Preprint manuscript No.
(will be inserted by the editor)

Minimal solvers for Relative Pose Estimation of Multi-Camera Systems using Affine Correspondences

Banglei Guan[1] · **Ji Zhao*** · **Daniel Barath**[2] · **Friedrich Fraundorfer**[3,4]

the date of receipt and acceptance should be inserted later

Abstract We propose three novel solvers for estimating the relative pose of a multi-camera system from affine correspondences (ACs). A new constraint is derived interpreting the relationship of ACs and the generalized camera model. Using the constraint, we demonstrate efficient solvers for two types of motions. Considering that the cameras undergo planar motion, we propose a minimal solution using a single AC and a solver with two ACs to overcome the degenerate case. Also, we propose a minimal solution using two ACs (a minimal number of one AC and one point correspondence) with known vertical direction, e.g., from an IMU. Since the proposed methods require significantly fewer correspondences than state-of-the-art algorithms, they can be efficiently used within RANSAC for outlier removal and initial motion estimation. The solvers are tested both on synthetic data and on three real-world scenes. It is shown that the accuracy of the estimated poses is superior to the state-of-the-art techniques. Source code is released at https://github.com/jizhaox/relative_pose_gcam_affine.

Keywords Relative pose estimation, Multi-camera system, Affine correspondence, Minimal solver

1 Introduction

Relative pose estimation from two views of a camera, or a multi-camera system is regarded as a fundamental problem in computer vision (Clipp et al. 2008; Hartley & Zisserman 2003; Scaramuzza & Fraundorfer 2011; Schönberger & Frahm 2016; Zhao et al. 2020) which plays an important role in simultaneous localization and mapping (SLAM) and structure-from-motion (SfM). Thus, improving the accuracy, efficiency and robustness of relative pose estimation algorithms is always an important research topic (Agarwal et al. 2017; Barath et al. 2020; Eichhardt & Barath 2020; Guan et al. 2018; Lee et al. 2014; Li et al. 2020; Ventura et al. 2015). Motivated by the fact that multi-camera systems are available in self-driving cars, micro aerial vehicles or AR headsets, this paper investigates the problem of estimating the relative pose of multi-camera systems from affine correspondences (ACs), see Fig. 1.

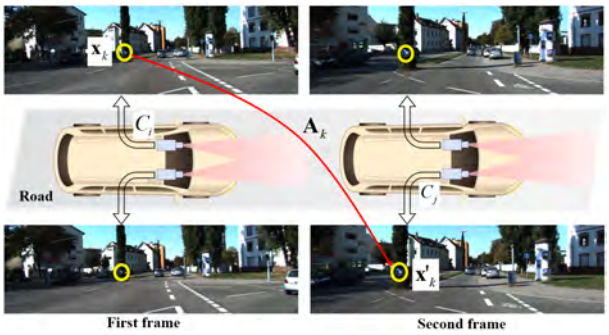


Fig. 1 An affine correspondence in a multi-camera system. It relates two perspective cameras C_i and C_j across two consecutive frames, where C_i and C_j can be the same or different cameras. The local affine transformation $\mathbf{A}_k$ relates the infinitesimal patches around correspondence $(\mathbf{x}_k, \mathbf{x}'_k)$.

Since a multi-camera system contains multiple individual cameras connected by being fixed to a single rigid body, it has the advantage of large field-of-view and high accuracy (Fragoso et al. 2020; Sweeney et al. 2015a). The main difference of a multi-camera system and a standard pinhole camera is the absence of a single projection center (Pless 2003). Due to the different camera model, the relative pose estimation problem of multi-camera systems (Stewénius et al. 2005) is different from the monocular cameras (Guan et al. 2020, 2021c; Nistér 2004), resulting in different equations. In order to remove outlier matches, most of the state-of-the-art SLAM and SfM pipelines using a multi-

* Corresponding author. E-mail: zhaoji84@gmail.com
[1] College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, 410073, China. E-mail: guanbanglei12@nudt.edu.cn
[2] Department of Computer Science, ETH Zürich, Zürich, 8092, Switzerland. E-mail: dbarath@ethz.ch
[3] Institute for Computer Graphics and Vision, Graz University of Technology, Graz, 8010, Austria. E-mail: fraundorfer@icg.tugraz.at
[4] Remote Sensing Technology Institute, German Aerospace Center, Weßling, 82234, Germany.

camera system (Häne et al. 2017; Heng et al. 2019) apply the relative pose estimation algorithms repeatedly in a robust estimation framework, *e.g.* the Random Sample Consensus (RANSAC) (Fischler & Bolles 1981). The outlier removal process has to be efficient since it has a large impact on the total run-time of the applied the SLAM and SfM pipeline. The computational complexity and, thus, the processing time of the RANSAC procedure depends exponentially on the number of points required for the relative pose estimation of multi-camera system.

Therefore, exploring the minimal solutions for relative pose estimation of multi-camera system is of significant importance and has received sustained attention (Clipp et al. 2008; Kim et al. 2009; Kneip et al. 2016; Li et al. 2008; Lim et al. 2010; Stewénius et al. 2005; Ventura et al. 2015). The idea of deriving minimal solutions for relative pose estimation of multi-camera systems ranges back to the work of Stewénius *et al.* with the 6-point method (Stewénius et al. 2005). Other classical works have been subsequently proposed, such as the 17-point linear method (Li et al. 2008) and techniques based on iterative optimization (Kneip & Li 2014). The minimal number of necessary points can be further reduced by taking additional motion constraints into account (Lee et al. 2013) or exploiting the measurements from other sensors, like an inertial measurement unit (IMU) (Lee et al. 2014; Liu et al. 2017; Martyushev & Li 2020; Sweeney et al. 2014, 2015b). Typically, the assumption of planar motion or considering known vertical direction are common for self-driving cars and ground robots (Choi & Kim 2018; Guan et al. 2020; Hajder & Barath 2020; Li et al. 2020; Saurer et al. 2016), which makes the outlier removal more efficient and numerically more stable.

All previously mentioned relative pose solvers estimate the pose parameters from a set of point correspondences (PCs), *e.g.*, coming from SIFT (Lowe 2004) or SURF (Bay et al. 2008) detectors. Due to containing more information about the underlying surface geometry than PCs, ACs enable to estimate the pose from fewer correspondences. The ACs can be established by applying the traditional affine-covariant feature detectors (Mikolajczyk & Schmid 2002) or view-synthesizing approaches, such as ASIFT (Morel & Yu 2009), MODS (Mishkin et al. 2015), and Hes-Aff-Net (Mishkin et al. 2018). An AC yields three independent constraints on the epipolar geometry estimation (Barath & Hajder 2018; Bentolila & Francos 2014; Raposo & Barreto 2016). These geometric constraints are the basis for relative pose estimation in two-view geometry. In this paper, we focus on the relative pose estimation of a multi-camera system from ACs, instead of PCs. We propose three novel minimal solutions for the relative pose estimation of a multi-camera system. The contributions of this paper are:

- A new constraint that interprets the relationship of ACs and the generalized camera model is derived under general motion. This constraint can be used in special cases of multi-camera motion, *e.g.*, planar motion and known vertical direction.
- When the motion is planar (*i.e.*, the body to which the cameras are fixed moves on a plane; 3DOF), a single AC is sufficient to recover the planar motion of a multi-camera system. In order to deal with the degenerate case of the 1AC solver, we also propose a new method to estimate the relative pose from two ACs. The point-based solver (Lee et al. 2013) requires at least two PCs and requires the Ackermann motion model to hold.
- A third solver is proposed for the case when the vertical direction is known (4DOF), *e.g.*, from an IMU attached to the multi-camera system. We show that two ACs are enough to recover the relative pose. In contrast, the point-based solver requires four PCs (Lee et al. 2014; Sweeney et al. 2014).

This work is the extension of our previous conference paper (Guan et al. 2021b). The main differences are: discussion of degenerate cases, additional comparisons and real-world experiments, and more detailed derivations.

2 Related Work

Due to the absence of a single center of projection, the camera model of multi-camera systems is different from the standard pinhole camera. Pless proposed to express the light rays using Plücker coordinates of lines and derived the generalized camera model which has become a standard representation for the multi-camera systems (Pless 2003). Stewénius *et al.* proposed the first minimal solution to estimate the relative pose of a multi-camera system from 6 PCs, which produces up to 64 solutions (Stewénius et al. 2005). Li *et al.* provided several linear solvers to compute the relative pose, among which the most commonly used one requires 17 PCs (Li et al. 2008). Kneip *et al.* proposed an iterative approach for the relative pose estimation based on eigenvalue minimization (Kneip & Li 2014). Ventura *et al.* used the first-order approximation of the rotation to simplify the problem and estimated the relative pose from 6 PCs (Ventura et al. 2015). By considering additional motion constraints or using additional information provided by an IMU, the number of required PCs can be further reduced. Lee *et al.* presented a minimal solution with two PCs for the ego-motion estimation of a multi-camera system, constraining the relative motion by the Ackermann motion model (Lee et al. 2013). In addition, a variety of algorithms have been proposed when a common direction of the multi-camera system is known, *i.e.* an IMU provides the roll and pitch angles of the multi-camera system. The relative pose estimation with known vertical direction requires a minimal number of 4 PCs (Lee et al. 2014; Liu et al. 2017; Sweeney et al. 2014).

Exploiting the additional affine parameters besides the image coordinates has been recently proposed for the relative pose estimation of monocular cameras, which reduces the number of required points significantly. Bentolila *et al.* estimated the fundamental matrix from three ACs (Bentolila & Francos 2014). Raposo *et al.* computed homography and essential matrix

Table 1 Relative pose solvers for multi-camera systems.

Solver	Motion	Feature	Point #
Li et.al. (2008)	6DOF	PCs	17
Kneip et.al. (2014)	6DOF	PCs	8
Stewenius et.al. (2005)	6DOF	PCs	6
Ventura et.al. (2015)	6DOF	PCs	6
Alyousefi et.al. (2020)	6DOF	ACs	6
Lee et.al. (2014) Sweeney et.al. (2014) Liu et.al. (2017)	4DOF	PCs	4
Guan et.al. (2021)	6DOF	ACs	2
Lee et.al. (2013)	2DOF	PCs	2
1AC plane	3DOF	ACs	1
2AC plane	3DOF	ACs	2
2AC vertical	4DOF	ACs	2

using two ACs (Raposo & Barreto 2016). Barath *et al.* derived the constraints between the local affine transformation and the essential matrix and recovered the essential matrix from two ACs (Barath & Hajder 2018). Hajder *et al.* (Hajder & Barath 2020) and Guan *et al.* (Guan et al. 2020, 2021c) proposed several minimal solutions for relative pose from a single AC under the planar motion assumption or with the knowledge of a vertical direction. The above mentioned works are only suitable for the monocular perspective cameras which is different from the camera model of multi-camera systems. For multi-camera systems, Alyousefi and Ventura recently proposed a linear solver to estimate the relative pose using 6 ACs (Alyousefi & Ventura 2020). Guan *et al.* estimated the relative pose from 2 ACs by utilizing the first-order rotation approximation (Guan et al. 2021a). The above relative pose estimation algorithms are derived from the same geometric constraints of AC observations. The main difference of the algorithms is the different modeling and equation solving methods. In this paper, we focus on the minimal number of ACs to estimate the relative pose of a multi-camera system. Table 1 shows a summary of the relative pose solvers for multi-camera systems, including the DOF of the motion, feature types and number of points required. Since the proposed methods require the fewest correspondences, they can be more efficiently used within RANSAC for outlier removal and initial motion estimation in comparison with state-of-the-art methods.

3 Geometric Constraints from ACs

A multi-camera system is made up of multiple perspective cameras, as shown in Fig. 1. An AC in a multi-camera system relates two perspective cameras C_i and C_j across two consecutive frames, where C_i and C_j can be the same or different cameras. The extrinsic parameters of cameras C_i and C_j expressed in a multi-camera reference frame are represented as $(\mathbf{R}_i, \mathbf{t}_i)$ and $(\mathbf{R}_j, \mathbf{t}_j)$, respectively. Rotation matrices $\mathbf{R}_i$ and $\mathbf{R}_j$ represent relative rotations to the multi-camera reference frame. Translation vectors $\mathbf{t}_i$ and $\mathbf{t}_j$ represent relative translations to the multi-camera reference frame.

An AC consists of a point pair and a 2×2 local affine transformation. Let us denote the k-th AC between consecutive frames as $(\mathbf{x}_k, \mathbf{x}'_k, \mathbf{A}_k)$, where $\mathbf{x}_k$ and $\mathbf{x}'_k$ are the homogeneous image coordinates of the k-th feature point, which are captured by the camera C_i in the first frame and the camera C_j in the second frame, respectively. $\mathbf{A}_k$ is the related local affine transformation, which maps the infinitesimally close vicinity of $\mathbf{x}_k$ to that of $\mathbf{x}'_k$ (Barath 2018).

For general motion, there is a 3DOF relative rotation $\mathbf{R}$ and a 3DOF relative translation $\mathbf{t}$ between two reference frames. Rotation $\mathbf{R}$ using Cayley parameterization and translation $\mathbf{t}$ can be written as:

$$\mathbf{R} = \frac{1}{1+q_x^2+q_y^2+q_z^2} \cdot \begin{bmatrix} 1+q_x^2-q_y^2-q_z^2 & 2q_xq_y-2q_z & 2q_y+2q_xq_z \\ 2q_xq_y+2q_z & 1-q_x^2+q_y^2-q_z^2 & 2q_yq_z-2q_x \\ 2q_xq_z-2q_y & 2q_x+2q_yq_z & 1-q_x^2-q_y^2+q_z^2 \end{bmatrix}, \tag{1}$$

$$\mathbf{t} = [t_x \;\; t_y \;\; t_z]^{\mathrm{T}}, \tag{2}$$

where $[1, q_x, q_y, q_z]^{\mathrm{T}}$ is a homogeneous quaternion vector. Note that 180 degree rotations are prohibited when using the Cayley parameterization, but this is a rare case for consecutive frames.

3.1 Generalized Epipolar Constraint

We give a brief description of the generalized camera model (GCM) (Pless 2003). The image coordinates $(\mathbf{p}_k, \mathbf{p}'_k)$ expressed in the multi-camera reference frame are written as

$$\mathbf{p}_k = \mathbf{R}_i \mathbf{x}_k, \qquad \mathbf{p}'_k = \mathbf{R}_j \mathbf{x}'_k. \tag{3}$$

The unit direction of rays $(\mathbf{u}_k, \mathbf{u}'_k)$ expressed in the multi-camera reference frame are given as: $\mathbf{u}_k = \mathbf{p}_k/\|\mathbf{p}_k\|$, $\mathbf{u}'_k = \mathbf{p}'_k/\|\mathbf{p}'_k\|$. The 6-dimensional Plücker coordinates corresponding to the rays are denoted as $\mathbf{l}_k = [\mathbf{u}_k^{\mathrm{T}}, \; (\mathbf{t}_i \times \mathbf{u}_k)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{l}'_k = [\mathbf{u}'_k{}^{\mathrm{T}}, \; (\mathbf{t}_j \times \mathbf{u}'_k)^{\mathrm{T}}]^{\mathrm{T}}$. The symbol $\times$ represents the cross product. The generalized epipolar constraint is written as (Pless 2003)

$$\mathbf{l}'^{\mathrm{T}}_k \begin{bmatrix} [\mathbf{t}]_\times \mathbf{R}, & \mathbf{R} \\ \mathbf{R}, & \mathbf{0} \end{bmatrix} \mathbf{l}_k = 0, \tag{4}$$

where $\mathbf{l}_k$ and $\mathbf{l}'_k$ are the Plücker coordinates of a line correspondence between two consecutive frames. The symbol $[\mathbf{t}]_\times$ represents the skew-symmetric matrix of the translation $\mathbf{t}$.

