Supplementary Material for Affine Correspondences between Multi-Camera Systems for 6DOF Relative Pose Estimation

Banglei Guan[1] and Ji Zhao ✉

[1]National University of Defense Technology, Changsha 410073, China
✉ corresponding author
guanbanglei12@nudt.edu.cn, zhaoji84@gmail.com

1 Relative Pose Estimation for Monocular Cameras

In this section, we show that our minimal solver generation framework can be easily extended to solve various relative pose estimation problems, *e.g.*, relative pose estimation for a monocular camera. It has been proved that a minimal number of two affine correspondences (ACs) is sufficient to recover the relative camera pose of a monocular camera [16,2]. Being different from [16,2] that solve the essential matrix firstly and then decompose it into the relative pose, we solve the relative rotation and translation between two views directly.

Consistent with the relative pose estimation for multi-camera systems, we also use the special parameterization to formulate the relative pose estimation problem for a monocular camera, see Fig. 1. The translation parameters can be eliminated using two depth parameters. We denote the j-th AC as $(\mathbf{x}_j, \mathbf{x}'_j, \mathbf{A}_j)$, where $\mathbf{x}_j$ and $\mathbf{x}'_j$ are the normalized homogeneous image coordinates of feature points in the view 1 and view 2, respectively. $\mathbf{A}_j$ is a 2×2 local affine transformation, which relates the infinitesimal patches around $\mathbf{x}_j$ and $\mathbf{x}'_j$. The corresponding unit direction vectors of feature points represented in two views can be computed as follows: $\mathbf{p}_j = \mathbf{x}_j / \|\mathbf{x}_j\|$ and $\mathbf{p}'_j = \mathbf{x}'_j / \|\mathbf{x}'_j\|$.

1.1 Parameterization for Relative Pose

We choose one AC to define a world reference system W, as shown in Fig. 1. Suppose the j-th AC is currently chosen. Let the origin of W as the position of the j-th AC in 3D space and the orientation of W is consistent with view 1. Denote the transformation between view 1 and view 2 as $[\mathbf{R}, \mathbf{t}]$, the transformation between view 1 and reference W as $[\mathbf{R}_1, \mathbf{t}_1]$, and the transformation between view 2 and reference W as $[\mathbf{R}_2, \mathbf{t}_2]$. Note that $\mathbf{R}_1 = \mathbf{I}$, $\mathbf{R}_2 = \mathbf{R}$. We also use Cayley parameterization to represent the rotation $\mathbf{R}$. Next, we parameterize $\mathbf{t}_1$ and $\mathbf{t}_2$ as linear functions of two unknown depth parameters $\{\lambda_{j1}, \lambda_{j2}\}$:

$$\mathbf{t}_1 = \lambda_{j1}\mathbf{p}_j, \qquad \mathbf{t}_2 = \lambda_{j2}\mathbf{p}'_j. \tag{1}$$

The relative pose between two views is determined by the composition of two transformations: (i) from view 1 to W, (ii) from W to view 2. There are unknowns

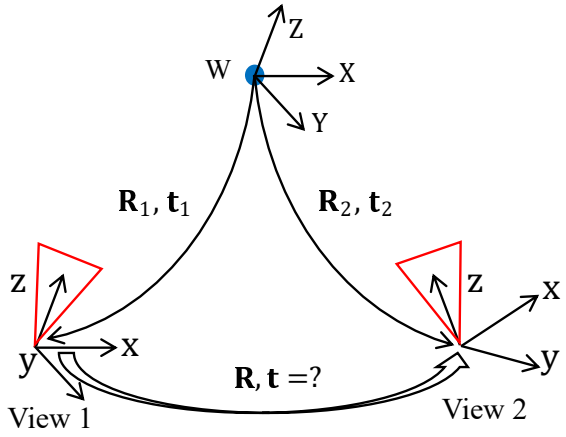


Fig. 1. Relative pose estimation from two ACs for a monocular camera. Red triangle represents a monocular camera. One AC is used to define a world reference system W.

$\mathbf{R}$, $\mathbf{t}_1$ and $\mathbf{t}_2$ which can be parameterized as $\{q_x, q_y, q_z, \lambda_{j1}, \lambda_{j2}\}$. Formally, the relative pose $[\mathbf{R}, \mathbf{t}]$ between view 1 and view 2 is represented as

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t}_2 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{t}_1 \\ \mathbf{0} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R} & \mathbf{t}_2 - \mathbf{R}\mathbf{t}_1 \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2}$$

The essential matrix can be represented as

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R} = -\mathbf{R}[\mathbf{t}_1]_\times + [\mathbf{t}_2]_\times \mathbf{R}. \tag{3}$$

By substituting Eq. (1) into Eq. (3), it can be verified that each entry in the essential matrix is linear with $\{\lambda_{j1}, \lambda_{j2}\}$. Then, we substitute Eq. (3) into Eqs.(8) and (9) in the paper. It can be seen that one AC yields three equations for the relative pose estimation of a monocular camera. Note that the special parameterization has been adopted by choosing one AC as the origin of world reference system, the PC derived from the chosen AC cannot contribute one constraint since the coefficients of the resulting equation are zero. Thus, when j-th AC is chosen to build up the world reference system W, five equations can be obtained from two ACs, which consist of two affine transformation constraints from j-th AC and three equations from the other AC. Based on the hidden variable technique [6], the five equations provided by two ACs can be written as

$$\underbrace{\mathbf{F}'_j(q_x, q_y, q_z)}_{5\times 2} \begin{bmatrix} \lambda_{j1} \\ \lambda_{j2} \end{bmatrix} = \mathbf{0}. \tag{4}$$

The entries in $\mathbf{F}'_j$ are quadratic in unknowns q_x, q_y, and q_z.

