# A Robust Least Squares Solution to the Calibrated Two-view Geometry with Two Known Orientation Angles

Gaku Nakano and Jun Takada

Information and Media Processing Laboratories, NEC Corporation,
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, Japan
g-nakano@cq.jp.nec.com, j-takada@bc.jp.nec.com

**Abstract.** This paper proposes a robust least squares solution to the calibrated two-view geometry with two known orientation angles. Using the knowledge reduces the degrees of freedom (DoF) from five to three: one from a remaining angle and two from a translation vector. This paper determines that the three parameters are obtained by solving a minimization problem of the smallest eigenvalue containing the unknown angle. The proposed solution minimizes a new simple cost function based on the matrix determinant in order to avoid the complicated eigenvalue computation. The estimated parameters are optimal since the cost function is minimized under three DoFs. Experimental results of synthetic data show that the robustness of the proposed solution is up to 1.5° angle noise, which is approximately three times that of a conventional solution. Moreover, 60 point correspondences, fewer than half those in conventional solutions, are sufficient to reach the performance boundary.

**Keywords:** Two-view Geometry, Relative Pose Problem, Essential Matrix, Structure from Motion, Two Known Orientation Angles

## 1 Introduction

The calibrated two-view geometry is an estimation problem of the relative pose between two cameras capturing the same scene from different positions. It is the most basic theory for an image based 3D reconstruction. ”Calibrated” means that the intrinsic camera parameters, e.g., the focal length, are assumed to be known.

The relative pose is generally expressed by five parameters, i.e., three orientation angles and a 3D translation vector up to scale. The absolute scale factor cannot be estimated without any prior knowledge about the scene. One point correspondence in the two images gives one constraint between the correspondence and the relative pose. Therefore, the calibrated two-view geometry is solved by at least five point correspondences. Many solutions based on point correspondences have been proposed, which are called the 5-point [1–7], 6-point [8], 7-point [9] and 8-point [9] algorithms.

The final published version is available at https://rd.springer.com/chapter/10.1007/978-3-662-44911-0_9.
DOI: 10.1007/978-3-662-44911-0_9

Meanwhile, a restricted relative pose problem has been raised in which two orientation angles are known. Two orientation angles are obtained by an internal measurement unit (IMU) sensor or a vanishing point. Using the known angles brings two great benefits. The one is that an angle measured by high accurate sensors is more reliable than that obtained by the point correspondence based algorithms. The other is that the relative pose problem becomes simpler since the total degrees of freedom (DoF) is reduced from five to three. Therefore, the relative pose problem is solved by at least three point correspondences. This reduces the computational cost of the pose estimation and also reduces the number of iterations of RANSAC [10].

Actual IMU sensors in many consumer products do not have the high accuracy needed in those solutions due to noise caused by camera shake and temperature change. Therefore, pragmatic solutions to the restricted pose problem must provide robustness to not only image noise but also sensor noise.

Although some solutions are proposed for using two known orientation angles, they are neither robust nor able to estimate the optimal pose. Kalantari et al. proposed a solution to the 3-point minimal case [11]. They formulate the problem as a system of multivariate polynomial equations and solve it by using a Gröbner basis method. Since the Gröbner basis method for that formulation requires large matrix decompositions, Kalantari et al.'s solution is difficult to extend to the least squares case in which the degree of polynomial equations becomes higher and the size of matrices becomes much larger. Fraundorfer et al. proposed three kind of solutions [12]. One is for the minimal case, and the others are for the least squares case of four and more than five point correspondences, respectively. Fraundorfer et al. show that the minimal solution is efficient for a RANSAC scheme. However, the two least squares solutions are neither optimal nor robust to noise because they do not exactly satisfy the nonlinear constraints that express three DoFs.

This paper proposes a robust least squares solution to the calibrated two-view geometry with two known orientation angles. The problem is formulated as a minimization problem of the smallest eigenvalue of a $3 \times 3$ matrix containing the unknown angle. The proposed solution minimizes a new simple cost function based on the matrix determinant in order to avoid the complicated eigenvalue computation. The unknown angle and translation vector are obtained as the root of an eighth degree univariate polynomial and the eigenvector corresponding to the smallest eigenvalue, respectively. Since the cost function is minimized under three DoFs, the proposed solution is optimal and robust to noisy data.

## 2 Problem Statement

This section describes the calibrated two-view geometry with two known orientation angles. Figure 1 shows an example such that the two orientation angles are obtained by the gravity direction $\boldsymbol{g}$.

