Globally Optimal Relative Pose Estimation with Gravity Prior

Yaqing Ding[1], Daniel Barath[2,3,4], Jian Yang[1], Hui Kong[1], Zuzana Kukelova[2]
[1] School of Computer Science and Engineering, Nanjing University of Science and Technology
[2] Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
[3] MPLab, SZTAKI, Budapest, [4] Department of Computer Science, ETH Zurich
dingyaqing@njust.edu.cn

Abstract

Smartphones, tablets and camera systems used, e.g., in cars and UAVs, are typically equipped with IMUs (inertial measurement units) that can measure the gravity vector accurately. Using this additional information, the y-axes of the cameras can be aligned, reducing their relative orientation to a single degree-of-freedom. With this assumption, we propose a novel globally optimal solver, minimizing the algebraic error in the least squares sense, to estimate the relative pose in the over-determined case. Based on the epipolar constraint, we convert the optimization problem into solving two polynomials with only two unknowns. Also, a fast solver is proposed using the first-order approximation of the rotation. The proposed solvers are compared with the state-of-the-art ones on four real-world datasets with approx. 50000 image pairs in total. Moreover, we collected a dataset, by a smartphone, consisting of 10933 image pairs, gravity directions and ground truth 3D reconstructions. The source code and dataset are available at https://github.com/yaqding/opt_pose_gravity

1. Introduction

Finding the relative pose between two cameras is one of the fundamental geometric vision problems with many applications, for example, in structure-from-motion [42, 40, 39, 2, 52, 44], visual localization [53, 48], and SLAM [34]. The robust relative pose estimation is usually done in a hypothesize and test framework, such as RANSAC [12].

Locally optimized (LO) RANSAC [7, 31] and its variants, such as Graph-Cut RANSAC [3], USAC [41], have shown a significant improvement in terms of accuracy and convergence compared to the standard RANSAC [12]. The key idea of these RANSAC variants is to apply a local optimization step to refine each so-far-the-best model using non-minimal samples. The main benefit of using a non-minimal sample is that it allows for averaging out observational noise in the measurements. Besides improving the accuracy, this often speeds up the robust estimation via finding an accurate model early and, thus, triggering the termination criterion. Both minimal and non-minimal solvers play important roles in the LO-RANSAC framework.

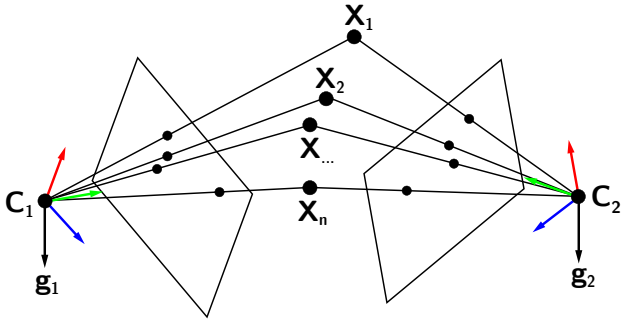


Figure 1: Two cameras $\mathbf{C}_1$ and $\mathbf{C}_2$ with known gravity directions $\mathbf{g}_1$, $\mathbf{g}_2$ observing $n \geq 4$ points $\mathbf{X}_1, ..., \mathbf{X}_n$. The objective is to find the relative pose of the cameras.

Minimal solutions to the relative pose estimation with different camera configurations have been well-studied in the literature. For instance, given two calibrated cameras, the relative pose can be efficiently recovered from five point correspondences [38, 28, 21]. Recently, point plus direction-based methods have shown benefits. The basic idea of such methods is to align one axis (*e.g.*, y-axis) of the cameras with the common reference direction, *e.g.*, the gravity vector. The relative orientation reduces to 1-DoF, and the relative pose can be estimated from only three point correspondences [13, 37, 49, 43, 10, 9].

While the problem of estimating the absolute camera pose from a non-minimal set of correspondences (the PnP problem) has received a lot of attention and many efficient and globally optimal solutions were published [24, 56, 25, 35, 36, 51, 57, 50], this does not hold for the relative pose estimation problem, also known as the *N-point problem*. The main reason is the fact that the N-point problem results in a more complicated optimization than the PnP problem.

The most commonly used non-minimal relative pose solver is the well-known direct linear transformation (DLT) method [22]. This method can be used to estimate the essential or the fundamental matrix from 8 (the well-known 8-point algorithm [17]) or more point correspondence. The DLT method assumes that the matrix elements are linearly independent. Therefore, this method does not minimize a properly defined error in the 5D space of essential matrices for calibrated cameras, and a correct essential matrix has to be recovered from the approximate DLT solution [22].

Therefore, several approaches that directly minimize an energy function defined in the 5D space of relative poses have been published recently. However, these approaches either do not guarantee a global optimum [18, 23, 32] or are too slow for practical applications [19, 6, 20].

Kneip et al. [26] presented an eigenvalue-based formulation that is minimizing an algebraic error for the relative pose problem, which is after eliminating the relative translation, parameterized in the 3-dimensional space of relative rotations. However, the proposed iterative Levenberg-Marquardt solution does not guarantee optimality and it requires a reasonably good initialization. The certifiably globally optimal solution to the eigenvalue-based formulation of relative pose estimation problem was proposed in [5]. This solution minimizes the algebraic error in the space of rotations and translations and formulates the problem as a quadratic program (QCQP) that is solved using a Semidefinite Program (SDP) relaxation. This method does not provide good pose estimates for forward motion. The solution was later extended to globally optimal essential matrix estimation for generalized cameras [55]. The globally optimal solution for the N-point problem [5] was recently improved in [54]. The proposed formulation results in QCQP, however, in fewer variables and constraints than the formulation in [5]. The solver is 2-3 orders of magnitude faster than [5].