1.2 Equation System Construction

The relative pose estimation problem of a monocular camera has 5DOF. However, two ACs provide six independent constraints. That means the number of constraints is greater than the number of unknowns, and there is a redundant constraint. Thus, we randomly choose four equations from Eq. (4) to explore the minimal case solution. For example, two affine transformation constraints of j-th AC, and the epipolar constraint and the first affine transformation constraint of the other AC are stacked into 4 equations in 5 unknowns, *i.e.*, the first four equations of Eq. (4):

$$\underbrace{\mathbf{F}_j(q_x, q_y, q_z)}_{4\times 2} \begin{bmatrix} \lambda_{j1} \\ \lambda_{j2} \end{bmatrix} = \mathbf{0}. \tag{5}$$

Since Eq. (5) has non-trivial solutions, the rank of $\mathbf{F}_j$ satisfies $\text{rank}(\mathbf{F}_j) \leq 1$. Thus, all the 2×2 sub-determinants of $\mathbf{F}_j$ must be zero. This gives six equations about three unknowns $\{q_x, q_y, q_z\}$. Up to now, we suppose j-th AC is chosen to build up the world reference system W. Since there are two ACs in the minimal solution case, we can also choose the other AC to build up the world reference system, and its orientation is also consistent with the reference of the multi-camera system in view 1. Suppose the j'-th AC is chosen, we obtain an new equation system about the same rotation parameters $\{q_x, q_y, q_z\}$, which is similar to Eq. (5):

$$\underbrace{\mathbf{F}_{j'}(q_x, q_y, q_z)}_{4\times 2} \begin{bmatrix} \lambda_{j'1} \\ \lambda_{j'2} \end{bmatrix} = \mathbf{0}. \tag{6}$$

Note that Eq. (6) provides new constraints which is different from Eq. (5). We use the computer algebra system `Macaulay 2` [7] to find that there are one dimensional families of extraneous roots if only Eq. (5) or Eq. (6) is used. Based on Eqs. (5) and (6), we have twelve equations with three unknowns $\{q_x, q_y, q_z\}$:

$$\begin{gathered} \det(\mathbf{N}(q_x, q_y, q_z)) = 0, \\ \mathbf{N} \in \{2 \times 2 \text{ submatrices of } \mathbf{F}_j\} \cup \{2 \times 2 \text{ submatrices of } \mathbf{F}_{j'}\}. \end{gathered} \tag{7}$$

These twelve equations have a degree of 4, *i.e.*, the highest of the degrees of the monomials with non-zero coefficients is 4. The Gröbner basis technique is also used to produce the solver for the polynomial equation system Eq. (7). Our monocular camera solver maximally has 20 complex solutions and the elimination template of size 36×56. Once the rotation parameters $\{q_x, q_y, q_z\}$ are obtained, $\mathbf{R}$ can be obtained immediately using Eq. (1) in the paper. Take the translation estimation using $\{\lambda_{jk}\}_{k=1,2}$ for an example. $\{\lambda_{jk}\}$ is determined by finding the null space of $\mathbf{F}_j$, see Eq. (5). Next we can calculate the translations $\mathbf{t}_1$ and $\mathbf{t}_2$ by Eq. (1). Finally, we calculate the relative pose $[\mathbf{R}, \mathbf{t}]$ of the monocular camera based on Eq. (2).

It should be noted the number of solutions obtained by our monocular camera solver is essentially the same as the solvers using essential matrix parametrization. The solvers using essential matrix parametrization have 10 solutions for the essential matrix [15,17]. For each essential matrix, there are four possible rotation-translation pairs [10]. Thus, there are 40 rotation-translation pair solutions for the solvers using essential matrix parametrization. Our monocular camera solver uses rotation and translation parametrization, and there are 20 solutions for the rotation. Since each rotation has two possible translations with opposite directions, the proposed monocular camera solver also has 40 rotation-translation pair solutions. Thus, we can see that the solvers using two different parametrizations have the same number of rotation-translation pair solutions.

2 Relative Pose Estimation for Multi-Camera Systems

2.1 Degenerated Configurations

In this section, we prove three cases of critical motions for the proposed solvers, including both the inter-camera solver and the intra-camera solver. In these critical configurations, the rotation and the translation direction between two views of the multi-camera system can be correctly recovered, but the metric scale of translation is unobtainable. The proofs of degenerated configurations modeled in different ways have been proposed in [8].

Proposition 1. *For inter-camera ACs, if a multi-camera system undergoes pure translation and the baseline of two camera is parallel with the translation direction, the metric scale of translation cannot be recovered.*

Proof. In the case of inter-camera ACs, each AC is seen by the different cameras over two consecutive views. For the pure translation case, the rotation between two views of the multi-camera system satisfies $\mathbf{R} = \mathbf{I}$. Since the baseline of two camera is parallel with the translation direction, the translation satisfies $\mathbf{s}_2 - \mathbf{s}_1 = a(\mathbf{t}_2 - \mathbf{t}_1)$, where a is a unknown number, $\mathbf{t}_2 - \mathbf{t}_1$ is the translation between two views of the multi-camera system. The essential matrix in Eq. (6) in the paper can be written as

$$\begin{aligned}\mathbf{E}' &= \mathbf{Q}_2^T \left([\mathbf{t}_2 - \mathbf{t}_1]_\times - [\mathbf{s}_2 - \mathbf{s}_1]_\times \right) \mathbf{Q}_1 \\ &= (1-a)\, \mathbf{Q}_2^T [\mathbf{t}_2 - \mathbf{t}_1]_\times \mathbf{Q}_1. \end{aligned} \tag{8}$$