3.2 Affine Transformation Constraint

We denote the transition matrix between the camera coordinate system C_i in the first frame and the camera coordinate system C_j in the second frame as $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$, which is represented as:

$$\begin{aligned} \begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_j & \mathbf{t}_j \\ \mathbf{0} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_j^{\mathrm{T}} \mathbf{R} \mathbf{R}_i & \mathbf{R}_j^{\mathrm{T}} (\mathbf{R}\mathbf{t}_i + \mathbf{t} - \mathbf{t}_j) \\ \mathbf{0} & 1 \end{bmatrix}. \end{aligned} \tag{5}$$

Essential matrix $\mathbf{E}_k$ of the two consecutive frames is

$$\mathbf{E}_k = [\mathbf{t}_{ij}]_\times \mathbf{R}_{ij} = \mathbf{R}_j^\mathrm{T}[\mathbf{R}_i\mathbf{t}_{ij}]_\times \mathbf{R}\mathbf{R}_i, \tag{6}$$

where $[\mathbf{R}_i\mathbf{t}_{ij}]_\times = \mathbf{R}[\mathbf{t}_i]_\times\mathbf{R}^\mathrm{T} + [\mathbf{t}]_\times - [\mathbf{t}_j]_\times$. The relationship of essential matrix $\mathbf{E}_k$ and local affine transformation $\mathbf{A}_k$ is formulated as follows (Barath & Hajder 2018):

$$(\mathbf{E}_k^\mathrm{T}\mathbf{x}'_k)_{(1:2)} = -(\hat{\mathbf{A}}_k^\mathrm{T}\mathbf{E}_k\mathbf{x}_k)_{(1:2)}, \tag{7}$$

where $\mathbf{E}_k^\mathrm{T}\mathbf{x}'_k$ and $\mathbf{E}_k\mathbf{x}_k$ denote the epipolar lines in their implicit form in the frames of cameras C_i and C_j. Subscripts 1 and 2 represent the first and second equations of the equation system, respectively. $\hat{\mathbf{A}}_k$ is a 3×3 matrix, which can be written as:

$$\hat{\mathbf{A}}_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}.$$

By substituting Eq. (6) into Eq. (7), we obtain:

$$\begin{aligned}(\mathbf{R}_i^\mathrm{T}\mathbf{R}^\mathrm{T}[\mathbf{R}_i\mathbf{t}_{ij}]_\times^\mathrm{T}\mathbf{R}_j\mathbf{x}'_k)_{(1:2)} \\ = -(\hat{\mathbf{A}}_k^\mathrm{T}\mathbf{R}_j^\mathrm{T}[\mathbf{R}_i\mathbf{t}_{ij}]_\times\mathbf{R}\mathbf{R}_i\mathbf{x}_k)_{(1:2)}.\end{aligned} \tag{8}$$

Based on Eq. (3), the above equation is reformulated and expanded as follows:

$$\begin{aligned}(\mathbf{R}_i^\mathrm{T}([\mathbf{t}_i]_\times\mathbf{R}^\mathrm{T} + \mathbf{R}^\mathrm{T}[\mathbf{t}]_\times - \mathbf{R}^\mathrm{T}[\mathbf{t}_j]_\times)\mathbf{p}'_k)_{(1:2)} = \\ (\hat{\mathbf{A}}_k^\mathrm{T}\mathbf{R}_j^\mathrm{T}(\mathbf{R}[\mathbf{t}_i]_\times + [\mathbf{t}]_\times\mathbf{R} - [\mathbf{t}_j]_\times\mathbf{R})\mathbf{p}_k)_{(1:2)}.\end{aligned} \tag{9}$$

Eq. (9) interprets the new epipolar constraints which a local affine transformation implies on cameras C_i and C_j in two consecutive frames. It can be seen that an AC yields three independent constraints from Eqs. (4) and (9). When an AC in a multi-camera system relates the same perspective camera across two consecutive frames, *i.e.*, C_i and C_j are the same camera ($\mathbf{R}_i = \mathbf{R}_j, \mathbf{t}_i = \mathbf{t}_j$), the constraints of Eqs. (4) and (9) still hold.

For each AC $(\mathbf{x}_k, \mathbf{x}'_k, \mathbf{A}_k)$, we get three polynomials for six unknowns $\{q_x, q_y, q_z, t_x, t_y, t_z\}$ based on Eqs. (4) and (9). After separating q_x, q_y, q_z from t_x, t_y, t_z, we arrive at equation system

$$\frac{1}{1+q_x^2+q_y^2+q_z^2}\underbrace{\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \end{bmatrix}}_{\mathbf{M}(q_x,q_y,q_z)}\begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} = \mathbf{0}, \tag{10}$$

where the elements of the coefficient matrix $\mathbf{M}(q_x, q_y, q_z)$ are formed by the polynomial coefficients and three unknown variables q_x, q_y, q_z. All the elements are quadratic polynomials with three variables q_x, q_y, q_z. Note that the multiple $1/(1+q_x^2+q_y^2+q_z^2)$ is not simply omitted in this paper. As we will see later, the multiple can be used to reduce the polynomial degree and improve the efficiency of the solution.

Equation (10) imposes three independent constraints on six unknowns $\{q_x, q_y, q_z, t_x, t_y, t_z\}$. Motivated by scenarios like self-driving cars, ground robots or AR headsets, we investigate relevant special cases of multi-camera motion, *i.e.*, planar motion and motion with known vertical direction, see Figs. 2 and 4. The constraint equations Eq. (10) can be used in special cases of multi-camera motion. We show that two special cases can be efficiently solved with ACs.

4 Relative Pose under Planar Motion

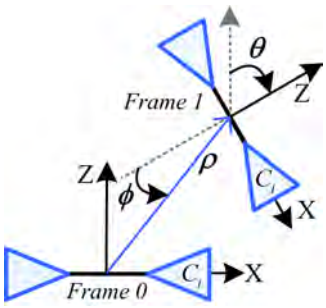


Fig. 2 Relative pose estimation under planar motion in top-view. There are three unknowns: yaw angle θ, translation direction ϕ and translation distance ρ.

When assuming that the body, to which the camera system is rigidly fixed, moves on a planar surface (as visualized in Fig. 2), there are only a y-axis rotation and 2D translation between reference frames. Similar to Eqs. (1) and (2), the rotation $\mathbf{R} = \mathbf{R}_y$ and the translation $\mathbf{t}$ from the first frame to the second frame is written as:

$$\begin{aligned}\mathbf{R}_y &= \frac{1}{1+q_y^2}\begin{bmatrix} 1-q_y^2 & 0 & -2q_y \\ 0 & 1+q_y^2 & 0 \\ 2q_y & 0 & 1-q_y^2 \end{bmatrix}, \\ \mathbf{t} &= [t_x \;\; 0 \;\; t_z]^\mathrm{T}.\end{aligned} \tag{11}$$

where $q_y = \tan(\frac{\theta}{2})$, $t_x = \rho\sin(\phi)$, $t_z = -\rho\cos(\phi)$, ρ is the distance between two multi-camera reference frames.

4.1 Solver for Planar Motion

By substituting Eq. (11) into Eqs. (4) and (9), we get an equation system of three polynomials for three unknowns q_y, t_x and t_z. Since an AC generally provides three independent constraints for relative pose, a single AC is sufficient to recover the planar motion of a multi-camera system. After separating q_y from t_x, t_z, the three independent constraints from an AC form matrix equation:

$$\frac{1}{1+q_y^2}\underbrace{\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}}_{\mathbf{M}(q_y)}\begin{bmatrix} t_x \\ t_z \\ 1 \end{bmatrix} = \mathbf{0}, \tag{12}$$

where the elements of the coefficient matrix $\mathbf{M}(q_y)$ are formed by the polynomial coefficients and one unknown variable q_y. All the elements are quadratic polynomials with variable q_y. Since $\mathbf{M}(q_y)$ is a square matrix, Eq. (12) has a non-trivial solution only if the determinant of $\mathbf{M}(q_y)/(1+q_y^2)$ vanishes. The expansion of

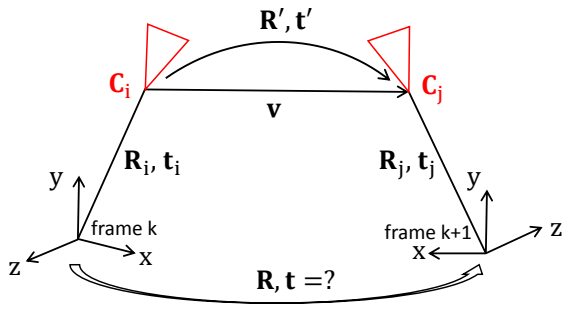


Fig. 3 Degenerate case under planar motion.

$\det(\mathbf{M}(q_y)/(1+q_y^2)) = 0$ gives a 4-degree univariate polynomial as follows:

$$\text{quot}(\sum_{i=0}^{6} w_i q_y^i, q_y^2+1) = 0, \tag{13}$$

where $\text{quot}(a,b)$ means calculating the quotient of a divided by b. $w_0, \ldots, w_6$ are formed by the Plücker coordinates of a line correspondence and the matrix elements of an affine transformation between the corresponding feature points.

Note that the coefficients are divided by q_y^2+1 which reduces the polynomial degree and improves the efficiency of the solution. The univariate polynomial Eq. (13) leads to an explicit analytic solution with a maximum of 4 real roots. Once the solutions for q_y are found, the remaining unknowns t_x and t_z are solved by substituting q_y into $\mathbf{M}(q_y)$ and solving the linear system via calculating its null vector. Finally, the rotation matrix $\mathbf{R}_y$ is recovered from Eq. (11).

4.2 Degenerate Case

In this section, we show that the solver using a single AC has a degenerate case under planar motion, *i.e.*, when the distances between the motion plane and optical centers of the cameras are equal.

Degenerate condition: *Consider a multi-camera system which is under planar motion. Assume the following three conditions are satisfied. (1) The rotation axis is the y-axis, and the translation is on xz-plane. (2) There is a single AC across camera C_i in the first frame and camera C_j in the second frame (C_i and C_j can be the same or different cameras). (3) The optical centers of cameras C_i and C_j have the same y-coordinate. Then this case is degenerate. Specifically, the rotation can be correctly recovered, while neither the translation direction nor the translation scale can be estimated.*

Interpretation: Fig. 3 illustrates the degenerate case under planar motion. Note that the multi-camera reference frame is established in the multi-camera system, not in a certain camera coordinate system. Our interpretation is based on the following observation: whether a case is degenerate does not depend on which relative pose estimation solver is used for recovering $(\mathbf{R}, \mathbf{t})$. Based on this point, we construct a new relative pose estimation solver which is different from the proposed solver in Section 4.1.

(i) Since the multi-camera system is rotated around the y-axis, camera C_i in the first frame and camera C_j in the second frame are under motion with known rotation axis. For the relative pose estimation problem of monocular cameras with known rotation axis, it has been proven that a single AC is sufficient to estimate the relative rotation and translation (only known up to scale) between C_i and C_j (Guan et al. 2020). This is a minimal solver since one AC provides 3 independent constraints and there are three unknowns (one unknown for rotation, two unknowns for translation by excluding scale-ambiguity). Denote the recovered rotation and translation between C_i and C_j as $(\mathbf{R}', \mathbf{t}')$, where $\mathbf{t}'$ is a unit vector. The scale of the translation vector cannot be recovered at this moment. Denote the unknown translation scale as λ.

(ii) From Fig. 3, we have

$$\begin{aligned}\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & 1\end{bmatrix} &= \begin{bmatrix}\mathbf{R}_j & \mathbf{t}_j\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}' & \lambda\mathbf{t}'\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}^{-1}\\ &= \begin{bmatrix}\mathbf{R}_j\mathbf{R}'\mathbf{R}_i^{\mathrm{T}} & \lambda\mathbf{R}_j\mathbf{t}' + \mathbf{t}_j - \mathbf{R}_j\mathbf{R}'\mathbf{R}_i^{\mathrm{T}}\mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}.\end{aligned} \tag{14}$$

From Eq. (14), we have

$$\mathbf{R} = \mathbf{R}_j\mathbf{R}'\mathbf{R}_i^{\mathrm{T}}, \tag{15}$$

$$\mathbf{t} = \lambda\mathbf{R}_j\mathbf{t}' + \mathbf{t}_j - \mathbf{R}_j\mathbf{R}'\mathbf{R}_i^{\mathrm{T}}\mathbf{t}_i. \tag{16}$$

From Eq. (15), the rotation $\mathbf{R}$ between the first frame and the second frame of the multi-camera system can be recovered. From Eq. (16), we have

$$\lambda(\mathbf{R}_j\mathbf{t}') - \mathbf{t} + (\mathbf{t}_j - \mathbf{R}\mathbf{t}_i) = \mathbf{0}. \tag{17}$$

In Eq. (17), note that $\mathbf{t} = [t_x, 0, t_z]^{\mathrm{T}}$ due to the planar motion assumption. Thus this linear equation system has 3 unknowns $\{\lambda, t_x, t_z\}$ and 3 equations. Usually, the unknowns can be uniquely determined by solving this equation system. However, if the second entry of $\mathbf{R}_j\mathbf{t}'$ is zero, three unknowns $\{\lambda, t_x, t_z\}$ cannot be uniquely computed. In other words, the translation direction and the translation scale cannot be determined. This is a degenerate case.

(iii) Finally, we exploit the geometric meaning of the degenerate case, *i.e.*, the second entry of $\mathbf{R}_j\mathbf{t}'$ is zero. Denote the normalized vector originated from C_i to C_j as $\mathbf{v}$. Since $\mathbf{v}$ represents the normalized translation vector between C_i and C_j, the coordinates of $\mathbf{v}$ in reference of camera C_j is $\mathbf{t}'$. Further, the coordinates of $\mathbf{v}$ in the second frame is $\mathbf{R}_j\mathbf{t}'$. The second entry of $\mathbf{R}_j\mathbf{t}'$ being zero means that the endpoints of $\mathbf{v}$ have the same y-coordinate in the second frame, which is the condition (3) in the degenerate condition.

This degenerate case might happen in the self-driving scenario, leading to the situation when neither the translation direction, nor its scale can be estimated from a single AC. For example, when a multi-camera system undergoes planar motion and a single AC is captured by the same camera over two consecutive frames, this case is degenerate. To overcome this issue, we use two ACs. For example, the first and second constraints of the first AC, and the first constraint of the second AC are selected as the three equations to be solved in the three unknowns, just as Eq. (12). Note that the

steps of the solver remain the same, except for the code constructing coefficient matrix $\mathbf{M}(q_y)$.

5 Relative Pose with Known Vertical Direction

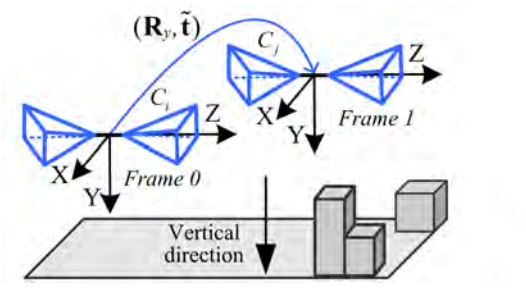


Fig. 4 Relative pose estimation with known vertical direction. There are four unknowns: a y-axis rotation $\mathbf{R}_y$ and 3D translation $\tilde{\mathbf{t}} = [\tilde{t}_x, \tilde{t}_y, \tilde{t}_z]^{\mathrm{T}}$.

