Relative Pose from a Calibrated and an Uncalibrated Smartphone Image

Yaqing Ding[1,4], Daniel Barath[2], Jian Yang[1], Zuzana Kukelova[3]
[1] School of Computer Science and Engineering, Nanjing University of Science and Technology
[2] Computer Vision and Geometry Group, Department of Computer Science, ETH Zürich
[3] Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
[4] Centre for Mathematical Sciences, Lund University
dingyaqing@njust.edu.cn

Abstract

In this paper, we propose a new minimal and a non-minimal solver for estimating the relative camera pose together with the unknown focal length of the second camera. This configuration has a number of practical benefits, e.g., when processing large-scale datasets. Moreover, it is resistant to the typical degenerate cases of the traditional six-point algorithm. The minimal solver requires four point correspondences and exploits the gravity direction that the built-in IMU of recent smart devices recover. We also propose a linear solver that enables estimating the pose from a larger-than-minimal sample extremely efficiently which then can be improved by, e.g., bundle adjustment. The methods are tested on 35654 image pairs from publicly available real-world and new datasets. When combined with a recent robust estimator, they lead to results superior to the traditional solvers in terms of rotation, translation and focal length accuracy, while being notably faster.

1. Introduction

Estimating the relative pose of two cameras using a, typically, minimal set of point correspondences is a classical computer vision problem [23]. It has a number of applications, including pose-graph initialization for global [5, 36, 50, 52] and incremental [44, 45, 53] Structure-from-Motion, Simultaneous Localization and Mapping algorithms [37, 38], augmented and virtual reality [33], multi-motion fitting in videos [51], and surveillance [34, 35]. Nowadays, with the popularity of smartphones equipped with various sensors, new possibilities arise for pose estimation exploiting the additional information provided by other built-in sensors, *e.g.*, Inertial Measurement Unit (IMU). In this paper, we focus on exploiting the gravity direction recovered by an IMU in the case when one of the cameras is fully calibrated, while the focal length of the other one is unknown.

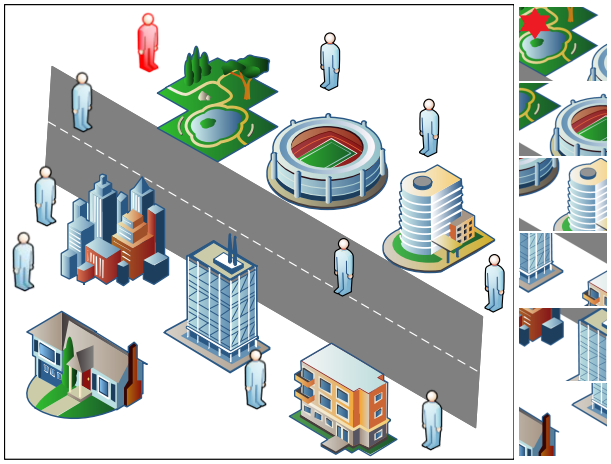

Figure 1. Almost everyone has a smartphone equipped with a camera and an IMU sensor. While the y-axes of cameras can be usually aligned using the gravity direction extracted from the IMU, the internal calibration of some cameras may be corrupted or not available. Here, by using as little as one calibrated camera (the person in red), we can estimate the relative poses and focal lengths of the remaining cameras.

Depending on the camera configuration, there have been a number of solutions proposed over the years. Assuming that both cameras are calibrated and, thus, the intrinsic matrices are known, one can estimate the relative pose from five correspondences [22, 27, 32, 40, 46]. This setting is used in many applications and is generally regarded as a solved problem with efficient stable solutions and only a few degeneracies. When we do not have access to an accurate calibration, *jpeg-exif* headers often contain useful information, *e.g.*, the focal length. This header is, however, sometimes corrupted, *e.g.*, for images from the Internet, or the included focal length is incorrect, *e.g.*, due to image resizing.

In the case when we are not given an accurate calibration a priori, it is safe, in practice, to assume that the pixels

are square-shaped and the principal point coincides with the image center. This implies that the only unknowns to be estimated are the focal lengths of the two images. When we assume that both cameras have a common unknown focal length, six correspondences are enough to solve the problem [22, 27, 28, 47]. The 6-point solver is, however, rarely used in practice due to problems with degeneracies, *e.g.*, when the optical axes of the cameras are parallel or intersecting. If the focal lengths of the two cameras are different and unknown, at least seven correspondences are needed to recover the relative pose and focal lengths [8, 21].

Another practically interesting case appears when one of the cameras is fully calibrated while the focal length of the other one is unknown. This problem requires six point correspondences and was solved by the Gröbner-basis method in [9, 28]. Such a situation often happens when processing large-scale datasets where some of the images either have their focal lengths available in the *exif* tag or the sensor type is known. For example, state-of-the-art structure from motion algorithms [44, 50], after recognizing the sensor type used for imaging, read the associated focal length from a database if it is available. While clearly having practical benefits when processing large-scale datasets, this configuration is also resistant to the typical degeneracies of the 6-point algorithm [46]. It works even if the optical axes of the cameras are parallel or intersecting.

Recent devices usually are equipped with an IMU sensor that measures the gravity direction accurately. Exploiting this gravity prior, the vertical axes of the cameras can be aligned, reducing their relative orientation to 1 degree of freedom (DOF). This prior not only simplifies the geometry and polynomial systems that have to be solved but, also, reduces that number of correspondences needed for the estimation. This is extremely important since the run-time of RANSAC-like robust estimation depends *exponentially* on the sample size. The gravity prior was used to simplify minimal relative pose [12,15,16,31,39,43,49] including the relative pose problem with two unknown equal or different focal lengths [14], absolute pose [2, 26, 48], and general radial distortion homography solvers [11, 41].