In this paper, to estimate the relative pose globally optimally in real-time, we assume that the views share a common reference direction. This case is relevant since smartphones, tablets and camera systems used, *e.g.*, in cars and UAVs, are typically equipped with IMUs (inertial measurement units) that can measure the gravity vector accurately. With this assumption, the relative rotation is reduced to 1-DoF and the N-point relative pose problem is significantly simplified. The main contributions of this paper are:

- We propose a **novel real-time globally optimal solver** that minimizes the algebraic error in the least squares sense and estimates the relative pose of calibrated cameras with known gravity direction from N-point correspondences ($N \geq 4$). Based on the epipolar constraint, we convert the optimization problem into solving two polynomials with only two unknowns and we propose two different efficient solutions to such a system.
- In addition, we consider that for mobile robots, autonomous driving cars and UAVs, the relative rotation is small and, thus, we propose a **solver estimating the linearized rotation efficiently**.
- In extensive real and synthetic experiments we show an **improvement in terms of accuracy and speed** over the state-of-the-art non-minimal N-point relative pose solvers. Moreover, we present a **novel dataset** with 10993 image pairs taken by a smartphone with known gravity direction and ground truth 3D reconstructions.

2. Background

Suppose that we are given 3D point $\mathbf{X}_i$ observed in two calibrated views. Let $\mathbf{m}_i = [u_i, v_i, 1]^\top$ and $\mathbf{m}'_i = [u'_i, v'_i, 1]^\top$ be its projections in the two images in their homogeneous form. Since the gravity direction can be calculated from, *e.g.* an IMU, we can, without loss of generality, align the y-axes of the cameras with the gravity direction. These alignments are done by rotation matrices $\mathbf{R}$ and $\mathbf{R}'$, giving rotated image points $\mathbf{p}'_i = \mathbf{R}'\mathbf{m}'_i$, $\mathbf{p}_i = \mathbf{R}\mathbf{m}_i$. After applying the rotations to the projected 2D points, we get

$$\lambda'_i \mathbf{p}'_i = \lambda_i \mathbf{R}_y \mathbf{p}_i + \mathbf{t}, \tag{1}$$

with

$$\mathbf{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{2}$$

where θ is the unknown rotation angle around the vertical axis after the alignment by the gravity direction, $\mathbf{t}$ is the unknown translational vector and λ_i, λ'_i are unknown depths. Vectors $\lambda'_i\mathbf{p}'_i$, $\lambda_i\mathbf{R}_y\mathbf{p}_i$ and $\mathbf{t}$ are always coplanar. Therefore, the scalar triple product of these three vectors should be zero, and we obtain

$$([\mathbf{p}'_i]_\times \mathbf{R}_y \mathbf{p}_i)^\top \mathbf{t} = 0. \tag{3}$$

In this case, the depth parameters λ_i, λ'_i are eliminated. Rotation matrix $\mathbf{R}_y$ (2) can be reparameterized as

$$\mathbf{R}_y = \frac{1}{1+y^2} \begin{bmatrix} 1-y^2 & 0 & 2y \\ 0 & 1+y^2 & 0 \\ -2y & 0 & 1-y^2 \end{bmatrix}, \tag{4}$$

where $y = \tan\frac{\theta}{2}$. This parameterization introduces a degeneracy for a 180° rotation which can be ignored in real applications [29, 11]. The objective is to estimate the relative pose parameters θ and $\mathbf{t}$ globally optimally in the overdetermined case, *i.e.*, from $N \geq 4$ point correspondences.

3. Problem Formulation

Let us assume that we are given N point correspondences. By stacking the constraints (3) for N correspondences in a matrix, we obtain

$$\mathbf{A}^\top \mathbf{t} = 0, \tag{5}$$

where $\mathbf{A}$ is a $3 \times N$ matrix with its i^{th} column of form

$$\mathbf{A}_i = [\mathbf{p}_i']_\times \mathbf{R}_y \mathbf{p}_i. \tag{6}$$

The minimal case $N = 3$ results in $\det(A) = 0$ with up to 4 solutions, which has been well-studied in the literature [13, 37, 9, 49]. We mainly focus on the case where $N \geq 4$. In the case when the point correspondences are contaminated by noise, the objective is to find the least squares optimal solution of (5). The problem is formalized as follows:

$$\arg_{\mathbf{R}_y, \mathbf{t}} \ \min \mathbf{t}^\top \mathbf{C}\, \mathbf{t}, \tag{7}$$

where $\mathbf{C} = \mathbf{A}\mathbf{A}^\top$ is a 3×3 symmetric matrix

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}. \tag{8}$$

The elements of C are univariate polynomials in y. Eq. (7) is equivalent to minimizing the smallest eigenvalue of $\mathbf{C}$ as

$$\arg_{\mathbf{R}_y} \ \min \alpha_{\min}(\mathbf{C}). \tag{9}$$

Note, that (9) is a similar eigenvalue-based formulation to the one used in [26]. However, thanks to the special form of $\mathbf{R}_y$, it results in a simpler optimization problem in 1D space instead of the 3D space of full rotations. Therefore, it can be solved globally optimally by efficiently computing all its stationary points, compared to [26] where an iterative solution was proposed. The three eigenvalues of 3×3 matrix $\mathbf{C}$ should satisfy the following cubic constraint

$$\alpha^3 - f_1 \alpha^2 + f_2 \alpha - f_3 = 0, \tag{10}$$

where

$$\begin{aligned} f_1 &= \mathrm{trace}(\mathbf{C}), \\ f_2 &= c_{11}c_{22} + c_{11}c_{33} + c_{22}c_{33} - c_{12}^2 - c_{13}^2 - c_{23}^2, \\ f_3 &= \det(\mathbf{C}), \end{aligned} \tag{11}$$

are univariate rational functions in y (Eq. (11) can be derived by expanding $\det(\mathbf{C} - \alpha \mathbf{I}) = 0$). For the sake of simplicity, let us replace α_{min} with α. The necessary condition for minimizing the smallest eigenvalue of $\mathbf{C}$ is that derivative $\frac{\mathrm{d}\alpha}{\mathrm{d}y}$ should be zero. In (10), f_1, f_2, f_3 and α are functions of y. Thus, using the assumption $\frac{\mathrm{d}\alpha}{\mathrm{d}y} = 0$, the derivative of (10) gives us

$$\frac{\mathrm{d}f_1}{\mathrm{d}y}\alpha^2 - \frac{\mathrm{d}f_2}{\mathrm{d}y}\alpha + \frac{\mathrm{d}f_3}{\mathrm{d}y} = 0. \tag{12}$$