The essential matrix $\mathbf{E}'$ is homogeneous with the translation between two views of the multi-camera system $\mathbf{t}_2 - \mathbf{t}_1$. We substitute Eq. (8) into Eqs. (8) and (9) in the paper. Then the geometric constraints provided by an AC become:

$$\mathbf{x}_j'^T \mathbf{Q}_2^T [\mathbf{t}_2 - \mathbf{t}_1]_\times \mathbf{Q}_1 \mathbf{x}_j = 0, \tag{9}$$

$$(\mathbf{Q}_1^T [\mathbf{t}_2 - \mathbf{t}_1]_\times \mathbf{Q}_2 \mathbf{x}_j')_{(1:2)} = \mathbf{A}_j^T (\mathbf{Q}_2^T [\mathbf{t}_2 - \mathbf{t}_1]_\times \mathbf{Q}_1 \mathbf{x}_j)_{(1:2)}. \tag{10}$$

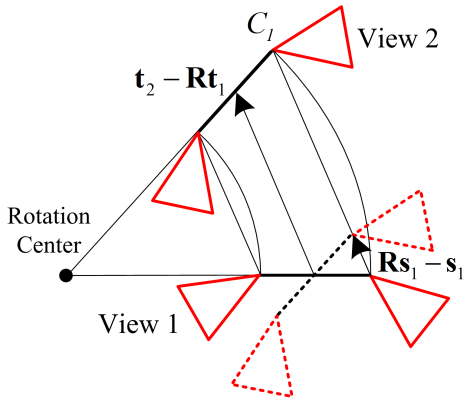


Fig. 2. Critical motion due to constant rotation rate.

Suppose κ is a free parameter, it can be verified that $\kappa(\mathbf{t}_2 - \mathbf{t}_1)$ satisfies Eqs. (9) and (10). Thus, the metric scale of translation between two views of the multi-camera system cannot be recovered.

Proposition 2. *For intra-camera ACs, when a multi-camera system undergoes pure translation or constant rotation rate, both cases are degenerate motions. Specifically, the metric scale of translation cannot be recovered.*

Proof. In the case of intra-camera ACs, each AC is seen by the same camera over two consecutive views. So we have $\mathbf{s}_1 = \mathbf{s}_2$ and $\mathbf{Q}_1 = \mathbf{Q}_2$.

(1) For the pure translation case, with the assumption that $\mathbf{R} = \mathbf{I}$, the essential matrix in Eq. (6) in the paper can be written as

$$\mathbf{E}' = \mathbf{Q}_1^T \left([\mathbf{t}_2 - \mathbf{t}_1]_\times \right) \mathbf{Q}_1. \tag{11}$$

The essential matrix is homogeneous with the translation between two views of the multi-camera system $\mathbf{t}_2 - \mathbf{t}_1$. Suppose κ is a free parameter, it can be verified that $\kappa(\mathbf{t}_2 - \mathbf{t}_1)$ invariably satisfies Eqs. (8) and (9) in the paper.

(2) For the constant rotation rate case, *i.e.*, both camera paths move along concentric circles, the proof is inspired by [5]. We take the camera C_1 in Fig. 2 as an example, the rotation induced translation $\mathbf{R}\mathbf{s}_1 - \mathbf{s}_1$ is aligned with the translation $\mathbf{t}_2 - \mathbf{R}\mathbf{t}_1$. Denote $\mathbf{R}\mathbf{s}_1 - \mathbf{s}_1 = a(\mathbf{t}_2 - \mathbf{R}\mathbf{t}_1)$ and substitute it to Eq. (6) in the paper, the essential matrix becomes

$$\mathbf{E}' = (1 + a)\, \mathbf{Q}_1^T \left([\mathbf{t}_2 - \mathbf{R}\mathbf{t}_1]_\times \mathbf{R} \right) \mathbf{Q}_1. \tag{12}$$

The essential matrix is homogeneous with the translation between two views of the multi-camera system $\mathbf{t}_2 - \mathbf{R}\mathbf{t}_1$. Suppose κ is a free parameter, it can be verified that $\kappa(\mathbf{t}_2 - \mathbf{R}\mathbf{t}_1)$ also satisfies Eqs. (8) and (9) in the paper.

To deal with these degenerate cases, we can use auxiliary sensors, such as integrating the acceleration over time from an IMU, to recover the metric scale of the translation [13,8,9]. Moreover, in the absence of auxiliary sensors, since the frame rate of current cameras is high and the multi-camera system usually moves at a constant speed within a short time, we can also use the metric scale of the previous image pairs to approximate the current metric scale.

2.2 Polynomial System Solving

In subsection 3.4 of the paper, we use all the equations $\mathcal{E}_1$ and $\mathcal{E}_2$ to construct polynomial systems and find solvers. It is possible to construct solvers using a subset of these equations. Specifically, denote $\mathcal{E}_{1,1}$ as

$$\det(\mathbf{N}(q_x, q_y, q_z)) = 0, \quad \mathbf{N} \in \{3 \times 3 \text{ submatrices of } \mathbf{F}_j\}, \tag{13}$$

and $\mathcal{E}_{1,2}$ as

$$\det(\mathbf{N}(q_x, q_y, q_z)) = 0, \quad \mathbf{N} \in \{3 \times 3 \text{ submatrices of } \mathbf{F}_{j'}\}. \tag{14}$$

We can see that $\mathcal{E}_1 = \mathcal{E}_{1,1} \cup \mathcal{E}_{1,2}$. Using different combinations of $\mathcal{E}_{1,1}$, $\mathcal{E}_{1,2}$, and $\mathcal{E}_2$, we have the following results for polynomial system solving. The dimension, degree, and number of solutions are shown in Table 1. When the dimension of the corresponding polynomial idea is zero, it means the number of solutions is finite. Otherwise, a positive dimension of the corresponding polynomial idea indicates infinite solutions.

