Relative Pose Estimation for Multi-Camera Systems from Affine Correspondences

Preprint · July 2020

CITATIONS
0

READS
132

4 authors, including:

Guan Banglei
National University of Defense Technology
33 PUBLICATIONS **145** CITATIONS

SEE PROFILE

Dániel Baráth
Hungarian Academy of Sciences
78 PUBLICATIONS **517** CITATIONS

SEE PROFILE

Friedrich Fraundorfer
Graz University of Technology
182 PUBLICATIONS **6,902** CITATIONS

SEE PROFILE

Relative Pose Estimation for Multi-Camera Systems from Affine Correspondences

Banglei Guan[1], Ji Zhao[*], Daniel Barath[2,3] and Friedrich Fraundorfer[4,5]

[1]College of Aerospace Science and Engineering, National University of Defense Technology, China

[2]Centre for Machine Perception, Czech Technical University, Czech Republic

[3]Machine Perception Research Laboratory, MTA SZTAKI, Hungary

[4]Institute for Computer Graphics and Vision, Graz University of Technology, Austria

[5]Remote Sensing Technology Institute, German Aerospace Center, Germany

guanbanglei12@nudt.edu.cn zhaoji84@gmail.com barath.daniel@sztaki.mta.hu

fraundorfer@icg.tugraz.at

Abstract

We propose four 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, it is shown that a minimum of two ACs are enough for recovering the 6DOF relative pose, i.e.*, 3D rotation and translation, of the system. Considering planar camera 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 with known gravity vector,* 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 real-world scenes from the KITTI benchmark. It is shown that the accuracy of the estimated poses is superior to the state-of-the-art techniques.*

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 [17, 40, 20, 41, 13], which plays an important role in simultaneous localization and mapping (SLAM), visual odometry (VO) and structure-from-motion (SfM). Thus, improving the accuracy, efficiency and robustness of relative pose estimation algorithms is always an important research topic [28, 46, 45, 1, 2, 42]. Motivated by the fact that multi-camera systems are already available in self-driving cars, micro aerial vehicles or augmented reality headsets, this paper investigates the problem of estimating the relative pose of multi-camera systems from affine correspondences, see Fig. 1.

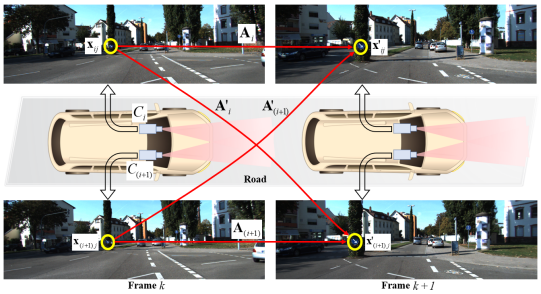


Figure 1. An affine correspondence in camera C_i between consecutive frames k and $k+1$. The local affine transformation $\mathbf{A}$ relates the infinitesimal patches around point correspondence $(\mathbf{x}_{ij}, \mathbf{x}'_{ij})$.

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. The main difference of a multi-camera system and a standard pinhole camera is the absence of a single projection center. A multi-camera system is modeled by the generalized camera model. The light rays that pass through a multi-camera system are expressed as Plücker lines and the epipolar constraint of the Plücker lines is described by the generalized essential matrix [37].

Most of the state-of-the-art SLAM and SfM pipelines using a multi-camera system [16, 18] follow the same procedure consisting of three major steps [40]: first, a feature matching algorithm is applied to establish image point correspondences between two frames. Then a robust es-

*Corresponding author.

timation framework, *e.g.* the Random Sample Consensus (RANSAC) [11], is applied to find the pose parameters and remove outlier matches. Finally, the final relative pose between the two frames is estimated using all RANSAC inliers. The reliability and robustness of such a scheme is heavily dependent on the outlier removal step. In addition, the outlier removal process has to be efficient, which directly affects the real-time performance of SLAM and SfM. The computational complexity and, thus, the processing time of the RANSAC procedure depends exponentially on the number of points required for the estimation. Therefore, exploring the minimal solutions for relative pose estimation of multi-camera system is of significant importance and has received sustained attention [19, 29, 28, 44, 46, 45, 25, 31].

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 [19]. Then other classical works have been subsequently proposed, such as the 17-point linear method [29] and techniques based on iterative optimization [24]. Moreover, the minimal number of necessary points can be further reduced by taking additional motion constraints into account or using other sensors, like an inertial measurement unit (IMU). For example, two point correspondences are sufficient for the ego-motion estimation of a multi-camera system by exploiting the Ackermann motion model constraints of wheeled vehicles [27]. For vehicles equipped with a multi-camera system and an IMU, the relative motion can be estimated from four point correspondences by exploiting the known vertical direction from the IMU measurements, *i.e.*, roll and pitch angles [28, 31].

All of the previously mentioned relative pose solvers estimate the pose parameters from a set of point correspondences, *e.g.*, coming from SIFT [32] or SURF [6] detectors. However, as it has been clearly shown in several recently published papers papers [7, 39, 3, 10], using more informative features, *e.g.* affine correspondences, improves the estimation procedure both in terms of accuracy and efficiency. An affine correspondence is composed of a point correspondence and a 2×2 affine transformation. Due to containing more information, than point correspondences, about the underlying surface geometry, the affine correspondences enable to estimate relative pose from fewer correspondences. In this paper, we focus on the relative pose estimation of a multi-camera system from affine correspondences, instead of point correspondences. Four novel solutions are proposed:

- A new minimal solver is proposed which requires two affine correspondences to estimate the general motion of a multi-camera system which has 6 degrees of freedom (6DOF). In contrast, state-of-the-art solvers use six point correspondences [19, 24, 46].

- When the motion is planar (*i.e.*, the body to which the cameras are fixed moves on a plane; 3DOF), a single affine correspondence is sufficient to recover the planar motion of a multi-camera system. In order to deal with the degenerate case of 1AC solver, we also propose a new method to estimate the relative pose from two affine correspondences. The point-based solution requires two point pairs, but only for the Ackermann motion model [27].

- A fourth 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 affine correspondences are required to recover the relative pose. In contrast, the point-based solver requires four correspondences [28, 44, 31].

2. Related Work

There has been much interest in using multi-camera systems in both academic and industrial communities. The most common case is that a set of cameras, particularly with non-overlapping views, are mounted rigidly on self-driving vehicles, unmanned aerial vehicles (UAV) or AR headsets.