Rational function f_1 (11) can be rewritten as $f_1 = \frac{g_1}{\delta^2}$, where g_1 is a quartic polynomial in y and $\delta = 1 + y^2$. On the other hand, due to the inner associations in the elements of matrix $\mathbf{C}$, f_2 can be rewritten as $f_2 = \frac{g_2}{\delta^3}$, and f_3 can be written as $f_3 = \frac{g_3}{\delta^4}$, where g_2, g_3 are polynomials of degree 6 and 8, respectively. Hence, the derivatives of f_1, f_2, f_3 can be rewritten as

$$\frac{\mathrm{d}f_1}{\mathrm{d}y} = \frac{h_1}{\delta^3}, \ \frac{\mathrm{d}f_2}{\mathrm{d}y} = \frac{h_2}{\delta^4}, \ \frac{\mathrm{d}f_3}{\mathrm{d}y} = \frac{h_3}{\delta^5}, \tag{13}$$

where h_1, h_2, h_3 are polynomials in y of degree $\{4, 6, 8\}$, respectively. Substituting the formulas into (10) and (12), multiplying (10) by δ^4 and (10) by δ^5, and substituting $\beta = \delta\alpha$, we obtain the following two polynomial equations in two unknowns $\{\beta, y\}$.

$$\begin{aligned} \delta\beta^3 - \beta^2 g_1 + \beta g_2 - g_3 = 0, \\ \beta^2 h_1 - \beta h_2 + h_3 = 0. \end{aligned} \tag{14}$$

Our objective is to solve these equations efficiently.

4. Globally Optimal Solver

A straightforward way of solving the system of two polynomial equations in two unknowns (14) is to use the Gröbner basis method [8] and generate a specific solver using an automatic generator, *e.g.*, [27, 29, 30]. Using [29], we obtained a Gröbner basis solver with a template of size 17×45 for the Gauss-Jordan elimination and 28 solutions. However, our experiments on real-world and synthetic data showed that this solver becomes unstable when we are given more than 30 point correspondences, which is almost always the case in real applications. The reason is that the matrix for the Gauss-Jordan elimination is often ill-conditioned. Therefore, to find a stable solution, we use the hidden variable technique [8] instead.

By treating y as a hidden variable, *i.e.*, by considering y as a coefficient and hiding it into the coefficient matrix, we can rewrite our system (14) as

$$\begin{bmatrix} -g_3 & g_2 & -g_1 & \delta \\ h_3 & -h_2 & h_1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \beta^3 \end{bmatrix} = 0, \tag{15}$$

where $g_1, \ldots, g_3, h_1, \ldots, h_3$ and δ are polynomials in y. With this formulation we have two equations with 4 monomials. Therefore, to easily solve this system by "linearizing" it, we need to add additional equations to (15) to obtain as many equations as monomials. In our case, we only need to multiply the first equation by β, and the second equation by $\{\beta, \beta^2\}$. In this way, we obtain 5 equations with 5 monomials, which can be written as

$$\mathbf{B}(y)\boldsymbol{\omega} = \mathbf{0}, \tag{16}$$

where $\boldsymbol{\omega} = [1\ \beta\ \beta^2\ \beta^3\ \beta^4]^\top$ and $\mathbf{B}(y)$ is the 5×5 matrix

$$\mathbf{B}(y) = \begin{bmatrix} -g_3 & g_2 & -g_1 & \delta & 0 \\ h_3 & -h_2 & h_1 & 0 & 0 \\ 0 & -g_3 & g_2 & -g_1 & \delta \\ 0 & h_3 & -h_2 & h_1 & 0 \\ 0 & 0 & h_3 & -h_2 & h_1 \end{bmatrix}, \tag{17}$$

whose elements are polynomials in y. Next, we will present two solutions to (16).

Sturm Sequence Solution. Since the matrix $\mathbf{B}(y)$ has a right null vector, the determinant of $\mathbf{B}(y)$ must vanish. The sparse structure of $\mathbf{B}(y)$ allows us to use the Laplace expansion to obtain $\det(\mathbf{B}(y))$, which is a univariate polynomial in y of degree 28. Then, the Sturm sequence method is used to find the real roots of the obtained univariate polynomial, which is fast and efficient. For more details about the Sturm root bracketing, we refer the reader to [15].

Polynomial Eigenvalue Solution. Note, that (16) is essentially a polynomial eigenvalue problem (PEP) of degree 8, which can be written as

$$(y^8\mathbf{B}_8 + y^7\mathbf{B}_7 + ... + \mathbf{B}_0)\,\boldsymbol{\omega} = 0, \tag{18}$$

where $\mathbf{B}_8, \mathbf{B}_7, ..., \mathbf{B}_0$ are 5×5 coefficient matrices. PEP (18) can be transformed to an eigenvalue problem $\mathbf{D}\boldsymbol{\psi} = z\boldsymbol{\psi}$, with $z = 1/y$, $\boldsymbol{\psi} = [\boldsymbol{\omega}, z\boldsymbol{\omega}, \ldots, z^7\boldsymbol{\omega}]^\top$, and 40×40 matrix $\mathbf{D}$ of the form

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & ... & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & ... & \mathbf{0} \\ ... & ... & ... & \mathbf{I} \\ -\mathbf{B}_0^{-1}\mathbf{B}_8 & -\mathbf{B}_0^{-1}\mathbf{B}_7 & ... & -\mathbf{B}_0^{-1}\mathbf{B}_1 \end{bmatrix}. \tag{19}$$

Note, that this eigenvalue formulation is a relaxation of the original problem (15) that does not consider the monomial dependences in $\boldsymbol{\psi}$, and therefore, it introduces some spurious solutions. Six of these spurious solutions can be, however, easily removed. These solutions correspond to zero eigenvalues of matrix $\mathbf{D}$ that are generated by zero columns in matrices $\mathbf{B}_2$ and $\mathbf{B}_1$. After removing these columns and corresponding rows, the size of $\mathbf{D}$ reduces to 34×34. The real eigenvalues of $\mathbf{D}$, which can be computed from real Schur decomposition, are the solutions to $z = 1/y$. Note, that we don't need to compute the eigenvectors of the 40×40 matrix $\mathbf{D}$ since α_{min} (9) and the translation vector can be extracted from the eigenvalues and eigenvectors of the 3×3 matrix $\mathbf{C}$. This polynomial eigevalue solution is slower than the solution based on Sturm sequences, however, as we will show in experiments, it is more stable.