Table 1. Different equation combinations for the multi-camera system solvers. `dimension` indicates the dimension of the corresponding polynomial ideal. `degree` indicates the degree of the algebraic variety. `#sol` indicates the number of solutions. `1-dim` indicates one dimensional families of extraneous roots.

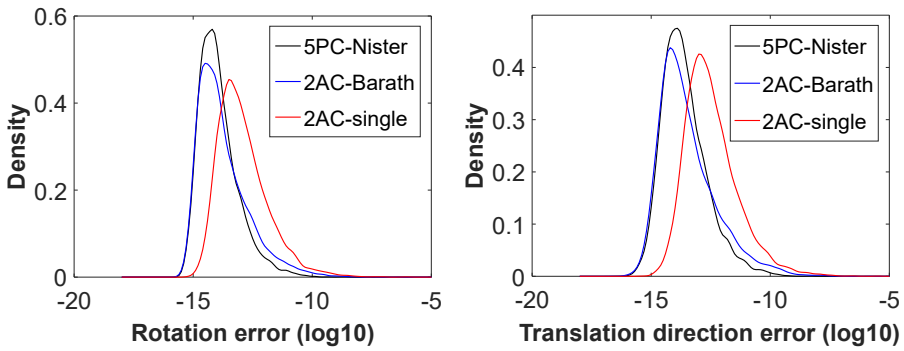
	Inter-camera			Intra-camera		
Equation	dimension	degree	#sol	dimension	degree	#sol
$\mathcal{E}_{1,1}$	1	2	1-dim	1	3	1-dim
$\mathcal{E}_{1,2}$	1	2	1-dim	1	3	1-dim
$\mathcal{E}_2$	1	16	1-dim	1	16	1-dim
$\mathcal{E}_1$	0	56	56	1	1	1-dim
$\mathcal{E}_{1,1}$+$\mathcal{E}_2$	0	56	56	0	56	56
$\mathcal{E}_{1,2}$+$\mathcal{E}_2$	0	56	56	0	56	56
$\mathcal{E}_1$+$\mathcal{E}_2$	0	48	48	0	48	48

3 Experiments

In this section, the experiment results of the proposed monocular camera solver are shown in subsection 3.1. The experiment results of the proposed multi-camera system solvers are shown in subsection 3.2, subsection 3.3, subsection 3.4, subsection 3.5 and subsection 3.6.

Methods	5PC-Nister [15]	2AC-Barath [2]	**2AC-single**
Runtime	5.5	11.0	127.1

Table 2. Runtime comparison of monocular camera solvers (unit: μs).



(a) (b)

Fig. 3. Probability density functions over relative pose estimation errors of the monocular camera solvers in noise-free cases (10,000 trials). The horizontal axis represents the $\log_{10}$ errors and the vertical axis represents the density.

3.1 Experiments for Monocular Cameras

In this set of experiments, we evaluate the performance of the proposed solver for monocular cameras in Section 1, which is referred to as `2AC-single`. The `2AC-single` solver is obtained as a side product of our minimal solver generation framework. It should be noted that the purpose of the following experiments is not to outperform the state-of-the-art methods using the essential matrix parametrization [15,2]. Instead, we illustrate the feasibility and practicality of the proposed solver.

The proposed solvers are evaluated on an Intel(R) Core(TM) i7-7800X 3.50GHz. All the solvers are implemented in C++. The code of `5PC-Nister` is provided by the `PoseLib`[1]. The `2AC-Barath` are publicly available from the code of [3]. Table 2 shows the average processing times of the monocular camera solvers over 10,000 runs. Since the solvers `5PC-Nister` and `2AC-Barath` solve the essential matrix firstly and then decompose it into the relative pose, the runtime of both methods is lower than the proposed `2AC-single` solver, which computes the relative rotation and translation directly. The `5PC-Nister` is most efficient, because it solves a univariate polynomial equation using the efficient Sturm sequence method. The proposed `2AC-single` solver takes about 0.127 milliseconds, and it is still applicable for common scenarios.

[1] `https://github.com/vlarsson/PoseLib`

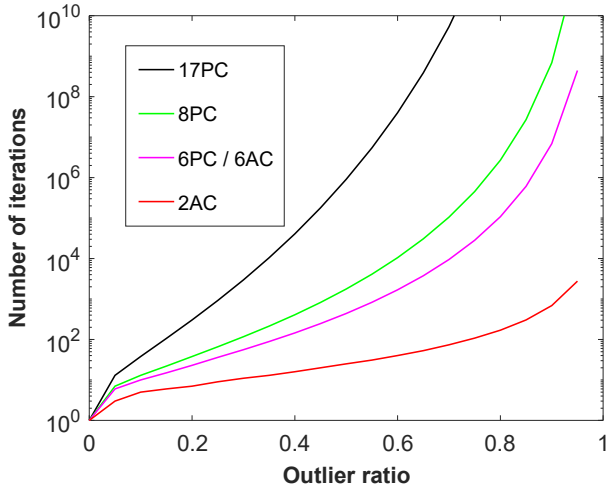


Fig. 4. RANSAC iteration number with respect to outlier ratio for success probability 99.9%. The number of RANSAC iterations increases exponentially with respect to the minimal number of feature correspondences.