In this paper, we fill the gap in existing relative pose solvers. We propose solvers exploiting the gravity direction for two practically interesting and previously unsolved settings. First, we propose several different minimal solvers for estimating the relative pose together with the unknown focal length of the second camera from a minimum of four point correspondences [1]. The proposed solvers are based on the state-of-the-art algebraic methods for generating efficient polynomial solvers [7,22,27,30], from which the hidden variable method provides the solver with the best trade-off between numerical stability and efficiency. When deriving these solutions, we provide an additional analysis of the Cayley parametrization used in these solvers. This allows us simplifying both the problem equations and the solvers. Second, we propose a solver for estimating the unknown parameters from a larger-than-minimal sample. This step is extremely important for state-of-the-arts RANSACs [25,42] where the accuracy is ensured by local optimization and final model polishing steps running non-minimal solvers.

[1]The solution without the gravity [9] requires six correspondences.

2. Problem Statement

Let us assume two cameras observing 3D points $\{\mathbf{X}_i\}$. Let $\mathbf{m}_i = [u_i, v_i, 1]^\top$ and $\mathbf{m}'_i = [u'_i, v'_i, 1]^\top$ be the homogeneous coordinates of the projections of the point $\mathbf{X}_i$ into the first and the second camera. The corresponding image points $\mathbf{m}_i$ and $\mathbf{m}'_i$ are related as

$$\lambda'_i \mathbf{K}_2^{-1} \mathbf{m}'_i = \lambda_i \mathbf{R} \mathbf{K}_1^{-1} \mathbf{m}_i + \mathbf{t}, \tag{1}$$

where $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ is the unknown relative rotation and translation between the two cameras, λ_i, λ'_i are the depths of the image points $\mathbf{m}_i, \mathbf{m}'_i$, and $\mathbf{K}_1, \mathbf{K}_2$ are the intrinsic matrices of the first and second camera, respectively.

In this paper, we assume that the two cameras have a common reference direction. This is a natural assumption, which occurs in many image capturing scenarios where we can extract the common reference direction from the built-in IMUs of smartphones and tablets. Without loss of generality, let us assume that the y-axes of the two cameras are aligned using roll and pitch angles calculated from the common reference direction. For this alignment, IMU-to-camera calibration is required. However, as it was shown in previous papers [12, 20] it is usually sufficient to assume that this calibration is known. Due to the way of how modern smart devices are constructed, the angle between the axes of the camera and the IMU is usually either $0°$, $\pm 90°$, or $180°$ and, therefore, it can be considered as known.

Let the rotation matrices used for the alignment of the y-axes of the two cameras be $\mathbf{R}_{align}$ and $\mathbf{R}'_{align}$. After the alignment the equation, (1) can be rewritten as

$$\lambda'_i \mathbf{R}'_{align} \mathbf{K}_2^{-1} \mathbf{m}'_i = \lambda_i \mathbf{R}_y \mathbf{R}_{align} \mathbf{K}_1^{-1} \mathbf{m}_i + \boldsymbol{\tau}, \tag{2}$$

where $\mathbf{R}_y$ is the rotation from the yaw angle (around axis y), and $\boldsymbol{\tau} = \mathbf{R}'_{align}\mathbf{t}$ is the translation after the alignment.

A common assumptions for modern cameras with CCD and CMOS sensors are square-shaped pixels, and the principal point coincident with the image center [22]. With this assumption the intrinsic calibration matrix is a diagonal matrix with the focal length as the only unknown parameter.

We assume that the focal length of the first camera is known. Therefore the equation (2) can be written as

$$\lambda'_i \mathbf{R}'_{align} \mathbf{K}_2^{-1} \mathbf{m}'_i = \lambda_i \mathbf{R}_y \mathbf{p}_i + \boldsymbol{\tau}, \tag{3}$$

where $\mathbf{p}_i = \mathbf{R}_{align} \mathbf{K}_1^{-1} \mathbf{m}_i$ are the known homogeneous coordinates of the calibrated image point in the first camera

after the alignment, and $\mathbf{K}_2^{-1} = \mathrm{diag}(1, 1, f)$, where f is the unknown focal length of the second camera.

The vectors $(\lambda_i' \mathbf{R}'_{align} \mathbf{K}_2^{-1} \mathbf{m}_i') \times (\lambda_i \mathbf{R}_y \mathbf{p}_i)$ are perpendicular to the translation vector $\boldsymbol{\tau}$. Therefore we can write

$$(\mathbf{R}'_{align}[u_i', v_i', f]^\top) \times (\mathbf{R}_y \mathbf{p}_i) \cdot \boldsymbol{\tau} = 0. \tag{4}$$

In this case, the depth parameters λ_i', λ_i are eliminated. Our objective is to estimate the unknown relative rotation $\mathbf{R}_y$, translation $\boldsymbol{\tau}$ and the focal length f of the second camera using equations (4).

3. Minimal 4-Point Solver

Each point correspondence $\mathbf{m}_i \leftrightarrow \mathbf{m}_i'$ gives a single constraint of the form (4). Since we have 4 DOFs (one for the unknown rotation parameter, two for the unknown translation that can be estimated only up to scale, and one for the focal length f of the second camera), we need at least four point correspondences to solve this problem. By stacking the equations for N point correspondences, constraint (4) can be written as

$$\mathbf{A}\boldsymbol{\tau} = 0, \tag{5}$$

where $\mathbf{A}$ is a $N \times 3$ polynomial matrix with the i^{th} row of $\mathbf{A}$ of the form

$$\mathbf{A}_{(i,:)} = (\mathbf{R}'_{align}[u_i', v_i', f]^\top) \times (\mathbf{R}_y \mathbf{p}_i). \tag{6}$$

The rotation matrix $\mathbf{R}_y$ can be parameterized using the Cayley parametrization as

$$\mathbf{R}_y = \frac{1}{1+\sigma^2} \begin{bmatrix} 1-\sigma^2 & 0 & 2\sigma \\ 0 & 1+\sigma^2 & 0 \\ -2\sigma & 0 & 1-\sigma^2 \end{bmatrix}, \tag{7}$$

where $\sigma = \tan \frac{\theta}{2}$, and θ is the rotation angle around the y-axes. The Cayley parameterization introduces a degeneracy for $180°$, however, it is frequently used in minimal solvers since it reduces number of unknowns. Moreover, this $180°$ degeneracy is not an issue in practice [29] and can be easily detected and filtered inside RANSAC.