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 as Plücker lines and derived the generalized camera model which has become a standard representation for the multi-camera systems [37]. Stewénius *et al.* proposed the first minimal solution to estimate the relative pose of a multi-camera system from 6 point correspondences, which produces up to 64 solutions [19]. Kim *et al.* later proposed several approaches for motion estimation using second-order cone programming [21] or branch-and-bound techniques [22]. Lim *et al.* presented the antipodal epipolar constraint and estimated the relative motion by using antipodal points [30]. Li *et al.* provided several linear solvers to compute the relative pose, among which the most commonly used one requires 17 point correspondences [29]. Kneip and Li proposed an iterative approach for the relative pose estimation based on eigenvalue minimization [24]. Ventura *et al.* used first-order approximation of the relative rotation to simplify the problem and estimated the relative pose from 6 point correspondences [46].

By considering additional motion constraints or using additional information provided by an IMU, the number of required point correspondences can be further reduced. Lee *et al.* presented a minimal solution with two point correspondences for the ego-motion estimation of a multi-camera system, which constrains the relative motion by the Ackermann motion model [27]. 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 rela-

tive pose estimation with known vertical direction requires a minimum of 4 point correspondences [28, 44, 31].

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 and Francos estimated the fundamental matrix from three ACs [7]. Raposo and Barreto computed homography and essential matrix using two ACs [39]. Barath and Hajder derived the constraints between the local affine transformation and the essential matrix and recovered the essential matrix from two ACs [3]. Eichhardt and Chetverikov [10] also estimated the relative pose from two ACs, which is applicable to arbitrary central-projection models. Hajder and Barath [15] and Guan *et al.* [14] proposed several minimal solutions for relative pose from a single AC under the planar motion assumption or with knowledge of a vertical direction. The above mentioned works are only suitable for the monocular perspective camera, rather than the multiple perspective cameras connected by being fixed to the single body. In this paper, we focuses on the minimal number of ACs to estimate the relative pose of a multi-camera system.

3. Relative Pose Estimation under General Motion

A multi-camera system is made up of individual cameras denoted by C_i, as shown in Fig. 1. Its extrinsic parameters expressed in a multi-camera reference frame are represented as $(\mathbf{R}_i, \mathbf{t}_i)$. For general motion, there is a 3DOF rotation and a 3DOF translation between two reference frames at time k and $k+1$. Rotation $\mathbf{R}$ using Cayley parameterization and translation $\mathbf{t}$ can be written as:

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

$$\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T, \tag{2}$$

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

3.1. Generalized camera model

We give a brief description of generalized camera model (GCM) [37]. Let us denote an affine correspondence in camera C_i between consecutive frames k and $k+1$ as $(\mathbf{x}_{ij}, \mathbf{x}'_{ij}, \mathbf{A})$, where $\mathbf{x}_{ij}$ and $\mathbf{x}'_{ij}$ are the normalized homogeneous image coordinates of feature point j and $\mathbf{A}$ is a 2×2 local affine transformation. Indices i and j are the camera and point index, respectively. The related local affine transformation $\mathbf{A}$ is a 2×2 linear transformation which relates the infinitesimal patches around $\mathbf{x}_{ij}$ and $\mathbf{x}'_{ij}$ [2].

The normalized homogeneous image coordinates $(\mathbf{p}_{ij}, \mathbf{p}'_{ij})$ expressed in the multi-camera reference frame are given as

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

The unit direction of rays $(\mathbf{u}_{ij}, \mathbf{u}'_{ij})$ expressed in the multi-camera reference frame are given as: $\mathbf{u}_{ij} = \mathbf{p}_{ij}/\|\mathbf{p}_{ij}\|$,$\mathbf{u}'_{ij} = \mathbf{p}'_{ij}/\|\mathbf{p}'_{ij}\|$. The 6-dimensional vector Plücker lines corresponding to the rays are denoted as $\mathbf{l}_{ij} = [\mathbf{u}_{ij}^T,\ (\mathbf{t}_i \times \mathbf{u}_{ij})^T]^T$, $\mathbf{l}'_{ij} = [\mathbf{u}'^T_{ij},\ (\mathbf{t}_i \times \mathbf{u}'_{ij})^T]^T$. The generalized epipolar constraint is written as [37]

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

where $\mathbf{l}'^T_{ij}$ and $\mathbf{l}_{ij}$ are Plücker lines between two consecutive frames at time k and $k+1$.

3.2. Affine transformation constraint

We denote the transition matrix of camera coordinate system C_i between consecutive frames k and $k+1$ as $(\mathbf{R}_{Ci}, \mathbf{t}_{Ci})$, which is represented as:

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

The essential matrix $\mathbf{E}$ between two frames of camera C_i is given as:

$$\mathbf{E} = [\mathbf{t}_{Ci}]_\times \mathbf{R}_{Ci} = \mathbf{R}_i^T [\mathbf{R}_i \mathbf{t}_{Ci}]_\times \mathbf{R} \mathbf{R}_i, \tag{6}$$

where $[\mathbf{R}_i \mathbf{t}_{Ci}]_\times = \mathbf{R}[\mathbf{t}_i]_\times \mathbf{R}^T + [\mathbf{t}]_\times - [\mathbf{t}_i]_\times$. The relationship of essential matrix $\mathbf{E}$ and local affine transformation $\mathbf{A}$ is formulated as follows [3]:

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

where $\mathbf{n}_{ij} \triangleq \mathbf{E}^T \mathbf{x}'_{ij}$ and $\mathbf{n}'_{ij} \triangleq \mathbf{E}\mathbf{x}_{ij}$ denote the epipolar lines in their implicit form in frames of camera C_i at time k and $k+1$. The subscript 1 and 2 represent the first and second equations of the equation system, respectively. $\hat{\mathbf{A}}$ is a 3×3 matrix: $\hat{\mathbf{A}} = [\mathbf{A}\ \mathbf{0}; \mathbf{0}\ 0]$. By substituting Eq. (6) into Eq. (7), we obtain:

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

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

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

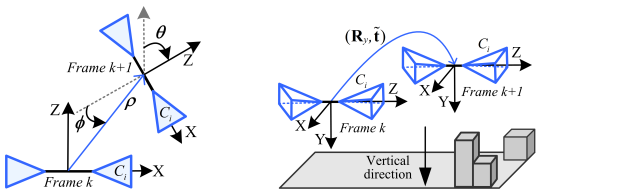
Equation (9) interprets the epipolar constraints which a local affine transformation implies on the i-th camera from a multi-camera system between two consecutive frames k and $k+1$.

3.3. Solution using Gröbner basis method

For affine correspondence $(\mathbf{x}_{ij}, \mathbf{x}'_{ij}, \mathbf{A})$, we get three polynomials for six unknowns $\{q_x, q_y, q_z, t_x, t_y, t_z\}$ from Eqs. (4) and (9). Thus two affine correspondences are enough to recover the relative pose of a multi-camera system under 6DOF general motion. The hidden variable resultant method [9] is used to solve for the unknowns, see supplementary material for details. The obtained solver is however too large and, therefore, slow and numerically unstable. Experiments confirmed that the solver is numerically unstable and, thus, no further experiments and comparisons are presented in the paper.