In this section a minimal solution using two ACs (at least one AC and one PC) is proposed for relative motion estimation for multi-camera systems with known vertical direction, see Fig. 4. In this case, an IMU is coupled with the multi-camera system and the relative rotation between the IMU and the reference frame is known. The IMU provides the known roll and pitch angles for the multi-camera reference frame.

5.1 Apply Roll and Pitch Angles

Based on the roll and pitch angles provided by the IMU, the multi-camera reference frame can be aligned with the measured vertical direction, such that the X-Z-plane of the aligned reference frame is parallel to the ground plane and the y-axis is parallel to the vertical direction. Let us denote the rotation matrices from the roll and pitch angles of the two corresponding multi-camera reference frames as $\mathbf{R}_{\mathrm{imu}}$ and $\mathbf{R}'_{\mathrm{imu}}$. Take the composition of the rotation matrix $\mathbf{R}_{\mathrm{imu}}$ for an example. Rotation matrix $\mathbf{R}_{\mathrm{imu}}$ for aligning the reference frame can be computed as follows:

$$\begin{aligned}\mathbf{R}_{\mathrm{imu}} &= \mathbf{R}_p\mathbf{R}_r \\ &= \begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\theta_p) & \sin(\theta_p)\\ 0 & -\sin(\theta_p) & \cos(\theta_p)\end{bmatrix}\begin{bmatrix}\cos(\theta_r) & \sin(\theta_r) & 0\\ -\sin(\theta_r) & \cos(\theta_r) & 0\\ 0 & 0 & 1\end{bmatrix},\end{aligned}$$

where θ_r and θ_p are roll and pitch angles provided by the IMU, respectively.

There are only a y-axis rotation $\mathbf{R} = \mathbf{R}_y$ and 3D translation $\tilde{\mathbf{t}} = [\tilde{t}_x, \tilde{t}_y, \tilde{t}_z]^{\mathrm{T}}$ to be estimated between the aligned multi-camera reference frames. By leveraging IMU measurement, the transition matrix between two reference frames can be represented as follows:

$$\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & 1\end{bmatrix} = \begin{bmatrix}\mathbf{R}'_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{R}_y & \tilde{\mathbf{t}}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}. \tag{18}$$

From Eq. (18), the relative pose between two reference frames can be written as:

$$\mathbf{R} = (\mathbf{R}'_{\mathrm{imu}})^{\mathrm{T}}\mathbf{R}_y\mathbf{R}_{\mathrm{imu}}, \tag{19}$$

$$\mathbf{t} = (\mathbf{R}'_{\mathrm{imu}})^{\mathrm{T}}\tilde{\mathbf{t}}. \tag{20}$$

5.2 Geometric Constraints with Known Vertical Direction

In this case, we show that the geometric constraints in Section 3 can be generalized to the multi-camera motion with known vertical direction. By substituting Eq. (19) into Eq. (4), the generalized epipolar constraint with known vertical direction is written as

$$\underbrace{\left(\begin{bmatrix}\mathbf{R}'_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}'_{\mathrm{imu}}\end{bmatrix}\mathbf{l}'_k\right)^{\mathrm{T}}}_{\tilde{\mathbf{l}}'_k}\begin{bmatrix}[\tilde{\mathbf{t}}]_\times\mathbf{R}_y & \mathbf{R}_y\\ \mathbf{R}_y & \mathbf{0}\end{bmatrix}\underbrace{\left(\begin{bmatrix}\mathbf{R}_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}_{\mathrm{imu}}\end{bmatrix}\mathbf{l}_k\right)}_{\tilde{\mathbf{l}}_k} = 0, \tag{21}$$

where $\tilde{\mathbf{l}}_k \leftrightarrow \tilde{\mathbf{l}}'_k$ are the corresponding Plücker coordinates of line correspondences expressed in the aligned multi-camera reference frame.

Next, we derive the affine transformation constraint with known vertical direction. Substituting Eq. (18) into Eq. (5) yields

$$\begin{bmatrix}\mathbf{R}_{ij} & \mathbf{t}_{ij}\\ \mathbf{0} & 1\end{bmatrix} = \left(\begin{bmatrix}\mathbf{R}'_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_j & \mathbf{t}_j\\ \mathbf{0} & 1\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{R}_y & \tilde{\mathbf{t}}\\ \mathbf{0} & 1\end{bmatrix}\left(\begin{bmatrix}\mathbf{R}_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}\right). \tag{22}$$

We denote that

$$\begin{aligned}\begin{bmatrix}\tilde{\mathbf{R}}_{\mathrm{imu}} & \tilde{\mathbf{t}}_{\mathrm{imu}}\\ \mathbf{0} & 1\end{bmatrix} &= \begin{bmatrix}\mathbf{R}_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix},\\ \begin{bmatrix}\tilde{\mathbf{R}}'_{\mathrm{imu}} & \tilde{\mathbf{t}}'_{\mathrm{imu}}\\ \mathbf{0} & 1\end{bmatrix} &= \begin{bmatrix}\mathbf{R}'_{\mathrm{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_j & \mathbf{t}_j\\ \mathbf{0} & 1\end{bmatrix}.\end{aligned} \tag{23}$$

By substituting Eq. (23) into Eq. (22), we obtain

$$\begin{aligned}&\begin{bmatrix}\mathbf{R}_{ij} & \mathbf{t}_{ij}\\ \mathbf{0} & 1\end{bmatrix}\\ &= \begin{bmatrix}(\tilde{\mathbf{R}}'_{\mathrm{imu}})^{\mathrm{T}}\mathbf{R}_y\tilde{\mathbf{R}}_{\mathrm{imu}} & (\tilde{\mathbf{R}}'_{\mathrm{imu}})^{\mathrm{T}}(\mathbf{R}_y\tilde{\mathbf{t}}_{\mathrm{imu}} + \tilde{\mathbf{t}} - \tilde{\mathbf{t}}'_{\mathrm{imu}})\\ \mathbf{0} & 1\end{bmatrix}.\end{aligned} \tag{24}$$

It can be seen that Eq. (24) has a similar composition to Eq. (5). Similar to Eq. (9), the affine transformation constraint with known vertical direction can be directly given as

$$\begin{aligned}&(\tilde{\mathbf{R}}^{\mathrm{T}}_{\mathrm{imu}}([\tilde{\mathbf{t}}_{\mathrm{imu}}]_\times\mathbf{R}_y^{\mathrm{T}} + \mathbf{R}_y^{\mathrm{T}}[\tilde{\mathbf{t}}]_\times - \mathbf{R}_y^{\mathrm{T}}[\tilde{\mathbf{t}}'_{\mathrm{imu}}]_\times)\tilde{\mathbf{p}}'_k)_{(1:2)} =\\ &(\hat{\mathbf{A}}_k^{\mathrm{T}}(\tilde{\mathbf{R}}'_{\mathrm{imu}})^{\mathrm{T}}(\mathbf{R}_y[\tilde{\mathbf{t}}_{\mathrm{imu}}]_\times + [\tilde{\mathbf{t}}]_\times\mathbf{R}_y - [\tilde{\mathbf{t}}'_{\mathrm{imu}}]_\times\mathbf{R}_y)\tilde{\mathbf{p}}_{ij})_{(1:2)}\end{aligned} \tag{25}$$

where $\tilde{\mathbf{p}}_k = \tilde{\mathbf{R}}_{\mathrm{imu}}\mathbf{x}_k$ and $\tilde{\mathbf{p}}'_k = \tilde{\mathbf{R}}'_{\mathrm{imu}}\mathbf{x}'_k$ are the image coordinates expressed in the aligned multi-camera reference frame.

5.3 Solver for Motion with Known Vertical Direction

Based on Eqs. (21) and (25), we get an equation system of three polynomials for four unknowns q_y, $\tilde{t}_x$, $\tilde{t}_y$ and $\tilde{t}_z$. Recall that there are three independent constraints provided by one AC. Thus, one more equation is required which can be taken from a second AC. In principle, one arbitrary equation can be chosen from Eqs. (21) and (25), for example, three constraints of the first AC, and the first constraint of the second AC, *i.e.*, four constraints provided by one AC and one PC, are stacked into 4 equations in 4 unknowns:

$$\frac{1}{1+q_y^2} \underbrace{\begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} & \tilde{M}_{13} & \tilde{M}_{14} \\ \tilde{M}_{21} & \tilde{M}_{22} & \tilde{M}_{23} & \tilde{M}_{24} \\ \tilde{M}_{31} & \tilde{M}_{32} & \tilde{M}_{33} & \tilde{M}_{34} \\ \tilde{M}_{41} & \tilde{M}_{42} & \tilde{M}_{43} & \tilde{M}_{44} \end{bmatrix}}_{\tilde{\mathbf{M}}(q_y)} \begin{bmatrix} \tilde{t}_x \\ \tilde{t}_y \\ \tilde{t}_z \\ 1 \end{bmatrix} = \mathbf{0}, \tag{26}$$

where the elements of the coefficient matrix $\tilde{\mathbf{M}}(q_y)$ are formed by the polynomial coefficients and one unknown variable q_y. All the elements are quadratic polynomials with variable q_y. Since $\tilde{\mathbf{M}}(q_y)/(1+q_y^2)$ is a square matrix, Eq. (26) has a non-trivial solution only if $\det(\tilde{\mathbf{M}}(q_y)/(1+q_y^2)) = 0$. The expansion of the determinant equation gives a 6-degree univariate polynomial:

$$\text{quot}(\sum_{i=0}^{8} w_i q_y^i, q_y^2+1) = 0, \tag{27}$$

where $\tilde{w}_0, \ldots, \tilde{w}_8$ are formed by the Plücker coordinates of two line correspondences and the matrix elements of two affine transformations between the corresponding feature points.

This univariate polynomial leads to a maximum of 6 solutions. Equation (27) can be efficiently solved by the companion matrix method (Cox et al. 2013) or Sturm bracketing method (Nistér 2004). Once q_y has been obtained, rotation matrix $\mathbf{R}_y$ is recovered from Eq. (11). For the relative pose between two multi-camera reference frames, the rotation matrix $\mathbf{R}$ is recovered from Eq. (19) and the translation is computed by $\mathbf{t} = (\mathbf{R}'_{\text{imu}})^{\text{T}}\tilde{\mathbf{t}}$ based on Eq. (20).

6 Experiments

In this section, we conduct extensive experiments on both synthetic and real-world data to evaluate the performance of the proposed methods. Our solvers are compared with state-of-the-art techniques.

For relative pose estimation under planar motion, the solvers using one AC and two ACs proposed in Section 4 are referred to as `1AC plane` method and `2AC plane` method, respectively. The accuracy of `1AC plane` and `2AC plane` are compared with the methods `17pt-Li` (Li et al. 2008), `8pt-Kneip` (Kneip & Li 2014), `6pt-Stewénius` (Stewénius et al. 2005), `2pt-Lee` (Lee et al. 2013) and `6AC-Ventura` (Alyousefi & Ventura 2020).

For relative pose estimation with known vertical direction, the solver proposed in Section 5 is referred to as `2AC vertical` method. We compare the accuracy of `2AC vertical` with the methods `17pt-Li` (Li et al. 2008), `8pt-Kneip` (Kneip & Li 2014), `6pt-Stewénius` (Stewénius et al. 2005), `6pt-Ventura` (Ventura et al. 2015), `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014), `4pt-Liu` (Liu et al. 2017), `6AC-Ventura` (Alyousefi & Ventura 2020) and `2AC-Guan` (Guan et al. 2021a).

In the experiments, all the solvers are integrated within RANSAC to reject outliers. For the point-based solvers, only the point coordinates of ACs are used. The relative pose which produces the highest number of inliers is chosen. The confidence of RANSAC is set to 0.99 and an inlier threshold angle is set to 0.1° by following the definition in OpenGV (Kneip & Furgale 2014). We also show the feasibility of our methods on the `KITTI` dataset (Geiger et al. 2013), `nuScenes` dataset (Caesar et al. 2020) and `EuRoc` MAV dataset (Burri et al. 2016). These experiment demonstrates that our methods are well suited for visual odometry in real scenarios.

6.1 Efficiency Comparison

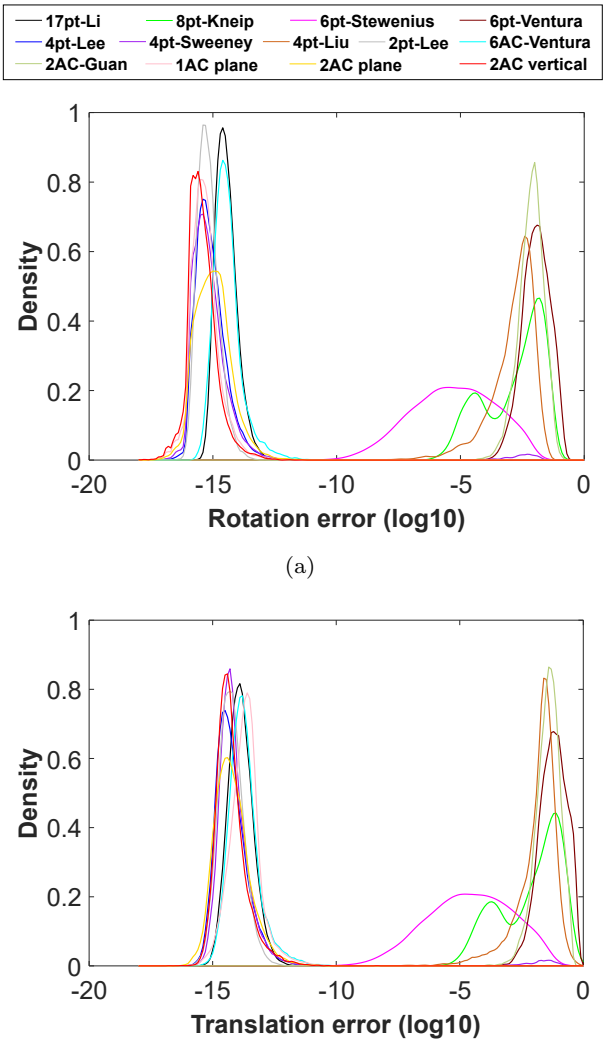
The runtimes of the solvers are evaluated on an Intel(R) Core(TM) i7-7800X 3.50GHz. All algorithms are implemented in C++. Methods `17pt-Li`, `8pt-Kneip` and `6pt-Stewenius` are provided in the OpenGV library (Kneip & Furgale 2014). We implemented the methods `4pt-Lee`, `2pt-Lee` and `2AC-Guan`. For methods `6pt-Ventura`, `4pt-Sweeney`, `4pt-Liu` and `6AC-Ventura`, we used their publicly available implementations from GitHub. The average, over 10,000 runs, processing times of the solvers are shown in Table 2. The runtimes of the methods `4pt-Liu`, `1AC plane` and `2AC plane` are the lowest, because these methods solve the 4-degree polynomial equation. The methods `2pt-Lee` and `2AC vertical` which solves the 6-degree polynomial equation also requires low computation time.