Since (5) has a non-trivial solution, the matrix $\mathbf{A}$ must be rank-deficient. This means that determinants of all 3×3 submatrices of matrix $\mathbf{A}$ must vanish. Note, that since the equations (6) are homogeneous, we can omit the scale factor $\frac{1}{1+\sigma^2}$ in the parameterization (7).

Moreover, determinants of 3×3 submatrices of the matrix $\mathbf{A}$ have the following property:

Property 1. *Determinants of all 3×3 submatrices $\mathbf{A}_I$ of the matrix $\mathbf{A}$ can be written as $\det(\mathbf{A}_I) = (1+\sigma^2) h_I(\sigma, f)$, where $h_I(\sigma, f)$ are polynomials in $\{\sigma, f\}$.*

Here I is an index set and $\mathbf{A}_I$ is a submatrix of the matrix $\mathbf{A}$ that contains rows complementary to I. Property 1 holds thanks to the Cayley representation used to parametrize the rotation matrix $\mathbf{R}_y$. A similar property has been recognized in several papers [14, 49, 55], however, none of them provides an exact proof. In this paper, we provide a proof of Property 1 by proving a stronger statement:

Property 2: *Determinants of 2×2 submatrices $\mathcal{A}_{12}, \mathcal{A}_{22}, \mathcal{A}_{32}$ of the matrix $\mathcal{A} = \mathbf{A}_I$ have $1 + \sigma^2$ as a common factor.*

Using the Laplace expansion by the second column, the determinant of the matrix $\mathcal{A}$ can be expressed as $\det(\mathcal{A}) = a_{22} \det(\mathcal{A}_{22}) - a_{12} \det(\mathcal{A}_{12}) - a_{32} \det(\mathcal{A}_{32})$. Thus, by proving Property 2, we directly obtain a proof of Property 1. The proof is in the supplementary material. Note that, a proof for the general rotation case can be found at [54].

Thanks to Property 1, we can reduce the degrees of polynomials used for solving the problem. For the minimum number of four point correspondences, the 4×3 matrix $\mathbf{A}$ in (5) has $\binom{4}{3} = 4$ subdeterminants of size 3×3 that has to vanish. These four subdeterminants give us four polynomials of degree 6 (the highest degree term is $\sigma^4 f^2$) in two unknowns $\{\sigma, f\}$ as follows:

$$h_k(\sigma, f) = \det(\mathbf{A}_k)/(1+\sigma^2). \tag{8}$$

In this way, we eliminated the unknown translation τ from our equations. The four polynomial equations $h_k(\sigma, f) = 0, k = 1, \ldots, 4$ can be rewritten as

$$\mathbf{B}\mathbf{w} = 0, \tag{9}$$

where $\mathbf{B}$ is a 4×15 coefficient matrix and

$$\mathbf{w} = [1, f, f^2, \sigma, \sigma f, \sigma f^2, ..., \sigma^4 f^2]^\top, \tag{10}$$

is a vector consisting of the 15 monomials.

The system of polynomial equations in (9) can be solved using different algebraic methods [10]. In this paper, we tested different state-of-the-art approaches for generating efficient algebraic solvers [7, 22, 27, 30]. Next, we describe these solutions, starting with the hidden variable solution that provides the best trade-off between stability and efficiency.

3.1. Hidden Variable solution

The polynomial system in (9) contains four polynomials in two unknowns (σ, f), and the highest degree of the unknown f is 2. In this case, σ can be chosen as the hidden variable, *i.e.* we can consider it as a parameter. Then the system of polynomial equations (9) can be rewritten as

$$\mathbf{M}(\sigma)\mathbf{v} = \mathbf{0}, \tag{11}$$

where $\mathbf{M}(\sigma)$ is a 4×3 polynomial matrix parameterized by σ, and $\mathbf{v} = [1, f, f^2]^\top$ is a vector of monomials in f without σ. $\mathbf{M}(\sigma)$ can be rewritten as

$$\mathbf{M}(\sigma) = \sigma^4 \mathbf{B}_4 + \sigma^3 \mathbf{B}_3 + \sigma^2 \mathbf{B}_2 + \sigma \mathbf{B}_1 + \mathbf{B}_0, \tag{12}$$

	F7	E6f_e	E6f_s	E4f_e	E5f_v	H4f_e	H4f_s	H4f_v	E4f_s	E6l
Reference	[23]	[22, 28]	[9, 28]	[14]	[14]	[12, 13]	[12, 13]	[12, 13]		
Different f	✔		○		✔		○	○	○	○
Pure rotation				✔	✔	○	○	○	✔	✔
Pure translation							○		✔	✔
Plane				✔	○	✔	✔	✔	✔	✔
Gravity prior				✔	✔	✔	✔	✔	✔	✔
DOF	7	6	6	4	5	7	7	8	4	4
No. of points	7	6	6	4	5	3.5	3.5	4	4	6
No. of solutions	3	15	9	20	24	24	12	8	10	1

Table 1. The properties of the proposed solvers (in gray) and the state-of-the-art solvers.

where $\mathbf{B}_4, \mathbf{B}_3, \mathbf{B}_2, \mathbf{B}_1, \mathbf{B}_0$ are some 4×3 coefficient matrices containing only numbers.

If the number of the rows of matrix $\mathbf{M}(\sigma)$ is equal to the number of the columns, *i.e.*, $N_{row} = N_{col}$, we can directly solve the system of polynomial equations (11) as a polynomial eigenvalue problem [27] or by computing the roots of the polynomial determinant $\det \mathbf{M}(\sigma) = 0$ [22]. If $N_{row} < N_{col}$, we can use the method from [27] to extend the system of equations to a system with square matrix $\mathbf{M}(\sigma)$. In our case, we have more rows than columns, *i.e.* $N_{row} > N_{col}$. In [27], the authors show that selecting a subset of polynomials to make $N_{row} = N_{col}$ can solve such systems. However, it may happen that a subset of the original polynomials leads to singular matrices $\mathbf{B}_4$ and $\mathbf{B}_0$, which is a degeneracy for [27]. In [22], two different approaches for non-square systems were illustrated. One approach is based on computing the the greatest common divisor of all $(N_{col}-1) \times (N_{col}-1)$ submatrices of $\mathbf{M}(\sigma)$, and the second one on computing $\det(\mathbf{M}(\sigma)^\top \mathbf{M}(\sigma))$. However, neither of these approaches are efficient.