We furthermore investigate the special cases of multi-camera motion, *i.e.*, planar motion and motion with known vertical direction, see Fig. 2. We will show that two special cases can be efficiently solved with affine correspondences.

4. Relative Pose Estimation Under Planar Motion



(a) Planar motion (b) Motion with known vertical direction

Figure 2. Special cases of multi-camera motion: (a) Planar motion between two multi-camera reference frames in top-view. There are three unknowns: yaw angle θ, translation direction ϕ and translation distance ρ. (b) Motion with known vertical direction. There are four unknowns: a Y-axis rotation $\mathbf{R}_y$ and 3D translation $\tilde{\mathbf{t}} = [\tilde{t}_x, \tilde{t}_y, \tilde{t}_z]^T$.

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

$$\mathbf{R}_y = \frac{1}{1+q_y^2}\begin{bmatrix} 1-q_y^2 & 0 & -2q_y \\ 0 & 1+q_y^2 & 0 \\ 2q_y & 0 & 1-q_y^2 \end{bmatrix}, \qquad \mathbf{t} = \begin{bmatrix} t_x & 0 & t_z \end{bmatrix}^T. \tag{10}$$

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. Solution by reduction to a single polynomial

By substituting Eq. (10) into Eqs. (4) and (9), we get an equation system of three polynomials for 3 unknowns q_y, t_x and t_z. Since an AC generally provides 3 independent constraints for relative pose, a single affine correspondence is sufficient to recover the planar motion of a multi-camera system. Three independent constraints from an affine correspondence are stacked into 3 equations in 3 unknowns:

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

where the elements M_{ij} $(i = 1,\ldots,3; j = 1,\ldots,3)$ of the coefficient matrix $\mathbf{M}(q_y)$ are formed by the polynomial coefficients and one unknown variable q_y, see supplementary material for details. Since $\mathbf{M}(q_y)/(1+q_y^2)$ is a square matrix, Eq. (11) has a non-trivial solution only if the determinant of $\mathbf{M}(q_y)/(1+q_y^2)$ is zero. The expansion of $\det(\mathbf{M}(q_y)/(1+q_y^2)) = 0$ gives an 4-degree univariate polynomial:

$$\mathrm{quot}(\textstyle\sum_{i=0}^{6} w_i q_y^i, q_y^2+1) = 0, \tag{12}$$

where $\mathrm{quot}(a,b)$ means calculating the quotient of a divided by b, $w_0,\ldots,w_6$ are formed by a Plücker line correspondence and a affine transformation between the corresponding feature points. This univariate polynomial 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. (10).

However, we proved that the solver relies on one AC has a degenerate case, *i.e.*, the distances between motion plane and optical centers of individual cameras are equal, see supplementary material for details. This degenerate case often happens in the self-driving scenario. To overcome this issue, two affine correspondences are used to estimate the relative pose. For example, the first and second constraints of the first affine correspondence, and the first constraint of the second affine correspondence are also stacked into 3 equations in 3 unknowns, just as Eq. (11). The solution procedure remains the same, except that the code for constructing the coefficient matrix $\mathbf{M}(q_y)$ is replaced.

An interesting fact in this case is that only three equations from two affine correspondences are used. Although two affine correspondences are required to sample for this solver in the RANSAC loop, it is possible to run a consistency check on two affine correspondences. To identify an outlier free planar motion estimation hypothesis, the three remaining equations of two affine correspondences have also to be fulfilled. The solutions which do not fulfill

the hypothesis would be preemptively rejected. This gives a significant computational advantage over the regular 2-point method, such as the solver with Ackermann motion assumption [27], because the inconsistent samples can be detected directly without testing on all the other affine correspondences.

5. Relative Pose Estimation with Known Vertical Direction

In this section a minimal solution using two affine correspondences is proposed for relative motion estimation for multi-camera systems with known vertical direction, see Fig. 2(b). 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 reference frame. So the reference frame can be aligned with the measured gravity 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 gravity direction. Rotation $\mathbf{R}_{\text{imu}}$ for aligning the reference frame to the aligned reference frame is written as:

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

where θ_r and θ_p are roll and pitch angles provided by the coupled IMU, respectively. Thus, there are only a Y-axis rotation $\mathbf{R} = \mathbf{R}_y$ and 3D translation $\tilde{\mathbf{t}} = \mathbf{R}'_{\text{imu}}\mathbf{t} = [\tilde{t}_x, \tilde{t}_y, \tilde{t}_z]^T$ to be estimated between the aligned multi-camera reference frames at time k and $k+1$.

5.1. Generalized camera model

Let us denote the rotation matrices from the roll and pitch angles of the two corresponding multi-camera reference frames at time k and $k+1$ as $\mathbf{R}_{\text{imu}}$ and $\mathbf{R}'_{\text{imu}}$. The relative rotation between two multi-camera reference frames can now be given as:

$$\mathbf{R} = (\mathbf{R}'_{\text{imu}})^T\mathbf{R}_y\mathbf{R}_{\text{imu}}. \tag{13}$$

We substitute Eq. (13) into Eq. (4) yields:

$$\underbrace{\left(\begin{bmatrix}\mathbf{R}'_{\text{imu}} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}'_{\text{imu}}\end{bmatrix}\mathbf{l}'_{ij}\right)^T}_{\tilde{\mathbf{l}}'_{ij}}\begin{bmatrix}[\tilde{\mathbf{t}}]_\times\mathbf{R}_y & \mathbf{R}_y\\ \mathbf{R}_y & \mathbf{0}\end{bmatrix}\cdot \underbrace{\left(\begin{bmatrix}\mathbf{R}_{\text{imu}} & \mathbf{0}\\ \mathbf{0} & \mathbf{R}_{\text{imu}}\end{bmatrix}\mathbf{l}_{ij}\right)}_{\tilde{\mathbf{l}}_{ij}} = 0, \tag{14}$$

where $\tilde{\mathbf{l}}_{ij} \leftrightarrow \tilde{\mathbf{l}}'_{ij}$ are the corresponding Plücker lines expressed in the aligned multi-camera reference frame.