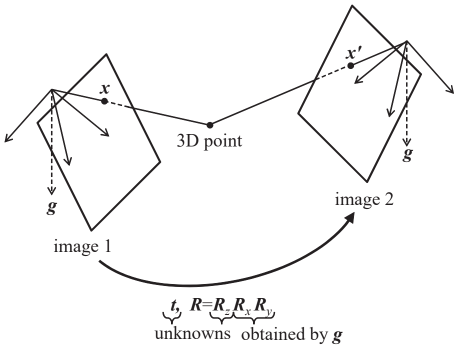

$\boldsymbol{x}$
$\boldsymbol{x}'$
3D point
$\boldsymbol{g}$
$\boldsymbol{g}$
image 1
image 2
$\underbrace{\boldsymbol{t},}_{\text{unknowns}}\ \boldsymbol{R}=\underbrace{\boldsymbol{R}_z}\underbrace{\boldsymbol{R}_x \boldsymbol{R}_y}_{\text{obtained by } \boldsymbol{g}}$

**Fig. 1.** Calibrated two-view geometry with two known orientation angles.

Let $\boldsymbol{x}$ and $\boldsymbol{x}'$ be point correspondences represented by 3D homogeneous coordinates in images 1 and 2, respectively. Then, the calibrated two-view geometry is written in the form

$$\boldsymbol{x}'^T [\boldsymbol{t}]_\times \boldsymbol{R}_z \boldsymbol{R}_y \boldsymbol{R}_x \boldsymbol{x} = 0. \tag{1}$$

where $\boldsymbol{t} = [t_x, t_y, t_z]^T$ denotes a 3D translation vector up to scale, $[\ ]_\times$ denotes a $3 \times 3$ skew symmetric matrix representation of the vector cross product, and $\boldsymbol{R}_x$, $\boldsymbol{R}_y$, and $\boldsymbol{R}_z$ are $3 \times 3$ rotation matrices around x, y, and z-axis, respectively. Equation (1) has five DoFs (two from $\boldsymbol{t}$ and three from $\boldsymbol{R}_x$, $\boldsymbol{R}_y$ and $\boldsymbol{R}_z$.)

Let $\phi$, $\psi$, and $\theta$ be the orientation angles around x, y, and z-axis, respectively. $\boldsymbol{R}_x$, $\boldsymbol{R}_y$, and $\boldsymbol{R}_z$ are expressed as

$$\boldsymbol{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \tag{2}$$

$$\boldsymbol{R}_y = \begin{bmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{bmatrix}, \tag{3}$$

$$\boldsymbol{R}_z = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

If an IMU sensor is embedded in the cameras or if a vanishing point is detected in the images, the two orientation angles around x- and y-axis, i.e., $\phi$ and $\psi$, are known. Since $\boldsymbol{R}_x$ and $\boldsymbol{R}_y$ are given by (2) and (3), $\boldsymbol{R}_y \boldsymbol{R}_x \boldsymbol{x}$ can be simply expressed by $\boldsymbol{x}$. Then, we have

$$\boldsymbol{x}'^T [\boldsymbol{t}]_\times \boldsymbol{R}_z \boldsymbol{x} = 0. \tag{5}$$

Equation (5) represents the relative pose problem with two known orientation angles. The total DoF of (5) is reduced to $5 - 2 = 3$.

Replacing $[\boldsymbol{t}]_\times \boldsymbol{R}_z$ by a $3 \times 3$ matrix $\boldsymbol{E}$, (5) can be written in the linear form

$$\boldsymbol{x}'^T \boldsymbol{E} \boldsymbol{x} = 0. \tag{6}$$

Here, $E_{1,1} = E_{2,2}$, $E_{1,2} = -E_{2,1}$ and $E_{3,3} = 0$. $E_{i,j}$ is the element of $\boldsymbol{E}$ at the $i$–th row and the $j$–th column.

$\boldsymbol{E}$ is called the essential matrix if and only if two of its singular values are nonzero and equal, and the third one is zero [13]. These constraints are expressed by

$$\det(\boldsymbol{E}) = 0, \tag{7}$$

$$\boldsymbol{E}\boldsymbol{E}^T\boldsymbol{E} - \frac{1}{2}\mathrm{trace}(\boldsymbol{E}\boldsymbol{E}^T)\boldsymbol{E} = \boldsymbol{0}_{3\times 3}. \tag{8}$$

$\boldsymbol{E}$ has six parameters. However, $\boldsymbol{E}$ has only three DoFs due to the scale ambiguity and the above constraints [12].

Solving a nonlinear equation (5) and solving a linear equation (6) with the nonlinear constraints ((7) and (8)) are identical.

## 3 Previous Work

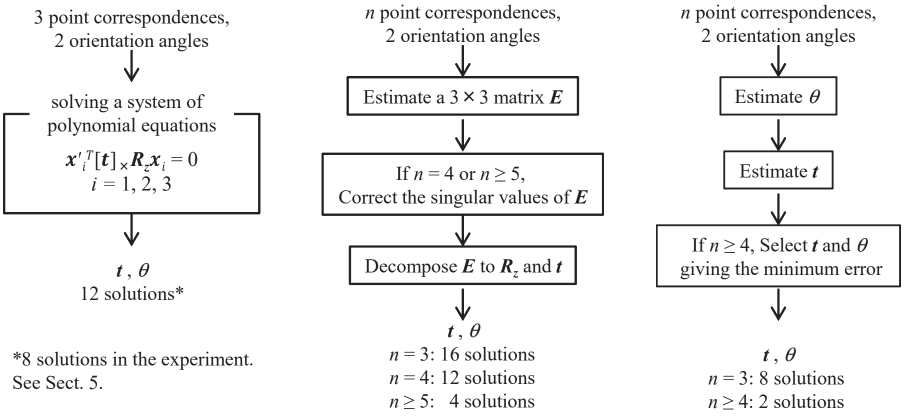
This section briefly describes the conventional solutions and points out their drawbacks. Their algorithm outlines are shown in Figs. 2(a) and 2(b).