Figure 3 reports the numerical accuracy comparison of the monocular camera solvers in noise-free cases. We repeat the procedure 10,000 times and plot the empirical probability density functions as the function of the $\log_{10}$ estimated errors. Numerical stability represents the round-off error of monocular camera solvers in noise-free cases. It is shown that the solvers `5PC-Nister` [15] and `2AC-Barath` [2] have comparable numerically stability. The numerical accuracy of the proposed `2AC-single` solver is sightly worse than the comparative solvers. Since the modes of the rotation error and translation error of the `2AC-single` solver are about 1×10^{-13}, and both the rotation error and translation error are basically below 1×10^{-8}, our method is also applicable for relative pose estimation of a monocular camera in the practical applications.

3.2 Efficiency Comparison in a RANSAC Framework

For the 6DOF relative pose estimation of multi-camera systems, we have evaluated the efficiency comparison and numerical stability of all the solvers in the paper. In addition to efficiency and numerical stability, another important factor for a solver is the minimal number of needed feature correspondences between two views. Because the minimal solvers are typically employed inside a RANSAC framework, and the computational complexity of the RANSAC estimator increases exponentially with respect to the number of feature correspondences needed. The number of iterations N required in RANSAC can be given by $N = \log(1-p)/\log(1-(1-\epsilon)^s)$, where s is the minimal number of feature correspondences needed for the solver, ϵ is the outlier ratio, and p is the success

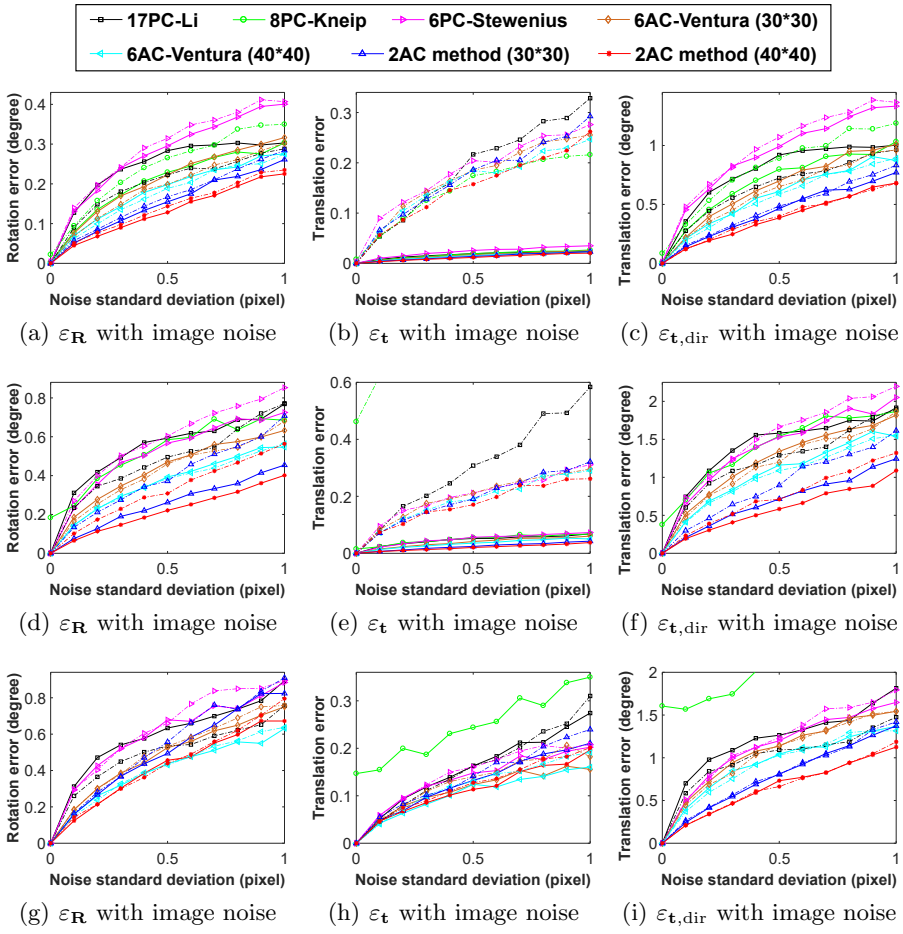
probability that all the selected feature correspondences are inlier. For a probability of success $p = 99.9\%$, the number of required RANSAC iterations with respect to the outlier ratio is shown in Fig. 4.

It can be seen that the number of iterations N increases exponentially with respect to the minimal number of feature correspondences s. For example, given the outlier ratio $\epsilon = 50\%$, when the solvers need 17, 8, 6 and 2 feature correspondences, the number of required RANSAC iterations is 905410, 1765, 439 and 25, respectively. Since the proposed solvers require only two ACs, the number of RANSAC iterations is obviously lower than both the PC-based methods and the AC-based linear method. Thus, our solvers have an advantage in detecting the outlier and estimating the initial motion efficiently when integrating them into the RANSAC framework. As we will see later, the proposed solvers have better overall efficiency than the comparative solvers in the experiments on real data.

3.3 Accuracy with Image Noise

In this scenario, the magnitude of image noise is set to Gaussian noise with a standard deviation ranging from 0 to 1.0 pixels. The directions of the multi-camera system are set to forward, random, and sideways motions, respectively. Figure 5 shows the performance of the proposed solvers with increasing image noise. All the solvers are evaluated on both inter-camera ACs and intra-camera ACs. The corresponding estimation results are represented by solid lines and dash-dotted lines, respectively. The `2AC method` indicates `2AC-inter-56` when using inter-camera ACs, and indicates `2AC-intra` when using intra-camera ACs. In this figure, the display range is limited so that some curves with large errors are invisible or partially invisible.

