, y^2, y, 1]^\top.\end{aligned} \tag{10}$$

We then arrive at a matrix equation $\mathbf{Gm} = \mathbf{0}$, where $\mathbf{G} \in \mathrm{R}^{10\times 23}$ is the coefficient matrix extracted from f_i, $i = 1, \cdots, 10$. Performing Gauss-Jordan elimination on $\mathbf{G}$ gives 10 new equations g_i, $i = 1, \cdots, 10$. g_i corresponds to the i-th row of the reduced form of $\mathbf{G}$. Because the ordering of the left 10 monomials in (10) is identical to that of [1], second-degree and third-degree terms in x and z can be eliminated using the same procedure. Despite the presence of 3 fourth-degree monomials in the right part of the ordering, we get the same type of equations as [1] if y is regarded as the hidden variable. In fact, the kept fourth-degree terms are linear in x and z after y is chosen as the hidden variable.

We remove high-degree terms in x and z from these equations by elimination

$$\begin{aligned} b_1 &= g_5 - y \cdot g_6,\\ b_2 &= g_7 - y \cdot g_8,\\ b_3 &= g_9 - y \cdot g_{10}.\end{aligned} \tag{11}$$

The obtained equations b_i, $i = 1, 2, 3$, are linear in x and z if y is considered as a constant. Thus by treating y as a hidden variable, one can further reduce (11) to a matrix equation

$$\mathbf{B}(y)\begin{bmatrix} x \\ z \\ 1 \end{bmatrix} = \mathbf{0}, \tag{12}$$

where $\mathbf{B}(y)$ is a 3×3 matrix in y only. It is worth noting that, owning to our special recipe for approximation, the degrees of the entries of $\mathbf{B}(y)$ are one greater than the degrees of the entries in [1]. Following the standard procedure in the hidden variable method, we reduce our problem to a single polynomial in y

$$\det(\mathbf{B}(y)) = 0. \tag{13}$$

Once solutions for y are found by Sturm sequences, the corresponding solutions for x and z are determined from the null vector of $\mathbf{B}(y)$. The rotation is obtained from the Cayley parameters through (5) and the translation is computed as in [17], [20]. Concretely, the translation is derived from the epipolar constraints as the right null vector of the matrix

$$\begin{bmatrix} \mathbf{p}_{11}^\top \mathbf{R}^\top [\mathbf{p}_{12}]_\times \\ \mathbf{p}_{21}^\top \mathbf{R}^\top [\mathbf{p}_{22}]_\times \\ \mathbf{p}_{31}^\top \mathbf{R}^\top [\mathbf{p}_{32}]_\times \\ \mathbf{p}_{41}^\top \mathbf{R}^\top [\mathbf{p}_{42}]_\times \\ \mathbf{p}_{51}^\top \mathbf{R}^\top [\mathbf{p}_{52}]_\times \end{bmatrix}, \tag{14}$$

where $\mathbf{R}$ is the estimated rotation matrix.

Our method simplifies the polynomial model only at the final stage and preserves more high-degree monomials than [6], while still allowing us to utilize the solution process designed for third-degree polynomials in [1]. As demonstrated in our experiments, our method consistently outperforms the method in [6].

V. EXPERIMENTS

In this section, we conduct experiments on synthetic and real-world data to evaluate the performance of the proposed solver in various aspects. In evaluating the performance of the proposed method, we compare the relative rotation and translation accuracy, following the geodesic distance criteria for quantitative evaluation. For rotation, geodesic distance between the estimated rotation $\mathbf{R}_{est}$ and ground truth $\mathbf{R}_{gt}$ is defined as

$$\epsilon_{rot}[degree] = \frac{180}{\pi}\arccos\left(\frac{\mathrm{trace}(\mathbf{R}_{gt}^\top \mathbf{R}_{est}) - 1}{2}\right). \tag{15}$$

Similarly, the translation direction error is defined as

$$\epsilon_{tran}[degree] = \frac{180}{\pi}\arccos\left(\frac{\mathbf{t}_{gt}^\top \mathbf{t}_{est}}{\|\mathbf{t}_{gt}\|\|\mathbf{t}_{est}\|}\right). \tag{16}$$

The two errors are measured in degrees. In the analysis, we mainly focus on the median errors because all solvers are used in a RANSAC framework in practice.