### 3.1 Kalantari et al.'s Solution [11]

Kalantari et al. proposed an algorithm to obtain all unknowns in (5) by solving a system of multivariate polynomial equations.

First, the Weierstrass substitution is used to express $\cos\theta$ and $\sin\theta$ without the trigonometric functions: $\cos\theta = \frac{1-p^2}{1+p^2}$ and $\sin\theta = \frac{2p}{1+p^2}$, where $p = \tan\frac{\theta}{2}$. By substituting three point correspondences into (5) and by adding a new scale constraint $\|\boldsymbol{t}\| = 1$, there are four polynomial equations in four unknowns $\{t_x, t_y, t_z, p\}$ of degree three. Kalantari et al. adopt a Gröbner basis method to solve the system of polynomial equations. The solutions are obtained by Gauss-Jordan elimination of $65 \times 77$ Macaulay matrix and eigenvalue decomposition of $12 \times 12$ Action matrix. Finally, at most 12 solutions are given from the eigenvectors.

Kalantari et al.'s 3-point algorithm is difficult to extend to the least squares case in which the degree of polynomial equations becomes higher and the size of matrices becomes a few hundred dimensions.



(a) Kalantari et al. (b) Fraundorfer et al. (c) Proposed

**Fig. 2.** Outlines of the conventional and proposed solutions.

In the experiments in this paper, the size of the decomposed matrices and the number of the solutions are not same as those in Kalantari et al.'s original implementation. The details are described in Sect.5.2.

### 3.2 Fraundorfer et al.'s Solution [12]

Fraundorfer et al. estimated the essential matrix in (6) instead of $\{t_x, t_y, t_z, p\}$. The most important contribution is to propose solutions to the least squares case.

Fraundorfer et al. proposed three algorithms for the cases of three, four and more than five point correspondences. The basic idea is very similar to the point correspondence based algorithms, i.e., Nistér's 5-point [3], Hartley's 7-point [9] and Hartley's 8-point [9] algorithm.

From a set of $n$ point correspondences, (6) can be equivalently written as

$$\boldsymbol{M}\mathrm{vec}(\boldsymbol{E}) = \mathbf{0}_{n\times 1}, \tag{9}$$

where $\boldsymbol{M} = \left[\boldsymbol{x}_1 \otimes \boldsymbol{x}'_1 \cdots \boldsymbol{x}_n \otimes \boldsymbol{x}'_n\right]^T$ , vec( ) denotes the vectorization of a matrix and $\otimes$ denotes the Kronecker product.

The solution of (9) is obtained by

$$\boldsymbol{E} = \sum_{i=1}^{6-n} a_i \boldsymbol{V}_i, \tag{10}$$

where $\boldsymbol{V}_i$ is the matrix corresponding to the generators of the right nullspace of the coefficient matrix $\boldsymbol{M}$, and $a_i$ is an unknown coefficient.

Estimating $\boldsymbol{E}$ is equivalent to calculate $a_i$. One of $a_i$ can be set to one to reduce the number of unknowns due to the scale ambiguity of $\boldsymbol{E}$. In the 3-point case, (7) and (8) are used to solve two unknowns. Similarly, (7) is used to solve one unknown in the 4-point case. For more than five point correspondences, the solution is obtained by taking the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{M}^T\boldsymbol{M}$.

An essential matrix can be decomposed to two $\boldsymbol{R}_z$s and $\pm\boldsymbol{t}$ [9, 14]. Fraundorfer et al.'s 3-point, 4-point, and 5-point algorithms estimate at most four, three and one essential matrices, respectively. Therefore, they give at most 16, 12, and four solutions.

Fraundorfer et al.'s 3-point algorithm satisfies all the constraints. However, the 4-point algorithm considers only one constraint, and the 5-point algorithm ignores all constraints. For this reason, an estimated $\boldsymbol{E}$ of the 4-point and 5-point algorithms may not be an essential matrix. To correct the estimated $\boldsymbol{E}$ to an essential matrix, constraints are enforced by replacing the singular values of $\boldsymbol{E}$ so that two are nonzero and equal, and the third one is zero. The enforcement does not guarantee to optimize $\theta$ and $\boldsymbol{t}$ that minimize (6), but optimizes the minimum change of the Frobenius norm. The 4-point and the 5-point algorithm do not minimize the residual (6) under three DoFs. Therefore, they are not optimal solutions.

