Minimal Solutions to Relative Pose Estimation From Two Views Sharing a Common Direction With Unknown Focal Length

Yaqing Ding* Jian Yang* Jean Ponce†‡ Hui Kong*

Abstract

We propose minimal solutions to relative pose estimation problem from two views sharing a common direction with unknown focal length. This is relevant for cameras equipped with an IMU (inertial measurement unit), e.g.*, smart phones, tablets. Similar to the 6-point algorithm for two cameras with unknown but equal focal lengths and 7-point algorithm for two cameras with different and unknown focal lengths, we derive new 4- and 5-point algorithms for these two cases, respectively. The proposed algorithms can cope with coplanar points, which is a degenerate configuration for these 6- and 7-point counterparts. We present a detailed analysis and comparisons with the state of the art. Experimental results on both synthetic data and real images from a smart phone demonstrate the usefulness of the proposed algorithms.*

1. Introduction

Estimating relative camera motion from two views using minimal point correspondences is a classical problem in computer vision. For example, given internally calibrated cameras, it is well known that the relative pose can be estimated using the 5-point algorithm [16, 21, 28]. An important case arises when the only unknown camera intrinsic parameter is the focal length (semi-calibrated case). It is important in practice since most modern CCD or CMOS cameras have square-shaped pixels and central principal point. If two cameras have shared and unknown focal length, the relative motion and common focal length can be estimated using six point correspondences [16, 21, 22, 30]. If the focal lengths of the two cameras are different and unknown, at least seven point correspondences are needed to recover the relative motion and focal lengths [3, 15]. Minimal-case algorithms can be used with RANSAC [10] in for example, SfM (structure-from-motion) pipelines. Using as few points as possible is of extreme importance to reduce processing time. However, this usually requires additional constraints or data. For example, if the common gravity direction between the two views are known, the number of needed points for the relative pose estimation can be reduced [9, 12, 27, 29, 31].

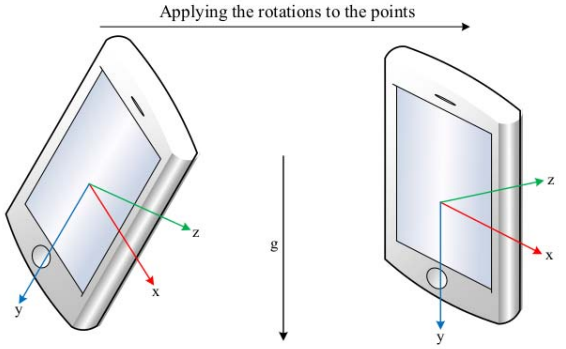

Figure 1: Visualization of the coordinate used in this paper. The cameras have a common gravity direction. The gravity vector $g = [g_x, g_y, g_z]$ can be obtained form IMU readings, and the roll and pitch angles w.r.t the world coordinate (gravity) can be extracted. Then the points can be rotated so that the y-axis matches the gravity direction.

In this paper, we *assume* that the views have a common direction. This case is relevant for smart devices, *e.g.*, smart phones, tablets, which have IMUs to measure the gravity direction. We can align one of the axes (*e.g.*, y-axis) of the cameras with this direction (Figure 1). Then there is only one unknown rotation parameter so that we can use fewer point correspondences.

Given this assumption, we propose minimal solutions to the two-view relative pose problems with unknown focal length. The main contributions of this paper are:

- For two cameras with the same unknown focal length, we propose a 4-point algorithm to estimate the camera

*Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology. {dingyaqing,csjyang,konghui}@njust.edu.cn
†INRIA, Paris, France. jean.ponce@inria.fr
‡Département d'informatique de l'ENS, ENS, CNRS, PSL University, Paris, France.

motion and focal length.

- For two cameras with different and unknown focal lengths, we show that only five points are required to recover the camera motion and the focal lengths.
- We present both the polynomial eigenvalue solution and action matrix solution to the proposed problems.

The most closely related work to ours is by Ding *et al*. [9], which solves the relative pose problem with unknown focal length using a homography and the gravity direction. Unlike them, we do not need to assume coplanar points, and the proposed methods are more general. Compared to the standard 6- and 7-point algorithms, the proposed approach has two advantages. (i) It requires fewer correspondences, which is important for RANSAC, (ii) They are more accurate with noisy data. In addition, the proposed 4- and 5-point algorithms can deal with the case that points are on a plane in 3D space, which is a degenerate configuration for existing 6- and 7-point algorithms [18].

2. Problem statement

Let $m_i = [u_i, v_i, 1]^T$ and $m'_i = [u'_i, v'_i, 1]^T$ be the homogeneous coordinates of corresponding points in the first and second images. The Euclidean transformation (R, T) of $SE(3)$ between the two frames can be expressed as:

$$\lambda_2 K_2^{-1} m'_i = \lambda_1 R K_1^{-1} m_i + T, \tag{1}$$

where λ_1, λ_2 are the depths of the image points m_i, m'_i, and K_1, K_2 are the camera intrinsic matrices of the first and second cameras, respectively.