We run all solvers 3000 times on a consumer CPU to calculate the average computation time, which is

shown in Table I. The computation time includes the time spent on all steps from input correspondences to output poses. For relative pose estimation, our method is already fast enough for usage. In Table I, we also show the average number of solutions (output poses) for all the methods in KITTI sequence 00. Our method has far fewer solutions than [1], [2], therefore it spends much less time on cheirality checking[2]. See following subsection for detailed descriptions of the compared methods. All simulation experiments in this section are repeated for 3000 times.

TABLE I

Computation time and number of solutions of several methods.

Method	Time(in us)	Number of solutions
Nister-5pt [1]	52.1	17.3
Ventura-5pt [6]	13.4	7.4
Stew-5pt [2]	40.2	40
Earp-5pt	24.3	8.0

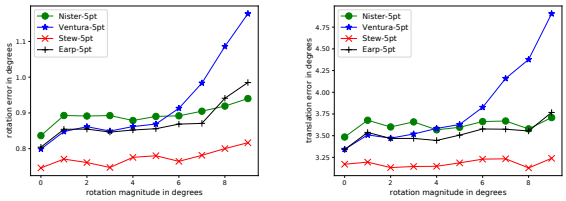
A. Synthetic data

We refer to the proposed 5-pt method as Earp-5pt. We compare our method against the following existing methods: Nister-5pt and Stew-5pt are the essential matrix solvers in [1] and [2] respectively, and Ventura-5pt is the first-order approximation method proposed by [6]. For Nister-5pt and Stew-5pt, we use the implementations in [22]. We also test our method with only third-degree or lower terms kept. This simpler method is more efficient but is less accurate for large motions, so we skip the analysis for it.

For synthetic data, we generate random problems in a similar manner to [18]. The random translation is constrained within a spherical shell with radius $[1, 2]$. We generate a set of 3D world points randomly around the origin with distances varying between 2 and 20, retaining points that lie fully inside the Field of View (FOV) of both cameras. Both cameras are assumed to have the same focal length 800 with image width 1600 pixels and height 900 pixels. When cameras rotate mainly around the y axis, the amount of rotation in the y axis is one order of magnitude greater than the total amount of rotation in the other two axes. We perturb each point pixel observation with Gaussian noise to make it consistent with the physical constraints. The standard deviation of the Gaussian noise is referred to as the noise level.

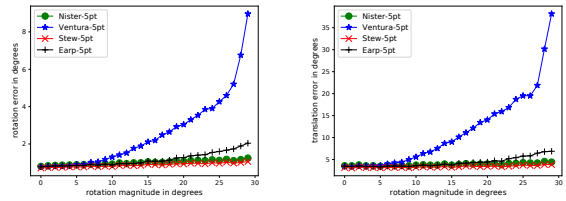
We first test the performance of several methods for rotation around an arbitrary axis with fixed noise level of 1 pixel and varying amount of rotation. The result is shown in Fig. 5. For general rotation, i.e. rotation around an arbitrary axis, our method consistently outperforms

[2]In practice, for relative pose problem, cheirality checking consumes the majority of the computing time in the RANSAC loop. As a result, the computation advantage of our method over [1], [2] is significant.



(a) Rotation error (b) Translation error

Fig. 5. The median rotation and translation errors for general rotations. The x-axis is the amount of relative rotation in degrees. The performance of approximated solvers degrade as the rotation becomes more significant. Our method outperforms Ventura-5pt, especially when rotation angle is greater than 3°. Our method is also better than Nister-5pt when rotation angle is at most 7°.



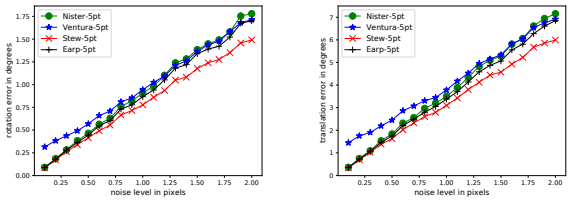
(a) Rotation error (b) Translation error

Fig. 6. For rotation mainly around y axis, the rotation and translation errors of several methods are presented. The x-axis is the amount of rotation in degrees. Our solver is still effective for a rotation angle of 20°, while Ventura-5pt is only suitable for angles less than 5°. Our method is more accurate than Nister-5pt when rotation angle is at most 15°.