5. Linearized Solver

In visual odometry and SLAM applications, the relative rotation between consecutive frames is often small or negligible. Therefore, the first-order Taylor expansion usually leads to a reasonably good approximation for rotation. In this case, the rotation matrix can be simplified as

$$\mathbf{R}_y = \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & 0 \\ -\theta & 0 & 1 \end{bmatrix}, \tag{20}$$

and the elements of matrix $\mathbf{C}$ (8) are quadratic polynomials in θ. Similar to (14), we may obtain two polynomials with respect to $\{\theta, \alpha\}$ as follows:

$$\begin{aligned} \alpha^3 - f_1\alpha^2 + f_2\alpha - f_3 = 0, \\ g_1\alpha^2 - g_2\alpha + g_3 = 0, \end{aligned} \tag{21}$$

where f_1, f_2, f_3 are polynomials in θ of degree $\{2, 4, 6\}$, and $g_1 = \frac{\mathrm{d}f_1}{\mathrm{d}\theta}, g_2 = \frac{\mathrm{d}f_2}{\mathrm{d}\theta}, g_3 = \frac{\mathrm{d}f_3}{\mathrm{d}\theta}$ are polynomials in θ of degree $\{1, 3, 5\}$, respectively. This formulation is simpler than (14), since the polynomials have lower degrees. This system has up to 15 real solutions and the Gröbner basis solver generated using [29] has a template matrix of size 15×30. This solver is, however, unstable. Therefore, we use the Sturm sequence and the polynomial eigenvalue technique [28] as for the non-linearized solver to generate a efficient and stable solvers for (20). The main steps are the same as in Section 4. The only difference is that we only need to find the roots of a univariate polynomial in degree 15 or eigenvalues of a 21×21 matrix.

6. Experiments

We studied the performance of the proposed PEP-based optimal (OPT) and linearized (LIN) solvers and their variants based on Sturm sequences (OPT(S)) and (LIN(S)) on synthetic and real-world images. In the comparison, we included the closely related work of Ding et al. (3PC) [9] and the globally optimal SDP solution (SDP) [54]. In addition, we compared the proposed solver with the 5PC algorithm [46] and the normalized 8PC solver [17] with and without a final Levenberg-Marquardt [33] numerical optimization (8PC+LM). We did not include 3PC+LM and 5PC+LM since their performance was similar to 8PC+LM.

6.1. Synthetic Evaluation

We chose the following setup to generate synthetic data. First, 200 random 3D points and 200 image pairs with random poses were generated. The focal length f_g of the camera was set to 1000 pixels. The parameters which were changed to test the performance are the noise level (σ) in the image point locations, the baseline ($\mathbf{t}$) between two cameras, the number of points (N) used for the solvers, the field of view (FOV) of the camera, and the noise level τ in the gravity direction. The default setting: σ = 1 pixel, $\mathbf{t}$ = 5% of the average scene depth, N = 20, FOV = 90°, and τ = 0°. It is reasonable to assume that we have an

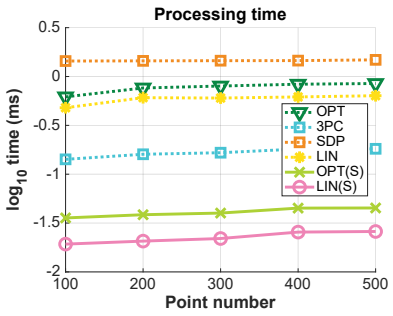


Figure 2: The $\log_{10}$ processing time in milliseconds plotted as the function of the point number used for the estimation.

almost perfect measurement of the gravity direction, since we obtain the gravity vector by applying a low-pass filter to isolate the force of gravity from the raw accelerometer data. The performance of the solvers was tested by modifying the value of a single parameter from the aforementioned ones. The rotation error was defined as the angle difference between the estimated rotation and the ground truth rotation as $\arccos\left(\left(\mathrm{tr}(\mathbf{R}_g\mathbf{R}_e^\top)-1\right)/2\right)$, where $\mathbf{R}_g$ and $\mathbf{R}_e$ are the ground truth and estimated rotations, respectively. The translation error was measured as the angle between the estimated and ground truth translation vectors, since the estimated translation is recovered only up to scale.

Solver accuracy. Fig. 3 and Fig. 4 report the rotation (top row) and translation (bottom) errors under different configurations. From Fig. 3, we can see that the proposed optimal solver (OPT) outperforms all the existing methods in the case when the gravity direction is perfect. However, in real applications, the gravity vector isolated from the raw accelerometer data may not be perfect, and we also studied the performance under increased gravity direction noise. Since accelerometers used in cars and modern smartphones have noise levels around $0.06°$ (or an expensive "good" accelerometers have less than $0.02°$), we added noise to roll and pitch angles for both views with a maximum value of $0.2°$. Fig. 4 shows that our OPT solver is still comparable to the SOTA even when the noise level is up to $0.2°$. Due to the lack of space and for batter readability of graphs we do not include results of OPT(S), LIN and LIN(S) solvers here. The results of these solvers, including the results for increasing relative rotation, forward motion, and stability experiments are in the supplementary material.

Complexity analysis and running times. The processing times of the solvers are shown in Fig. 2. The proposed Sturm sequences-based optimal OPT(S) and linearized LIN(S) ones are two orders of magnitude faster than the SDP optimal solver of [54]. Also, they are an order of magnitude faster than the minimal 3PC solver [9]. The following table contains the main operations performed by the proposed solvers, together with their running times in μs. The second column shows the size of the matrix for the Gauss-Jordan elimination, the third column shows the size of the matrix for the eigenvalue decomposition, and the fourth one shows the degree of the univariate polynomial solved by Sturm sequences.

Solver	G-J	Eigen	Sturm	Time (μs)
OPT	5×34	34×34	-	115
OPT(S)	-	-	28	24
LIN	5×21	21×21	-	54
LIN(S)	-	-	15	13

6.2. Real-world Experiments

In order to test the proposed technique on real-world data, we chose the `Malaga` [4][1], `KITTI` [14][2] and `ETH3D` [45][3] datasets. `Malaga` was gathered entirely in urban scenarios with car-mounted sensors, including one high-resolution stereo camera and five laser scanners. We used the sequences of one high-resolution camera and every 10th frame from each sequence. The proposed solvers were applied to every consecutive image pair. The ground truth paths were composed using the GPS coordinates provided in the dataset. In total, $9,064$ image pairs were used from the dataset. The `KITTI` odometry benchmark consists of 22 stereo sequences. Only 11 sequences (00–10) are provided with ground truth trajectories for training. We therefore used these 11 sequences to evaluate the compared solvers. In total, 23190 image pairs were used. The `ETH3D` dataset covers a variety of indoor and outdoor scenes. Ground truth geometry has been obtained using a high-precision laser scanner. A DSLR camera as well as a synchronized multi-camera rig with varying field-of-view was used to capture images. In total, we used 5162 image pairs.