5.2. Affine transformation constraint

In this case, the transition matrix of the camera coordinate system C_i between consecutive frames k and $k+1$ is represented as

$$\begin{aligned}\begin{bmatrix}\mathbf{R}_{Ci} & \mathbf{t}_{Ci}\\ \mathbf{0} & 1\end{bmatrix} = &\left(\begin{bmatrix}\mathbf{R}'_{\text{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}\right)^{-1}\cdot \\ &\begin{bmatrix}\mathbf{R}_y & \tilde{\mathbf{t}}\\ \mathbf{0} & 1\end{bmatrix}\left(\begin{bmatrix}\mathbf{R}_{\text{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}\right),\end{aligned} \tag{15}$$

we denote that

$$\begin{aligned}\begin{bmatrix}\tilde{\mathbf{R}}_{\text{imu}} & \tilde{\mathbf{t}}_{\text{imu}}\\ \mathbf{0} & 1\end{bmatrix} &= \begin{bmatrix}\mathbf{R}_{\text{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}, \\ \begin{bmatrix}\tilde{\mathbf{R}}'_{\text{imu}} & \tilde{\mathbf{t}}'_{\text{imu}}\\ \mathbf{0} & 1\end{bmatrix} &= \begin{bmatrix}\mathbf{R}'_{\text{imu}} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}.\end{aligned} \tag{16}$$

By substituting Eq. (16) into Eq. (15), we obtain

$$\begin{aligned}&\begin{bmatrix}\mathbf{R}_{Ci} & \mathbf{t}_{Ci}\\ \mathbf{0} & 1\end{bmatrix} \\ &= \begin{bmatrix}(\tilde{\mathbf{R}}'_{\text{imu}})^T\mathbf{R}_y\tilde{\mathbf{R}}_{\text{imu}} & (\tilde{\mathbf{R}}'_{\text{imu}})^T(\mathbf{R}_y\tilde{\mathbf{t}}_{\text{imu}} + \tilde{\mathbf{t}} - \tilde{\mathbf{t}}'_{\text{imu}})\\ \mathbf{0} & 1\end{bmatrix}.\end{aligned} \tag{17}$$

The essential matrix $\mathbf{E}$ between two frames of camera C_i is given as

$$\mathbf{E} = [\mathbf{t}_{Ci}]_\times\mathbf{R}_{Ci} = (\tilde{\mathbf{R}}'_{\text{imu}})^T[\tilde{\mathbf{R}}'_{\text{imu}}\mathbf{t}_{Ci}]_\times\mathbf{R}_y\tilde{\mathbf{R}}_{\text{imu}}, \tag{18}$$

where $[\tilde{\mathbf{R}}'_{\text{imu}}\mathbf{t}_{Ci}]_\times = \mathbf{R}_y[\tilde{\mathbf{t}}_{\text{imu}}]_\times\mathbf{R}_y^T + [\tilde{\mathbf{t}}]_\times - [\tilde{\mathbf{t}}'_{\text{imu}}]_\times$. By substituting Eq. (18) into Eq. (7), we obtain

$$\begin{aligned}&(\tilde{\mathbf{R}}^T_{\text{imu}}\mathbf{R}_y^T[\tilde{\mathbf{R}}'_{\text{imu}}\mathbf{t}_{Ci}]_\times^T\tilde{\mathbf{R}}'_{\text{imu}}\mathbf{x}'_{ij})_{(1:2)} = \\ &\quad -(\hat{\mathbf{A}}^T(\tilde{\mathbf{R}}'_{\text{imu}})^T[\tilde{\mathbf{R}}'_{\text{imu}}\mathbf{t}_{Ci}]_\times\mathbf{R}_y\tilde{\mathbf{R}}_{\text{imu}}\mathbf{x}_{ij})_{(1:2)}.\end{aligned} \tag{19}$$

We denote the normalized homogeneous image coordinates expressed in the aligned multi-camera reference frame as $(\tilde{\mathbf{p}}_{ij}, \tilde{\mathbf{p}}'_{ij})$, which are given as

$$\tilde{\mathbf{p}}_{ij} = \tilde{\mathbf{R}}_{\text{imu}}\mathbf{x}_{ij}, \qquad \tilde{\mathbf{p}}'_{ij} = \tilde{\mathbf{R}}'_{\text{imu}}\mathbf{x}'_{ij}. \tag{20}$$

Based on the above equation, Eq. (19) is rewritten and expanded as follows:

$$\begin{aligned}&(\tilde{\mathbf{R}}^T_{\text{imu}}([\tilde{\mathbf{t}}_{\text{imu}}]_\times\mathbf{R}_y^T + \mathbf{R}_y^T[\tilde{\mathbf{t}}]_\times - \mathbf{R}_y^T[\tilde{\mathbf{t}}'_{\text{imu}}]_\times)\tilde{\mathbf{p}}'_{ij})_{(1:2)} = \\ &(\hat{\mathbf{A}}^T(\tilde{\mathbf{R}}'_{\text{imu}})^T(\mathbf{R}_y[\tilde{\mathbf{t}}_{\text{imu}}]_\times + [\tilde{\mathbf{t}}]_\times\mathbf{R}_y - [\tilde{\mathbf{t}}_{\text{imu}}]_\times\mathbf{R}_y)\tilde{\mathbf{p}}_{ij})_{(1:2)}\end{aligned} \tag{21}$$

5.3. Solution by reduction to a single polynomial

Based on Eqs. (14) and (21), we get an equation system of three polynomials for 4 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 affine correspondence. In principle, one arbitrary equation can be chosen from Eqs. (14) and (21), for example, three constraints of the first affine correspondence, and the first constraint of the second affine correspondence are stacked into 4 equations in 4 unknowns:

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

where the elements $\tilde{M}_{ij}(i = 1,\dots,4; j = 1,\dots,4)$ of the coefficient matrix $\tilde{\mathbf{M}}(q_y)$ are formed by the polynomial coefficients and one unknown variable q_y, see supplementary material for details. Since $\tilde{\mathbf{M}}(q_y)/(1+q_y^2)$ is a square matrix, Eq. (22) has a non-trivial solution only if the determinant of $\tilde{\mathbf{M}}(q_y)/(1+q_y^2)$ is zero. The expansion of $\det(\tilde{\mathbf{M}}(q_y)/(1+q_y^2)) = 0$ gives a 6-degree univariate polynomial:

$$\mathrm{quot}(\textstyle\sum_{i=0}^{8} w_i q_y^i, q_y^2+1) = 0, \quad (23)$$

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

This univariate polynomial leads to a closed-form solution with a maximum of 6 real roots. Equation (23) can be efficiently solved by the companion matrix method [9] or Sturm bracketing method [35]. Once q_y has been obtained, the rotation matrix $\mathbf{R}_y$ is recovered from Eq. (10). For the relative pose between two multi-camera reference frames at time k and $k+1$, the rotation matrix $\mathbf{R}$ is recovered from Eq. (13) and the translation is computed by $\mathbf{t} = (\mathbf{R}'_{\text{imu}})^T\tilde{\mathbf{t}}$. Note that two remaining equations of the second affine correspondence can be also used in the preemptive hypothesis tests, which detect and reject inconsistent samples directly.