## 4 Proposed Solution

This section first describes the basic idea of the proposed solution in the minimal case, and then, explains how to extend the idea to the least squares case. The algorithm outline is shown in Fig. 2(c).

### 4.1 3-point Algorithm for the Minimal Case

Equation (5) can be equivalently written as

$$\boldsymbol{v}^T\boldsymbol{t} = 0, \tag{11}$$

where $\boldsymbol{v} = [\boldsymbol{x}']_\times^T \boldsymbol{R}_z\boldsymbol{x}$.

Given three point correspondences, we have

$$\boldsymbol{A}\boldsymbol{t} = \boldsymbol{0}_{3\times1}, \tag{12}$$

where $\boldsymbol{A} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]^T$ is a $3 \times 3$ matrix containing the unknown $\theta$.

Since $\boldsymbol{t}$ must not be the trivial solution $\boldsymbol{t} = \boldsymbol{0}_{3\times1}$, (12) shows that $\boldsymbol{A}$ is singular and $\boldsymbol{t}$ is the nullspace of $\boldsymbol{A}$. Consequently, $\theta$ is the solution of $\det(\boldsymbol{A}) = 0$.

In the proposed 3-point algorithm, $\cos\theta$ and $\sin\theta$ are replaced by new unknowns $c$ and $s$, respectively, instead of using the Weierstrass substitution. The reason is that the Weierstrass substitution changes the range of values from $-\pi \leq \theta \leq +\pi$ to $-\infty < p < +\infty$. This may cause computational instability.

Furthermore, a symbolic fractional calculation complicates polynomial equations in the least squares case.

The unknowns $c$ and $s$ are obtained by solving the following system of polynomial equations:

$$\begin{cases} f_1(c,s) = \det(\boldsymbol{A}) = 0, \\ g(c,s) = c^2 + s^2 - 1 = 0. \end{cases} \tag{13}$$

Equation (13) can be solved by the resultant based method, which is also known as the hidden variable method [15]. Let $f_1$ and $g$ be polynomial equations of $s$ and $c$ be regarded as a constant, and the resultant $\mathrm{Res}(f_1, g, c) = 0$ is a fourth degree univariate polynomial in $c$. We obtain at most four solutions as the real roots of $\mathrm{Res}(f_1, g, c) = 0$.

As a result, $\theta$ is obtained by

$$\theta = \mathrm{atan2}(s, c). \tag{14}$$

Subsisting estimated $\theta$ into (12), $\boldsymbol{t}$ is obtained by the cross product of two arbitrary rows of $\boldsymbol{A}$. The largest of these three cross products should be chosen for numerical stability [14].

If $\boldsymbol{v}_i \times \boldsymbol{v}_j$ is the largest, we obtain $\boldsymbol{t}$ up to scale,

$$\boldsymbol{t} = \pm \frac{\boldsymbol{v}_i \times \boldsymbol{v}_j}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|}. \tag{15}$$

The proposed 3-point algorithm gives at most eight possible combinations of four $\theta$s and $\pm\boldsymbol{t}$.

### 4.2 4-point Algorithm for the Least Squares Case

This section describes how to extend the proposed 3-point algorithm to the least squares case.

Given more than four point correspondences, the pose estimation problem is expressed by an optimization problem:

$$\begin{aligned} \underset{\boldsymbol{t},\theta}{\text{minimize}} \quad & \|\boldsymbol{Bt}\|^2 \\ \text{subject to} \quad & \|\boldsymbol{t}\| = 1 \end{aligned} \tag{16}$$

where $\boldsymbol{B} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n]^T$ is an $n \times 3$ matrix containing the unknown $\theta$, and $\|\boldsymbol{t}\| = 1$ is a constraint to avoid the trivial solution $\boldsymbol{t} = \boldsymbol{0}_{3\times1}$.

As known in Hartley's 8-point algorithm [9], the optimal $\boldsymbol{t}$ is the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{B}^T\boldsymbol{B}$, and the minimum error in the cost function $\|\boldsymbol{Bt}\|^2$ is equal to the smallest eigenvalue of $\boldsymbol{B}^T\boldsymbol{B}$. The optimization problem (16) is essentially identical to the eigenvalue problem. However, the smallest eigenvalue of $\boldsymbol{B}^T\boldsymbol{B}$ represented by $\theta$ and imaginary numbers is difficult to compute directly.

To avoid the eigenvalue computation, this paper proposes a new cost function, $\det(\boldsymbol{B}^T\boldsymbol{B})$. The determinant of a square matrix is equal to the product of all it's eigenvalues, and $\boldsymbol{B}^T\boldsymbol{B}$ is positive-semidefinite. Therefore, $\|\boldsymbol{Bt}\|^2$ is expected to be minimized if $\det(\boldsymbol{B}^T\boldsymbol{B})$ is minimized. The proposed 4-point algorithm minimizes $\det(\boldsymbol{B}^T\boldsymbol{B})$ instead of $\|\boldsymbol{Bt}\|^2$.