For testing non-minimal solvers on real-world data, we chose a locally optimized RANSAC, *i.e.*, Graph-Cut RANSAC[4] [3] (GC-RANSAC). In GC-RANSAC (and other locally optimized RANSACs), two different solvers are used: (a) one for estimating the pose from a minimal sample and (b) one for fitting to a larger-than-minimal sample when doing final pose polishing on all inliers or in the local optimization step. For (a), the main objective is to solve the problem using as few correspondences as possible since the processing time depends exponentially on the number of correspondences required for the pose estimation. We tested two minimal solvers, the 5PC algorithm of Stewenius et al. [47] and the 3PC method of Ding et al.

[1] https://www.mrpt.org/MalagaUrbanDataset
[2] http://www.cvlibs.net/datasets/kitti
[3] https://www.eth3d.net/
[4] https://github.com/danini/graph-cut-ransac

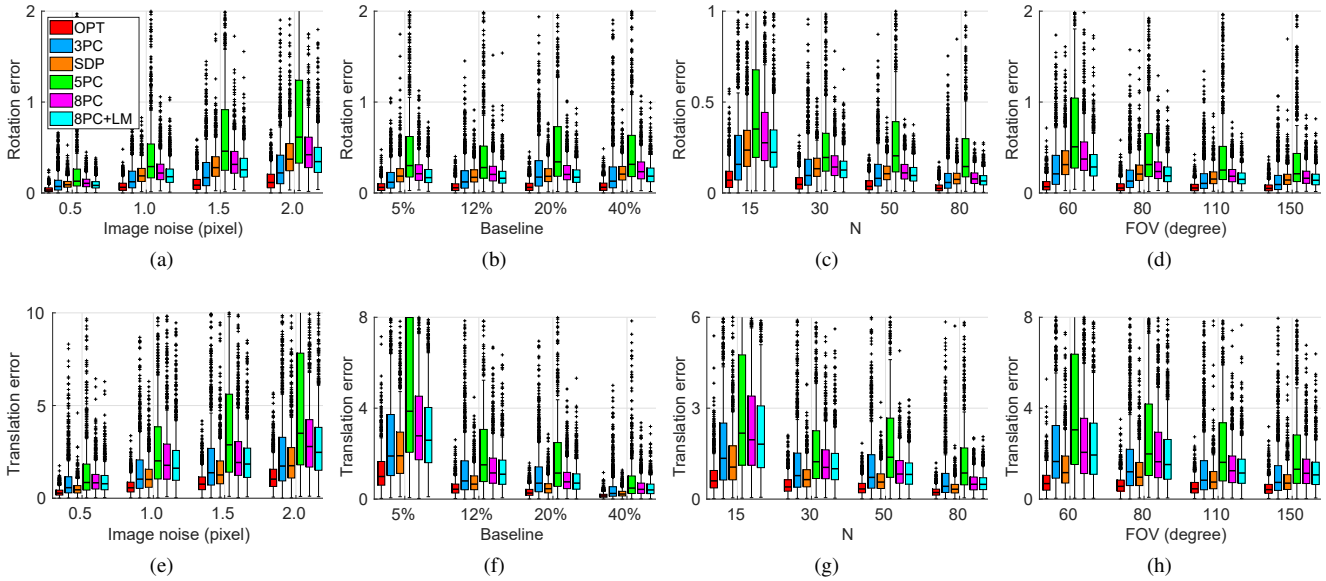


Figure 3: **Top**: rotation error in degrees. **Bottom**: translation error in degrees. The columns show the error of the solvers against increasing (a) image noise, (b) baseline, (c) correspondence number, and (d) field-of-view. We use the following default values for the parameters not tested in a figure: std. of the image noise is $\sigma = 1$ px; length of the baseline = 10% scene depth; number of correspondences N = 20; field of view FOV = 90°, the std. of the gravity directional noise $\tau = 0°$.

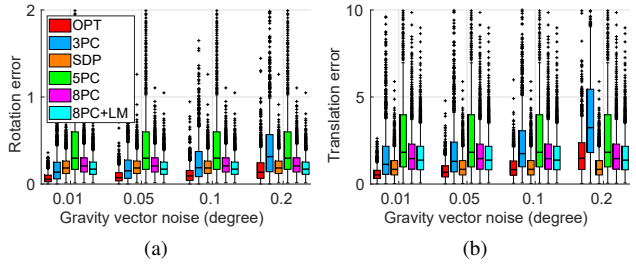


Figure 4: Rotation and translation errors in degrees. Default settings with increased roll and pitch noise.