6.2 Numerical Stability

Figure 5 reports the numerical stability of the solvers in the noise-free case. The procedure is repeated 10,000 times. The empirical probability density functions (vertical axis) are plotted as the function of the $\log_{10}$ estimated errors (horizontal axis). Methods `1AC plane`, `2AC plane`, `2AC vertical`, `17pt-Li`, `4pt-Lee`, `4pt-Sweeney`, `2pt-Lee` and `6AC-Ventura` are numerically stable. The `4pt-Sweeney` method has a small peak, both in the rotation and translation error curves, around 10^{-2}. The corresponding density of the small peak is about 0.02. The `8pt-Kneip` method based on iterative optimization is susceptible to falling into local minima. Due to the use of first-order approximation of the relative rotation, the methods `6pt-Ventura`, `4pt-Liu` and `2AC-Guan` inevitably has greater than zero error in the noise-free case.

Table 2 Run-time comparison of relative pose estimation algorithms (unit: μs).

Methods	17pt-Li	8pt-Kneip	6pt-St.	6pt-Ven.	4pt-Lee	4pt-Sw.	4pt-Liu	2pt-Lee	6AC-Ven.	2AC-Guan	**1AC plane**	**2AC plane**	**2AC vertical**
Timings	43.3	102.0	3275.4	29.8	26.5	22.2	3.7	5.3	38.1	28.6	**3.6**	**3.6**	17.8



(a)

(b)

Fig. 5 Probability density functions over estimation errors in the noise-free case (10 000 runs). The horizontal axis represents the $\log_{10}$ errors and the vertical axis represents the density. Plot (a) reports the rotation error. Plot (b) reports the translation error. The proposed `1AC plane` method, `2AC plane` method and `2AC vertical` are compared against `17pt-Li` (Li et al. 2008), `8pt-Kneip` (Kneip & Li 2014), `6pt-Stewénius` (Stewénius et al. 2005), `6pt-Ventura` (Ventura et al. 2015), `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014), `4pt-Liu` (Liu et al. 2017), `2pt-Lee` (Lee et al. 2013), `6AC-Ventura` (Alyousefi & Ventura 2020) and `2AC-Guan` (Guan et al. 2021a).

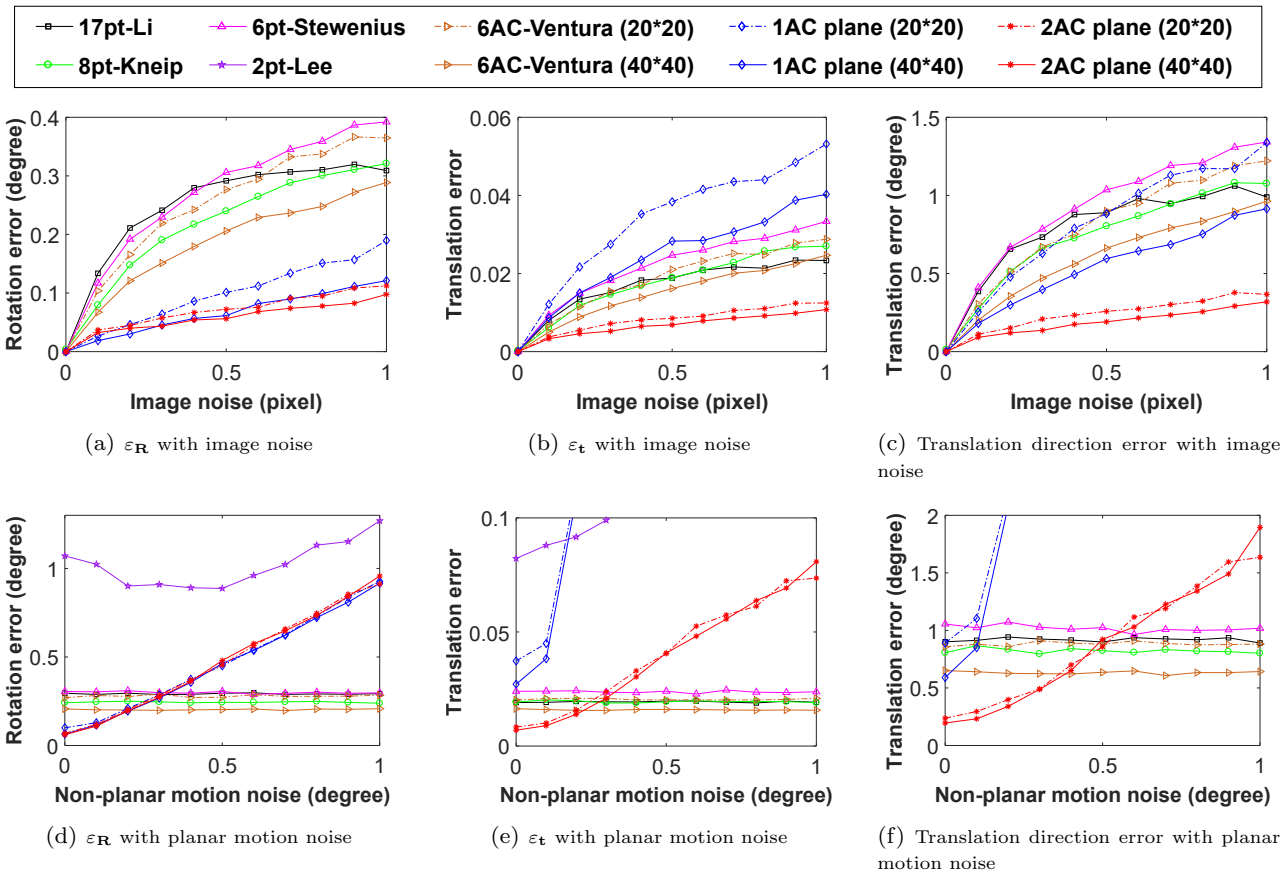
6.3 Experiments on Synthetic Data

We made a simulated 2-camera rig system by following the KITTI autonomous driving platform. The baseline length between two simulated cameras is set to 1 meter and the cameras are installed at different altitudes. The multi-camera reference frame is set at the center of the camera rig and the translation between two multi-camera reference frames is 3 meters. The resolution of the cameras is 640 $\times$ 480 pixels and the focal lengths are 400 pixels. The principal points are set to the image center (320, 240).

The synthetic scene is composed of a ground plane and 50 random planes. All 3D planes are randomly generated within the range of -5 to 5 meters (along axes X and Y), and 10 to 20 meters (Z-axis direction), that are expressed in the respective axis of the multi-camera reference frame. The equation of 3D plane can be represented as $AX + BY + CZ + D = 0$. The normal vector to the 3D plane is given by $\mathbf{N} = [A, B, C]^T$. For a ground plane, the corresponding normal vector is set to $[0, 1, 0]^T$, which is consistent with the Y-axis direction of the multi-camera reference frame. For 50 random planes, the corresponding normal vectors are randomly generated. Then, we choose a random 3D point (X_0, Y_0, Z_0) in the range of the synthetic scene. Finally, the parameter D can be computed based on the normal vector and the chosen 3D point. Thus, by the above procedure, the ground plane and 50 random planes can be randomly generated in the synthetic scene.

We choose 50 ACs from the ground plane and an AC from each random plane randomly, thus, having a total of 100 ACs. For each AC, a random 3D point from a 3D plane (X_0, Y_0, Z_0) is reprojected onto two cameras to get the image point pair $(\mathbf{x}_k, \mathbf{x}'_k)$. The corresponding affine transformation $(\mathbf{A}_k)$ is obtained by the following procedure. First, four sampled image points are chosen as the vertices of a square in the 2D image plane of the first frame, where the center of the square is the point coordinates of AC. The side length of the square W is set as 20 or 40 pixels. A larger side length causes smaller noise of affine transformation. The four sampled image point in the first frame can be computed as follows: $(\mathbf{x}_k + [-W/2,\ -W/2]^T, \mathbf{x}_k + [W/2,\ -W/2]^T, \mathbf{x}_k + [-W/2,\ W/2]^T, \mathbf{x}_k + [W/2,\ W/2]^T)$. Second, the four corresponding sampled image points in the second frame are directly calculated by the ground truth homography. Third, four sampled point pairs are contaminated by Gaussian noise, which is similar to the noise added to the coordinates of image point pair. Fourth, the noisy homography matrix is estimated using the four sampled point pairs. The noisy affine transformation is the first-order approximation of the noisy homography matrix. The implied noisy local affine frame is then calculated via projective geometry. This can be seen as perturbing the 3D plane centered on the observed 3D point. This procedure enables an indirect but geometrically interpretable way of adding noise to the affine transformation (Barath & Kukelova 2019).

A total of 1000 trials are carried out in the synthetic experiment. In each test, 100 ACs are generated randomly. The ACs for the methods are selected randomly and the error is measured on the relative pose which produces the most inliers within the RANSAC



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

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

(c) Translation direction error with image noise

(d) $\varepsilon_{\mathbf{R}}$ with planar motion noise

(e) $\varepsilon_{\mathbf{t}}$ with planar motion noise

(f) Translation direction error with planar motion noise

Fig. 6 Rotation and translation error under planar motion. (a–c): varying image noise under perfect planar motion. (d–f): varying planar motion noise and fixed 0.5 pixel std. image noise.

scheme. This also allows us to select the best candidate from multiple solutions by counting their inliers in a RANSAC-like procedure. The median of errors are used to assess the rotation and translation error. The rotation error is computed as the angular difference between the ground truth rotation and the estimated rotation: $\varepsilon_{\mathbf{R}} = \arccos((\text{trace}(\mathbf{R}_{gt}\mathbf{R}^{\mathrm{T}}) - 1)/2)$, where $\mathbf{R}_{gt}$ and $\mathbf{R}$ are the ground truth and estimated rotation matrices. Following the definition in (Lee et al. 2014; Quan & Lan 1999), the translation error is defined as: $\varepsilon_{\mathbf{t}} = 2\,\|(\mathbf{t}_{gt} - \mathbf{t})\| / (\|\mathbf{t}_{gt}\| + \|\mathbf{t}\|)$, where $\mathbf{t}_{gt}$ and $\mathbf{t}$ are the ground truth and estimated translations. Due to the limited display range of the figures, some curves with large errors are invisible or partially invisible.

6.3.1 Planar Motion Estimation

In this scenario, the planar motion of the multi-camera system is described by (θ, ϕ), see Fig. 2. The magnitudes of both angles ranges from $-10°$ to $10°$. Suppose we are given Gaussian image noise with a standard deviation ranging from 0 to 1 pixel. Fig. 6(a–c) shows the performance of the proposed `1AC plane` and `2AC plane` methods against image noise. Since the noise magnitude of affine transformation is influenced by the support region of sampled points, the AC-based methods have better performance with larger support region at the same magnitude of image noise. It can be seen that `2AC plane` performs better than the other compared methods under perfect planar motion, even though the size of the square is 20 pixels. The `1AC plane` method performs better than the PC-based methods and the `6AC-Ventura` method in rotation estimation, but has worse performance in translation estimation. Since the planar motion of the multi-camera system does not satisfy the Ackermann motion

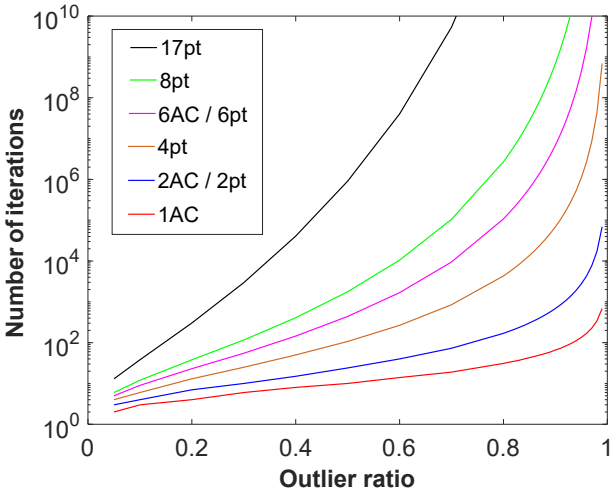


Fig. 7 Comparison of the RANSAC iteration number for 99% of success probability.

assumption, the `2pt Lee` method has large errors and its error curves are out of the display range. As shown later in Section 6.3.3, the `2pt Lee` method performs well when the Ackermann motion holds. In Fig. 6(c), we plot the translation direction error as an additional evaluation. It is interesting to see that when the side length of the square is 40 pixels, the `1AC plane` method performs better than the PC-based methods and the `6AC-Ventura` method in translation direction estimation.

We also evaluate the accuracy of the proposed methods `1AC plane` and `2AC plane` for increasing planar motion noise. To test such noise, we added a small randomly generated X-axis, Z-axis rotation and a YZ-plane translation (Choi & Kim 2018) to the motion of the multi-camera system. The magnitude of non-planar motion noise ranges from 0° to 1° and the standard deviation of the image noise is set to 0.5 pixel. Figures 6(d–f) show the performance of the proposed `1AC plane` method and `2AC plane` method against planar motion noise. Methods `17pt-Li`, `8pt-Kneip`, `6pt-Stewénius` and `6AC-Ventura` deal with the 6DOF motion case and, thus they are not affected by the noise in the planarity assumption. The `2pt Lee` method does not have an obvious trend with non-planar motion noise levels, because the accuracy of this method mainly depends on whether the Ackermann motion assumption is well fulfilled. It can be seen that the rotation accuracy of the `2AC plane` method performs better than comparative methods when the planar motion noise is less than 0.2°. Since the estimation accuracy of translation direction of the `2AC plane` method in Fig. 6(f) performs satisfactory, the main reason for poor performance of translation estimation is that the metric scale estimation is sensitive to the planar motion noise. In comparison with the `2AC plane` method, the `1AC plane` method has similar performance in rotation estimation, but performs poorly in translation estimation. The translation accuracy decreases significantly with the increase of the planar motion noise.

In addition to efficiency and numerical stability, another important factor for a solver is the minimal number of required image points. The iteration number N of RANSAC can be computed by $N = \log(1-p)/\log(1-(1-\epsilon)^s)$, where s is the number of minimal image points, ϵ is the outlier ratio, and p is the success probability. For a probability of success $p = 99\%$, the RANSAC iterations needed with respect to the outlier ratio needed are shown in Figure 7. It can be seen that the iteration number of the RANSAC estimator increases exponentially with respect to the number of image points needed. For example, in a percentage of outliers $\epsilon = 50\%$, when the solvers require 1, 2, 4, 6, 8 and 17 points, the RANSAC estimator need 7, 16, 71, 292, 1177 and 603607 iterations, respectively. The proposed `1AC plane` method which only uses a single AC requires the lowest number of RANSAC iterations. Since the proposed `2AC plane` method need two ACs, the iteration number of RANSAC is also low in comparison to PC-based methods. Thus, our solvers can be used efficiently for detecting a correct inlier set when integrating them into the RANSAC framework.