In this paper, we *assume* that the views have a common reference direction. We can use the gravity direction computed by an IMU on mobile phones or tablets for this reference direction. Without loss of generality, we can align the y-axis of the two cameras with the common reference direction. Using this direction, we can compute the roll and pitch angles of the two correspondence cameras for the alignment. Usually if we want to know the extrinsic parameters between the camera and the IMU, we need to know the intrinsic parameters of the camera first. However, smart devices such as smart phones and tablets are special, because the relationship between the axes of the camera and the IMU are usually approximate to $0°, \pm 90°, 180°$ [9, 14]. In this case, we can directly obtain the rotation between the camera and the IMU. Let's denote the rotation matrices from the roll and pitch angles of the two camera frames as $(R_r, R_p) \leftrightarrow (R'_r, R'_p)$. In this case, (1) can be written as

$$\lambda_2 R'_p R'_r K_2^{-1} m'_i = \lambda_1 R_y R_p R_r K_1^{-1} m_i + \widetilde{T}, \tag{2}$$

where R_y is the rotation from the yaw angle (around the y-axis), and $\widetilde{T} = R'_p R'_r T$ is the translation after the alignment. For most modern CCD or CMOS cameras, it is reasonable to assume that the cameras have square-shaped pixels, and the principal point coincides with the image center [16]. Hence, we can let $K_1^{-1} = \mathrm{diag}(1, 1, f_1), K_2^{-1} = \mathrm{diag}(1, 1, f_2)$ for the first and second cameras. Since the cross product of $\lambda_2 R'_p R'_r K_2^{-1} m'_i$ and $\lambda_1 R_y R_p R_r K_1^{-1} m_i$ is perpendicular to the translation $\widetilde{T}$, we obtain

$$(R'_p R'_r [u'_i, v'_i, f_2]^\top) \times (R_y R_p R_r [u_i, v_i, f_1]^\top) \cdot \widetilde{T} = 0. \tag{3}$$

In this case, the depth parameters λ_1, λ_2 are eliminated. The rotation matrix R_y can be written as

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{4}$$

where θ is the rotation angle around the gravity direction. The rotation matrix R_y can thus be rewritten as

$$R_y = \frac{1}{1+s^2} \begin{bmatrix} 1-s^2 & 0 & 2s \\ 0 & 1+s^2 & 0 \\ -2s & 0 & 1-s^2 \end{bmatrix}, \tag{5}$$

where $s = \tan\frac{\theta}{2}$. This formulation introduces a degeneracy for a $180°$ rotation, which can almost be ignored in real application. Since (3) is homogeneous, the scale factor $\frac{1}{1+s^2}$ in (5) can be omitted. Our aim is to estimate the rotation, translation and focal lengths of the cameras using the minimal number of point correspondences.

3. Shared and unknown focal length - E4f

The first situation is that the two cameras have the same focal length. For example, the scenes are captured by a smart phone with constant focal length. In this case, we have a 4-DOF problem with respect to $\{s, f, t_x, t_y, t_z\}$ (there are 5 unknowns, but the translation is up to a scale factor). Since each point correspondence gives one constraint, we need at least 4 points. By stacking the constraint rows for 4 point correspondences, constraint (3) can be written as

$$A\,\widetilde{T} = 0, \tag{6}$$

where A is a 4×3 data matrix and the i^{th} row of A is

$$A_i = (R'_p R'_r [u'_i, v'_i, f]^\top) \times (R_y R_p R_r [u_i, v_i, f]^\top). \tag{7}$$

Since (6) has non-trivial solutions, the matrix A must be rank-deficient, which means that every determinant of all its 3×3 submatrices must vanish.

Property 1. *The determinant of the* 3×3 *submatrices of* A *can be written as* $det(A_{ijk}) = (1 + s^2)h(s, f)$*, where* $h(s, f)$ *are polynomials in* $\{s, f\}$ *(subscripts* $_{ijk}$ *indicate the rows of the matrix* A*).*

One can verify this property using Matlab or other symbolic computation softwares. In this case, we can reduce

the degrees of the polynomials. In particularly, we obtain $C_3^4 = 4$ polynomials of degree 8 (the highest degree term is s^4f^4) in the two unknowns $\{s, f\}$

$$\begin{aligned} h_1(s,f) &= \det(A_{123})/(1+s^2), \\ h_2(s,f) &= \det(A_{124})/(1+s^2), \\ h_3(s,f) &= \det(A_{134})/(1+s^2), \\ h_4(s,f) &= \det(A_{234})/(1+s^2). \end{aligned} \tag{8}$$

Next we describe how to solve this system of equations using the polynomial eigenvalue method and action matrix method, respectively.

3.1. Polynomial eigenvalue solution

Polynomial eigenvalue methods have been successfully used for many minimal problems in computer vision, such as the 9-point one-parameter radial distortion problem [11], the 5- and 6-point relative pose problems [21], the 6-point one unknown focal length problem [4], and the self-calibration problems [17]. We show that the system of polynomials (8) can be efficiently solved using the polynomial eigenvalue method with some modifications.