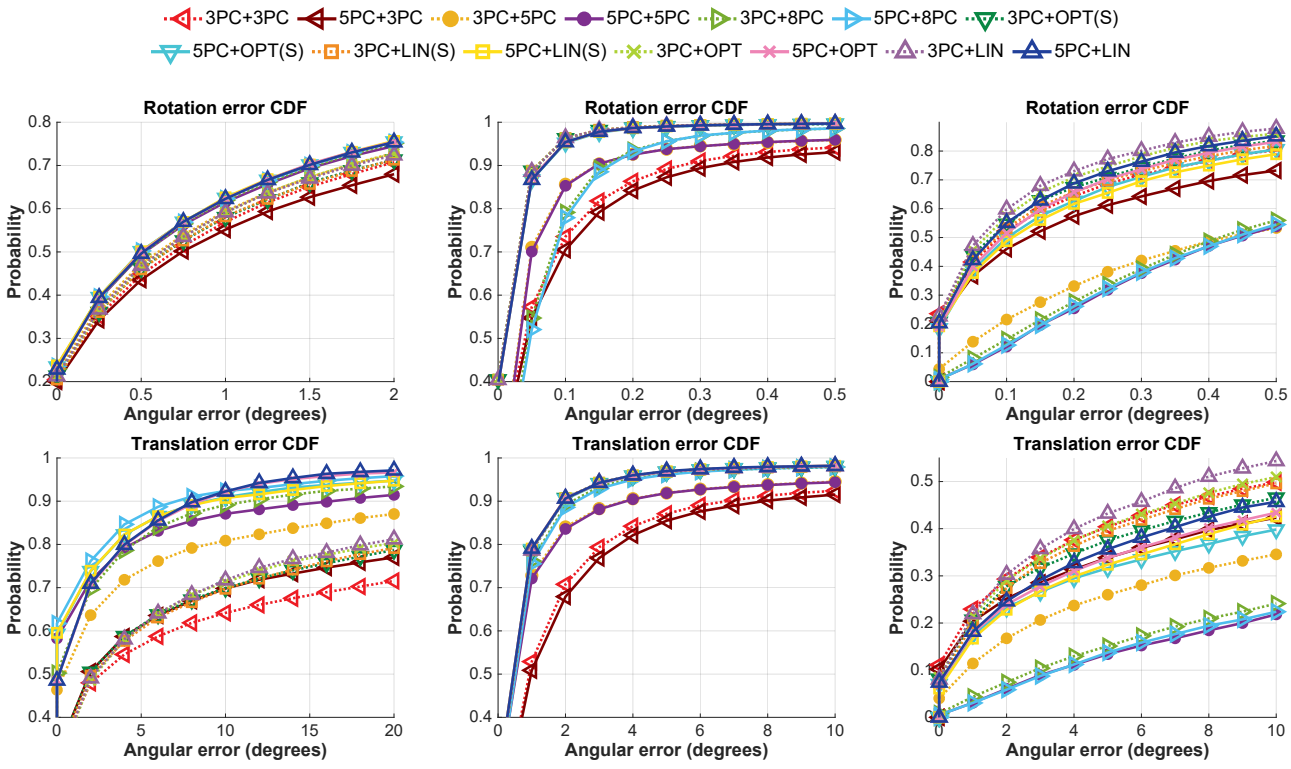
[9] exploiting the gravity direction similarly to our method. The purpose of (b) is to estimate the pose parameters as accurately as possible. In (b), we tested the proposed solvers, the normalized 8PC algorithm [17], the 5PC solver [47], and the 3PC method of Ding et al. [9]. We excluded [54] from the real-world tests since it is extremely slow, even when implemented in C++, as reported in Fig. 2.

The cumulative distribution functions (CDF) of the rotation and translation errors (in degrees) on the three tested datasets are shown in Fig. 5. Being accurate is interpreted as a curve close to the top-left corner. The proposed OPT, LIN, OPT(S), and LIN(S) solvers are always among the top-performing methods. Interestingly, the LIN solver leads to the most accurate rotations and translations on these scenes. Since these datasets contain image pairs with relatively small baseline and, thus, small pose change between the frames, it is not a surprise that LIN is accurate. Moreover, since the OPT solver finds the optimum of an algebraic error, which might not always coincide with the minimum of the geometric one, it can happen that LIN solver leads to better pose estimates than the OPT solver. The average errors are reported in Table 1. All of the proposed solvers lead to more accurate results than the other compared methods. The most accurate results are obtained by the combination of 5PC and LIN.

6.3. Phone Dataset

To further illustrate the usefulness of the proposed solvers, we tested them on images from a phone where the gravity directions are obtained from the built-in IMU. Since we have not found such datasets, we decided to build one using images captured by a smartphone (iPhone 6s). We captured 6 sequences at @30Hz with the rear camera. The corresponding IMU data were captured at @100Hz with the phone's sensor (costs less than a dollar). We then synchronized the images and IMU data based on their timestamps. In order to obtain a ground truth, which can be used to test pose solvers, we applied the RealityCapture [1] software. A total of 10993 image pairs with synchronized gravity direction, ground truth poses, calibrations and 3D reconstructions were obtained. Example images are shown in Fig. 6.

The CDFs of the compared solvers are shown in the left



(a) Malaga (b) KITTI (c) ETH3D

Figure 5: The cumulative distribution functions of the rotation and translation errors (in degrees) of GC-RANSAC combined by different minimal and non-minimal solvers on datasets Malaga (9,064 image pairs), KITTI (23,190) and ETH3D (5,162). Being accurate is interpreted as a curve close to the top-left corner.

two plots of Fig. 7a. The same trend can be seen as before, *i.e.*, the proposed optimal and linearized solvers lead to the most accurate rotations and translations. However, OPT and LIN leads to similar accuracy on this dataset. This is due to having slightly bigger view changes than in the datasets designed for autonomous driving. The bigger view changes increase the approximative nature of solver LIN and, thus, make its results marginally less accurate. Still, it is one of the top-performing methods in terms of accuracy.

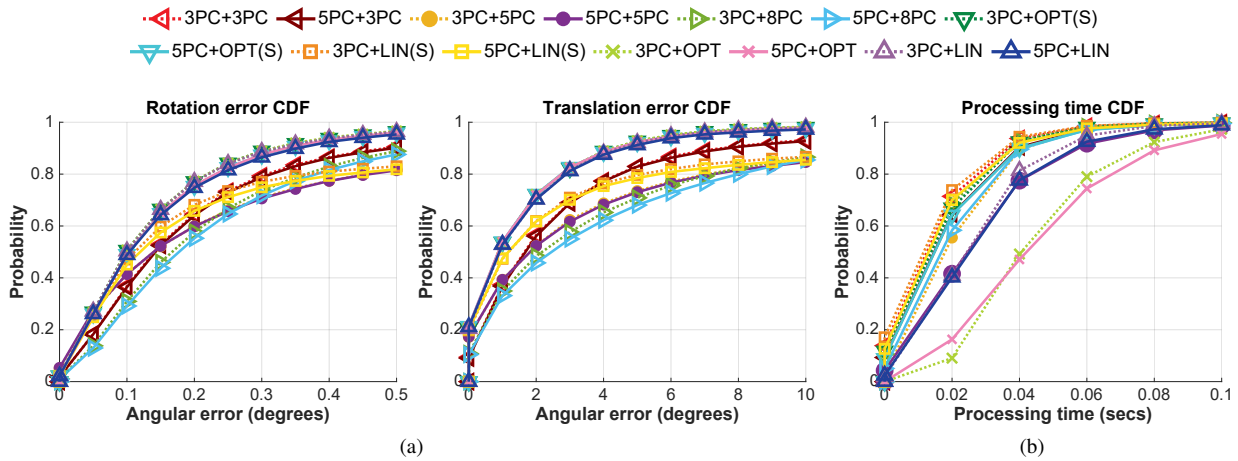
Figure 6: Images from the captured phone dataset. It consists a total of 10993 image pairs with synchronized gravity directions, ground truth poses and 3D reconstructions.