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 methods.

For relative pose estimation under planar motion, the solvers using 1 AC and 2 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 `17pt-Li` [29], `8pt-Kneip` [24] and `6pt-Stewénius` [19], which are provided in the OpenGV library [23]. Since the Ackermann motion model is restrictive in practice and usually requires a post-relaxation [27, 31], the methods using the Ackermann motion model are not compared in this paper.

For relative pose estimation with known vertical direction, the solver proposed in Section 5 is referred to as the `2AC method`. We compare the accuracy of `2AC method` with `17pt-Li` [29], `8pt-Kneip` [24], `6pt-Stewénius` [19], `4pt-Lee` [28], `4pt-Sweeney` [44] and `4pt-Liu` [31].

The proposed methods `1AC plane`, `2AC plane` and `2AC method` take about 3.6, 3.6 and 17.8 μs in C++. Due to space limitations, the efficiency comparison and stability study are provided in the supplementary material. In the experiments, all the solvers are implemented within RANSAC to reject outliers. 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 [23]. We also show the feasibility of our methods on the `KITTI` dataset [12]. This experiment demonstrates that our methods are well suited for visual odometry in road driving scenarios.

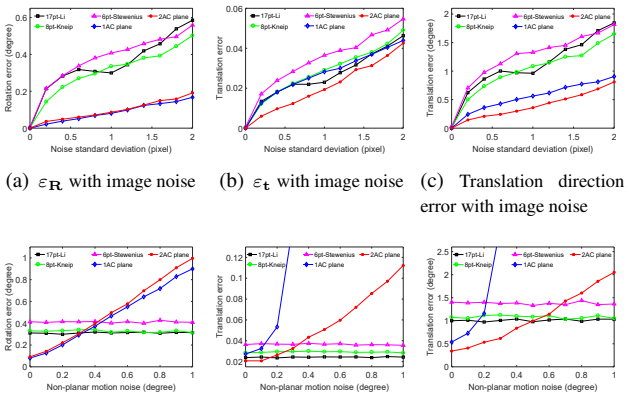
6.1. Experiments on synthetic data

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 heights. The multi-camera reference frame is built at the middle of 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 (X-axis direction), -5 to 5 meters (Y-axis direction), and 10 to 20 meters (Z-axis direction), which are expressed in the respective axis of the multi-camera reference frame. We choose 50 ACs from the ground plane and an AC from each random plane randomly. Thus, there are 100 ACs generated randomly in the synthetic data. For each AC, a random 3D point from a plane is reprojected onto two cameras to get the image point pair. The corresponding affine transformation is obtained by the following procedure. First, the implicit homography is calculated for each plane by four random, not col-linear, additional 3D points from the same plane; projecting them to the cameras; adding Gaussian noise with a standard deviation to the image coordinates, which is similar to the noise added to the coordinates of image point pair; and, finally, estimating the homography. The affine parameters is the first-order approximation of the noisy homography matrix, which the plane implies at the image point pair. The 3D points initializing both the image point pair and the homography are

selected randomly considering both the image size and the range of the synthetic scene. Note that the homography can be calculated directly from the plane normal and distance. However, using four projected additional random 3D points enables an indirect but geometrically interpretable way of adding noise to the affine transformation [4].

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 scheme. This also allows us to select the best candidate from multiple solutions. 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: $\varepsilon_{\mathbf{R}} = \arccos((\text{trace}(\mathbf{R}_{gt}\mathbf{R}^T) - 1)/2)$, where $\mathbf{R}_{gt}$ and $\mathbf{R}$ are the ground truth and estimated rotation matrices. Following the definition in [38, 28], 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.



(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-planar motion noise (e) $\varepsilon_{\mathbf{t}}$ with non-planar motion noise (f) Translation direction error with non-planar motion noise

Figure 3. Rotation and translation error under planar motion. (a) $\sim$ (c): vary image noise under perfect planar motion. (d) $\sim$ (f): vary non-planar motion noise and fix the standard deviation of image noise at 1.0 pixel.

6.1.1 Planar motion estimation

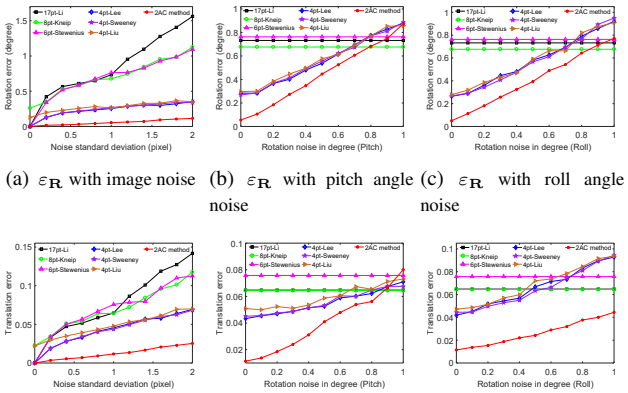
In this scenario, the planar motion of the multi-camera system is described by (θ, ϕ), see Fig. 2(a). The magnitudes of both angles ranges from $-10°$ to $10°$. The magnitude of image noise is set to Gaussian noise with a standard deviation ranging from 0 to 2 pixel. Figure 3(a) $\sim$ (c) show the performance of the proposed `1AC plane` method and `2AC plane` method against image noise. The `2AC plane` method performs better than comparative methods under perfect planar motion. In comparison with the `2AC plane` method, the `1AC plane` method has similar performance in rotation estimation, but performs slightly worse in translation estimation. As shown in Fig. 3(c) and (f), we plot the translation direction error as an additional evaluation. It is interesting to see that the `1AC plane` method also performs better than comparative methods in translation direction estimation.

We also evaluate the accuracy of the proposed `1AC plane` method and `2AC plane` method for increasing non-planar motion noise. The non-planar components of a 6DOF relative pose including X-axis rotation, Z-axis rotation and direction of YZ-plane translation [8] are randomly generated and added 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 1.0 pixel. Figures 3(d) $\sim$ (f) show the performance of the proposed `1AC plane` method and `2AC plane` method against non-planar motion noise. Methods `17pt-Li`, `8pt-Kneip` and `6pt-Stewénius` deal with the 6DOF motion case and, thus they are not affected by the noise in the planarity assumption. It can be seen that the rotation accuracy of `2AC plane` method performs better than comparative methods when the non-planar motion noise is less than $0.3°$. Since the estimation accuracy of translation direction of the `2AC plane` method in Fig. 3(f) performs satisfactory, the main reason for poor performance of translation estimation is that the metric scale estimation is sensitive to the non-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 when the non-planar motion noise is more than $0.2°$.