As shown in [2], polynomial eigenvalue problems are problems of the form

$$\mathrm{M}(\lambda)\mathbf{v} = 0, \tag{9}$$

where $\mathrm{M}(\lambda)$ is a square matrix parameterized by λ, and $\mathbf{v}$ is a vector of monomials in all variables without λ. $\mathrm{M}(\lambda)$ is defined as

$$\mathrm{M}(\lambda) \equiv \lambda^l \mathrm{C}_l + \lambda^{l-1}\mathrm{C}_{l-1} + \cdots + \lambda \mathrm{C}_1 + \mathrm{C}_0, \tag{10}$$

where $\mathrm{C}_l, \mathrm{C}_{l-1}, ..., \mathrm{C}_1, \mathrm{C}_0$ are square coefficient matrices. Note that, first (8) can be written as

$$\mathrm{B}\mathbf{X} = 0, \tag{11}$$

where B is a 4×25 coefficient matrix and

$$\begin{aligned} \mathbf{X} = (&1, f, f^2, f^3, f^4, s, sf, sf^2, sf^3, sf^4, s^2, s^2f, s^2f^2, s^2f^3, \\ &s^2f^4, s^3, s^3f, s^3f^2, s^3f^3, s^3f^4, s^4, s^4f, s^4f^2, s^4f^3, s^4f^4) \end{aligned}$$

is a vector of all the 25 monomials. The unknowns s and f both appear in degree four monomials, so we can choose either of them, *e.g.*, s as λ in (9). The four polynomials can be rewritten as

$$\mathrm{M}(s)\mathbf{v} = 0, \tag{12}$$

where $\mathbf{v} = (1, f, f^2, f^3, f^4)^\top$ is a 5×1 vector of monomials in f. There are four polynomials, but v has five elements. Therefore, we need to add additional polynomials by multiplying the original ones with f (it is enough for this problem). We obtain four additional polynomials and select two of them. In this case, (11) and (12) can be rewritten as

$$\mathrm{B}'\mathbf{X}' = 0, \tag{13}$$

where B' is a 6×30 coefficient matrix and $\mathbf{X}'$ is a vector if all the 30 monomials, and (12) can be rewritten as

$$(s^4\mathrm{C}_4 + s^3\mathrm{C}_3 + s^2\mathrm{C}_2 + s\mathrm{C}_1 + \mathrm{C}_0)\mathbf{v} = 0, \tag{14}$$

where $\mathbf{v} = (1, f, f^2, f^3, f^4, f^5)^\top$, $\mathrm{C}_4, \mathrm{C}_3, \mathrm{C}_2, \mathrm{C}_1$, and C_0 are 6×6 coefficient matrices:

$$\begin{aligned} \mathrm{C}_0 &\equiv (b_1, b_2, b_3, b_4, b_5, b_6), \\ \mathrm{C}_1 &\equiv (b_7, b_8, b_9, b_{10}, b_{11}, b_{12}), \\ \mathrm{C}_2 &\equiv (b_{13}, b_{14}, b_{15}, b_{16}, b_{17}, b_{18}), \\ \mathrm{C}_3 &\equiv (b_{19}, b_{20}, b_{21}, b_{22}, b_{23}, b_{24}), \\ \mathrm{C}_4 &\equiv (b_{25}, b_{26}, b_{27}, b_{28}, b_{29}, b_{30}), \end{aligned} \tag{15}$$

where b_n is the nth column of the 6×30 matrix B'. Based on [2], the solutions of s are the eigenvalues of the 24×24 matrix:

$$\mathrm{D} = \begin{bmatrix} 0 & \mathrm{I} & 0 & 0 \\ 0 & 0 & \mathrm{I} & 0 \\ 0 & 0 & 0 & \mathrm{I} \\ \text{-}\mathrm{C}_4^{-1}\mathrm{C}_0 & \text{-}\mathrm{C}_4^{-1}\mathrm{C}_1 & \text{-}\mathrm{C}_4^{-1}\mathrm{C}_2 & \text{-}\mathrm{C}_4^{-1}\mathrm{C}_3 \end{bmatrix}. \tag{16}$$

We obtain 24 eigenvalues which are the solutions to s. The corresponding eigenvectors contain solutions for f, *i.e.*, the second rows of the eigenvectors divided by the first rows. However, four of the solutions do not satisfy the inner constraints of the vector $\mathbf{v}$, *e.g.*, $\mathbf{v}_3 = \mathbf{v}_2^2$. It is because we are solving a relaxed version of the original one so that there are redundant solutions. Hence, there are up to 20 possible solutions (include complex ones). The computer algebra system Macaulay2 [13] also shows that there are in general 20 solutions. Once $\{s, f\}$ are calculated, the translation can be extracted from the null space of the matrix A. In practice, we only need to calculate the null space of one of the 3×3 submatrices of A. Then we may obtain 20 possible focal lengths (including negative and complex ones) and rotations, which correspond to 40 possible translations (each rotation corresponds to two possible opposite translations). Among the solutions we are only interested in the real ones with positive focal length. Finally the full rotation and translation between the two views are given by

$$\begin{aligned} R &= R_r'^\top R_p'^\top R_y R_p R_r, \\ T &= R_r'^\top R_p'^\top \widetilde{T}. \end{aligned} \tag{17}$$

3.2. Action matrix solution

The system of polynomials (8) can also be solved using the action matrix method [6]. We present only the simple implementation of the method and refer the reader to [6, 19, 5, 23, 25] for details. As described in [5], in order to create the action matrix, we need to generate additional polynomials by multiplying the initial polynomials

with monomials. The degree of the additional polynomials are up to ten, which means that we need to multiply the original polynomials with $\{s, f, s^2, sf, f^2\}$. Then there are totally 24 polynomials with 46 monomials. After removing 8 unnecessary polynomials and 10 monomials which are not used in the action matrix, we obtain a 16×36 template for the Gauss-Jordan elimination.