6.4. Processing Times

The CDFs of the processing times of the robust estimation on all the 47863 image pairs from all datasets are in Fig. 7b. The proposed solvers based on Sturm sequences are the fastest ones. The optimal solver (OPT), compared to the other ones are the slowest method, while the linearized (LIN) solver has comparable speed to the other methods. It is important to mention that even the slowest combination (*i.e.*, 5PC+OPT) has an average run-time of 61 ms. Therefore, all tested combinations lead to *real-time* performance.

7. Extensions and Discussion

General case. The proposed technique can be, in theory, applied to the general case (without gravity prior). In this case, for the Cayley parameterization of the rotation, the elements of the matrix $\mathbf{C}$ are polynomials in x, y, z. The three eigenvalues of $\mathbf{C}$ have to satisfy (10), but now there are three partial derivatives $\frac{\partial \alpha}{\partial x}$, $\frac{\partial \alpha}{\partial y}$, $\frac{\partial \alpha}{\partial z}$ that should be zero. This gives four polynomials with respect to four unknowns $\{\alpha, x, y, z\}$. Using Macaulay2 [16], we found that this system of polynomials has more than 2,000 solutions. A solver to such a system is impractical since the eigenvalue decomposition of a matrix of size $2,000 \times 2,000$ is extremely slow. However, it means that there are more than 2,000 stationary points. Hence, locally optimized methods (*e.g.*, [26] or



(a) (b)

Figure 7: **(a)** The cumulative distribution functions (CDF) of the rotation and translation errors (in degrees) of GC-RANSAC combined by different minimal and non-minimal solvers on the captured phone dataset consisting of 10993 image pairs. Being accurate or fast is interpreted as a curve close to the top-left corner. **(b)** The CDF of the processing times (in seconds) of the robust estimation on all datasets (47, 863 image pairs).

		Malaga		KITTI		ETH3D		avg
m	$> m$	$\epsilon_\mathbf{R}$	$\epsilon_\mathbf{t}$	$\epsilon_\mathbf{R}$	$\epsilon_\mathbf{t}$	$\epsilon_\mathbf{R}$	$\epsilon_\mathbf{t}$	
3PC	OPT	2.61	9.86	**0.04**	1.97	**0.33**	**23.94**	6.46
3PC	OPT(S)	3.02	10.17	**0.04**	1.99	0.38	27.45	7.18
3PC	LIN	2.61	9.58	**0.04**	1.97	**0.31**	**22.31**	**6.14**
3PC	LIN(S)	2.99	10.17	**0.04**	1.96	0.40	25.59	6.86
3PC	3PC	2.99	12.39	1.00	4.75	0.70	24.99	7.80
3PC	5PC	2.89	6.77	0.25	3.22	1.82	34.60	8.26
3PC	8PC	**2.43**	4.46	0.10	2.11	1.14	37.80	8.01
5PC	OPT	**2.41**	3.59	**0.05**	1.90	0.41	28.71	6.18
5PC	OPT(S)	2.49	3.73	**0.05**	1.91	0.44	31.11	6.62
5PC	LIN	**2.41**	**3.49**	**0.05**	**1.89**	0.37	26.86	**5.85**
5PC	LIN(S)	2.47	3.74	**0.04**	**1.88**	0.46	29.32	6.32
5PC	3PC	3.69	10.44	1.13	5.18	0.94	29.39	8.46
5PC	5PC	2.72	4.79	0.27	3.30	2.15	36.74	8.33
5PC	8PC	2.49	**3.23**	0.11	2.11	1.18	39.46	8.10

Table 1: The avg. rotation and translation errors (in degrees) of GC-RANSAC [3] combined with different pose solvers are reported on datasets `Malaga`, `KITTI`, `ETH3D` (in total, 47, 863 image pairs). Columns 1–2 show the solvers used for minimal (m) and non-minimal ($> m$) fitting: the proposed optimal solvers OPT/OPT(S), LIN/LIN(S), 3PC [9], 5PC [47], and 8PC [17]. The best results are shown in red, the second bests are in blue. The cumulative distribution functions of the errors are shown in Fig. 5.

8PC+LM) may get stucked in these minima.

Planar motion. In case of planar camera motion, which is usual scenario for unmanned ground vehicles with rigidly mounted cameras, we have constraint $t_y = 0$. Therefore, the middle row and column of the matrix $\mathbf{C}$ in (9) can be removed. Matrix $\mathbf{C}$ becomes a 2×2 matrix, where the two eigenvalues should satisfy quadratic constraint $\alpha^2 - f_1\alpha + f_2 = 0$, where $f_1 = \text{trace}(\mathbf{C})$ and $f_2 = \det(\mathbf{C})$. The remaining steps are similar to Sec. 4, and there are up to 8 solutions. In this case, the polynomial eigenvalue solution needs to find the eigenvalues of a 10×10 matrix.

Forward motion. In our experiments, we observed that the globally optimal solvers [54, 5] estimate inaccurate poses in the case of forward motion. For this motion, the proposed globally optimal solvers return the best results. Due to the lack of space we included the experiments for forward motion in the supplementary material.

8. Conclusions

We propose globally optimal solutions for estimating the relative pose of two cameras, aligned by the gravity direction, from a larger-than-minimal set of point correspondences. All of the proposed solvers, even the linearized ones, lead to results *superior* to the state-of-the-art in terms of accuracy on approx. 50k image pairs from four widely-used datasets. The techniques based on linearization or Sturm sequences lead to extremely fast estimation with an average of 22 ms run-time (*i.e.*, total RANSAC time) with no or negligible deterioration in the accuracy. Moreover, we captured a new dataset consisting of more than 10k photos taken by a smartphone with a gravity sensor.