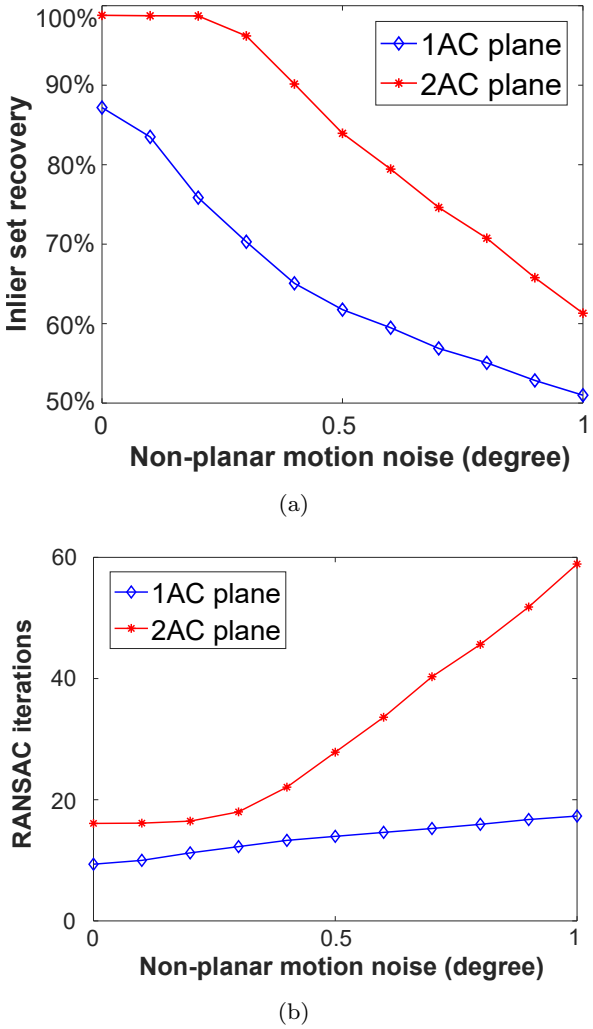
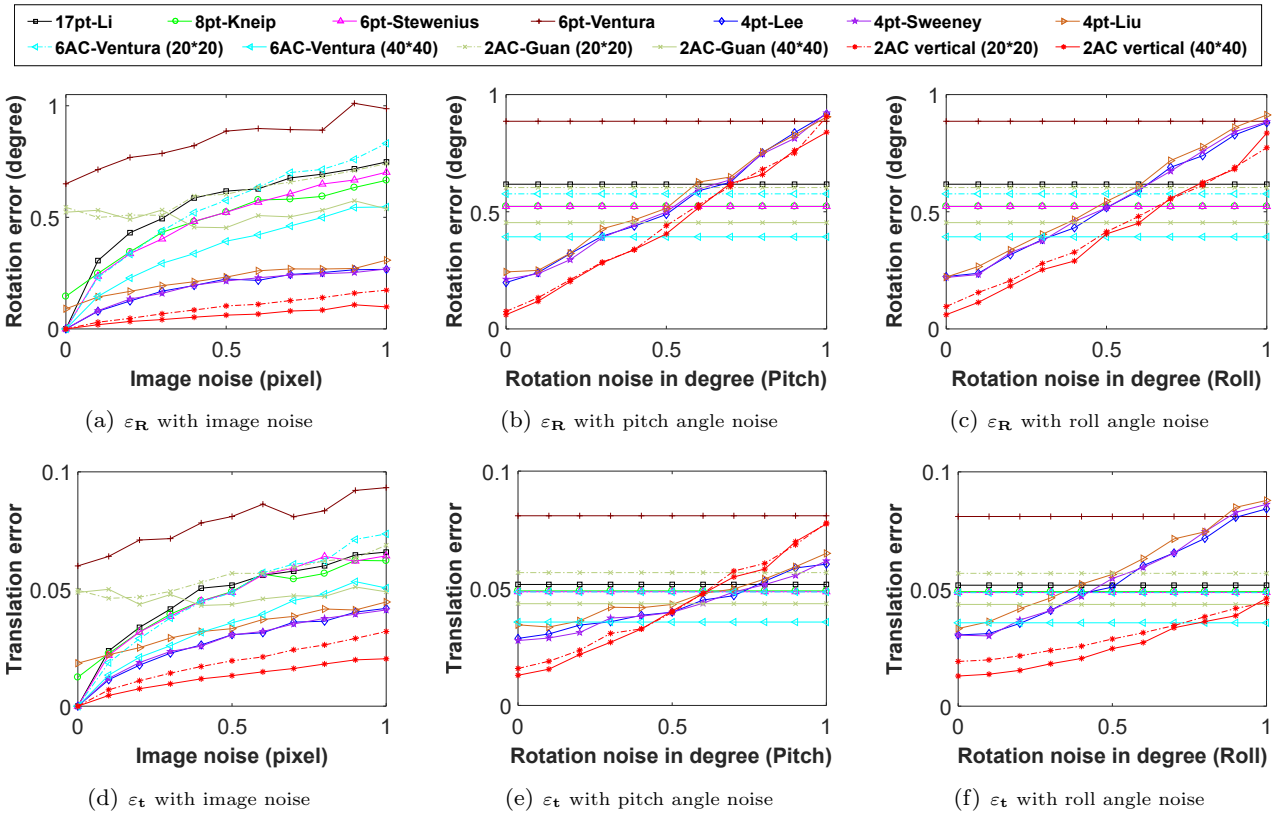


Fig. 8 Rotation and translation error with varying planar motion noise. The image noise is fixed at 0.5 pixel and the outlier ratio is set to 50%.

We evaluate the performance of the proposed `1AC plane` method and `2AC plane` method for outlier detection in presence of outliers. The outlier ratio is set to 50%. The other configurations of this synthetic experiment are set as same as using in Figure 6(d–f). Figure 8 shows the performance of the proposed methods against planar motion noise. It is interesting to see that the `1AC plane` method recovers more than 50% inliers and requires fewer number of RANSAC iterations, even though it performs poorly in translation estimation as shown in Figure 6(e–f). Thus, the `1AC plane` method has the advantage of detecting a correct inlier set efficiently, which can then be used for accurate motion estimation with non-linear optimization.

6.3.2 Motion with Known Vertical Direction

In this set of experiments, the translation direction of two multi-camera reference frames is chosen to produce either forward, sideways or random motions. The second reference frame is rotated around three axes randomly with angles ranging from $-10°$ to $10°$. Assuming known roll and pitch angles, the multi-camera reference



(a) $\varepsilon_{\mathbf{R}}$ with image noise (b) $\varepsilon_{\mathbf{R}}$ with pitch angle noise (c) $\varepsilon_{\mathbf{R}}$ with roll angle noise

(d) $\varepsilon_{\mathbf{t}}$ with image noise (e) $\varepsilon_{\mathbf{t}}$ with pitch angle noise (f) $\varepsilon_{\mathbf{t}}$ with roll angle noise

Fig. 9 Rotation and translation error under random motion with known vertical direction. Upper row: rotation error. Bottom row: translation error. (a,d): varying image noise. (b,e) and (c,f): varying IMU angle noise and fixed 0.5 pixel std. image noise.

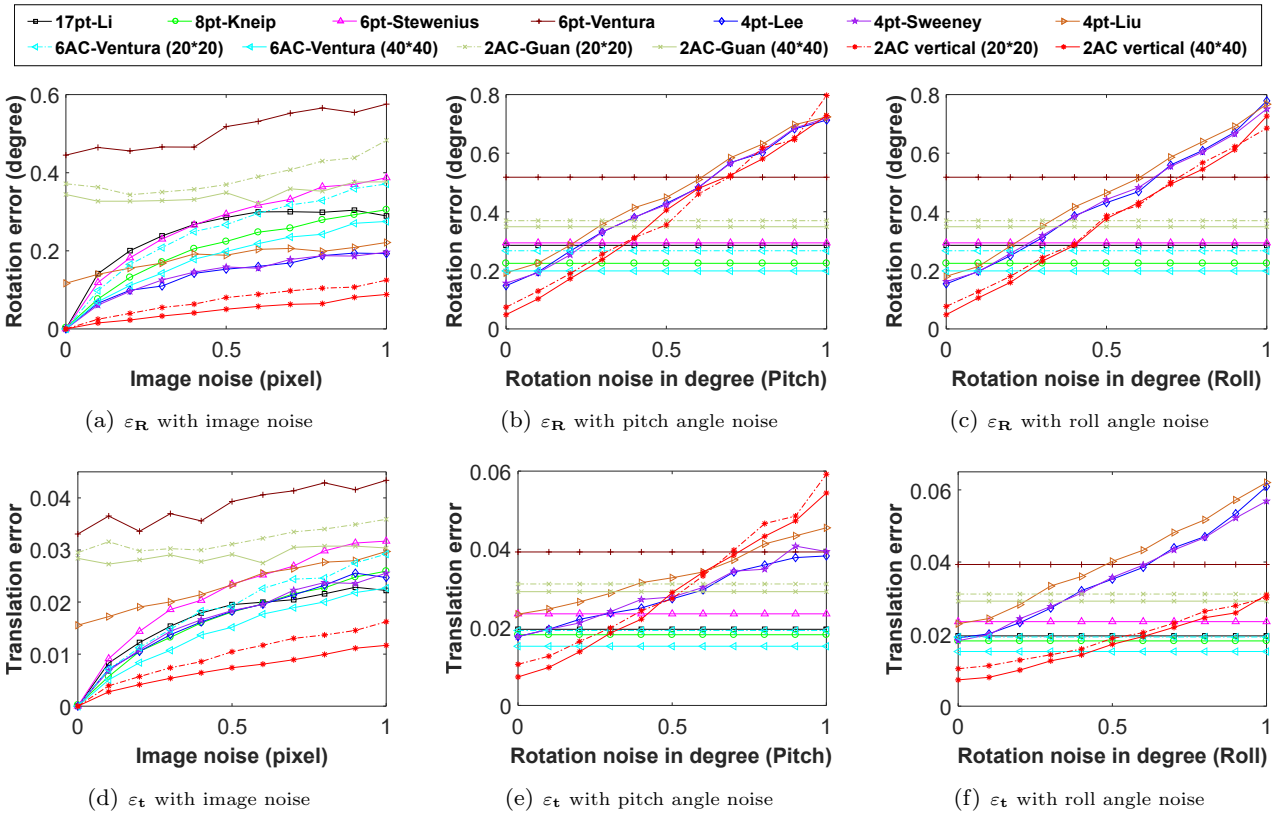
frame is aligned with the vertical direction. Figs. 9(a) and (d) show the performance of `2AC vertical` against image noise with perfect IMU data under random motion. The proposed method is robust to image noise and performs better than the other methods. The iterative optimization in `8pt-Kneip` is prone to falling into local minima. Since the methods `6pt-Ventura`, `4pt-Liu` and `2AC-Guan` use the first-order approximation of the relative rotation, the error of these methods is not zero even for image noise-free input.

Figs. 9(b,e) and (c,f) show the performance of `2AC vertical` against IMU noise in the random motion case, while the standard deviation of the image noise is fixed at 0.5 pixel. Note that the methods `17pt-Li`, `8pt-Kneip`, `6pt-Stewénius`, `6pt-Ventura`, `6AC-Ventura` and `2AC-Guan` are not influenced by IMU noise, because these methods do not use the known vertical direction as a prior. The methods `4pt-Lee`, `4pt-Sweeney` and `4pt-Liu` use the known vertical direction as a prior. It is interesting to see that the proposed method outperforms the comparative methods in the random motion case, even though the IMU noise is around 0.4°.

Figure 10 shows the performance of the proposed `2AC vertical` under forward motion. It can be seen that `2AC vertical` outperforms the comparative methods against image noise and provides comparable accuracy for increasing IMU noise, even though the size of the square is 20 pixels. Figure 11 shows the performance of the proposed `2AC vertical` under sideways motion. The results demonstrate that when the side length of the square is 40 pixels, the `2AC vertical` performs basically better than all compared methods against image noise and achieves comparable performance for increasing noise on the IMU data.

6.3.3 Ackermann Motion Case

In this scenario, we evaluate the accuracy of the proposed methods `1AC plane` and `2AC plane` under Ackermann motion. The relative motion of the multi-camera system is constrained by the Ackermann motion model (Scaramuzza et al. 2009), which is described by a rotation angle and a translation distance. Specifically, the multi-camera system moves along circular trajectories about the instantaneous center of rotation and the translation direction satisfies the circular motion constraint. The magnitude of the rotation angle ranges from $-10°$ to $10°$. The other configurations of this synthetic experiment are set as same as using in Figure 6. This scenario is suitable for the methods with Ackermann motion assumption, such as `2pt-Lee`. Suppose we are given Gaussian image noise with a standard deviation ranging from 0 to 1.0 pixel. Fig. 12(a–c) shows the performance of the proposed `1AC plane` and `2AC plane` methods against image noise under perfect Ackermann Motion. When the Ackermann motion assumption is well fulfilled, the `2pt Lee` outperforms the other methods in rotation estimation and translation direction estimation. The methods `2AC plane` and `2pt Lee` have similar performance in translation estimation. Moreover, it can be seen that `2AC plane` per-



(a) $\varepsilon_{\mathbf{R}}$ with image noise (b) $\varepsilon_{\mathbf{R}}$ with pitch angle noise (c) $\varepsilon_{\mathbf{R}}$ with roll angle noise

(d) $\varepsilon_{\mathbf{t}}$ with image noise (e) $\varepsilon_{\mathbf{t}}$ with pitch angle noise (f) $\varepsilon_{\mathbf{t}}$ with roll angle noise

Fig. 10 Rotation and translation error under forward motion with known vertical direction. Upper row: rotation error. Bottom row: translation error. (a,d): varying image noise. (b,e) and (c,f): varying IMU angle noise and fixed 0.5 pixel std. image noise.

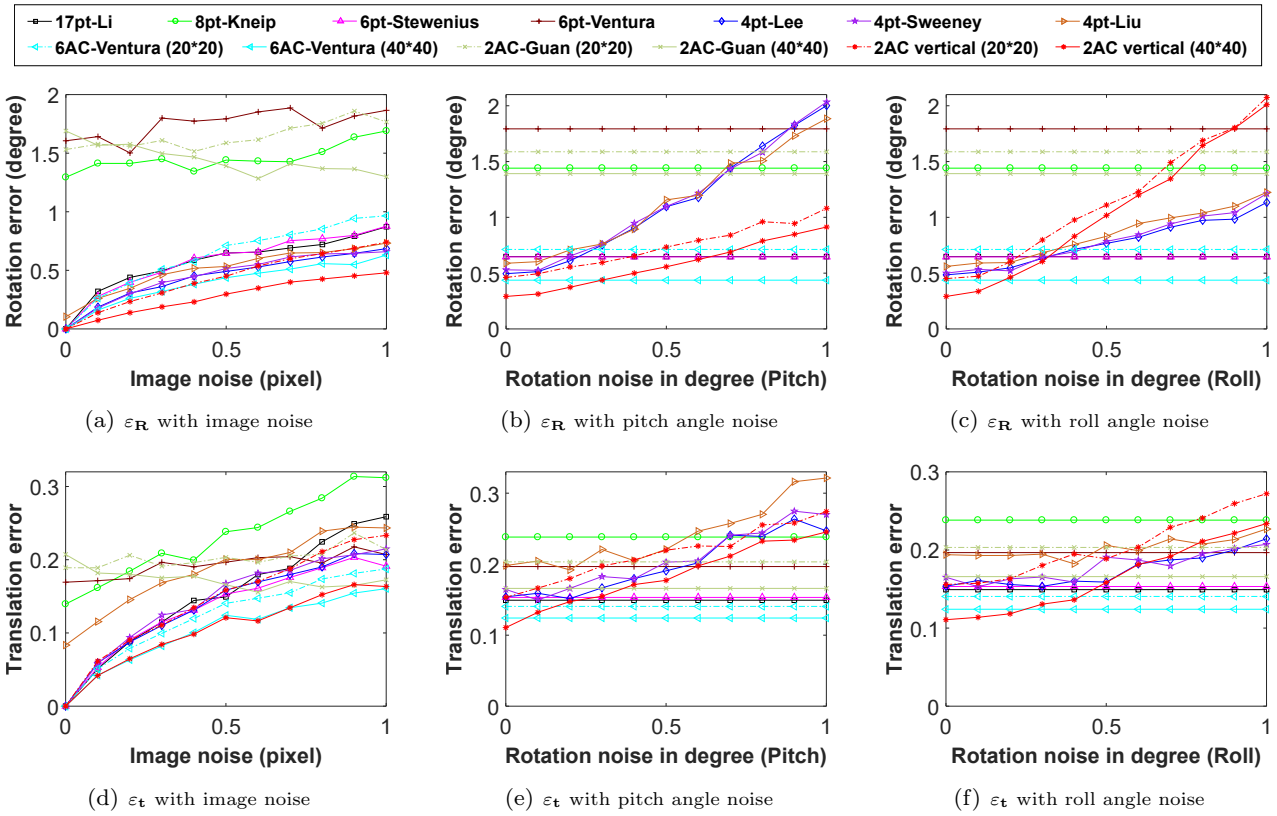
forms better than the comparative methods which do not constrain the relative motion by the Ackermann motion model.