We have the following observations. (1) The solvers using inter-camera ACs generally have better performance than intra-camera ACs, especially in recovering the metric scale of translation. (2) The performance of AC-based methods is influenced by the noise magnitude of affine transformation, which is determined by the support region of sampled points. Thus, the AC-based methods have better performance with larger support regions at the same magnitude of image noise. (3) When the side length of the square is 40 pixels, the proposed `2AC method` provides better results than the comparative methods with both inter-camera ACs and intra-camera ACs. (4) The `8PC-Kneip` performs well in the forward motion of the multi-camera systems, but it performs poorly in the random and sideways motions. The probable reason may be the iterative optimization which is susceptible to falling into local minima [20]. (5) The linear solvers `17PC-Li` and `6AC-Ventura` with fewer calculations have less round-off error than the proposed `2AC method` in noise-free cases, see Fig. 3. However, our method has better accuracy than the linear solvers with the influence of image noise. This is also consistent with the real-world data experiments.



(a) $\varepsilon_{\mathbf{R}}$ with image noise (b) $\varepsilon_{\mathbf{t}}$ with image noise (c) $\varepsilon_{\mathbf{t},\mathrm{dir}}$ with image noise

(d) $\varepsilon_{\mathbf{R}}$ with image noise (e) $\varepsilon_{\mathbf{t}}$ with image noise (f) $\varepsilon_{\mathbf{t},\mathrm{dir}}$ with image noise

(g) $\varepsilon_{\mathbf{R}}$ with image noise (h) $\varepsilon_{\mathbf{t}}$ with image noise (i) $\varepsilon_{\mathbf{t},\mathrm{dir}}$ with image noise

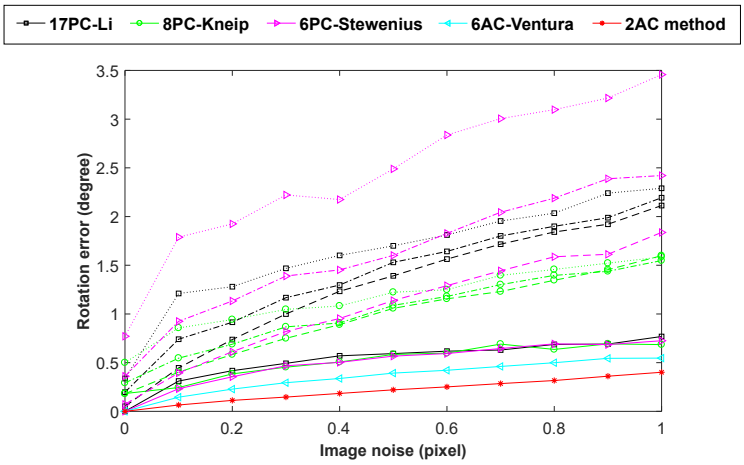
Fig. 5. Rotation and translation error with increasing image noise. The first, second and third rows reports the performance of the proposed solvers under forward, random and sideways motions, respectively. Solid line indicates using inter-camera ACs, and dash-dotted line indicates using intra-camera ACs.

3.4 Evaluation of PC-based Solvers using ACs

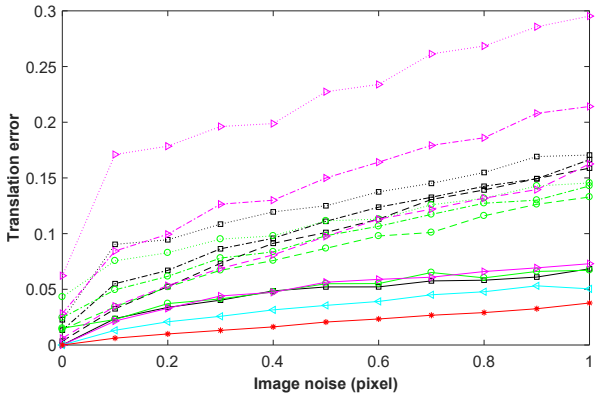
In this experiment, we test the performance of PC-based solvers for the multi-camera relative pose estimation using ACs. An AC can be converted into three PCs, which are then used as the input of the PC-based solvers. Three generated PCs converted from an AC consist of a PC $(\mathbf{x}_j, \mathbf{x}'_j)$ and two hallucinated PCs calculated by the local affine transformation $\mathbf{A}_j$. However, the hallucinated PCs inevitably have errors even for noise-free input. Because the local affine transformation is only valid in the distribution area, where it is infinitesimally close to the image coordinates of AC [2]. Following the conversion equation in [3], we can compute three approximate PCs converted from one AC: $(\mathbf{x}_j, \mathbf{x}_j + [s,\ 0]^T, \mathbf{x}_j + [0,\ s]^T)$ and $(\mathbf{x}'_j, \mathbf{x}'_j + \mathbf{A}_j[s,\ 0]^T, \mathbf{x}'_j + \mathbf{A}_j[0,\ s]^T)$, where s represents the size of the distribution area of the generated PCs. It can be found that the size of s determines the magnitude of the conversion error of the hallucinated PCs. In this experiment, we set s to 1, 5, and 10 pixels, respectively. The performance of PC-based solvers is evaluated with the different sizes of the distribution area.