Acknowledgments. This research was supported by the National Science Fund of China (Grant No. U1713208), the "111" Program B13022, the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics", the ERC-CZ grant MSMT LL1901, and by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program.

References

[1] RealityCapture. http://www.capturingreality.com. 6

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 2011. 1

[3] Daniel Barath and Jiří Matas. Graph-cut RANSAC. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 5, 8

[4] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. G. Jiménez. The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *International Journal of Robotics Research (IJRR)*, 2014. 5

[5] Jesus Briales, Laurent Kneip, and Javier Gonzalez-Jimenez. A certifiably globally optimal solution to the non-minimal relative pose problem. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8

[6] Graziano Chesi. Camera displacement via constrained minimization of the algebraic error. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2008. 2

[7] Ondej Chum, Jií Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, 2003. 1

[8] David A. Cox, John Little, and Donal O'shea. *Using algebraic geometry*. Springer Science & Business Media, 2006. 3

[9] Yaqing Ding, Jian Yang, and Hui Kong. An efficient solution to the relative pose estimation with a common direction. In *2020 IEEE International Conference on Robotics and Automation*, 2020. 1, 3, 4, 5, 6, 8

[10] Yaqing Ding, Jian Yang, Jean Ponce, and Hui Kong. An efficient solution to the homography-based relative pose problem with a common reference direction. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[11] Yaqing Ding, Jian Yang, Jean Ponce, and Hui Kong. Minimal solutions to relative pose estimation from two views sharing a common direction with unknown focal length. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1

[13] Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *European Conference on Computer Vision (ECCV)*, 2010. 1, 3

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[15] Walter Gellert, M. Hellwich, H Kästner, and H Küstner. *The VNR concise encyclopedia of mathematics*. Springer Science & Business Media, 2012. 4

[16] Daniel R. Grayson and Michael E. Stillman. Macaulay 2, a software system for research in algebraic geometry, 2002. 7

[17] Richard Hartley. In defence of the 8-point algorithm. In *International Conference on Computer Vision (ICCV)*, 1995. 2, 4, 6, 8

[18] Richard Hartley. Minimizing algebraic error in geometric estimation problems. In *International Conference on Computer Vision (ICCV)*, 1998. 2

[19] Richard Hartley and Fredrik Kahl. Global optimization through searching rotation space and optimal estimation of the essential matrix. In *International Conference on Computer Vision (ICCV)*, 2007. 2

[20] Richard Hartley and Fredrik Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 2009. 2

[21] Richard Hartley and Hongdong Li. An efficient hidden variable approach to minimal-case camera motion estimation. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2012. 1

[22] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[23] Uwe Helmke, Knut Hüper, Pei Yean Lee, and John Moore. Essential matrix estimation using gauss-newton iterations on a manifold. *International Journal of Computer Vision*, 2007. 2

[24] Joel A. Hesch and Stergios I. Roumeliotis. A direct least-squares (dls) method for pnp. In *International Conference on Computer Vision (ICCV)*, 2011. 1

[25] Laurent Kneip, Hongdong Li, and Yongduek Seo. Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability. In *European Conference on Computer Vision (ECCV)*, 2014. 1

[26] Laurent Kneip and Simon Lynen. Direct optimization of frame-to-frame rotation. In *International Conference on Computer Vision (ICCV)*, 2013. 2, 3, 7

[27] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *European Conference on Computer Vision (ECCV)*, 2008. 3

[28] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to minimal problems in computer vision. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2012. 1, 4

[29] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 4

[30] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla. Beyond gröbner bases: Basis selection for minimal solvers. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[31] Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized RANSAC–full experimental evaluation. In *British Machine Vision Conference (BMVC)*, 2012. 1

[32] Yi Ma, Jana Košecká, and Shankar Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 2001. 2

[33] Jorge J. Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*. 1978. 4

[34] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 2017. 1

[35] Gaku Nakano. Globally optimal dls method for pnp problem with cayley parameterization. In *British Machine Vision Conference (BMVC)*, 2015. 1

[36] Gaku Nakano. A versatile approach for solving pnp, pnpf, and pnpfr problems. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[37] Oleg Naroditsky, Xun S. Zhou, Jean Gallier, Stergios I. Roumeliotis, and Kostas Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2012. 1, 3

[38] David Nistér. An efficient solution to the five-point relative pose problem. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2004. 1

[39] Marc Pollefeys, David Nistér, J-M Frahm, Amir Akbarzadeh, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, Paul Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision (IJCV)*, 2008. 1

[40] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision (IJCV)*, 2004. 1

[41] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: a universal framework for random sample consensus. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2013. 1

[42] Carsten Rother. *Multi-view reconstruction and camera recovery using a real or virtual reference plane*. PhD thesis, Numerisk analys och datalogi, 2003. 1

[43] Olivier Saurer, Pascal Vasseur, Rémi Boutteau, Cédric Demonceaux, Marc Pollefeys, and Friedrich Fraundorfer. Homography based egomotion estimation with a common direction. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 1

[44] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[45] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[46] H. Stewenius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *Journal of Photogrammetry and Remote Sensing*, 2006. 4

[47] Henrik Stewénius, David Nistér, Fredrik Kahl, and Frederik Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *Computer Vision and Pattern Recognition (CVPR)*, 2005. 5, 6, 8

[48] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2016. 1

[49] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. *International Conference on 3D Vision (3DV)*, 2014. 1, 3

[50] George Terzakis and Manolis Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[51] Ping Wang, Guili Xu, Yuehua Cheng, and Qida Yu. A simple, robust and fast method for the perspective-n-point problem. *Pattern Recognition Letters*, 2018. 1

[52] Changchang Wu. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision (3DV)*, 2013. 1

[53] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[54] Ji Zhao. An efficient solution to non-minimal case essential matrix estimation. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2020. 2, 4, 5, 6, 8

[55] Ji Zhao, Wanting Xu, and Laurent Kneip. A certifiably globally optimal solution to generalized essential matrix estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[56] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Astrom, and Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *International Conference on Computer Vision (ICCV)*, 2013. 1

[57] Lipu Zhou and Michael Kaess. An efficient and accurate algorithm for the perspecitve-n-point problem. In *International Conference on Intelligent Robots and Systems (IROS)*, 2019. 1