We also evaluate the accuracy of the proposed methods `1AC plane` and `2AC plane` for increasing non-Ackermann motion noise. To test such noise, we added a small randomly generated angle error to the translation direction of the multi-camera system in XZ-plane. The relative motion of the multi-camera system is deviating from Ackermann motion, but still satisfying planar motion. The magnitude of non-Ackermann motion noise ranges from 0° to 1° and the standard deviation of the image noise is set to 0.5 pixel. Figures 12(d–f) show the performance of the proposed `1AC plane` method and `2AC plane` method against non-Ackermann motion noise. Methods `17pt-Li`, `8pt-Kneip`, `6pt-Stewénius`, `6AC-Ventura`, `1AC plane` and `2AC plane` deal with the planar motion case and, thus they are not affected by non-Ackermann motion noise. The accuracy of the `2pt Lee` method decreases significantly with the increase of the non-Ackermann motion noise. It can be seen that the `2AC plane` method performs better than comparative methods when the non-Ackermann motion noise is more than 0.4°. In comparison with the `2AC plane` method, the `1AC plane` method has similar performance in rotation estimation, but performs poorly in translation estimation.

6.3.4 Small Rotation Case

In this scenario, we evaluate the accuracy of the proposed `2AC vertical` method under small rotation motion. The rotation angles between two multi-camera reference frames are kept constant at 1° (Ventura et al. 2015). The translation direction of two multi-camera reference frames is chosen to produce random motion. The other configurations of this synthetic experiment are set as same as using in Figure 9. Since the relative rotation between two consecutive frames is small, several methods with first-order approximation to relative rotation are suitable, such as `6pt-Ventura` (Ventura et al. 2015), `4pt-Liu` (Liu et al. 2017) and `2AC-Guan` (Guan et al. 2021a). Assuming known roll and pitch angles, the multi-camera reference frame is aligned with the vertical direction.

Figs. 13(a) and (d) show the performance of `2AC vertical` against image noise with perfect IMU data under small rotation motion. It can be seen that the proposed `2AC vertical` method performs better than the other methods. The methods `6pt-Ventura`, `4pt-Liu` and `2AC-Guan` achieve good performance when the small rotation motion assumption is well fulfilled. Figs. 13(b,e) and (c,f) show the performance of `2AC vertical` against IMU noise under small rotation motion, while the standard deviation of the image noise is fixed at 0.5 pixel. The results demonstrate that when the side length of the square is 40 pixels, the `2AC vertical` basically outperforms the comparative



(a) $\varepsilon_{\mathbf{R}}$ with image noise (b) $\varepsilon_{\mathbf{R}}$ with pitch angle noise (c) $\varepsilon_{\mathbf{R}}$ with roll angle noise

(d) $\varepsilon_{\mathbf{t}}$ with image noise (e) $\varepsilon_{\mathbf{t}}$ with pitch angle noise (f) $\varepsilon_{\mathbf{t}}$ with roll angle noise

Fig. 11 Rotation and translation error under sideways motion with known vertical direction. Upper row: rotation error. Bottom row: translation error. (a,d): varying image noise. (b,e) and (c,f): varying IMU angle noise and fixed 0.5 pixel std. image noise.

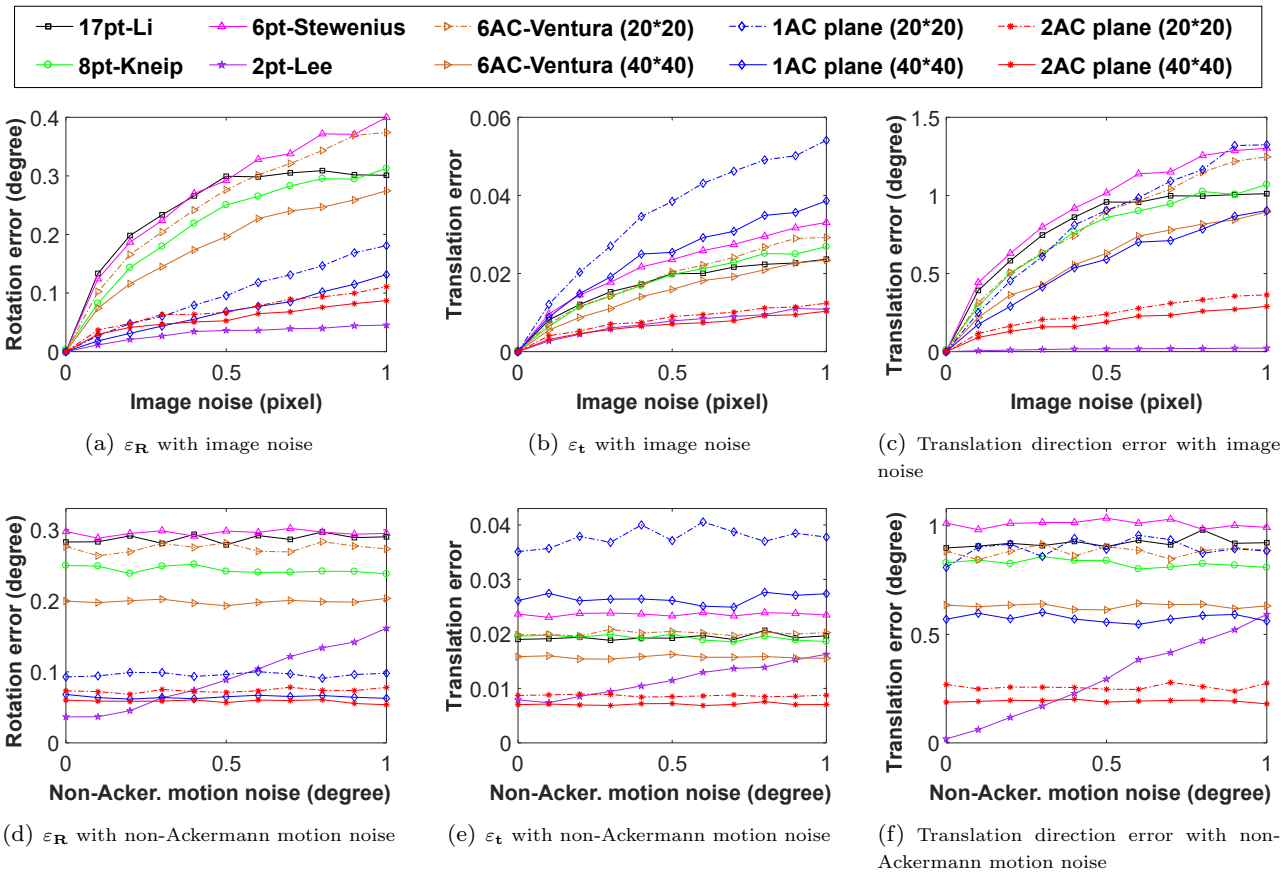
methods, even though the IMU noise is around 0.2°.

6.3.5 Using PCs converted from ACs

In this set of experiments, we evaluate the performance of PC-based solvers using the PCs converted from ACs. Three generated PCs include an image point pair of AC and two hallucinated image point pairs calculated by the local affine transformation. Since local affine transformations are defined as the partial derivative, w.r.t. the image directions, of the related homography, they are valid only infinitesimally close to the image coordinates of AC. Thereby, one AC can only provide three approximate PCs – the error is not zero even for noise-free input (Barath & Hajder 2018). Three approximate PCs converted from one AC can be computed as follows (Barath et al. 2020): $(\mathbf{x}_k, \mathbf{x}_k+[w,\ 0]^T, \mathbf{x}_k+[0,\ w]^T)$ and $(\mathbf{x}'_k, \mathbf{x}'_k+\mathbf{A}_k[w,\ 0]^T, \mathbf{x}'_k+\mathbf{A}_k[0,\ w]^T)$, where w determines the distribution area of the generated PCs. To evaluate the performance of PC-based solvers with different distribution area, w is set to 1, 5 and 10 pixels, respectively.

Take relative pose estimation with known vertical direction for an example. A total of 1000 trials are carried out in the synthetic experiment. In each test, 100 ACs are generated randomly with 40*40 support region. In the RANSAC loop, six ACs and two ACs are selected randomly for the `6AC-Ventura` method and the proposed `2AC vertical` method, respectively. The hallucinated PCs converted from a minimal number of ACs are used as input for the PC-based solvers. Thus, 6, 3 and 2 ACs are selected randomly for the `17pt-Li` solver (Li et al. 2008), the `8pt-Kneip` solver (Kneip & Li 2014), and the solvers `6pt-Stewénius` (Stewénius et al. 2005), `6pt-Ventura` (Ventura et al. 2015), `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014) and `4pt-Liu` (Liu et al. 2017), respectively. Note that the hallucinated PCs converted from ACs are only used for hypothesis generation, and the inlier set is found by evaluating the image point pairs of ACs. The solution which produces the highest number of inliers is chosen. The other configurations of this synthetic experiment are set as same as using in Figure 9.

Figure 14 shows the performance of the PC-based solvers against image noise in the random motion case. The estimation results using the image point pairs of ACs are represented by solid lines. The estimation results using the hallucinated PCs generated with different distribution area are represented by dashed line (w = 1 pixel), dash-dotted line (w = 5 pixels) and dotted line (w = 10 pixels), respectively. We have the following observations. (1) The PC-based solvers using the hallucinated PCs perform worse than using the image point pairs of AC. Because the conversion error between each AC and three PCs is newly introduced. It can be seen that the estimation error of PC-based solvers using the hallucinated PCs is not zero even for image noise-free input. Moreover, the hallucinated PCs generated by each AC are near each other which may be a degenerate case for the PC-based solvers. The error curves of the



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

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

(c) Translation direction error with image noise

(d) $\varepsilon_{\mathbf{R}}$ with non-Ackermann motion noise

(e) $\varepsilon_{\mathbf{t}}$ with non-Ackermann motion noise

(f) Translation direction error with non-Ackermann motion noise

Fig. 12 Rotation and translation error under Ackermann motion. (a–c): varying image noise under perfect Ackermann motion. (d–f): varying non-Ackermann motion noise and fixed 0.5 pixel std. image noise.

`6pt-Ventura` method are out of the display range when the hallucinated PCs is used. (2) The performance of PC-based solvers is influenced by the different distribution area of hallucinated PCs. Since a smaller distribution area causes smaller conversion error between ACs and PCs, the PC-based solvers have better performance with smaller distribution area. (3) The performance of the proposed `2AC vertical` method is best. Because the AC-based solvers use the relationship between local affine transformations and epipolar lines, *i.e.*, Eq. (9). This is a strictly satisfied constraint and does not result in any error for noise-free input. In addition, the `2AC vertical` method is robust to image noise and performs better than the methods `6AC-Ventura` and `2AC-Guan`.

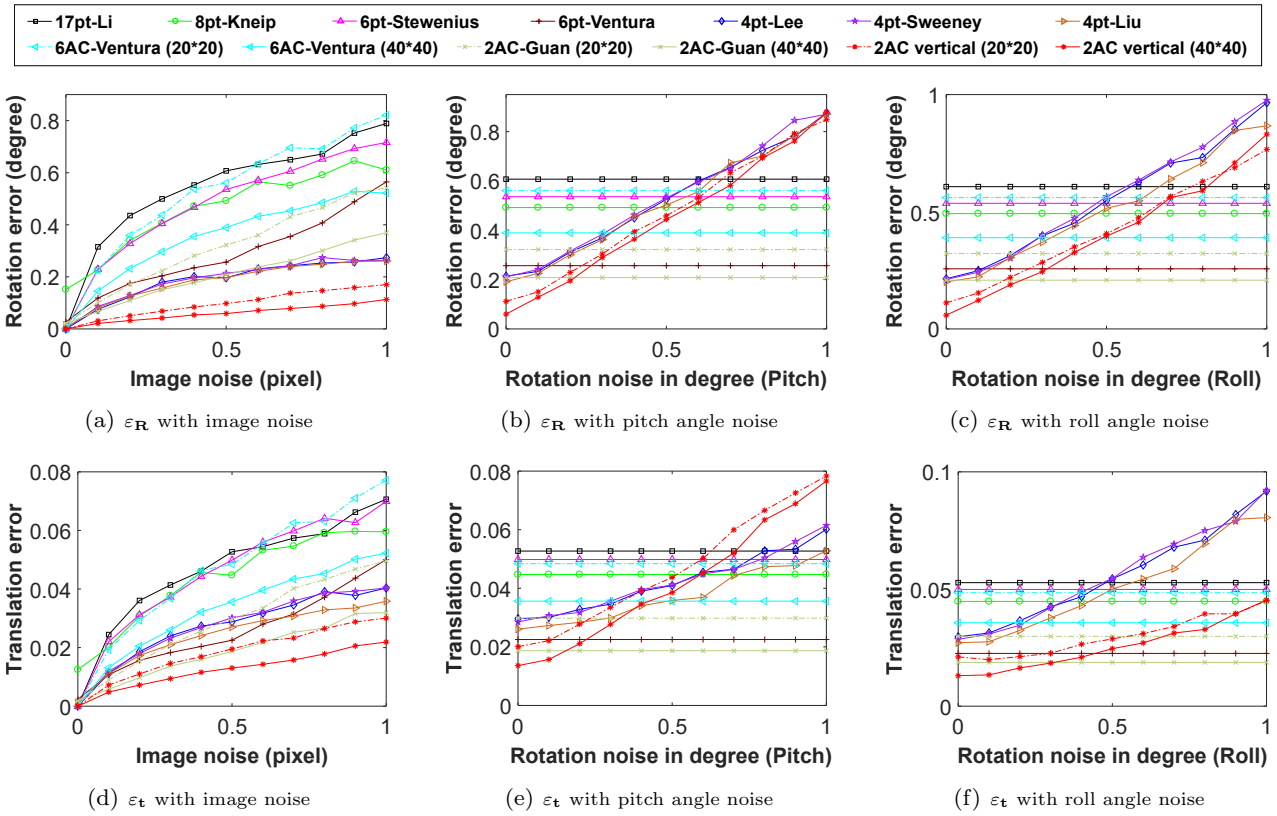
6.4 Experiments on Real Data

To demonstrate the suitability of our methods in real scenarios, we validate the performance of the proposed solvers on three public datasets. The `KITTI` dataset (Geiger et al. 2013) and the `nuScenes` dataset (Caesar et al. 2020) are collected on an autonomous driving environment. The `EuRoc` MAV dataset (Burri et al. 2016) are collected on an unmanned aerial vehicle environment. We compare our solvers against state-of-the-art techniques in these two popular modern robot applications.