In this paper, we use a very simple approach that is both efficient and that also avoids possible degeneracies caused by selecting only a subset of original equations. Since in general, either $\mathbf{B}_4$ or $\mathbf{B}_0$ is non-singular, in this approach we multiply (11) by $\mathbf{B}_4^\top$ (or $\mathbf{B}_0^\top$). This actually generates three new polynomials that are linear combinations of the original four ones. Let $\mathbf{C}_i = \mathbf{B}_4^\top \mathbf{B}_i, i = 1, \dots 4$. Matrices $\mathbf{C}_i$ are 3×3 square matrices, thus, polynomial matrix $\mathbf{B}_4^\top \mathbf{M}(\sigma)$ becomes a 3×3 square matrix. If we consider (11) as a polynomial eigenvalue problem [3], the solutions to σ are the eigenvalues of 12×12 matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ -\mathbf{C}_4^{-1}\mathbf{C}_0 & -\mathbf{C}_4^{-1}\mathbf{C}_1 & -\mathbf{C}_4^{-1}\mathbf{C}_2 & -\mathbf{C}_4^{-1}\mathbf{C}_3 \end{bmatrix}. \quad (13)$$

An efficient way to find the eigenvalues of such matrix is to use the real Schur decomposition [24]: $\mathbf{Q} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ where $\mathbf{U}$ is a real orthogonal matrix and $\mathbf{\Lambda}$ is a real quasi-triangular matrix. A quasi-triangular matrix is a block-triangular matrix whose diagonal consists of 1×1 blocks and 2×2 blocks with complex eigenvalues which can be omitted. The eigenvalues of the blocks on the diagonal of $\mathbf{\Lambda}$ are the same as the eigenvalues of the matrix $\mathbf{Q}$. Once we have solutions to σ, the focal length f can be extracted from the null vector of $\mathbf{M}(\sigma)$ based on (11).

In this way, we obtain 12 possible solutions. Note that we solved a relaxed version of the original problem (11) [2], leading to two redundant solutions which do not ensure the elements of $\mathbf{v}$ to satisfy $\mathbf{v} = [1, f, f^2]^\top$.

Once $\{\sigma, f\}$ are calculated, the translation is extracted from the null space of the matrix $\mathbf{A}$ (5). In practice, we only need to calculate the null space of one of the 3×3 submatrices of the matrix $\mathbf{A}$. Among the solutions we are only interested in the real ones with positive focal length. Finally, the full relative rotation and translation can be found as $\mathbf{R} = \mathbf{R}_{align}^{'\top} \mathbf{R}_y \mathbf{R}_{align}$ and $\mathbf{T} = \mathbf{R}_{align}^{'\top} \boldsymbol{\tau}$.

Sturm sequences solution. Another way of solving the system in (11) is to compute the polynomial determinant [22] of the 3×3 polynomial matrix $\mathbf{B}_4^\top \mathbf{M}(\sigma)$. Since (11) has a non-trivial solution, the matrix $\mathbf{B}_4^\top \mathbf{M}(\sigma)$ should be rank deficient, *i.e.* $\det(\mathbf{B}_4^\top \mathbf{M}(\sigma)) = 0$. This is a univariate polynomial of degree 12 in σ (there are also two redundant solutions), which can be efficiently solved using the Sturm sequences [18].

3.2. Other Solutions

We have tested two additional state-of-the-art methods for generating efficient polynomial solvers. One is based on u-resultants [7] and one on Gröbner bases [30]. The u-resultant approach [7] generates a solver of size 22×32 with 10 solutions that can be extracted as the eigenvalues of a 10×10 matrix. The state-of-the art [30] Gröbner basis method generates a more efficient solver with a template of size 8×18 for the Gauss-Jordan elimination and with 10 solution that are extracted from the eigenvalues of a 10×10 matrix. In the synthetic experiments we show, that the hidden variable solver is more stable (see Fig. 2) than this

[2] The original system (11) has 10 solutions as it can be shown, *e.g.*, using computer algebra system Macaulay2 [19].

Gröbner basis solver. Hence, for practical applications, we recommend the user to use the hidden variable solver.

4. Linear Non-minimal Solver

In this section, we focus on solving the relative pose and focal length estimation problem when having a larger-than-minimal sample. This is particularly useful in the final model polishing step or in the local optimization of modern RANSACs, *e.g.*, MAGSAC++ [6]. Even when the final model accuracy is ensured by applying bundle adjustment, such fast non-minimal estimators are extremely important both in the local optimization or to provide an initial estimate for the numerical parameter refinement [25].

Since the vector $\mathbf{w}$ in (9) is of size 15×1, we need at least 15 equations for a linearization of the system (9). Equations in the system (9) are obtained as determinants of 3×3 submatrices of $N \times 3$ matrix $\mathbf{A}$ in (5). This matrix contains $\binom{N}{3}$ submatrices of size 3×3. Therefore, for $N \geq 6$, we obtain enough equations to linearize (9), *i.e.* we obtain at least 15 equations[3]. Such an overdetermined system can be considered as a linear system by ignoring the monomial dependencies in the vector $\mathbf{w}$. Therefore, the values of σ and f can be found as standard least square solutions to an overdetermined linear system[4]. Note that in this case the point normalization is important to improve the numerical stability of the solver.