4. Different and unknown focal lengths - $\mathbf{E5f_1f_2}$

The other important case is when the two cameras have different and unknown focal lengths, *e.g.*, scenes are captured by multiple smart phones or a smart phone with a zoom camera. It is a 5-DOF problem with respect to $\{s, f_1, f_2, t_x, t_y, t_z\}$. In this case, we need at least 5 points. The i^{th} row of $A_{5\times 3}$ can be formulated as

$$A_i = (R'_p R'_r [u'_i, v'_i, f_2]^\top) \times (R_y R_p R_r [u_i, v_i, f_1]^\top). \quad (18)$$

Since every determinant of the 3×3 submatrices of A must vanish, we can obtain $C_3^5 = 10$ polynomials:

$$\begin{aligned} h_1(s, f) &= \det(A_{123})/(1 + s^2), \\ h_2(s, f) &= \det(A_{124})/(1 + s^2), \\ &\dots \\ h_{10}(s, f) &= \det(A_{345})/(1 + s^2). \end{aligned} \quad (19)$$

Based on **Property 1**, the polynomials are of degree 8 (the highest degree term is $s^4 f_1^2 f_2^2$).

Polynomial eigenvalue solution

We can first rewrite the system of ten polynomial equations as

$$\mathrm{B}\mathbf{X} = 0, \quad (20)$$

where B is a 10×45 coefficient matrix and $\mathbf{X} = \{s^i f_1^j f_2^k \mid i = 0, 1, 2, 3, 4; j = 0, 1, 2; k = 0, 1, 2\}$ is the vector formed by all 45 monomials. We still choose s as λ in (9). The ten polynomials can be rewritten as $\mathrm{M}(s)\mathbf{v} = 0$, where $\mathbf{v} = (1, f_2, f_2^2, f_1, f_1 f_2, f_1 f_2^2, f_1^2, f_1^2 f_2, f_1^2 f_2^2)^\top$ is a vector of all 9 monomials in f_1, f_2. There are ten polynomials, and v has only nine elements. It seems that we can choose nine polynomials to build a solver just as Sec. 3.1. However, we find that the maximum rank of both the 9×9 coefficient matrices $\mathrm{C}_4, \mathrm{C}_0$ is 8, which makes the system ill-conditioned. So we need to add additional polynomials to obtain well-conditioned matrices $\mathrm{C}_4, \mathrm{C}_0$. In practice, multiplying the original polynomials with f_1 (or f_2) is enough. In this case, there are 20 polynomials and 60 monomials in total and we can select 12 of polynomials to ensure that $\mathrm{C}_4, \mathrm{C}_0$ are well-conditioned. Then, these 12 polynomials can be written as

$$\mathrm{B}'\mathbf{X}' = 0, \quad (21)$$

where B′ is a 12×60 coefficient matrix and $\mathbf{X}' = \{s^i f_1^j f_2^k \mid i = 0, 1, 2, 3, 4; j = 0, 1, 2, 3; k = 0, 1, 2\}$ is a vector of all the 60 monomials. Eq (21) can be rewritten as

$$(s^4\mathrm{C}_4 + s^3\mathrm{C}_3 + s^2\mathrm{C}_2 + s\mathrm{C}_1 + \mathrm{C}_0)\mathbf{v} = 0, \quad (22)$$

where $\mathbf{v} = (1, f_2, f_2^2, f_1, f_1 f_2, f_1 f_2^2, f_1^2, f_1^2 f_2, f_1^2 f_2^2, f_1^3, f_1^3 f_2, f_1^3 f_2^2)^\top$ is a vector of 12 monomials, and C_4, C_3, C_2, C_1, and C_0 are 12×12 coefficient matrices

$$\begin{aligned} \mathrm{C}_0 &\equiv (b_1, b_2, ..., b_{12}), \\ \mathrm{C}_1 &\equiv (b_7, b_8, ..., b_{24}), \\ \mathrm{C}_2 &\equiv (b_{13}, b_{14}, ..., b_{36}), \\ \mathrm{C}_3 &\equiv (b_{19}, b_{20}, ..., b_{48}), \\ \mathrm{C}_4 &\equiv (b_{25}, b_{26}, ..., b_{60}), \end{aligned} \quad (23)$$

where b_n is the nth column of the 12×60 matrix B′. We obtain the solutions of s as the 48 eigenvalues of a 48×48 matrix (similar to D in (16)). After normalizing the first entry of the eigenvector to 1, the second and the fourth ones are the solutions to f_2 and f_1. The computer algebra system Macaulay2 [13] shows that there are in general 24 solutions. The redundant solutions can be eliminated by checking the inner constraints of the vector $\mathbf{v}$ as before, *e.g.*, $\mathbf{v}_3 = \mathbf{v}_2^2$. Finally, there are up to 24 possible solutions (include complex ones). The solutions with negative focal lengths can be eliminated. Once $\{s, f_1, f_2\}$ are calculated, the remaining steps are equal to the shared and unknown focal length case.

Action matrix solution

The system of polynomials (19) can also be solved using the action matrix method. Since the polynomials are much more complex, we recommend using an online automatic generator, *e.g.*, [25]. We obtain a template of size 67×91 for the Gauss-Jordan elimination.