the first-order method [6] and has better accuracy than Nister-5pt when rotation angle is less than 7°, whereas Stew-5pt performs the best among all the methods. The rotation and translation errors show similar characteristics. We also conduct experiments for rotation mainly around y axis. In Fig. 6, our method can operate over a large range of motions and is still effective for a rotation angle of 20°, whereas the approximated solver [6] is only accurate when rotation angle is less than 5°. Our method is more accurate than Nister-5pt when rotation angle is at most 15°. We also test the performance of several methods with varying noise level in the range $[0.1, 2.0]$ with step 0.1 pixel. The result is depicted in Fig. 7. In low-noise cases, the method in [6] is less accurate than the other methods, as the model approximation error in [6] starts to dominate when the noise is very small. This validates the fact that our model is more accurate than [6]. The rotation angle is fixed to 6° in Fig. 7.

B. Real data

In order to test the proposed technique on real-world data, we choose two sequences (sequence 00 and 01) in the KITTI dataset [5], where the cameras primarily rotate along the y axis. To generate the correspondences, we extract FAST features and obtain feature matches between consecutive images in the left camera from forward and backward optical flow. Point correspondences with correct optical flow are further filtered by the



(a) Rotation error (b) Translation error

Fig. 7. The median rotation and translation errors of several methods for increased noise levels. The x-axis is the noise levels in pixels. In low-noise cases, the method in [6] is less accurate than the other methods, as the model approximation error starts to dominate when the noise is very small.

fundamental matrix. Next, we obtain the corresponding bearing vectors by using the pinhole camera model with the intrinsic parameters provided in [5]. To prevent the baseline between two consecutive images from being too small, we reduce the number of images by taking only every third image in the sequences. For each image pair, we sample five point correspondences for 3 times to avoid singular configurations and save the best estimation result for comparison. After that, we compute the rotation and translation errors for each pair of consecutive frames using the ground truth. The mean rotation and translation errors in degrees are presented in Table II. As can be seen from the result, our method is more accurate than [1] and [6]. To further demonstrate the superiority of our method over [6] for large rotations, we show the rotation errors for various amounts of rotations in sequence 00 of KITTI dataset in Table III. We partition image pairs in sequence 00 based on the amount of relative rotation, and calculate the rotation error for each part. Our method is less sensitive to the amount of rotation, whereas the performance of [6] degrades rapidly for large rotations.

TABLE II

Accuracy of relative pose solvers on KITTI. Both rotation and translation errors are in degrees. Rot.(00) denotes the rotation error for sequence 00, and similarly for others.

Method	Rot.(00)	Tran.(00)	Rot.(01)	Tran.(01)
Nister-5pt	0.361	4.08	0.30	8.07
Ventura-5pt	0.367	4.12	0.29	8.06
Stew-5pt	0.277	3.45	0.233	8.289
Earp-5pt	0.350	4.03	0.286	8.09

We also test our method on two sequences from the EuRoc dataset [23], where the camera can rotate along an arbitrary axis. The settings are similar to those used for the KITTI dataset, with the exception that we take only every sixth image in the sequences from the left camera. Table IV shows the mean rotation and translation errors in degrees. The result supports the efficacy of our method for general type of rotations.

TABLE III

Rotation errors in degrees for all the solvers on KITTI sequence 00 for various amounts of rotations. 0°-4° denotes the specific rotation range, and similarly for others. The last row counts the number of effective image pairs that fall within the specified rotation range.

Method	0°-4°	4°-8°	8°-12°	12°-16°
Nister-5pt	0.286	0.519	0.652	0.671
Ventura-5pt	0.274	0.639	0.784	0.943
Stew-5pt	0.225	0.438	0.478	0.492
Earp-5pt	0.274	0.467	0.565	0.586
Num. images	1253	121	128	11

TABLE IV

Accuracy of relative pose solvers on EuRoc dataset. All errors are in degrees. Rot.(01) denotes the rotation error for sequence MH01, and similarly for others.