5. Comparison with Existing Solvers

In this section, we show the properties of the existing state-of-the-art solvers including the well-known 7-point fundamental matrix solver (F7) [23], the 6-point solver assuming equal and unknown focal length (E6f_e) [22,28], the 6-point solver with a single known focal length (E6f_s) [9, 28], the essential matrix-based solvers with known gravity direction (E4f_e, E5f_v) [14], and the homography-based solvers with known gravity direction (H4f_s, H4f_s, H4f_s) [12, 13]. All these solvers have some degenerate configurations. For example, the standard solvers (F7, E6f_e, E6f_s) can not deal with pure rotation, pure translation and planar scenes. The homography-based solvers (H4f_s, H4f_s, H4f_s) can solve the previously mentioned special cases, but they require the underlying 3D points to be co-planar which is a fairly strong assumption in practice. The essential matrix-based solvers (E4f_e, E5f_v) using the gravity prior can deal with pure rotation and planar scenes. However, they are time consuming in practice (too many possible solutions) and can not handle pure translation. By contrast, the proposed methods are more efficient and do not have any of the aforementioned degenerate configurations. The comparisons of different solvers are shown in Table 1. Note that, ✔ means that the solvers can solve a particular special case without any additional assumptions. Circle ◯ means that the solver can solve a particular case but only when introducing additional assumptions, *e.g.*, co-planar points. The proposed minimal (E4f_s) and the linear non-minimal (E6l) solver are in gray.

Complexity analysis and running times. The following table contains the main operations performed by the proposed and the state-of-the-art solvers, together with their average running times in μs. The second column shows the size of the matrix for the SVD which is used to extract the null vector. The third column shows the size of the matrix for the Gauss-Jordan elimination. The fourth one reports the size of the matrix for the eigenvalue decomposition. The fifth one contains the degree of the univariate polynomial solved by Sturm sequences.

Solver	SVD	G-J	Eigen	Sturm	Time (μs)
E4f_s (polyeig)	-	3×12	12×12	-	51
E4f_s (GB)	-	8×18	10×10	-	38
E4f_s (Sturm)	-	-	-	12	24
E6l	20×15	-	-	-	40
F7	7×9	-	-	-	11
E6f_e	6×9	21×36	15×15	-	72
E6f_s	6×9	6×15	9×9	-	37
E4f_e	-	6×24	24×24	-	110
E5f_v	-	12×48	48×48	-	310

6. Synthetic evaluation

In this section, we compare the proposed solvers with the SOTA. We assume that the focal lengths of the cameras are different, but one of them is known. In this case, we compare with F7, E6f_s and E5f_v. Other solvers which need stronger assumptions, *e.g.* planarity, are omitted from these experiments. We use C++-mex implementations for all the solvers in this evaluation. The synthetic data are generated in the following setup. We randomly sample 200 3D points distributed in a 3D cube of size $[-3, 3] \times [-3, 3] \times [3, 8]$. The focal lengths of cameras are uniformly randomly set to $f_g \in [300, 3000]$ pixels, and the resolution of the image is 1000×1000 pixels. The parameters which were changed to test the performance are the noise level in the image point locations, the field of view (FOV) of cameras, the baseline between two cameras, and the noise level in the gravity direction. For the gravity noise, we noise the roll and pitch angles of both views. The default setting was: image noise = 1 pixel, FOV = 90°, baseline = 5% of the average scene depth, and the gravity vector noise = 0°.

The performance of the solvers is tested by modifying the value of a single parameter from the aforementioned ones while keeping the others constant. The rotation error is defined as the angle difference between the estimated rotation and the ground truth rotation as

[3]For 6 point correspondences we obtain $\binom{6}{3} = 20$ equations

[4]Note, that the obtained solution is not a least square solution to the original system (9), since we are ignoring monomial dependencies in $\mathbf{w}$.