6.4.1 Experiments on KITTI Dataset

We test the performance of our methods on the `KITTI` dataset (Geiger et al. 2013) that consists of successive video frames from a forward facing stereo camera. The ground truth pose is provided from the built-in GPS/IMU units. We ignore the overlap in their fields of view and treat it as a general multi-camera system. The sequences labeled from 0 to 10, which have ground truth, are used for the evaluation. Therefore, the methods were tested on a total of 23000 image pairs. The ACs between consecutive frames in each camera are established by applying the ASIFT (Morel & Yu 2009) detector. The extraction of ACs can also be sped up by MSER (Matas et al. 2004), GPU acceleration, or approximating ACs from SIFT features for subsequent video frames. The ACs across the two cameras are not matched and the metric scale is not estimated as the movement between consecutive frames is small. Besides, integrating the acceleration over time from an IMU is more suitable for recovering the scale (Nützi et al. 2011). All the solvers have been integrated into a RANSAC scheme.

The proposed methods `2AC plane` and `2AC vertical` are compared against `17pt-Li` (Li et al. 2008), `8pt-Kneip` (Kneip & Li 2014), `6pt-Stewénius` (Stewénius et al. 2005), `6pt-Ventura` (Ventura et al. 2015), `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014),



(a) $\varepsilon_{\mathbf{R}}$ with image noise (b) $\varepsilon_{\mathbf{R}}$ with pitch angle noise (c) $\varepsilon_{\mathbf{R}}$ with roll angle noise

(d) $\varepsilon_{\mathbf{t}}$ with image noise (e) $\varepsilon_{\mathbf{t}}$ with pitch angle noise (f) $\varepsilon_{\mathbf{t}}$ with roll angle noise

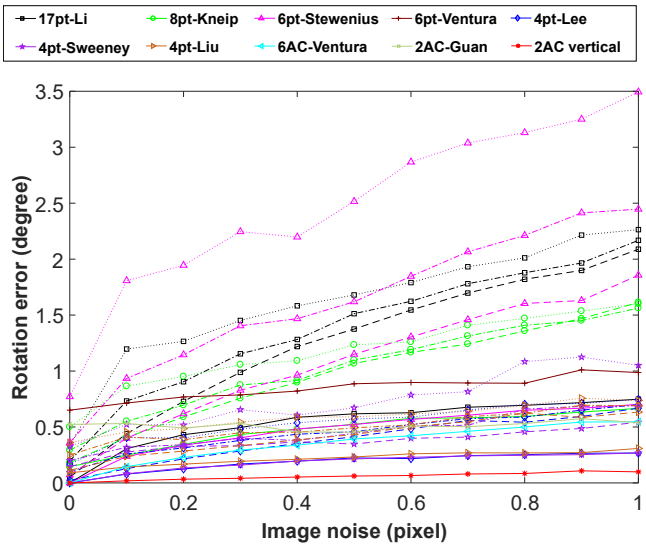
Fig. 13 Rotation and translation error under small rotation motion. Upper row: rotation error. Bottom row: translation error. (a,d): varying image noise. (b,e) and (c,f): varying IMU angle noise and fixed 0.5 pixel std. image noise.

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

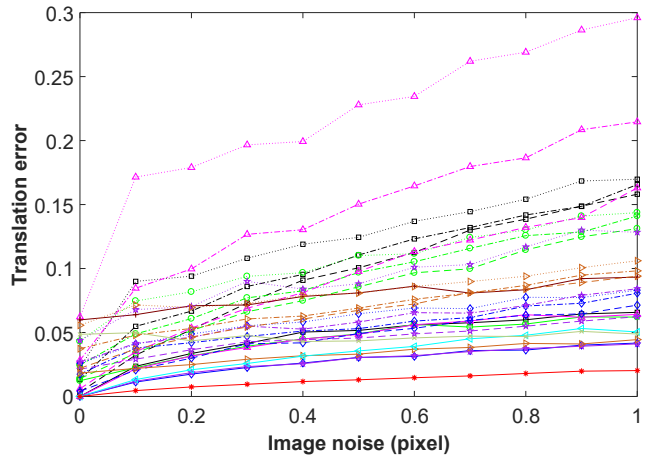
	Seq.	17pt-Li	8pt-Kneip	6pt-St.	6pt-Ven.	4pt-Lee	4pt-Sw.	4pt-Liu	2pt-Lee	6AC-Ven.	2AC-Guan	**2AC plane**	**2AC vertical**
$\varepsilon_{\mathbf{R}}$	00	0.139	0.130	0.229	0.125	0.065	0.050	0.066	0.301	0.142	0.118	0.280	**0.031**
	01	0.158	0.171	0.762	0.178	0.137	0.125	0.105	0.189	0.146	0.140	0.168	**0.025**
	02	0.123	0.126	0.186	0.248	0.057	0.044	0.057	0.254	0.121	0.208	0.213	**0.030**
	03	0.115	0.108	0.265	0.210	0.064	0.069	0.062	0.261	0.113	0.172	0.238	**0.037**
	04	0.099	0.116	0.202	0.212	0.050	0.051	0.045	0.131	0.100	0.098	0.116	**0.020**
	05	0.119	0.112	0.199	0.157	0.054	0.052	0.056	0.199	0.116	0.122	0.185	**0.022**
	06	0.116	0.118	0.168	0.168	0.053	0.092	0.056	0.145	0.115	0.111	0.137	**0.023**
	07	0.119	0.112	0.245	0.188	0.058	0.065	0.054	0.202	0.137	0.141	0.173	**0.023**
	08	0.116	0.111	0.196	0.166	0.051	0.046	0.053	0.225	0.108	0.146	0.203	**0.024**
	09	0.133	0.125	0.179	0.274	0.056	0.046	0.058	0.234	0.124	0.169	0.189	**0.027**
	10	0.127	0.115	0.201	0.195	0.052	0.040	0.058	0.265	0.203	0.174	0.223	**0.025**
$\varepsilon_{\mathbf{t}}$	00	2.412	2.400	4.007	2.272	2.469	2.190	2.519	2.746	2.499	2.133	2.243	**1.738**
	01	5.231	4.102	41.19	4.217	4.782	11.91	3.781	2.179	3.654	3.012	2.486	**1.428**
	02	1.740	1.739	2.508	2.422	1.825	1.579	1.821	2.506	1.702	1.891	1.975	**1.558**
	03	2.744	2.805	6.191	4.208	3.116	3.712	3.258	2.065	2.731	2.571	**1.849**	1.888
	04	1.560	1.746	3.619	2.966	1.564	1.708	1.635	2.385	1.725	1.892	1.768	**1.228**
	05	2.289	2.281	4.155	3.013	2.337	2.544	2.406	2.735	2.273	2.279	2.354	**1.532**
	06	2.071	1.862	2.739	2.675	1.757	2.721	1.760	2.543	1.956	1.978	2.247	**1.303**
	07	3.002	3.029	6.397	4.354	2.810	4.554	3.048	3.105	2.892	2.601	2.902	**1.820**
	08	2.386	2.349	3.909	2.537	2.433	2.422	2.457	3.200	2.344	2.572	2.569	**1.911**
	09	1.977	1.806	2.592	2.947	1.838	1.656	1.793	2.673	1.876	1.901	1.997	**1.440**
	10	1.889	1.893	2.781	2.659	1.932	1.658	1.888	2.955	2.057	2.230	2.296	**1.586**

Table 4 Runtime of RANSAC averaged over KITTI sequences combined with different solvers (unit: s).

Methods	17pt-Li	8pt-Kneip	6pt-St.	6pt-Ven.	4pt-Lee	4pt-Sw.	4pt-Liu	2pt-Lee	6AC-Ven.	2AC-Guan	**2AC plane**	**2AC vertical**
Mean time	52.82	10.36	79.76	5.71	0.85	0.63	0.45	0.11	6.83	0.59	**0.07**	0.09
Standard deviation	2.62	1.59	4.52	0.73	0.093	0.057	0.058	0.014	0.61	0.067	**0.0071**	0.0086



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



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

Fig. 14 Rotation and translation error with varying image noise under random motion with known vertical direction. Solid line indicates using image point pairs of ACs. Dashed line, dash-dotted line and dotted line indicate using the hallucinated PCs, which are generated with different distribution area $w = 1$, 5, 10 pixels, respectively.

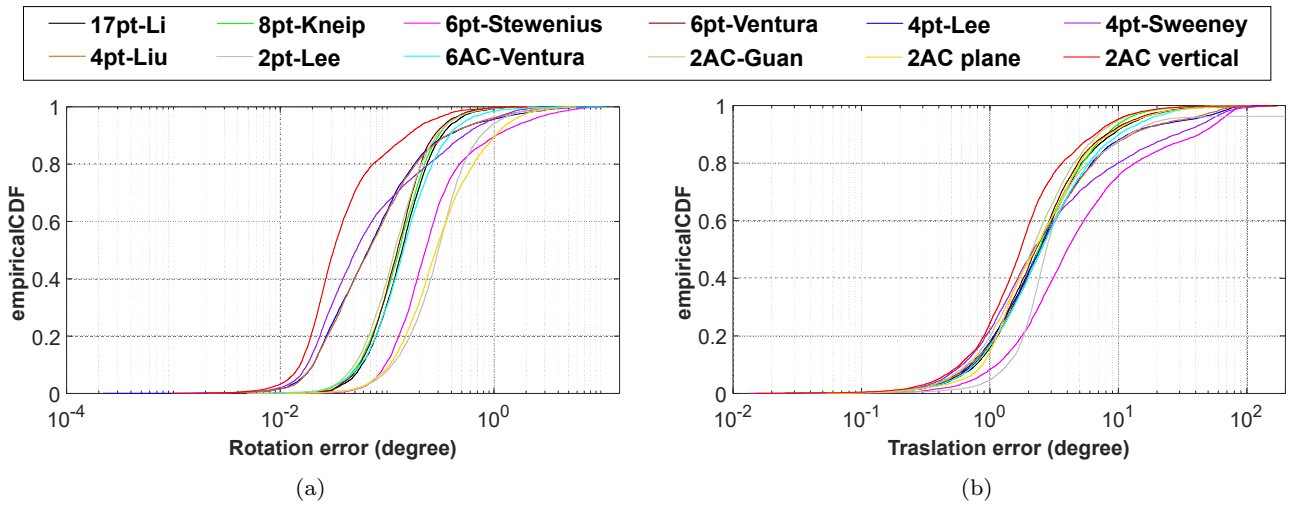
`4pt-Liu` (Liu et al. 2017), `2pt-Lee` (Lee et al. 2013), `6AC-Ventura` (Alyousefi & Ventura 2020) and `2AC-Guan` (Guan et al. 2021a). Since the `KITTI` dataset is captured by a stereo rig with both cameras having the same altitude, that is a degenerate case for the `1AC plane` method, it is not performed in the experiment. For the `2AC plane` method, the results are also compared to the ground truth of the 6DOF relative pose, even though this method only estimates two angles (θ, ϕ) with the plane motion assumption. For the `2AC vertical` method, the roll and pitch angles obtained from the GPS/IMU units are used to align the multi-camera reference frame with the vertical direction (Guan et al. 2020; Li et al. 2020; Saurer et al. 2016). To ensure the fairness of the experiment, the roll and pitch angles are also provided for the methods `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014) and `4pt-Liu` (Liu et al. 2017), which use the known vertical direction as a prior. Table 3 shows the results of the rotation and translation estimation. The median error for each individual sequence is used to evaluate the estimation accuracy. The runtime of RANSAC averaged over `KITTI` sequences combined with different solvers is shown in Table 4. The reported runtimes include the robust relative pose estimation without feature extraction, *i.e.*, recovering the relative pose by RANSAC combined with a minimal solver.

The proposed `2AC vertical` method offers the best overall performance among all the methods. The `6pt-Stewénius` method performs poorly on sequence 01, because this sequence is a highway with few trackable close objects, and this method always fails to select the best candidate from multiple solutions under forward motion in the RANSAC scheme. Besides, it is interesting to see that the translation accuracy of the `2AC plane` method basically outperforms the methods `6pt-Stewénius` and `6pt-Ventura`, even though the planar motion assumption does not fit the `KITTI` dataset well. Because the KITTI dataset has obvious ups and downs, which will affect the accuracy of relative pose estimation under the planar motion assumption. We also show the empirical cumulative error distributions for KITTI sequence 00. These values are calculated from the same values which were used for creating Table 3. Figure 15 shows the proposed `2AC vertical` method offers the best overall performance in comparison to state-of-the-art methods.

To visualize the comparison results, the estimated trajectory for sequence 00 is plotted in Fig. 16. We are directly concatenating frame-to-frame relative pose measurements without any post-refinement. The trajectory for `2AC vertical` is compared with the two best performing comparison methods in sequence 00 based on Table 3: `2AC-Guan` in 6DOF motion case and `4pt-Sweeney` in 4DOF motion case. Since all methods were not able to estimate the scale correctly, in particular for the many straight parts of the trajectory, the ground truth scale is used to plot the trajectories. Then the trajectories are aligned with the ground truth and the color along the trajectory encodes the absolute trajectory error (ATE) (Sturm et al. 2012). Even though all trajectories have a significant accumulation of drift, it can still be seen that the `2AC vertical` method has the smallest ATE among the compared trajectories. Due to the benefits of computational efficiency, both the `2AC plane` method and the `2AC vertical` method are quite suitable for finding a correct inlier set, which is then used for accurate motion estimation in visual odometry.

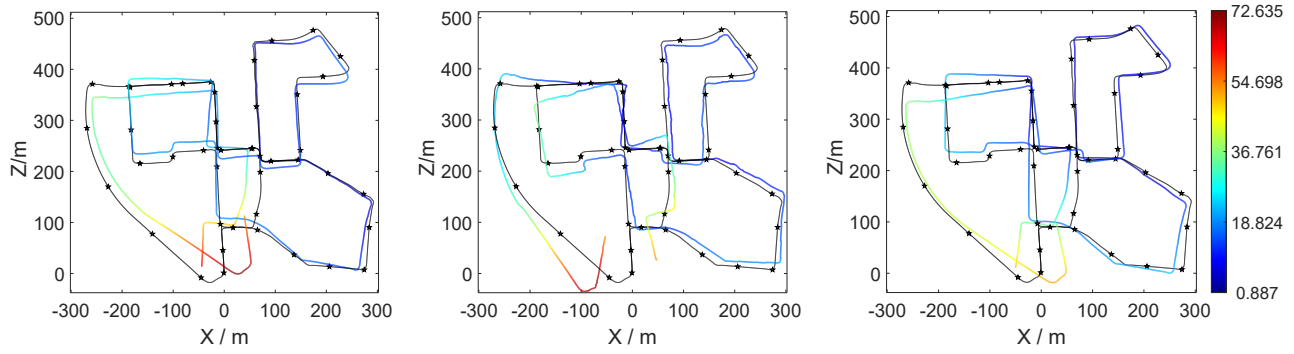
6.4.2 Experiments on nuScenes Dataset

We also test the performance of our methods on the `nuScenes` dataset (Caesar et al. 2020), which consists of consecutive keyframes from 6 cameras. All the keyframes of Part 1 are used for the evaluation and there are 3376 images in total. The ground truth pose is provided from a lidar map-based localization scheme. Similar to the experiments on `KITTI` dataset, the ACs between consecutive keyframes in each camera are established by applying



(a) (b)

Fig. 15 Empirical cumulative error distributions for KITTI sequence 00. (a) reports the rotation error. (b) reports the translation error. The proposed `2AC plane` method and `2AC vertical` are compared against `17pt-Li` (Li et al. 2008), `8pt-Kneip` (Kneip & Li 2014), `6pt-Stewénius` (Stewénius et al. 2005), `6pt-Ventura` (Ventura et al. 2015), `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014), `4pt-Liu` (Liu et al. 2017), `2pt-Lee` (Lee et al. 2013), `6AC-Ventura` (Alyousefi & Ventura 2020) and `2AC-Guan` (Guan et al. 2021a).