Similar to the proposed 3-point algorithm, $\theta$ is obtained by solving the following polynomial system of equations:

$$\begin{cases} f_2(c,s) = \dfrac{d}{d\theta}\det(\boldsymbol{B}^T\boldsymbol{B})\Big|_{\substack{\cos\theta=c,\\ \sin\theta=s}} = 0, \\ g(c,s) = c^2+s^2-1 = 0. \end{cases} \tag{17}$$

Here, $\dfrac{d}{d\theta}\det(\boldsymbol{B}^T\boldsymbol{B})\Big|_{\substack{\cos\theta=c,\\ \sin\theta=s}}$ denotes that $\cos\theta$ and $\sin\theta$ in $\dfrac{d}{d\theta}\det(\boldsymbol{B}^T\boldsymbol{B})$ are replaced by $c$ and $s$, respectively.

The resultant $\mathrm{Res}(f_2, g, c) = 0$ is an eighth degree univariate polynomial in $c$. We select the optimal $\theta$ from the real roots so that it minimizes $\det(\boldsymbol{B}^T\boldsymbol{B})$ or the smallest eigenvalue of $\boldsymbol{B}^T\boldsymbol{B}$.

Finally, we obtain the optimal $\boldsymbol{t}$ by taking the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{B}^T\boldsymbol{B}$. The proposed 4-point algorithm gives at most two possible combinations of one $\theta$ and $\pm\boldsymbol{t}$.

Moreover, the proposed 4-point algorithm includes the solutions of the proposed 3-point algorithm. For this reason, the proposed 4-point algorithm is a true extension of the 3-point algorithm. The proof is described in Appendix.

## 5 Experiments

This section evaluates the performance of the proposed solutions on synthetic data. The proposed solutions were compared with Kalantari et al.'s and Fraundorfer et al.'s solutions as well as Nistér's 5-point and Hartley's 8-point algorithms, which are standard methods for using only point correspondences. All program codes were written in MATLAB 2012b and implemented by the authors of this paper except for Nistér's 5-point algorithm[1]. Kukelova's automatic generator of Gröbner basis solvers [16][2] was used to implement the conventional 3-point algorithms. Kalantari et al.'s 3-point algorithm in the experiments computed 58 × 66 Macaulay matrix and 8 × 8 Action matrix due to the difference in the definition of the unknown orientation angle[3]. The simulations were performed on a windows 7 SP1 with a Core i7-3770 processor.

### 5.1 Synthetic Data

The robustness of the proposed solutions was evaluated under various image and angle noises on synthetic data.

[1] `http://www.vis.uky.edu/~stewe/FIVEPOINT/`

[2] `http://cmp.felk.cvut.cz/minimal/automatic_generator.php`

[3] The original derivation of Kalantari et al. [16] assumes that the unknown orientation angle is around y-axis, not z-axis as in this paper.

3D points were generated randomly similar to Fraundorfer et al. [12] so that the 3D points have a depth of 50% of the distance of the first camera to the scene. In the work of Fraundorfer et al. [12], two camera configurations are performed, i.e., sideway and forward motion with random rotation. To simulate a more realistic environment, random motion with random rotation was performed in these experiments. The baseline between the two cameras was 10% of the distance to the scene.

Gaussian noise was added to two known orientation angles and the image points, which are projections of the 3D points onto the cameras. For an image noise test, the standard deviation of Gaussian noise was changed $0 \leq \sigma_{image} \leq 3$ pixel for the image points and fixed $\sigma_{angle} = 0.5^\circ$ for the angles. Similarly, for an angle noise test, the standard deviation of Gaussian noise was changed $0^\circ \leq \sigma_{angle} \leq 3^\circ$ for the angles and fixed $\sigma_{image} = 0.5$ pixel for the point correspondences.

Kalantari et al. and Fraundorfer et al. assume that the error in the two orientation angles measured by a low cost sensor is from 0.5° to at most 1.0°. However, information of the accuracy of almost all low cost sensors is not published. Some may have noise larger than 1.0°. Therefore, the error was assumed to be at most 3.0° in the experiments.

The estimation errors in $\theta$ and $\boldsymbol{t}$ were evaluated as follows:

$$Error(\theta_{est}, \theta_{true}) = \mathrm{abs}(\theta_{est} - \theta_{true}), \tag{18}$$

$$Error(\boldsymbol{t}_{est}, \boldsymbol{t}_{true}) = \cos^{-1}\left(\frac{\boldsymbol{t}_{est}^T \boldsymbol{t}_{true}}{\|\boldsymbol{t}_{est}\| \, \|\boldsymbol{t}_{true}\|}\right), \tag{19}$$