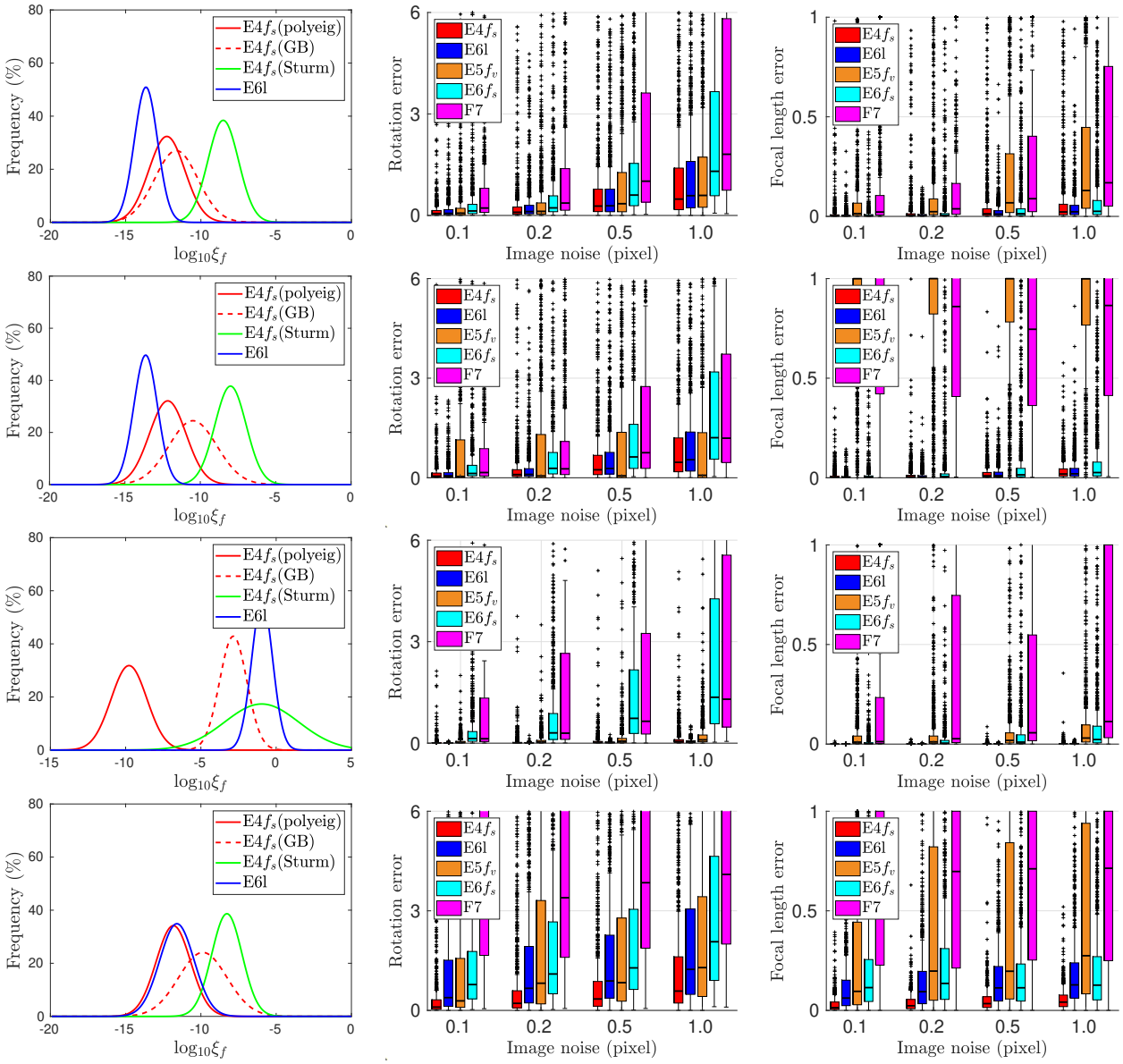

Figure 2. From **top row** to **bottom row**: performance under general motion, pure translation, pure rotation and planar structures, respectively. **Left column**: numerical stability of the proposed solvers on noise-free data. **Middle column**: rotation error of the solvers w.r.t. increasing image noise. **Right column**: focal length error of the solvers w.r.t. increasing image noise.

$\arccos\left(\left(\text{tr}(\mathbf{R}_g\mathbf{R}_e^\top)-1\right)/2\right)$, where $\mathbf{R}_g$ and $\mathbf{R}_e$ are the ground truth and the estimated rotation, respectively. The translation error is measured as the angle between the estimated and the ground truth translation vectors, since the estimated translation is recovered only up to scale. The focal length error was measured as $\xi_f = |f_e - f_g|/f_g$, where f_g and f_e are the ground truth and the estimated focal length, respectively. We focus on four practical cases which are very common in real applications: **general motion**, **pure translation**, **pure rotation** and **planar scenes**.

The left column of Fig. 2 shows the numerical stability of the proposed solvers in four different configurations (from top row to bottom one: general motion, pure translation, pure rotation and planar scenes, respectively). The proposed solvers are stable for all tested configurations, except for the pure rotation, where the proposed 6-point linear non-minimal solver E6l and the minimal solver based on Sturm sequences provide slightly unstable results on the noise-free data. However, these solvers provide very accurate results in the presence of image noise.

From three tested minimal solvers, the polynomial eigenvalue solver is more stable than the solver based on Sturm sequences and the Gröbner basis solver. The middle and right columns of Fig. 2 report the rotation and the focal

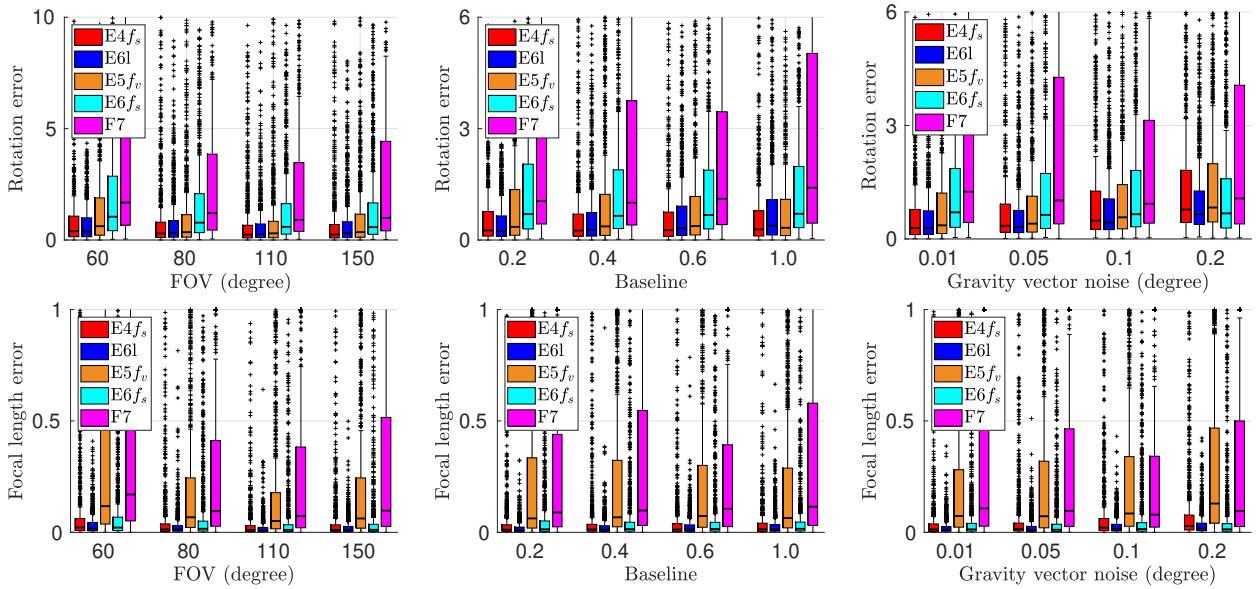

Figure 3. Rotation (**top row**) and focal length (**bottom row**) error under general motion. *From left to right*: the columns show the error of the solvers w.r.t. increasing field-of-view, baseline, and gravity vector noise.

length error w.r.t. increasing image noise. Based on the experimental results, we can see that the proposed solvers outperform the existing methods on different configurations. Due to the lack of space and for the better readability these graphs contain results only for one new minimal 4-point solver, *i.e.* the polynomial eigenvalue solver (E4f_s). The translation error is reported in the supplementary material.

Fig. 3 shows errors for general camera motion as a function of the field-of-view, baseline, and the gravity vector noise. The proposed solvers (E4f_s) and (E6l) lead to the most accurate results for most of the configurations. Even when the noise level in the roll and the pitch angle (for both cameras) is 0.2°, the proposed solvers are comparable to the SOTA solvers. Note that accelerometers used in cars and modern smartphones have noise levels around 0.06° [16].