Both the `1AC plane` method and the `2AC plane` method have a significant computational advantage over comparative methods, because the efficient solver for 4-degree polynomial equation takes only about 3.6 μs. A more interesting fact for the `2AC plane` method is the speed-up gained by the preemptive hypothesis tests, which detect and reject inconsistent samples directly. Compared with testing on the other affine correspondences, the preemptive hypothesis tests sped up the procedure by more than three times while leading to the same accuracy of relative pose estimation.

6.1.2 Motion with known vertical direction

In this set of experiments, the translation direction between two multi-camera reference frames is chosen to produce either forward, sideways or random motions. In addition, the second reference frame is rotated around three axes in order and the rotation angles range from $-10°$ to $10°$. With the assumption that the roll and pitch angles are known, the



(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

Figure 4. Rotation and translation error under random motion with known vertical direction. The upper row: rotation error, the bottom row: translation error. (a)(d): vary image noise. (b)(e) and (c)(f): vary IMU angle noise and fix the standard deviation of image noise at 1.0 pixel.

multi-camera reference frame is aligned with the gravity direction. Due to space limitations, we only show the results for random motion. The results for forward and sideways motions are shown in the supplementary material. Figure 4(a) and (d) show the performance of the `2AC method` against image noise with perfect IMU data in the random motion case. It can be seen that the proposed method is robust to image noise and performs better than the comparative methods.

Figure 4(b)(e) and (c)(f) show the performance of the proposed `2AC method` against IMU noise in the random motion case, while the standard deviation of the image noise is fixed at 1.0 pixel. Note that the methods `17pt-Li`, `8pt-Kneip` and `6pt-Stewénius` are not influenced by IMU noise, because these methods do not use the known vertical direction as a prior. It is interesting to see that our method outperforms the methods `17pt-Li`, `8pt-Kneip` and `6pt-Stewénius` in the random motion case, even though the IMU noise is around $0.8°$. In addition, the proposed `2AC method` performs better than the methods `4pt-Lee`, `4pt-Sweeney` and `4pt-Liu` as well, which also use the known vertical direction as a prior. The results under forward and sideways motion also demonstrate that the `2AC method` performs basically better than all comparative methods against image noise and provides comparable accuracy for increasing IMU noise. It is worth to mention that, with the help of preemptive hypothesis tests, the relative pose estimation with the proposed `2AC method` solver sped up the procedure by more than three times while leading to similarly accurate relative poses.

6.2. Experiments on real data

We test the performance of our methods on the `KITTI` dataset [12], which consists of successive video frames from a forward facing stereo camera. 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 affine correspondences between consecutive frames in each camera are established by applying the ASIFT [34]. It can also be obtained by MSER [33] which will be slightly less accurate but much faster to obtain [5]. The affine correspondences 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 IMU is more suitable for recovering the metric scale [36]. All the solvers have been integrated into a RANSAC scheme.

The proposed `2AC plane` method and `2AC method` are compared against `17pt-Li` [29], `8pt-Kneip` [24], `6pt-Stewénius` [19], `4pt-Lee` [28], `4pt-Sweeney` [44] and `4pt-Liu` [31]. Since the `KITTI` dataset is captured by the stereo camera with the same height, which is a degenerate case for the `1AC plane` method, this method is not performed in the experiment. For the `2AC plane` method, the estimation results are also compared with the 6DOF ground truth of relative pose, even though this method only estimates two angles (θ, ϕ) with the plane motion assumption. For the `2AC method`, the roll and pitch angles obtained from the ground truth data are used to simulate IMU measurements, which align the multi-camera reference frame with the gravity direction. To ensure the fairness of the experiment, the roll and pitch angles are also provided for the methods `4pt-Lee` [28], `4pt-Sweeney` [44] and `4pt-Liu` [31]. The results of the rotation and translation estimation are shown in Table 1. The runtime of RANSAC averaged over `KITTI` sequences combined with different solvers is shown in Table 2.

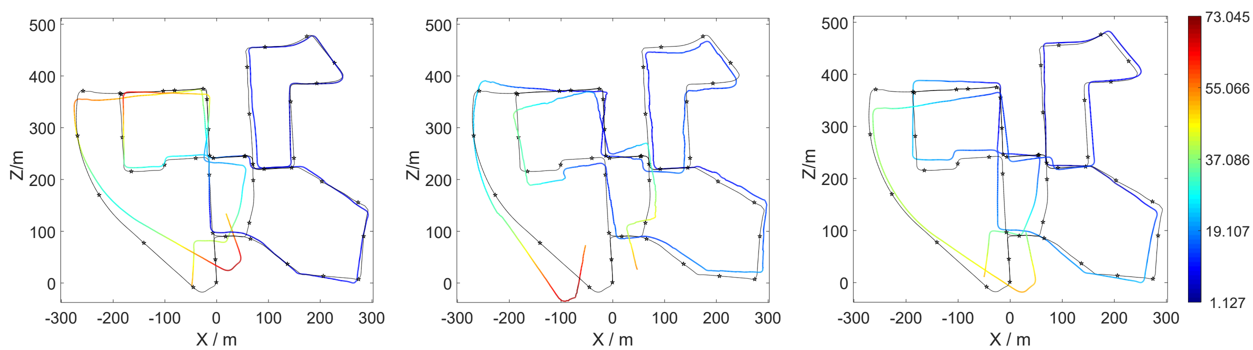
The `2AC method` offers the best overall performance among all the methods. The `6pt-Stewénius` method performs poorly on sequence 01, because this sequence is a highway with few tractable close objects, and this method always fails to select the best candidate from multiple solutions under forward motion in the RANSAC scheme. Besides, it is interesting to see that the translation accuracy of `2AC plane` method basically outperforms the `6pt-Stewénius` method, even though the planar motion assumption does not fit the `KITTI` dataset well. Due to the benefits of computational efficiency, both the `2AC plane` method and the `2AC method` are quite suitable for finding a correct inlier set, which is then used for accurate motion estimation in visual odometry.

Table 1. Rotation and translation error on KITTI sequences (unit: degree).