Take the relative pose estimation using inter-camera ACs for an example. The synthetic data is generated by following the configuration in Section 4.1 in the paper. We carry out a total of 1000 trials in the synthetic experiment. The rotation and translation errors are assessed by the median of errors. In each test, 100 ACs are generated randomly, which includes 50 ACs from a ground plane and 50 ACs from 50 random planes. The support region for generating the ACs is set to 40*40 pixels. In this experiment, the `2AC method` indicates the proposed `2AC-inter-56` solver. The required ACs are selected randomly for the AC-based solvers within the RANSAC scheme. So, 6 and 2 ACs are selected randomly for the `6AC-Ventura` [1] method and the proposed `2AC method`, respectively. For the PC-based solvers, the hallucinated PCs converted from a minimal number of ACs are used as input. Thus, 6, 3 and 2 ACs are selected randomly for the solvers `17PC-Li` [12], `8PC-Kneip` [11], and `6PC-Stewénius` [18], respectively. It should be noted that we only use the hallucinated PCs converted from ACs for hypothesis generation. The corresponding inlier set of the estimated relative pose is still determined by evaluating the image point pairs of ACs. The relative pose which produces the most inliers is used to measure the error. This also allows us to select the best candidate from multiple solutions.

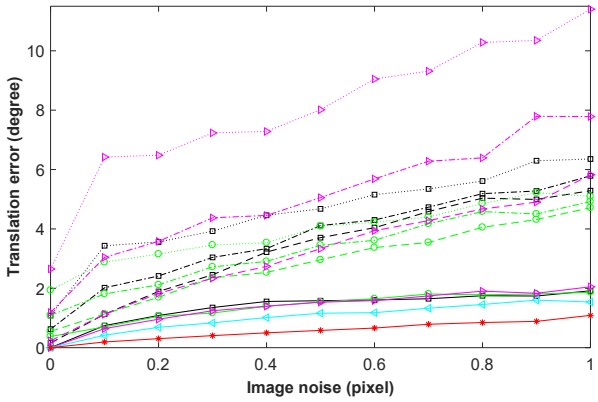
Figure 6 shows the performance of the PC-based solvers with increasing image noise under random motion. Solid lines represent the estimation results using the image point pairs of ACs. Dashed lines, dash-dotted lines, and dotted lines represent the estimation results using the hallucinated PCs converted from ACs, when the size of the distribution area is set to 1, 5, and 10 pixels, respectively. We have the following observations. (1) The PC-based solvers using the hallucinated PCs have worse performance than using the image point pairs of the ACs. Because the conversion error is newly introduced while the hallucinated PCs are generated by the ACs. In addition, since the hallucinated PCs generated by each AC are close to each other, this may be a degenerate case for the PC-based solvers. (2) Even though the image noise is zero, the rotation and translation



(a) $\varepsilon_{\mathbf{R}}$ with image noise



(b) $\varepsilon_{\mathbf{t}}$ with image noise



(c) $\varepsilon_{\mathbf{t},\mathrm{dir}}$ with image noise

Fig. 6. Rotation and translation error of the PC-based solvers using ACs with increasing image noise. Solid lines indicate using the image point pairs of ACs. Dashed lines, dash-dotted lines, and dotted lines indicate using the hallucinated PCs converted from ACs, when the size of the distribution area is set to 1, 5, and 10 pixels, respectively.

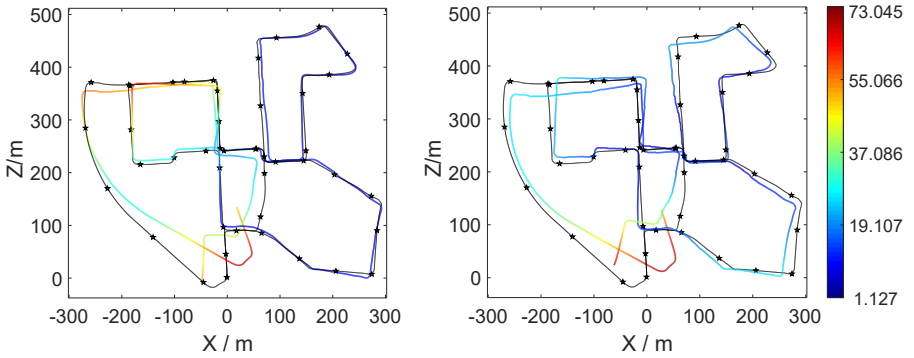
error of the PC-based solvers is not zero when using the hallucinated PCs. This also shows that the local affine transformation is only valid in the infinitesimal patches around the image point pairs of ACs. (3) The PC-based solvers have better performance with smaller distribution areas at the same magnitude of image noise. Because the conversion error between ACs and hallucinated PCs is determined by the size of the distribution area, and the smaller distribution area causes the smaller conversion error. (4) The proposed `2AC method` provides better estimation results than the comparative methods. Compared with the PC-based solvers, the AC-based solvers use the affine transformation constraints as expressed in Eq. (9) in the paper. These affine transformation constraints describe the strictly satisfied geometric relationship between the essential matrix and the local affine transformation. The affine transformation constraints have not any conversion error. It is an advantage compared to using the epipolar constraints of the hallucinated PCs.

3.5 Experiments on KITTI Dataset

In order to visualize the comparison results, we also show the estimated trajectory for `KITTI` sequence 00. Figure 7 shows the estimated trajectories without any post-refinement. The estimated trajectory of the proposed `2AC method` is compared with the best performing comparison method `8PC-Kneip` [11], which has been shown in Table 3 of the paper. Note that the frame-to-frame relative pose estimation results are directly concatenated without any post-refinement. We align both estimated trajectories with the ground truth. The trajectories on X-Z plane are displayed in Fig. 7. It is worth mentioning that our `2AC method` has a smaller error than the `8PC-Kneip` method in Y-axis. Moreover, the absolute trajectory error (ATE) is encoded by the color along the estimated trajectory [19]. It is shown that the proposed `2AC method` has a smaller ATE than the `8PC-Kneip` method.