7. Real-world Experiments

In order to show the practical benefits of the proposed methods in real applications, we test the solvers on the `KITTI` [17][5] datasets. Moreover, we collected the new `PHONE` dataset. The `KITTI` odometry benchmark provides 22 sequences, but only 11 sequences (00–10) are provided with ground truth obtained by GPS and IMU for training. We therefore used these 11 sequences to evaluate the compared solvers. In total, 23190 image pairs were used. The `PHONE` dataset is recorded by using different smartphones (iPhone 6s and iPhone 11). The sequences were captured at @30Hz with the rear camera, and the corresponding IMU data were captured at @100Hz with the built-in sensor. In addition, the sequences cover all the camera configurations we discussed in the synthetic evaluation: general motion, pure translation and rotation, and planar scenes. To obtain a ground truth, we calibrated the phones and use the RealityCapture [1] software to obtain camera poses and 3D reconstructions. In total, 12464 image pairs with synchronized gravity directions, ground truth poses, calibrations and 3D reconstructions were generated. Example images are shown in the supplementary material.

For testing the proposed solvers on real-world data, we chose a state-of-the-art RANSAC, *i.e.*, Graph-Cut RANSAC[6] [4] (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. We use the proposed solvers in (a). Note that the E5f_v solver is filling large matrices with complex symbolic coefficients computed from symbolic determinants. Thus, the C++ implementation, which is huge (23.9 MB), crashed in our experiments. In this case, we provide extra Matlab (C++-mex) tests with the E5f_v solver.

The cumulative distribution functions (CDF) of the rotation, translation and focal length errors, run-time, iteration number, and inlier number on the `KITTI` dataset are shown in Fig. 4. Being accurate is interpreted as a curve close to the top-left corner. Both proposed solvers lead to more accurate rotation, translation, and focal length estimates than the tested SOTA ones. At the same time, the new solvers require fewer RANSAC iterations and, thus, they are faster. While SOTA methods provide more in-

[5] http://www.cvlibs.net/datasets/kitti

[6] https://github.com/danini/graph-cut-ransac

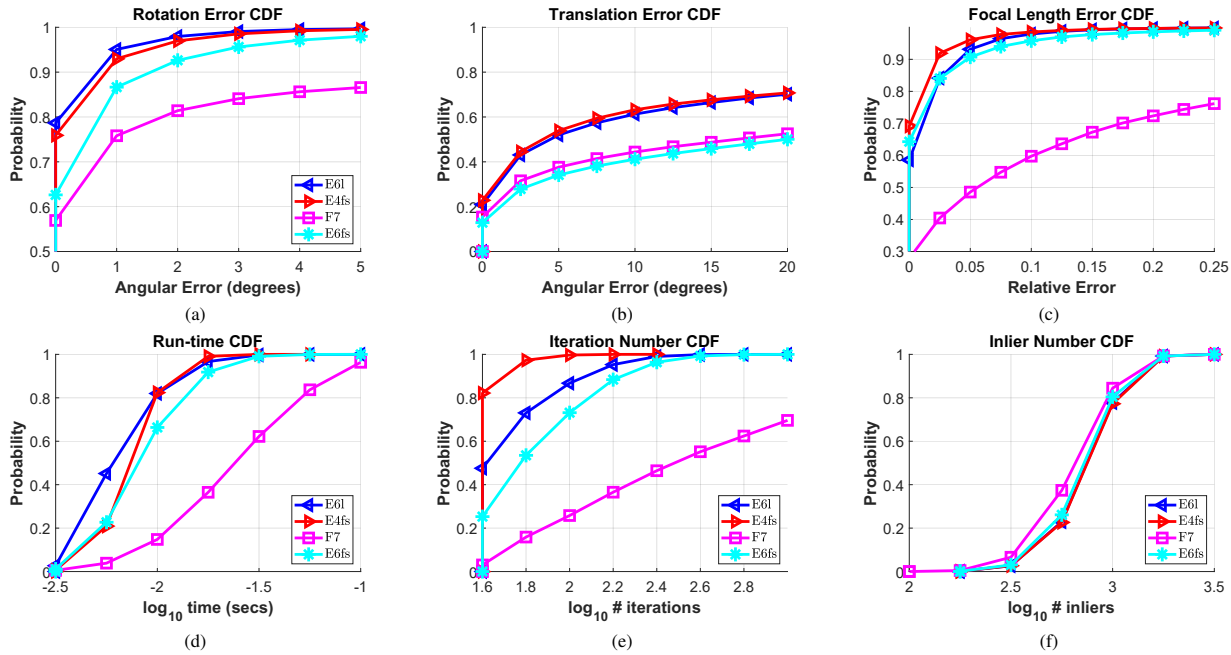

Figure 4. The CDFs of the (a) rotation, (b) translation (both in degrees), (c) focal length errors, (d) run-times, (e) iteration, and (f) inlier numbers of GC-RANSAC on the KITTI datasets (23 190 images). Being accurate is interpreted as a curve close to the top-left corner.

		E4f_s	E6l	E6f_s	F7
$\xi_{\mathbf{R}}$ (°)	MED	0.59	**0.57**	2.31	4.93
	AVG	1.17	**1.05**	4.75	20.39
$\xi_{\mathbf{T}}$ (°)	MED	**16.55**	17.84	30.65	29.81
	AVG	**23.10**	23.16	39.10	37.81
ξ_f (%)	MED	**0.30**	0.49	0.50	1.18
	AVG	**0.74**	1.25	1.58	11.05
# iterations	AVG	**101**	120	150	776
# inliers	AVG	**1370**	1345	1275	1100

Table 2. Solver comparison on the captured PHONE dataset (12 464 image pairs). The rotation ($\xi_{\mathbf{R}}$), translation ($\xi_{\mathbf{t}}$) and relative focal length ($\xi_{\mathbf{f}}$) errors are reported. The best results are marked bold.

liers, the pose accuracy is usually more important in most of the real-world applications. Table 2 shows the median and mean rotation, translation and focal length errors for the PHONE dataset. We used every 10th frame for this dataset. Again the proposed solvers lead to the most accurate results, here, with more inliers and fewer iterations than the SOTA solvers. Table 3 shows the comparison of E4f_s and E5f_v (pre-compiled C++-mex implementation with Matlab) on the first sequence from the PHONE dataset (1535 images). Additional results are in the Supp. material.