(a) `2AC-Guan` (b) `4pt-Sweeney` (c) `2AC vertical`

Fig. 16 Estimated trajectories without any post-refinement. The relative pose measurements between consecutive frames are directly concatenated. The colorful curves are the trajectories estimated by `2AC-Guan` (Guan et al. 2021a), `4pt-Sweeney` (Sweeney et al. 2014) and `2AC vertical`. Black curves with stars are the ground truth trajectories. Best viewed in color.

Table 5 Rotation and translation error on `nuScenes` sequences (unit: degree).

Part		17pt-Li	8pt-Kneip	6pt-St.	6pt-Ven.	4pt-Lee	4pt-Sw.	4pt-Liu	2pt-Lee	6AC-Ven.	2AC-Guan	**2AC plane**	**2AC vertical**
$\varepsilon_{\mathbf{R}}$	01	0.161	0.156	0.203	0.179	0.083	0.078	0.108	0.371	0.143	0.127	0.344	**0.057**
$\varepsilon_{\mathbf{t}}$	01	2.680	2.407	2.764	2.521	1.780	1.659	1.941	2.327	2.366	2.195	2.284	**1.469**

the ASIFT (Morel & Yu 2009) detector. The proposed methods `2AC plane` and `2AC vertical` are compared against `17pt-Li` (Li et al. 2008), `8pt-Kneip` (Kneip & Li 2014), `6pt-Stewénius` (Stewénius et al. 2005), `6pt-Ventura` (Ventura et al. 2015), `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014), `4pt-Liu` (Liu et al. 2017), `2pt-Lee` (Lee et al. 2013), `6AC-Ventura` (Alyousefi & Ventura 2020) and `2AC-Guan` (Guan et al. 2021a). All solvers are used within RANSAC.

Table 5 shows the results of the rotation and translation estimation for the Part1 of `nuScenes` dataset. The median error is used to evaluate the estimation accuracy. It can be seen that the proposed `2AC vertical` method offers the best performance among all the methods. This experiment also demonstrates that both planar motion and known vertical direction assumptions are met in practical self-driving situations.

6.4.3 Experiments on EuRoC Dataset

We further evaluate the performance of the proposed solvers in an unmanned aerial vehicle environment. The `EuRoC` MAV dataset (Burri et al. 2016) is used for the evaluation in this experiment, which is collected with a stereo camera mounted on a micro aerial vehicle. The ground truth pose is provided from the nonlinear least-squares batch solution over the Leica position and IMU

Table 6 Rotation and translation error on `EuRoC` sequences (unit: degree).

	Seq.	17pt-Li	8pt-Kneip	6pt-St.	6pt-Ven.	4pt-Lee	4pt-Sw.	4pt-Liu	6AC-Ven.	2AC-Guan	**2AC vertical**
$\varepsilon_{\mathbf{R}}$	MH01	0.113	0.109	0.124	0.623	0.030	0.027	0.029	0.106	0.443	**0.022**
	MH02	0.106	0.112	0.144	0.636	0.022	0.027	0.022	0.102	0.394	**0.020**
	MH03	0.137	0.148	0.181	0.835	0.039	0.040	0.052	0.133	0.561	**0.034**
	MH04	0.154	0.170	0.175	0.745	0.043	0.041	0.045	0.165	0.806	**0.033**
	MH05	0.167	0.158	0.179	0.852	0.038	0.040	0.035	0.176	0.718	**0.029**
$\varepsilon_{\mathbf{t}}$	MH01	2.928	2.865	3.555	7.348	1.947	2.170	2.075	2.858	4.682	**1.792**
	MH02	2.494	2.553	2.908	6.339	1.573	1.786	1.707	2.483	4.045	**1.489**
	MH03	2.412	2.276	3.068	5.104	2.177	1.787	1.977	2.075	3.728	**1.675**
	MH02	2.950	3.127	5.531	6.369	2.261	2.098	2.591	2.966	5.945	**1.949**
	MH03	3.071	2.753	4.275	7.971	1.957	2.130	2.004	2.904	5.034	**1.751**

measurements. The sequences labeled from MH01 to MH05, which are collected in a large industrial machine hall, are used for performance comparison. Since the industrial environment is unstructured and cluttered, it renders these sequences challenging to process. Considering that the movement between consecutive frames is small, we choose the part of image pairs for relative pose estimation by an amount of one out of every four images. Besides, we crop the image pairs with insufficient motion in this experiment. Therefore, the methods were tested on a total of 3000 image pairs.

Since the Ackermann motion assumption and the planar motion assumption do not fit the `EuRoC` MAV dataset, the methods `2pt-Lee`, `1AC plane` and `2AC plane` are not performed in the experiment. The `2AC vertical` method are compared against `17pt-Li` (Li et al. 2008), `8pt-Kneip` (Kneip & Li 2014), `6pt-Stewénius` (Stewénius et al. 2005), `6pt-Ventura` (Ventura et al. 2015), `4pt-Lee` (Lee et al. 2014), `4pt-Sweeney` (Sweeney et al. 2014), `4pt-Liu` (Liu et al. 2017), `6AC-Ventura` (Alyousefi & Ventura 2020) and `2AC-Guan` (Guan et al. 2021a). Similar to the experiments on `KITTI` dataset, all the solvers have been integrated into a RANSAC scheme. The ACs between consecutive frames in each camera are established by applying the ASIFT (Morel & Yu 2009) detector. Table 6 shows the results of the rotation and translation estimation for `EuRoC` sequences. The median error for each individual sequence is used to evaluate the estimation accuracy. The proposed `2AC vertical` method offers the best performance among all the methods. This experiment demonstrates that the `2AC vertical` method is well suited for relative pose estimation in the unmanned aerial vehicle environment.

7 Conclusion

By exploiting the geometric constraints which interprets the relationship of ACs and the generalized camera model, we have proposed three solutions for the relative pose estimation of a multi-camera system. Under the planar motion assumption, we present two solvers to recover the relative pose of a multi-camera system, including a minimal solver using a single AC and a solver based on two ACs. In addition, a minimal solution with two ACs is proposed to solve for the relative pose of the multi-camera system with known vertical direction. Both planar motion and known vertical direction assumptions are realistic in popular modern robot applications. We evaluate the proposed solvers on synthetic data and three real image sequence datasets. The experimental results clearly showed that the proposed methods provide better efficiency and accuracy for relative pose estimation in comparison to state-of-the-art methods.

Acknowledgments

This work has been partially funded by the National Natural Science Foundation of China (Grant Nos. 11902349 and 11727804).

References

Agarwal, S., Lee, H.-L., Sturmfels, B., & Thomas, R. R. (2017). On the existence of epipolar matrices. *International Journal of Computer Vision*, *121*(3), pp. 403–415.

Alyousefi, K., & Ventura, J. (2020). Multi-camera motion estimation with affine correspondences. In *International Conference on Image Analysis and Recognition*. (pp. 417–431).

Barath, D. (2018). Five-point fundamental matrix estimation for uncalibrated cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 235–243).

Barath, D., & Hajder, L. (2018). Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, *27*(11), pp. 5328–5337.

Barath, D., & Kukelova, Z. (2019). Homography from two orientation-and scale-covariant features. In *IEEE International Conference on Computer Vision*. (pp. 1091–1099).

Barath, D., Polic, M., FÃűrstner, W., Sattler, T., Pajdla, T., & Kukelova, Z. (2020). Making affine correspondences work in camera geometry computation. In *European Conference on Computer Vision*. (pp. 723–740).

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*(3), pp. 346–359.

Bentolila, J., & Francos, J. M. (2014). Conic epipolar constraints from affine correspondences. *Computer Vision and Image Understanding*, *122*, pp. 105–114.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., et al. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, *35*(10), pp. 1157–1163.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., et al. (2020). nuScenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 11621–11631).

Choi, S., & Kim, J.-H. (2018). Fast and reliable minimal relative pose estimation under planar motion. *Image and Vision Computing*, *69*, pp. 103–112.

Clipp, B., Kim, J.-H., Frahm, J.-M., Pollefeys, M., & Hartley, R. (2008). Robust 6dof motion estimation for non-overlapping, multi-camera systems. In *IEEE Workshop on Applications of Computer Vision*. IEEE, (pp. 1–8).

Cox, D., Little, J., & O'Shea, D. (2013). *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media.

Eichhardt, I., & Barath, D. (2020). Relative pose from deep learned depth and a single affine correspondence. In *European Conference on Computer Vision*. (pp. 627–644).

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), pp. 381–395.

Fragoso, V., DeGol, J., & Hua, G. (2020). gdls*: Generalized pose-and-scale estimation given scale and gravity priors. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 2210–2219).

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, *32*(11), pp. 1231–1237.

Guan, B., Vasseur, P., Demonceaux, C., & Fraundorfer, F. (2018). Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions. In *IEEE International Conference on Robotics and Automation*. (pp. 2320–2327).

Guan, B., Zhao, J., Barath, D., & Fraundorfer, F. (2021a). Efficient recovery of multi-camera motion from two affine correspondences. In *IEEE International Conference on Robotics and Automation*. (pp. 1305–1311).

Guan, B., Zhao, J., Barath, D., & Fraundorfer, F. (2021b). Minimal cases for computing the generalized relative pose using affine correspondences. In *IEEE International Conference on Computer Vision*. (pp. 6068–6077).

Guan, B., Zhao, J., Li, Z., Sun, F., & Fraundorfer, F. (2020). Minimal solutions for relative pose with a single affine correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 1929–1938).

Guan, B., Zhao, J., Li, Z., Sun, F., & Fraundorfer, F. (2021c). Relative pose estimation with a single affine correspondence. *IEEE Transactions on Cybernetics*, pp. 1–12.

Hajder, L., & Barath, D. (2020). Relative planar motion for vehicle-mounted cameras from a single affine correspondence. In *IEEE International Conference on Robotics and Automation*. (pp. 8651–8657).

Häne, C., Heng, L., Lee, G. H., Fraundorfer, F., Furgale, P., Sattler, T., et al. (2017). 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, *68*, pp. 14–27.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

Heng, L., Choi, B., Cui, Z., Geppert, M., Hu, S., Kuan, B., et al. (2019). Project AutoVision: Localization and 3D scene perception for an autonomous vehicle with a multi-camera system. In *IEEE International Conference on Robotics and Automation*. (pp. 4695–4702).

Kim, J.-H., Li, H., & Hartley, R. (2009). Motion estimation for nonoverlapping multicamera rigs: Linear algebraic and L_∞ geometric solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(6), pp. 1044–1059.

Kneip, L., & Furgale, P. (2014). OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In *IEEE International Conference on Robotics and Automation*. (pp. 12034–12043).

Kneip, L., & Li, H. (2014). Efficient computation of relative pose for multi-camera systems. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 446–453).

Kneip, L., Sweeney, C., & Hartley, R. (2016). The generalized relative pose and scale problem: View-graph fusion via 2D-2D registration. In *IEEE Winter Conference on Applications of Computer Vision*. (pp. 1–9).

Lee, G. H., Faundorfer, F., & Pollefeys, M. (2013). Motion estimation for self-driving cars with a generalized camera. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 2746–2753).

Lee, G. H., Pollefeys, M., & Fraundorfer, F. (2014). Relative pose estimation for a multi-camera system with known vertical direction. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 540–547).

Li, B., Martyushev, E., & Lee, G. H. (2020). Relative pose estimation of calibrated cameras with known SE(3) invariants. In *European Conference on Computer Vision*. (pp. 215–231).

Li, H., Hartley, R., & Kim, J.-h. (2008). A linear approach to motion estimation using generalized camera models. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 1–8).

Lim, J., Barnes, N., & Li, H. (2010). Estimating relative camera motion from the antipodal-epipolar constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(10), pp. 1907–1914.

Liu, L., Li, H., Dai, Y., & Pan, Q. (2017). 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), pp. 2432–2444.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), pp. 91–110.

Martyushev, E., & Li, B. (2020). Efficient relative pose estimation for cameras and generalized cameras in case of known relative rotation angle. *Journal of Mathematical Imaging and Vision*, *62*, pp. 1076–1086.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, *22*(10), pp. 761–767.

Mikolajczyk, K., & Schmid, C. (2002). An affine invariant interest point detector. In *European conference on computer vision*. Springer, (pp. 128–142).

Mishkin, D., Matas, J., & Perdoch, M. (2015). MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, *141*, pp. 81–93.

Mishkin, D., Radenovic, F., & Matas, J. (2018). Repeatability is not enough: Learning affine regions via discriminability. In *European Conference on Computer Vision*. (pp. 284–300).

Morel, J.-M., & Yu, G. (2009). ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, *2*(2), pp. 438–469.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(6), pp. 756–777.

Nützi, G., Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Journal of intelligent & robotic systems*, *61*(1-4), pp. 287–299.

Pless, R. (2003). Using many cameras as one. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 1–7).

Quan, L., & Lan, Z. (1999). Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(8), pp. 774–780.

Raposo, C., & Barreto, J. P. (2016). Theory and practice of structure-from-motion using affine correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 5470–5478).

Saurer, O., Vasseur, P., Boutteau, R., Demonceaux, C., Pollefeys, M., & Fraundorfer, F. (2016). Homography based egomotion estimation with a common direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(2), pp. 327–341.

Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry: The first 30 years and fundamentals. *IEEE Robotics & Automation Magazine*, *18*(4), pp. 80–92.

Scaramuzza, D., Fraundorfer, F., & Siegwart, R. (2009). Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *IEEE International Conference on Robotics and Automation*. (pp. 4293–4299).

Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 4104–4113).

Stewénius, H., Oskarsson, M., Aström, K., & Nistér, D. (2005). Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision in conjunction with ICCV*. (pp. 1–8).

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on*

Intelligent Robots and Systems. (pp. 573–580).

Sweeney, C., Flynn, J., Nuernberger, B., Turk, M., & Höllerer, T. (2015a). Efficient computation of absolute pose for gravity-aware augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality*. (pp. 19–24).

Sweeney, C., Flynn, J., & Turk, M. (2014). Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *International Conference on 3D Vision*. (pp. 483–490).

Sweeney, C., Kneip, L., Hollerer, T., & Turk, M. (2015b). Computing similarity transformations from only image correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 3305–3313).

Ventura, J., Arth, C., & Lepetit, V. (2015). An efficient minimal solution for multi-camera motion. In *IEEE International Conference on Computer Vision*. (pp. 747–755).

Zhao, J., Xu, W., & Kneip, L. (2020). A certifiably globally optimal solution to generalized essential matrix estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 12034–12043).