Seq.	17pt-Li [29]		8pt-Kneip [24]		6pt-St. [19]		4pt-Lee [28]		4pt-Sw. [44]		4pt-Liu [31]		**2AC plane**		**2AC method**	
	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$	$\varepsilon_{\mathbf{R}}$	$\varepsilon_{\mathbf{t}}$
00	0.139	2.412	0.130	2.400	0.229	4.007	0.065	2.469	0.050	2.190	0.066	2.519	0.280	2.243	**0.031**	**1.738**
01	0.158	5.231	0.171	4.102	0.762	41.19	0.137	4.782	0.125	11.91	0.105	3.781	0.168	2.486	**0.025**	**1.428**
02	0.123	1.740	0.126	1.739	0.186	2.508	0.057	1.825	0.044	1.579	0.057	1.821	0.213	1.975	**0.030**	**1.558**
03	0.115	2.744	0.108	2.805	0.265	6.191	0.064	3.116	0.069	3.712	0.062	3.258	0.238	**1.849**	**0.037**	1.888
04	0.099	1.560	0.116	1.746	0.202	3.619	0.050	1.564	0.051	1.708	0.045	1.635	0.116	1.768	**0.020**	**1.228**
05	0.119	2.289	0.112	2.281	0.199	4.155	0.054	2.337	0.052	2.544	0.056	2.406	0.185	2.354	**0.022**	**1.532**
06	0.116	2.071	0.118	1.862	0.168	2.739	0.053	1.757	0.092	2.721	0.056	1.760	0.137	2.247	**0.023**	**1.303**
07	0.119	3.002	0.112	3.029	0.245	6.397	0.058	2.810	0.065	4.554	0.054	3.048	0.173	2.902	**0.023**	**1.820**
08	0.116	2.386	0.111	2.349	0.196	3.909	0.051	2.433	0.046	2.422	0.053	2.457	0.203	2.569	**0.024**	**1.911**
09	0.133	1.977	0.125	1.806	0.179	2.592	0.056	1.838	0.046	1.656	0.058	1.793	0.189	1.997	**0.027**	**1.440**
10	0.127	1.889	0.115	1.893	0.201	2.781	0.052	1.932	0.040	1.658	0.058	1.888	0.223	2.296	**0.025**	**1.586**

Table 2. Runtime of RANSAC averaged over KITTI sequences combined with different solvers (unit: s).

Methods	17pt-Li [29]	8pt-Kneip [24]	6pt-St. [19]	4pt-Lee [28]	4pt-Sw. [44]	4pt-Liu [31]	**2AC plane**	**2AC method**
Mean time	52.82	10.36	79.76	0.85	0.63	0.45	**0.07**	0.09
Standard deviation	2.62	1.59	4.52	0.093	0.057	0.058	**0.0071**	0.0086



(a) 8pt-Kneip (b) 4pt-Sweeney (c) 2AC method

Figure 5. Estimated trajectories without any post-refinement. The relative pose measurements between consecutive frames are directly concatenated. Colorful curves are estimated trajectories with 8pt-Kneip [24], 4pt-Sweeney [44] and 2AC method. Black curves with stars are ground truth trajectories. Best viewed in color.

To visualize the comparison results, the estimated trajectory for sequence 00 is plotted in Fig. 5. We are directly concatenating frame-to-frame relative pose measurements without any post-refinement. The trajectory for the 2AC method is compared with the two best performing comparison methods in sequence 00 based on Table 1: the 8pt-Kneip method in 6DOF motion case and the 4pt-Sweeney method 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) [43]. Even though all trajectories have a significant accumulation of drift, it can still be seen that the proposed 2AC method has the smallest ATE among the compared trajectories.

7. Conclusion

By exploiting the affine parameters, we have proposed four solutions for the relative pose estimation of a multi-camera system. A minimum of two affine correspondences is needed to estimate the 6DOF relative pose of a multi-camera system. Under the planar motion assumption, we present two solvers to recover the planar motion of a multi-camera system, including a minimal solver with a single affine correspondence and a solver with two affine correspondences. In addition, a minimal solution with two affine

correspondences is also proposed to solve for the relative pose of the multi-camera system with known vertical direction. The assumptions taken in these solutions are commonly met in road driving scenes. We evaluate the latter two solutions on synthetic data and 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.

References

[1] Sameer Agarwal, Hon-Leung Lee, Bernd Sturmfels, and Rekha R. Thomas. On the existence of epipolar matrices. *International Journal of Computer Vision*, 121(3):403–415, 2017.

[2] Daniel Barath. Five-point fundamental matrix estimation for uncalibrated cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 235–243, 2018.

[3] Daniel Barath and Levente Hajder. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, 27(11):5328–5337, 2018.

[4] Daniel Barath and Zuzana Kukelova. Homography from two orientation-and scale-covariant features. In *IEEE International Conference on Computer Vision*, pages 1091–1099, 2019.

[5] Daniel Barath, Jiri Matas, and Levente Hajder. Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry. In *British Machine Vision Conference*, 2016.

[6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[7] Jacob Bentolila and Joseph M Francos. Conic epipolar constraints from affine correspondences. *Computer Vision and Image Understanding*, 122:105–114, 2014.

[8] Sunglok Choi and Jong-Hwan Kim. Fast and reliable minimal relative pose estimation under planar motion. *Image and Vision Computing*, 69:103–112, 2018.

[9] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[10] Iván Eichhardt and Dmitry Chetverikov. Affine correspondences between central cameras for rapid relative pose estimation. In *European Conference on Computer Vision*, pages 482–497, 2018.

[11] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[13] Banglei Guan, Pascal Vasseur, Cédric Demonceaux, and Friedrich Fraundorfer. Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions. In *IEEE International Conference on Robotics and Automation*, pages 2320–2327, 2018.

[14] Banglei Guan, Ji Zhao, Zhang Li, Fang Sun, and Friedrich Fraundorfer. Minimal solutions for relative pose with a single affine correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1929–1938, 2020.

[15] Levente Hajder and Daniel Barath. Relative planar motion for vehicle-mounted cameras from a single affine correspondence. In *IEEE International Conference on Robotics and Automation*, pages 8651–8657, 2020.

[16] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017.

[17] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[18] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Geppert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, Gim Hee Lee, Marc Pollefeys, and Torsten Sattler. Project AutoVision: Localization and 3D scene perception for an autonomous vehicle with a multi-camera system. In *IEEE International Conference on Robotics and Automation*, pages 4695–4702, 2019.

[19] Stewénius Henrik, Oskarsson Magnus, Kalle Aström, and David Nistér. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision in conjunction with ICCV*, pages 1–8, 2005.

[20] Tim Kazik, Laurent Kneip, Janosch Nikolic, Marc Pollefeys, and Roland Siegwart. Real-time 6D stereo visual odometry with non-overlapping fields of view. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1529–1536, 2012.

[21] Jae-Hak Kim, Richard Hartley, Jan-Michael Frahm, and Marc Pollefeys. Visual odometry for non-overlapping views using second-order cone programming. In *Asian Conference on Computer Vision*, pages 353–362, 2007.

[22] Jae-Hak Kim, Hongdong Li, and Richard Hartley. Motion estimation for nonoverlapping multicamera rigs: Linear algebraic and L_∞ geometric solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1044–1059, 2009.