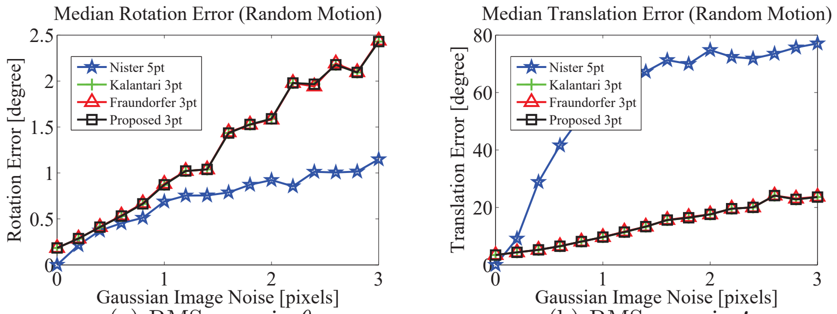
where the subscript *est* and *true* denote the estimated and the ground truth value, respectively. If an algorithm found multiple solutions, the one that had the minimum error was selected. The root mean square (RMS) errors in degrees are plotted over in 500 independent trials for each noise level in the result figures.

### 5.2 Estimation Error for the Minimal Case

The robustness of the proposed 3-point algorithm was evaluated in the minimal case and compared with the two conventional 3-point algorithms and Nistér's 5-point algorithm.

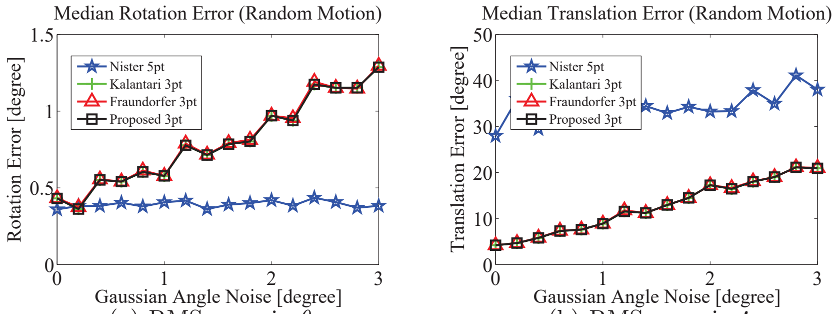
Figs. 3 and 4 indicate that all the 3-point algorithms have almost the same performance. There is no difference between the DoFs of each algorithm. Thus, they solve the mathematically identical problems and estimate almost the same values. Nistér's algorithm is not influenced by the angle noise in Fig. 4 because it uses only point correspondences.

Nistér's algorithm is slightly better for estimating the rotation than the other methods. The difference is only approximately 1.0°. On the other hand, the 3-point algorithms estimate translation vectors much better than Nistér's algorithm. Hence, the 3-point algorithms are better to use if the two angles are known.



(a) RMS error in $\theta$ (b) RMS error in $\boldsymbol{t}$

**Fig. 3.** Results of the minimal case with variable image noise ($0 \leq \sigma_{image} \leq 3$ pixel) and fixed angle noise ($\sigma_{angle} = 0.5°$.)



(a) RMS error in $\theta$ (b) RMS error in $\boldsymbol{t}$

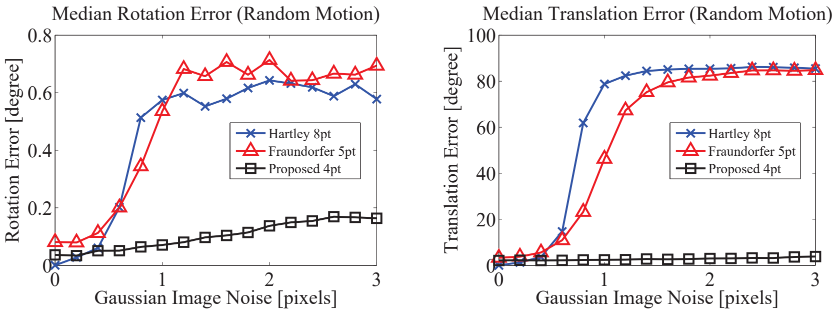
**Fig. 4.** Results of the minimal case with fixed image noise ($\sigma_{image} = 0.5$ pixel) and variable angle noise ($0° \leq \sigma_{angle} \leq 3°$.)

### 5.3 Estimation Error for the Least Squares Case

The robustness of the proposed 4-point algorithm was evaluated in the least squares case and compared with Fraundorfer et al.'s 5-point algorithm and Hartley's 8-point algorithm. The number of point correspondences was 100 in this experiment.

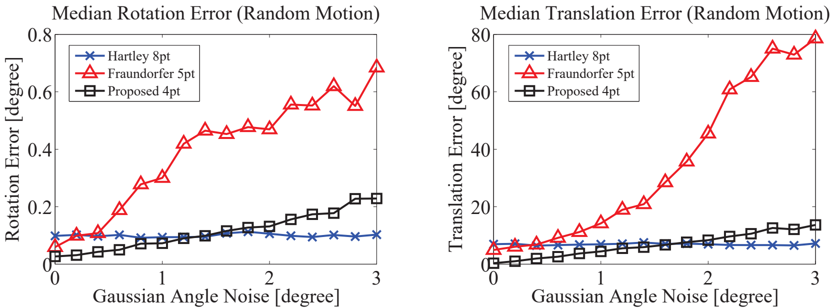
Figure 5 shows that the proposed algorithm is more stable than the others if the image noise is raised. From 0.4 to 1.0 pixel noises, the estimation errors in the algorithms of Fraundorfer et al. and Hartley increase considerably. There is no significant difference between them in a high level image noise. However, the proposed algorithm is robust in such cases.