5. Degenerate configurations

Degenerate configurations may be caused by data or critical motions. Four or more collinear points will be redundant, since there are up to three linearly independent constraints for collinear points [12]. This case is quite unusual in practice and can be handled by robust estimators, such as RANSAC [10]. Arbitrary planar motions when the optical axes lie in the plane are critical motions for the standard 6 and 7-point algorithms [18]. For the proposed 4- and 5-point algorithms, the degenerate configuration can be expressed as $R_r = R_p = I$ and $R'_r = R'_p = I$, which means that the y-axes of the two cameras are coincided with the gravity direction. This leads to a different system of polynomials (smaller degree), which will make the matrices C_4 and C_0 singular. However, the probability for the roll and pitch angles to be both equal to zero is very low, so we do not discuss this case. On the other hand, our 5-point algorithm has an extra degenerate case: $R_r = R'_r$ and $R'_p = R'_p$, *i.e.*, the roll and pitch angles are equal for the two views. It

is a special case for the optical axes intersect, which is a degenerate case for the different and unknown focal lengths problem [18]. Pure rotation, which makes the essential matrix become zero matrix, is also a degenerate case for the standard essential matrix based algorithms. By contrast, our methods can deal with pure rotation since the constraints are direct on the rotation matrix. Experimental results are provided in Sec. 6.2.

6. Simulation results

In this section, we evaluate the performance of the proposed algorithms on synthetic data under increased image and IMU noise. The synthetic data are generated in the following setup. We randomly sample 200 3D points distributed on a 3D cube $[-3,3] \times [-3,3] \times [3,8]$. On the other hand, to illustrate that the proposed methods can cope with planar scenes, we also sample 200 3D points distributed on a plane. The focal length of the camera is set to $f_g \in [300, 3000]$ pixels, and the resolution of the image is set to $2f \times 2f$. Each 3D point is observed by two cameras with random but feasible poses. Similar to [28, 12, 29, 9], we focus on two important practical motions: sideways motion (parallel to the scene) and forward motion (along the z-axis). The distance between the two cameras is set to be 10 percent of the average scene depth. Additionally, the first and second cameras are rotated around every axis. We generate 10,000 pairs of images with different transformations. Note that all the algorithms have multiple solutions, so we need to choose the solution which can be recovered in real applications. So we calculate the geometry error of each solution with respect to the set of points, and choose the one with the minimum error.

6.1. Numerical stability

We first evaluate the numerical stability of the proposed algorithms with noise-free data. The rotation error ξ_R, translation error ξ_t and focal length error ξ_f are defined as follows

- $\xi_R = \arccos((\text{trace}(R_g R_e^T) - 1)/2)$,
- $\xi_t = \arccos((t_g^T t_e)/(\|t_g\|\|t_e\|))$,
- $\xi_f = |f_e - f_g|/f_g$,

where R_g, t_g, f_g represent the ground-truth rotation, translation and focal length, and R_e, t_e, f_e are the corresponding estimated rotation, translation and focal length, respectively. We measure the angular error between the estimated translation direction and the true direction since the estimated translation is up to scale. For the different and unknown focal lengths problems, we compute the geometric mean of the focal length errors $\xi_f = \sqrt{\xi_{f_1}\xi_{f_2}}$. This measurement has been widely used in camera pose estimations [28, 4, 21, 29, 24, 9]. In this experiment, 4pt-polyeig and 4pt-action denote the 4-point polynomial eigenvalue solution and 4-point action matrix solution, 5pt-polyeig and 5pt-action denote the proposed 5-point polynomial eigenvalue solution and 5-point action matrix solution, respectively. To save space we only show the errors in the focal length for stability evaluation since the other errors are qualitatively similar. Figure 2 shows the kernel smoothed histograms of the focal length errors under sideways motion (left) and forward motion (right) for both the shared and unknown focal length and different and unknown focal lengths problems. As we can see, all the proposed algorithms are numerically stable and do not contain large errors. The polynomial eigenvalue solutions perform slightly better than the action matrix solutions, so we use them in our next experiments.

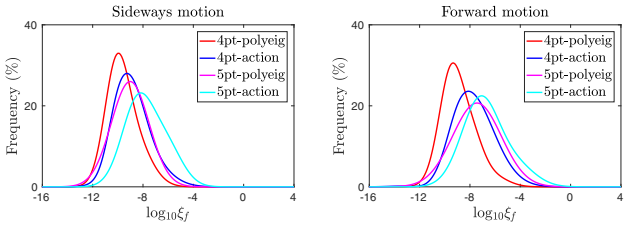

Figure 2: Kernel smoothed histograms of the focal length errors for 10,000 runs of the proposed algorithms with noise-free data. The polynomial eigenvalue solution is slightly better than the action matrix solution.

6.2. Pure rotation

In this section, we show that the proposed solvers are compatible with the pure rotation case without knowing the prior knowledge of the motion. Figure 3 reports the focal length and rotation errors of our polynomial eigenvalue solvers with noise-free data under pure rotation. It seems that the precision is not as good as the sideways motion or forward motion, but they are good enough for real applications. On the other hand, the translation may not be exactly equal to zero in practice, so the performance is acceptable.

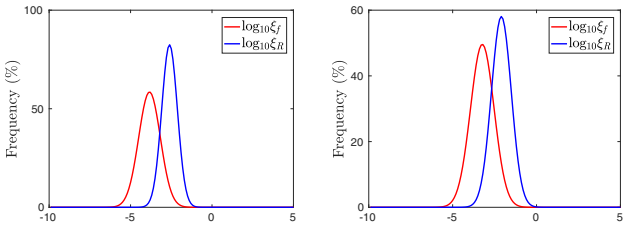

Figure 3: Kernel smoothed histograms of the errors for 10,000 runs with noise-free data under pure rotation. *Left*: Shared and unknown focal length. *Right*: Different and unknown focal lengths.