Method	Rot.(01)	Tran.(01)	Rot.(02)	Tran.(02)
Nister-5pt	0.97	15.3	1.43	17.7
Ventura-5pt	0.96	15.1	1.45	17.6
Stew-5pt	0.85	14.2	1.29	17.1
Earp-5pt	0.94	14.7	1.43	17.3

VI. CONCLUSIONS

In this work we present a novel solution to the 5-pt relative pose problem. We achieve an efficient solution to relative pose by using a different approximation scheme that is effective for large motions. Our solver uses minimal correspondences and can handle large motions when the camera rotates primarily around one coordinate axis. Our algorithm consistently outperforms previous approximated solver, and is more efficient than essential matrix solvers. By carefully selecting kept terms in the polynomial system, we obtain a significantly more accurate model than the first-order method while the solution procedure remains largely the same. We intend to apply this idea to other computationally complex geometric problems in the future.

References

[1] D. Nistér, "An efficient solution to the five-point relative pose problem," IEEE transactions on pattern analysis and machine intelligence, vol. 26, no. 6, pp. 756–770, 2004.

[2] H. Stewenius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 60, no. 4, pp. 284–294, 2006.

[3] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. New York, NY, USA: Cambridge University Press, 2 ed., 2003.

[4] 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, vol. 24, no. 6, pp. 381–395, 1981.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.

[6] J. Ventura, C. Arth, and V. Lepetit, "Approximated relative pose solvers for efficient camera motion estimation," in European Conference on Computer Vision, pp. 180–193, Springer, 2014.

[7] J. Ventura, C. Arth, and V. Lepetit, "An efficient minimal solution for multi-camera motion," in Proceedings of the IEEE International Conference on Computer Vision, pp. 747–755, 2015.
[8] L. Liu, H. Li, and Y. Dai, "Robust and efficient relative pose with a multi-camera system for autonomous vehicle in highly dynamic environments," arXiv preprint arXiv:1605.03689, 2016.
[9] R. Hartley and H. Li, "An efficient hidden variable approach to minimal-case camera motion estimation," IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 12, pp. 2303–2314, 2012.
[10] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in European Conference on Computer Vision, pp. 302–315, Springer, 2008.
[11] M. Kalantari, F. Jung, J.-P. Guedon, and N. Paparoditis, "The five points pose problem: A new and accurate solution adapted to any geometric configuration," in Pacific-Rim Symposium on Image and Video Technology, pp. 215–226, Springer, 2009.
[12] Z. Kukelova, M. Bujnak, and T. Pajdla, "Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems.," in BMVC, vol. 2, p. 2008, 2008.
[13] H. Li and R. Hartley, "Five-point motion estimation made easy," in 18th International Conference on Pattern Recognition (ICPR'06), vol. 1, pp. 630–633, IEEE, 2006.
[14] L. Kneip, R. Siegwart, and M. Pollefeys, "Finding the exact rotation between two images independently of the translation," in European conference on computer vision, pp. 696–709, Springer, 2012.
[15] L. Kneip and S. Lynen, "Direct optimization of frame-to-frame rotation," in Proceedings of the IEEE International Conference on Computer Vision, pp. 2352–2359, 2013.
[16] Y. Ding, J. Yang, J. Ponce, and H. Kong, "Minimal solutions to relative pose estimation from two views sharing a common direction with unknown focal length," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7045–7053, 2020.
[17] E. Martyushev and B. Li, "Efficient relative pose estimation for cameras and generalized cameras in case of known relative rotation angle," Journal of Mathematical Imaging and Vision, vol. 62, no. 8, pp. 1076–1086, 2020.
[18] J. Briales, L. Kneip, and J. Gonzalez-Jimenez, "A certifiably globally optimal solution to the non-minimal relative pose problem," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 145–154, 2018.
[19] Y. Ding, D. Barath, J. Yang, H. Kong, and Z. Kukelova, "Globally optimal relative pose estimation with gravity prior," arXiv preprint arXiv:2012.00458, 2020.
[20] J. Zhao and B. Guan, "On relative pose recovery for multi-camera systems," arXiv preprint arXiv:2102.11996, 2021.
[21] D. Cox, J. Little, and D. OShea, Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra. Springer Science & Business Media, 2013.
[22] L. Kneip and P. Furgale, "Opengv: A unified and generalized approach to real-time calibrated geometric vision," in 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–8, IEEE, 2014.
[23] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," The International Journal of Robotics Research, vol. 35, no. 10, pp. 1157–1163, 2016.