Figure 6 shows that the proposed algorithm is more robust than that of Fraundorfer et al. in a practical scene. Their algorithm is much more sensitive to the angle noise and less accurate than Hartley's algorithm, which is not influenced by the angle noise, similarly to Nistér's algorithm. Fraundorfer et al.'s algorithm is effective for only less than 0.4° angle noise. On the other hand, the proposed algorithm is robust against to the angle noise up to 1.5°. This is



(a) RMS error in $\theta$ (b) RMS error in $\boldsymbol{t}$

**Fig. 5.** Results of the least squares case with variable image noise ($0 \leq \sigma_{image} \leq 3$ pixel) and fixed angle noise ($\sigma_{angle} = 0.5°$.)



(a) RMS error in $\theta$ (b) RMS error in $\boldsymbol{t}$

**Fig. 6.** Results of the least squares case with fixed image noise ($\sigma_{image} = 0.5$ pixel) and variable angle noise ($0° \leq \sigma_{angle} \leq 3°$.)

3 times as much as Fraundorfer et al.'s algorithm. Additionally, the estimation errors in the proposed algorithm rise gradually.

### 5.4 Estimation Error for the Number of Point Correspondences

The influence of changing the number of point correspondences was evaluated in the least squares case. The image and angle noise were fixed $\sigma_{image} = 0.5$ pixel and $\sigma_{angle} = 0.5°$, respectively. From four to 200 point correspondences were evaluated.

As shown in Fig. 7, the proposed algorithm outperforms the others regardless of the number of the point correspondences. It is notable that the proposed algorithm reaches the performance boundary at 60 point correspondences, whereas the conventional algorithms needs more than 100 point correspondences. This is very important for practical use since a few dozen point correspondences are generally obtained. Moreover, for more than 40 point correspondences, Fraundorfer et al.'s algorithm is worse than Hartley's algorithm, which uses only point correspondences. This result indicates that two known angles are not useful for Fraundorfer et al.'s algorithm in the case of many point correspondences.

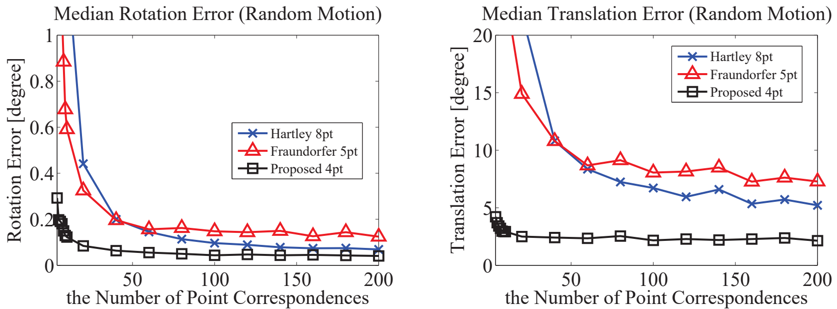
### 5.5 Computation Time

The comparison of the computation time is summarized in Table 1. Note that the all computation times were measured in MATLAB 2012b.

The proposed 4-point algorithm is slightly slower than Fraundorfer et al.'s 5-point algorithm since the proposed algorithm solves an eighth degree polynomial by using `roots` command. The increase in the number of point correspondences seems to have no influence. According to the analysis of MATLAB profiler, most of the computation time was spent running `svd` command in Fraundorfer et al.'s 5-point algorithm and `roots` command and calculating the coefficients of the polynomial in the proposed 4-point algorithm. The increase in the number of point correspondences is not significant since matrix multiplication is well optimized in MATLAB. The conventional 3-point algorithms take much longer computation time to perform Gauss-Jordan elimination, `rref` command in MATLAB. However, Kalantari et al. [11] report that the total time of their C++ implementation is 0.002 milliseconds on a laptop PC with a 1.6 GHz processor. Although they do not refer to the trade name of the CPU and the code optimization, they suggest that writing C++ improves the efficiency of the proposed algorithms dramatically. Generating point correspondences (point detection and matching), is one of the most time-consuming processes in practical 3D reconstruction. It takes more than 10 milliseconds to detect points from a VGA image even if GPU implementation [17, 18]. For this reason, the computation time of the proposed solutions would be negligible.