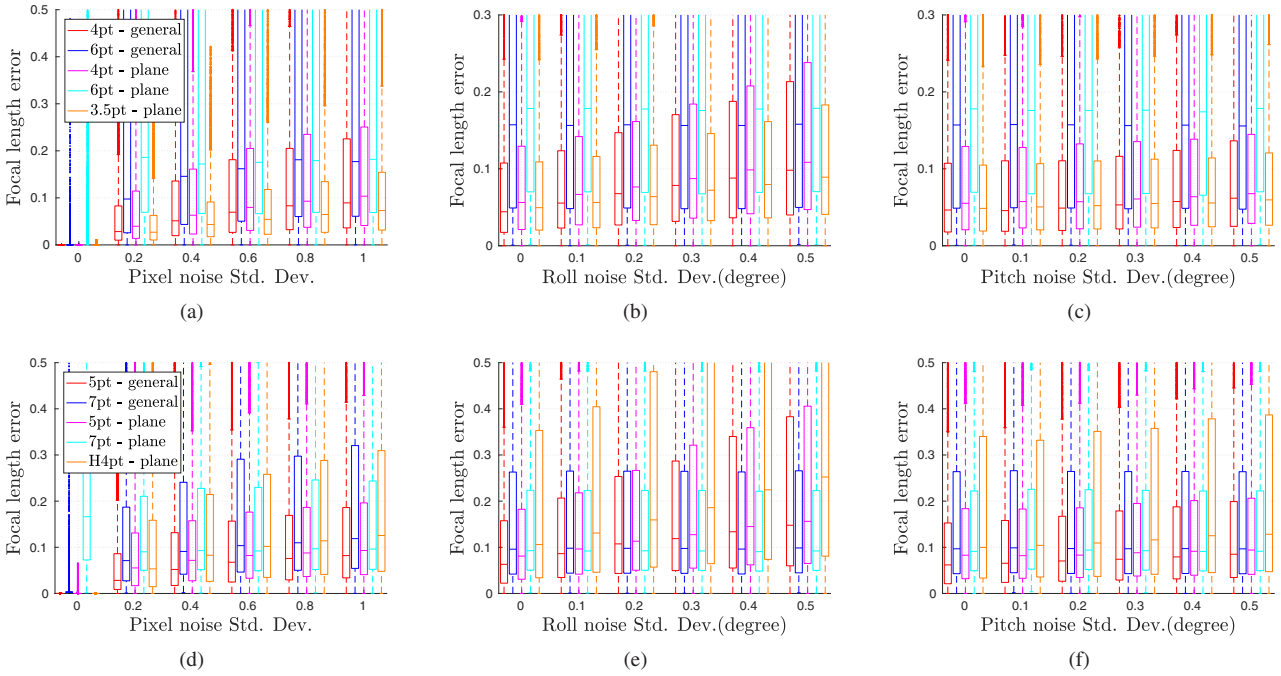

Figure 4: Performance under forward motion with different noisy conditions. **Top row**: Shared and unknown focal length. **Bottom row**: Different and unknown focal lengths. From left to right: (a)(d) Increased image noise; (b)(e) Increased roll noise with constant image noise of 0.5 pixel standard deviation; (c)(f) Increased pitch noise with constant image noise of 0.5 pixel standard deviation.

6.3. Noise resilience

To evaluate the sensitivity to image noise, we add Gaussian noise with standard deviation ranging from 0 to 1 pixels to the image points. In addition, the gravity direction measured by the IMU readings is not perfect in real applications. Errors might be introduced by dynamics in the accelerometer readings and rotation alignment of the camera and IMU. Since the angular accuracy of roll and pith angle in low cost IMUs is about 0.5°, and is less than 0.02° in high accuracy IMUs [20]. We also simulate the noisy case where the (roll, pitch) noise ranging from 0 to 0.5 degrees standard deviation with constant image noise of 0.5 pixel standard deviation. To compare with the standard algorithms fairly, we use (17) to compute the error of the full rotation and translation for our algorithms.

In this experiment, 4pt and 5pt denote the 4-point and 5-point polynomial eigenvalue solutions, respectively. 6pt denotes the state-of-the-art 6-point algorithm proposed in [22], 7pt denotes the 7-point algorithm which extracts two different focal lengths from the fundamental matrix using Bougnoux formula [3], 3.5pt and H4pt denote the state-of-the-art homography-based algorithms for the shared and unknown and varying focal lengths problems proposed in [9], respectively. The suffixes -general and -plane denote that the methods are evaluated with general and planar scenes, respectively. To save space, we only show the results of the focal length errors under forward motion since other errors are qualitatively similar. Results for the rotation and translation errors and sideways motion are given in the supplemental material.