Limitations. In this paper, we assume that the two cameras have a common direction, which can be extracted, *e.g.*, from the IMU readings. Since cameras used in modern smartphones, tablets, and robots are usually equipped with IMUs, we believe that this assumption is reasonable and practical.

		$\xi_{\mathbf{R}}$(°)	$\xi_{\mathbf{T}}$(°)	ξ_f (%)
E4f_s	MED	**0.88**	**3.62**	**1.16**
	AVG	**1.33**	**5.60**	**1.63**
E5f_v	MED	2.05	5.05	20.35
	AVG	5.13	9.52	33.88

Table 3. Comparison of E4f_s and E5f_v on the first sequence from the PHONE dataset. The best results are marked bold.

8. Conclusion

In this paper, we focus on the case when one of the cameras is fully calibrated while the focal length of the other one is unknown. Assuming a known common reference direction, we propose new minimal solvers that estimate the relative pose together with the unknown focal length from a minimum of four point correspondences. We also propose a linear solver that allows for estimating the pose from a larger-than-minimal sample efficiently. The configuration with one calibrated and one camera with unknown focal length is resistant to the typical degenerate cases of the traditional six-point algorithm [46] and has a number of practical benefits, *e.g.*, when processing large-scale datasets. We demonstrate on thousands of image pairs from publicly available datasets and on a new PHONE dataset, that the proposed solvers are superior to the state-of-the-art both in terms of accuracy and processing time. The source code and the PHONE dataset are available at https://github.com/yaqding/relative-pose-E4f.

Acknowledgments. This work was supported by the ETH Zurich Postdoctoral Fellowship, and by the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

References

[1] Realitycapture. http://www.capturingreality.com. 7

[2] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. Rolling shutter absolute pose problem with known vertical direction. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[3] Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM, 2000. 4

[4] Daniel Barath and Jiří Matas. Graph-cut RANSAC. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[5] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient initial pose-graph generation for global sfm. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14546–14555, 2021. 1

[6] Daniel Barath, Jana Noskova, and Jiri Matas. Marginalizing sample consensus. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2021. 5

[7] Snehal Bhayani, Zuzana Kukelova, and Janne Heikkila. A sparse resultant based method for efficient minimal solvers. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4

[8] Sylvain Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *International Conference on Computer Vision (ICCV)*, 1998. 2

[9] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. 3d reconstruction from image collections with a single known focal length. In *International Conference on Computer Vision (ICCV)*, pages 1803–1810. IEEE, 2009. 2, 4, 5

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

[11] Yaqing Ding, Daniel Barath, and Zuzana Kukelova. Minimal solutions for panoramic stitching given gravity prior. In *International Conference on Computer Vision (ICCV)*. 2

[12] 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. 2, 4, 5

[13] Yaqing Ding, Jian Yang, Jean Ponce, and Hui Kong. Homography-based minimal-case relative pose estimation with known gravity direction. *Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2020. 4, 5

[14] 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, 3, 4, 5

[15] Jan-Michael Frahm and Reinhard Koch. Camera calibration with known rotation. In *International Conference on Computer Vision (ICCV)*, volume 3, pages 1418–1418. IEEE Computer Society, 2003. 2

[16] 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. 2, 7

[17] 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. 7

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

[19] Daniel R Grayson and Michael E Stillman. Macaulay 2, a software system for research in algebraic geometry, 2002. 4

[20] Banglei Guan, Qifeng Yu, and Friedrich Fraundorfer. Minimal solutions for the rotational alignment of imu-camera systems using homography constraints. *Computer vision and image understanding*, 2018. 2

[21] Richard Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conference on Computer Vision (ECCV)*, 1992. 2

[22] 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, 2, 3, 4, 5

[23] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 4, 5

[24] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. 4

[25] Maksym Ivashechkin, Daniel Barath, and Jiri Matas. VSAC: Efficient and accurate estimator for h and f. 2021. 2, 5

[26] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *Asian Conference on Computer Vision (ACCV)*, 2010. 2

[27] 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, 2, 3, 4

[28] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4, 5

[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. 3

[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. 2, 3, 4

[31] Gim Hee Lee, Marc Pollefeys, and Friedrich Fraundorfer. Relative pose estimation for a multi-camera system with known vertical direction. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[32] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 630–633. IEEE, 2006. 1

[33] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey.

IEEE transactions on visualization and computer graphics, 22(12):2633–2651, 2015. 1

[34] Branislav Micusik. Relative pose problem for non-overlapping surveillance cameras with known gravity vector. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3105–3112. IEEE, 2011. 1

[35] Jun Minagawa, Kohei Okahara, Kento Yamazaki, and Tsukasa Fukasawa. A camera recalibration method for a top-view surveillance system based on relative camera pose and structural similarity. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 1–8. IEEE, 2019. 1

[36] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 1

[37] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[38] 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*, 33(5):1255–1262, 2017. 1

[39] 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. 2

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

[41] Marcus Valtonen Ornhag, Patrik Persson, Marten Wadenback, Kalle Astrom, and Anders Heyden. Efficient real-time radial distortion correction for uavs. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 1751–1760, 2021. 2

[42] 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. 2

[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. 2

[44] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1, 2

[45] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 1

[46] Henrik Stewenius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006. 1, 2, 8

[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. 2

[48] Chris Sweeney, John Flynn, Benjamin Nuernberger, and Matthew Turk. Efficient computation of absolute pose for gravity-aware augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality*, 2015. 2

[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. 2, 3

[50] Christopher Sweeney, Tobias Hollerer, and Matthew Turk. Theia: A fast and scalable structure-from-motion library. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 693–696, 2015. 1, 2

[51] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 1

[52] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision (ECCV)*, pages 61–75. Springer, 2014. 1

[53] Changchang Wu et al. VisualSFM: A visual structure from motion system. 2011. 1

[54] Ji Zhao and Banglei Guan. On relative pose recovery for multi-camera systems. *arXiv preprint arXiv:2102.11996*, 2021. 3

[55] Ji Zhao, Laurent Kneip, Yijia He, and Jiayi Ma. Minimal case relative pose computation using ray-point-ray features. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1176–1190, 2019. 3