## 6 Conclusion

This paper has proposed a robust least squares solution to the calibrated two-view geometry with two known orientation angles. Using the knowledge reduces the DoFs from five to three: one from a remaining angle and two from a translation vector. This paper had determined that the three parameters are obtained by solving a minimization problem of the smallest eigenvalue containing the unknown angle. The proposed solution minimizes a new simple cost function based on the matrix determinant in order to avoid the complicated eigenvalue computation. The estimated parameters are optimal since the cost function is minimized under three DoFs. Experimental results for synthetic data showed that the robustness of the proposed solution is up to 1.5° angle noise, which is approximately three times that of a conventional solution. Moreover, 60 point correspondences, fewer than half those in conventional solutions, are sufficient to reach the performance boundary. The proposed solution is applicable for consumer IMU sensors to achieve highly accurate 3D reconstruction. Demonstrating in a practical scene is a future work.



(a) RMS error in $\theta$ (b) RMS error in $\boldsymbol{t}$

**Fig. 7.** Results of changing the number of point correspondences with $\sigma_{image} = 0.5$ pixel and $\sigma_{angle} = 0.5°$.

**Table 1.** Comparison of Mean Computation Time [msec]

| | the Number of Point Correspondences | | | | | | |
|---|---|---|---|---|---|---|---|
| | 3 | 4 | 5 | 10 | 50 | 100 | 500 |
| Kalantari et al. | 75.64 | n/a | n/a | n/a | n/a | n/a | n/a |
| Fraundorfer et al. | 3.62 | 0.69 | 0.33 | 0.32 | 0.33 | 0.33 | 0.38 |
| Proposed | 0.38 | 0.41 | 0.40 | 0.40 | 0.41 | 0.41 | 0.48 |

## References

1. Philip, J.: A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. The Photogrammetric Record **15**(88) (1996) 589–599
2. Triggs, B.: Routines for relative pose of two calibrated cameras from 5 points. Technical Report, INRIA (2000)
3. Nistér, D.: An efficient solution to the five-point relative pose problem. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. Volume 2., IEEE (2003) II–195
4. Stewénius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing **60**(4) (2006) 284–294
5. Li, H., Hartley, R.: Five-point motion estimation made easy. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. Volume 1., IEEE (2006) 630–633
6. Kukelova, Z., Bujnak, M., Pajdla, T.: Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. BMVC 2008 **2**(5) (2008)
7. Kalantari, M., Jung, F., Guedon, J., Paparoditis, N.: The five points pose problem: A new and accurate solution adapted to any geometric configuration. Advances in Image and Video Technology (2009) 215–226
8. Pizarro, O., Eustice, R., Singh, H.: Relative pose estimation for instrumented, calibrated imaging platforms. Proceedings of Digital Image Computing Techniques and Applications, Sydney, Australia (2003) 601–612
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)

10. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6) (June 1981) 381–395
11. Kalantari, M., Hashemi, A., Jung, F., Guédon, J.P.: A new solution to the relative orientation problem using only 3 points and the vertical direction. CoRR **abs/0905.3964** (2009)
12. Fraundorfer, F., Tanskanen, P., Pollefeys, M.: A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. Computer Vision–ECCV 2010 (2010) 269–282
13. Faugeras, O.: Three-dimensional computer vision: a geometric viewpoint. the MIT Press (1993)
14. Horn, B.K.P.: Recovering baseline and orientation from essential matrix. J. Optical Society of America (1990)
15. Cox, D.A., Little, J.B., O'Shea, D.: Using Algebraic Geometry. 2nd edn. Springer (2005)
16. Kukelova, Z., Bujnak, M., Pajdla, T.: Automatic generator of minimal problem solvers. Computer Vision–ECCV 2008 (2008) 302–315
17. Sinha, S.N., Frahm, J.M., Pollefeys, M., Genc, Y.: Gpu-based video feature tracking and matching. In: EDGE, Workshop on Edge Computing Using New Commodity Architectures. Volume 278. (2006) 4321
18. Terriberry, T.B., French, L.M., Helmsen, J.: Gpu accelerating speeded-up robust features. In: Proceedings of 3DPVT '08. (2008) 355–362

## Appendix

The proof of that the proposed 4-point algorithm including the 3-point algorithm is as follows.

Substituting three point correspondences into (17), we have

$$\begin{aligned}\frac{d}{d\theta}\det(\boldsymbol{B}^T\boldsymbol{B}) &= \frac{d}{d\theta}\det(\boldsymbol{A}^T\boldsymbol{A})\\ &= \frac{d}{d\theta}\det(\boldsymbol{A})^2\\ &= 2\det(\boldsymbol{A})\frac{d}{d\theta}\det(\boldsymbol{A}).\end{aligned} \tag{20}$$

We can construct a system of polynomial equations as follows:

$$\begin{cases} f_3(c,s) = \det(\boldsymbol{A})\dfrac{d}{d\theta}\det(\boldsymbol{A})\Big|_{\substack{\cos\theta=c,\\ \sin\theta=s}} = 0,\\ g(c,s) = c^2+s^2-1=0. \end{cases} \tag{21}$$

The solutions of $\mathrm{Res}(f_3, g, c) = 0$ include that of $\mathrm{Res}(f_1, g, c) = 0$.