Shared and unknown focal length

Figure 4(a) shows the boxplot of the focal length errors with increased image noise under forward motion for the shared and unknown focal length problem. For perfect planar scenes without image noise, the standard 6-point algorithm fails since it is a degenerate case. Our 4-point algorithm performs better than the 6-point algorithm under different levels of image noise for both general and planar scenes. The 3.5-point homography-based algorithm performs best with planar scenes, but it needs to assume that points lie on a plane. Figure 4(b) and Figure 4(c) show the focal length errors under increased roll and pitch noise, respectively. When the roll or pitch noise is up to 0.5 degrees, our 4-point algorithm is still better than the 6-point algorithm, and comparable to the homography-based algorithm with planar scenes.

Different and unknown focal lengths

Figure 4(d) shows the boxplot of the focal length errors with increased image noise under forward motion for the different and unknown focal lengths problem. Our 5-point

Algorithm	SVD	QR	Eigen	Time (us)	Max No. of real solutions	Iterations with outliers ($1-w$) 0.30	0.50	0.70	0.90
4pt-polyeig	-	6×6	24×24	110	20	17	72	566	4.6×10^4
3.5pt [9]	8×9	152×152	24×24	1200	20	17	72	566	4.6×10^4
6pt [22]	6×9	21×21	15×15	72	15	37	292	6315	4.6×10^6
5pt-polyeig	-	12×12	48×48	310	24	25	145	1893	4.6×10^5
H4pt [9]	8×9	33×33	8×8	50	8	17	72	566	4.6×10^4
7pt [3]	7×9	-	-	11	3	54	587	21055	4.6×10^7

Table 1: Efficiency comparison of the proposed algorithms (gray) vs. state of the art.

algorithm performs better than the 7-point algorithm under different levels of image noise for both general and planar scenes. It also has advantage over the 4-point homography-based algorithm with planar scenes. Figure 4(e) and Figure 4(f) show the focal length errors under increased roll and pitch noise, respectively. It seems that our 5-point algorithm is slightly sensitive to roll noise. When the pitch noise is up to 0.5 degrees, our 5-point algorithm is still better than the 7-point algorithm and the 4-point homography-based algorithm. In general, with accurate gravity information the proposed methods are better than the standard 6- and 7-point algorithms, and can cope with coplanar points. Note that the 6- and 7-point algorithms are general methods, while ours need gravity information, which is largely motivated by its availability for smart phones, tablets.

6.4. Computational Complexity

Table 1 reports the major steps and run-time (averaged by 10,000 trials) of all the algorithms on an Intel i7-8700K 3.7GHz based desktop using Matlab. We used C++-mex implementations for all the polynomial solvers (based on Eigen linear algebra library). For the shared and unknown focal length problem, the timing of 4pt-polyeig for one hypothesis estimation is $110\mu s$. For the different and unknown focal lengths problem, the timing of 5pt-polyeig is $310\mu s$. The high run-time of the our 5-point algorithm is due to the large size of matrix for the eigenvalue computation, but it is fast enough for real applications. The fewer max number of real solutions the better, since every real solution needs to be evaluated with a set of points within RANSAC. However, since we have constraints on the focal length, many meaningless real solutions can be abandoned.

Since in practice the algorithms need to be used within RANSAC or other robust statistics to reject outliers, the number of necessary iterations is very important for efficiency consideration. We also report the theoretical RANSAC iteration number based on $\frac{log(1-p)}{log(1-w^q)}$ with $p = 0.99$, where p is the desired probability that the RANSAC provides a useful result after running, w is the percent of inliers in data and q is the number of point correspondences needed for the algorithm. With the rate of outliers increased, the proposed algorithms significantly reduce the number of iterations. We show that the improvement holds up in the real experiments as well. Although the homography based algorithms only need 4 points, they are based on the assumption that points are coplanar.

7. Real data from a smart phone

To evaluate the proposed algorithms on real data, we have recorded 8 sequences at the resolution of 1920×1080 and the IMU data with an iphone 6s. Then we synchronize the frames and IMU data based on their timestamps. The raw data contains both the gravity and the acceleration, so we need to apply a high-pass filter to isolate the force of gravity from the raw accelerometer data [8, 7]. We used every 10th image from each sequence (*i.e.*, {1,11},{2,12},{3,13},$\cdots$), since the relative translation may be very small for consecutive frames. Example images of the sequences are shown in Figure 5. We extract SIFT [26] feature points and descriptors of the images. Since we only want to give a fair comparison, we use the standard RANSAC [10] without any optimizations to estimate the focal length and relative pose. The RANSAC confidence is set to 0.99 and the maximum number of iterations is set to 5000. The distance threshold is set to 2 pixels for fundamental matrix and 2.5 pixels for homography, respectively. We use the focal length calculated from the CMOS parameters (1610 pixels) and the motion parameters obtained from RealityCapture [1] (the intrinsic parameters were fixed based on the focal length and central principal point) as the ground truth. Due to the page limitation, we only show the results of the first sequence. Results for other sequences are given in the supplemental material.