3.6 Experiments on EuRoC Dataset

To validate the proposed solver in an unmanned aerial vehicle environment, we further use the `EuRoC` MAV dataset [4] to evaluate the 6DOF relative pose estimation. The `EuRoC` MAV dataset is recorded using a stereo camera mounted on a micro aerial vehicle. We test the `2AC-intra` solver on all the available 5 sequences, which are collected in a large industrial machine hall. Each sequence contains synchronized stereo images, accurate position, and IMU measurements. The spatio-temporally aligned ground truth is provided from the nonlinear least-squares batch solution over the Leica position and IMU measurements. Since the industrial environment is unstructured and cluttered, it renders these sequences challenging to process. In order to prevent the movement of the image pair from being too small, the images for relative pose estimation are thinned out from the consecutive image sequences by an amount of one out of every four images. Besides, the image pairs with insufficient motion are cropped in this experiment. The ACs between the consecutive views in each camera are also established by



(a) `8PC-Kneip` (b) `2AC method`

Fig. 7. Estimated trajectories without any post-refinement. The relative pose measurements between consecutive frames are directly concatenated. The trajectories estimated by `8PC-Kneip` [11] and `2AC method` are represented by the colorful curves. The ground truth trajectory is represented by the black curves with stars. Best viewed in color.

Table 3. Rotation and translation error on `EuRoC` sequences (unit: degree).

Seq.	17PC-Li [12]		8PC-Kneip [11]		6PC-Stew. [18]		6AC-Vent. [1]		**2AC method**	
	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t},\text{dir}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t},\text{dir}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t},\text{dir}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t},\text{dir}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t},\text{dir}}$
MH01 (788 images)	0.113	2.928	0.109	2.865	0.124	3.555	0.106	2.858	**0.092**	**2.519**
MH02 (675 images)	0.106	2.494	0.112	2.553	0.144	2.908	0.102	2.483	**0.086**	**2.242**
MH03 (605 images)	0.137	2.412	0.148	2.276	0.181	3.068	0.133	2.075	**0.125**	**1.928**
MH04 (449 images)	0.154	2.950	0.170	3.127	0.175	5.531	0.165	2.966	**0.139**	**2.609**
MH05 (514 images)	0.167	3.071	0.158	2.753	0.179	4.275	0.176	2.904	**0.146**	**2.714**

the ASIFT [14]. For the PC-based solvers, only the PCs derived from the ACs are used. All the solvers are tested on about 3000 image pairs in total.

Table 3 shows the rotation and translation error of the proposed `2AC method` for `EuRoC` sequences. It is shown that the `2AC method` provides better results than the comparative methods `17PC-Li`, `8PC-Kneip`, `6PC-Stewénius` and `6AC-Ventura`. This experiment also demonstrates that our `2AC method` is well suited for the relative pose estimation in the unmanned aerial vehicle environment.

References

1. Alyousefi, K., Ventura, J.: Multi-camera motion estimation with affine correspondences. In: International Conference on Image Analysis and Recognition. pp. 417–431 (2020)
2. Barath, D., Hajder, L.: Efficient recovery of essential matrix from two affine correspondences. IEEE Transactions on Image Processing **27**(11), 5328–5337 (2018)

3. Barath, D., Polic, M., FÃűrstner, W., Sattler, T., Pajdla, T., Kukelova, Z.: Making affine correspondences work in camera geometry computation. In: European Conference on Computer Vision. pp. 723–740 (2020)
4. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The EuRoC micro aerial vehicle datasets. The International Journal of Robotics Research **35**(10), 1157–1163 (2016)
5. Clipp, B., Kim, J.H., Frahm, J.M., Pollefeys, M., Hartley, R.: Robust 6dof motion estimation for non-overlapping, multi-camera systems. In: IEEE Workshop on Applications of Computer Vision. pp. 1–8. IEEE (2008)
6. Cox, D.A., Little, J., O'Shea, D.: Using algebraic geometry. Springer Science & Business Media (2006)
7. Grayson, D.R., Stillman, M.E.: Macaulay 2, a software system for research in algebraic geometry. `https://faculty.math.illinois.edu/Macaulay2/` (2002)
8. Guan, B., Zhao, J., Barath, D., Fraundorfer, F.: Efficient recovery of multi-camera motion from two affine correspondences. In: IEEE International Conference on Robotics and Automation. pp. 1305–1311 (2021)
9. Guan, B., Zhao, J., Barath, D., Fraundorfer, F.: Minimal cases for computing the generalized relative pose using affine correspondences. In: IEEE International Conference on Computer Vision. pp. 6068–6077 (2021)
10. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)
11. Kneip, L., Li, H.: Efficient computation of relative pose for multi-camera systems. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 446–453 (2014)
12. Li, H., Hartley, R., Kim, J.h.: A linear approach to motion estimation using generalized camera models. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008)
13. Liu, L., Li, H., Dai, Y., Pan, Q.: Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments. IEEE Transactions on Intelligent Transportation Systems **19**(8), 2432–2444 (2017)
14. Morel, J.M., Yu, G.: ASIFT: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences **2**(2), 438–469 (2009)
15. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(6), 756–777 (2004)
16. Raposo, C., Barreto, J.P.: Theory and practice of structure-from-motion using affine correspondences. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5470–5478 (2016)
17. Stewénius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing **60**(4), 284–294 (2006)
18. Stewénius, H., Oskarsson, M., Aström, K., Nistér, D.: Solutions to minimal generalized relative pose problems. In: Workshop on Omnidirectional Vision in conjunction with ICCV. pp. 1–8 (2005)
19. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580 (2012)
20. Ventura, J., Arth, C., Lepetit, V.: An efficient minimal solution for multi-camera motion. In: IEEE International Conference on Computer Vision. pp. 747–755 (2015)