Table 2(a) shows the median and mean errors in the estimated rotation, translation and focal length for the shared and unknown focal length case. As we can see, the proposed 4-point algorithm outperforms the standard algorithms. We also report the mean number of iterations and inliers. For

		3.5pt [9]	6pt [22]	4pt
Rotation error in degree	median	0.5311	0.8607	**0.1465**
	mean	3.2840	1.0708	**0.2224**
Translation error in degree	median	5.0219	3.9352	**1.2284**
	mean	10.2042	5.9041	**1.7173**
Focal length error (%)	median	24.72	38.03	**5.21**
	mean	42.02	41.15	**11.31**
Number of iterations	mean	5000+12	1423	**237**
Number of inliers	mean	156	**733**	730

(a)

H4pt [9]	7pt [3]	5pt
1.1858	3.4309	**0.3793**
2.8567	4.0895	**0.5607**
9.5434	7.7842	**2.8436**
16.8343	13.5088	**4.8110**
52.21	44.18	**26.11**
57.47	43.35	**31.37**
5000+11	2474 (85)	**452 (180)**
157	696	**707**

(b)

Table 2: Comparison of different algorithms on the first sequence (1162 images) under forward motion. (a) Shared and unknown focal length. (b) Different and unknown focal lengths. The best results are marked bold. See text for details.

the homography-based method, we first use the standard 4-point homography algorithm to reject outliers as suggested by [9]. Then we use the 3.5-point homography-based solver to find the best solution within the inliers. Since the scene is general and does not contain a dominant plane, RANSAC for the standard 4-point homography algorithm always reaches the maximum (5000). However, the 3.5-point homography-based algorithm only need 12 iterations using the inliers. Compared to the 6-point algorithm, our 4-point algorithm needs many fewer iterations: 237 vs 1423, and the number of inliers are almost the same. Our 4-point algorithm has three fewer inliers on the average, which might be due to the noise in the IMU readings.

Table 2(b) shows the results for the different and unknown focal lengths problem. In this case, the proposed algorithm still performs better than the 7-point algorithm and the homography-based 4-point algorithm. However, we find that the focal length error is large without the equal constraint on the focal length. It is possibly due to the motion, which used to capture the sequence, is close to the degenerate configuration (see Sec. 5) and the translation is much smaller than the scene depth. Note that, for the 7-point algorithm, sometimes the inliers and the fundamental matrix seem to be correct, but the extracted focal lengths are invalid. So we also report the number of samples yield physically possible focal length (positive and larger than one fifth of the image width). Compared to the 7-point algorithm, our 5-point algorithm needs many fewer iterations: 452 vs 2474. Among the samples, 180 of ours have physically possible focal length, while there are only 85 for the 7-point algorithm. On the other hand, our 5-point algorithm has more inliers than the 7-point algorithm. It is because that sometimes the fundamental matrix has good inliers, but the focal length is meaningless. In this case, such solutions should be abandoned. In short, the proposed algorithms are more accurate and need fewer iterations, but recall of course that the standard algorithms are general methods, while the proposed algorithms use the gravity direction, which is widely available for smart devices.

Figure 5: Example images of the 8 sequences recorded with an iphone 6s.

8. Conclusion

In this paper we propose minimal solutions to estimate relative pose for the case of sharing a common direction with unknown focal length. It is a practically relevant case for smart devices which can provide the gravity direction using IMU readings. We have discussed both the shared and unknown, and different and known focal lengths cases. The synthetic evaluation and the real experiments with smart phone show that the proposed algorithms are stable enough for real applications. We believe that the proposed algorithms are promising, since it is conceivable that cameras will always be coupled with IMU's in the future.

Acknowledgments. The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by the National Science Fund of China under Grant No. U1713208, "111" Program B13022, the Inria/NYU collaboration and the Louis Vuitton/ENS chair on artificial intelligence. In addition, this work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

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

[2] 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. 3

[3] Sylvain Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *The IEEE International Conference on Computer Vision (ICCV)*, 1998. 1, 6, 7, 8

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

[5] Martin Byröd, Klas Josephson, and Kalle Åström. Fast and stable polynomial equation solving and its application to computer vision. *International Journal of Computer Vision*, 2009. 3

[6] David A Cox, John Little, and Donal O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006. 3

[7] Android developer. http://developer.android.com. 7

[8] IOS developer. https://developer.apple.com. 7

[9] 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 *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 1, 2, 5, 6, 7, 8

[10] 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, 4, 7

[11] Andrew Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1. 3

[12] 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 *The European Conference on Computer Vision (ECCV)*, 2010. 1, 4, 5

[13] Daniel R Grayson and Michael E Stillman. Macaulay 2, a software system for research in algebraic geometry, 2002. 3, 4

[14] 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

[15] Richard Hartley. Estimation of relative camera positions for uncalibrated cameras. In *The European Conference on Computer Vision (ECCV)*, 1992. 1

[16] Richard Hartley and Hongdong Li. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2012. 1, 2

[17] Janne Heikkila. Using sparse elimination for solving minimal problems in computer vision. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[18] Fredrik Kahl and Bill Triggs. Critical motions in euclidean structure from motion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999. 2, 4, 5

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

[20] 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*, 2010. 6

[21] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to minimal problems in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 1, 3, 5

[22] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 6, 7, 8

[23] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2017. 3

[24] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Camera pose estimation with unknown principal point. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5

[25] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla. Beyond gröbner bases: Basis selection for minimal solvers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 4

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 7

[27] Oleg Naroditsky, Xun S Zhou, Jean Gallier, Stergios I Roumeliotis, and Kostas Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 2012. 1

[28] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 2004. 1, 5

[29] Olivier Saurer, Pascal Vasseur, Rémi Boutteau, Cédric Demonceaux, Marc Pollefeys, and Friedrich Fraundorfer. Homography based egomotion estimation with a common direction. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 1, 5

[30] Henrik Stewénius, David Nistér, Fredrik Kahl, and Frederik Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 1

[31] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. *3DV*, 2014. 1