[23] Laurent Kneip and Paul Furgale. OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In *IEEE International Conference on Robotics and Automation*, pages 1–8, 2014.

[24] Laurent Kneip and Hongdong Li. Efficient computation of relative pose for multi-camera systems. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 446–453, 2014.

[25] Laurent Kneip, Chris Sweeney, and Richard Hartley. The generalized relative pose and scale problem: View-graph fusion via 2D-2D registration. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9, 2016.

[26] Viktor Larsson, Kalle Aström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 820–828, 2017.

[27] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013.

[28] Gim Hee Lee, Marc Pollefeys, and Friedrich Fraundorfer. Relative pose estimation for a multi-camera system with known vertical direction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 540–547, 2014.

[29] Hongdong Li, Richard Hartley, and Jae-hak Kim. A linear approach to motion estimation using generalized camera models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[30] John Lim, Nick Barnes, and Hongdong Li. Estimating relative camera motion from the antipodal-epipolar constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1907–1914, 2010.

[31] Liu Liu, Hongdong Li, Yuchao Dai, and Quan Pan. Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2432–2444, 2017.

[32] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[33] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[34] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[35] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):0756–777, 2004.

[36] Gabriel Ntzi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Journal of intelligent & robotic systems*, 61(1-4):287–299, 2011.

[37] Robert Pless. Using many cameras as one. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–587, 2003.

[38] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, 1999.

[39] Carolina Raposo and Joao P Barreto. Theory and practice of structure-from-motion using affine correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5470–5478, 2016.

[40] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry: The first 30 years and fundamentals. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011.

[41] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[42] Thiago L. T. da Silveira and Claudio R. Jung. Perturbation analysis of the 8-point algorithm: A case study for wide FoV cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11757–11766, 2019.

[43] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[44] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *IEEE International Conference on 3D Vision*, pages 483–490, 2014.

[45] Chris Sweeney, Laurent Kneip, Tobias Hollerer, and Matthew Turk. Computing similarity transformations from only image correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3305–3313, 2015.

[46] Jonathan Ventura, Clemens Arth, and Vincent Lepetit. An efficient minimal solution for multi-camera motion. In *IEEE International Conference on Computer Vision*, pages 747–755, 2015.

Supplementary Material

A. Relative Pose Estimation under General Motion

A.1. Solution using Gröbner basis method

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

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

where the elements M_{ij} $(i = 1, \ldots, 3; j = 1, \ldots, 4)$ 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:

$$\mathbf{M}(q_x, q_y, q_z) = \begin{bmatrix} [2] & [2] & [2] & [2] \\ [2] & [2] & [2] & [2] \\ [2] & [2] & [2] & [2] \end{bmatrix}, \tag{25}$$

where $[N]$ denotes a polynomial of degree N in variables q_x, q_y, q_z.

For the general case, Eq. (24) imposes three independent constraints on six unknowns $\{q_x, q_y, q_z, t_x, t_y, t_z\}$. Thus two affine correspondences are enough to recover the relative pose of a multi-camera system under 6DOF general motion. Hence, we get an equation system of 6 independent

constraints from 2 affine correspondences in similar form as Eq. (24). These constraints are stacked into six equations in six unknowns:

$$\frac{1}{1+q_x^2+q_y^2+q_z^2}\underbrace{\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \\ M_{51} & M_{52} & M_{53} & M_{54} \\ M_{61} & M_{62} & M_{63} & M_{64} \end{bmatrix}}_{\mathbf{M}_{6\times 4}} \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} = \mathbf{0}, \tag{26}$$

Since $\mathbf{M}_{6\times 4}/(1+q_x^2+q_y^2+q_z^2)$ has a null vector, its rank must be at most three. Thus, all the 4×4 sub-determinants of $\mathbf{M}_{6\times 4}/(1+q_x^2+q_y^2+q_z^2)$ must be zero. In this paper, three sub-matrices which give three equations in three unknowns q_x, q_y, q_z are choose as follows:

$$\begin{cases} \det(\frac{1}{1+q_x^2+q_y^2+q_z^2}\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}) = 0 \\ \det(\frac{1}{1+q_x^2+q_y^2+q_z^2}\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{51} & M_{52} & M_{53} & M_{54} \end{bmatrix}) = 0 \\ \det(\frac{1}{1+q_x^2+q_y^2+q_z^2}\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{61} & M_{62} & M_{63} & M_{64} \end{bmatrix}) = 0 \end{cases} \tag{27}$$

The hidden variable resultant method [9] is used to solve for the unknowns of Eq. (27). By grouping the unknowns q_x, q_y, q_z with the known coefficients, we obtain an equation system with 84 monomials which consist of q_x, q_y, q_z. Note that the coefficients are divided by $1+q_x^2+q_y^2+q_z^2$, which reduces the polynomial degree and improves the efficiency of the solution. The final solver was obtained by the automatic solver generator of [26]. This equation system has a maximum polynomial degree of 6.

B. Relative Pose Estimation Under Planar Motion

B.1. Details about the coefficient matrix $\mathbf{M}(q_y)$

Refer to Eq. (12) in the paper, three constraints obtained from two affine correspondences are stacked into 3 equations in 3 unknowns. The elements M_{ij} $(i=1,\ldots,3; j=1,\ldots,3)$ of the coefficient matrix $\mathbf{M}(q_y)$ are formed by the polynomial coefficients and one unknown variable q_y, which can be described as:

$$\mathbf{M}(q_y) = \begin{bmatrix} [2] & [2] & [2] \\ [2] & [2] & [2] \\ [2] & [2] & [2] \end{bmatrix}, \tag{28}$$

where $[N]$ denotes a polynomial of degree N in variable q_y.

B.2. Degenerate Case

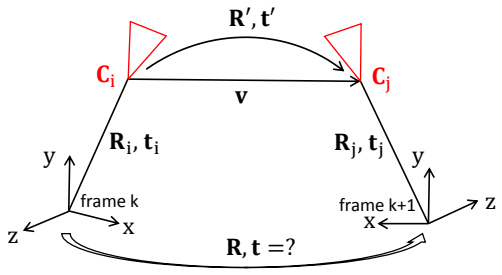


Figure 6. Relative pose estimation for a multi-camera system.

Proposition 1. *Consider a multi-camera system which is under planar motion. Assume the following three conditions are satisfied. (1) The rotation axis is y-axis, and the translation is on xz-plane. (2) There is one affine correspondence across camera C_i in frame k and camera C_j in frame $k+1$ (C_i and C_j can be the same or not). (3) The optical centers of camera C_i and C_j have the same y-coordinate. Then this case is degenerate. Specifically, the rotation can be correctly recovered, while the translation can not.*

Proof. Figure 6 illustrates the case described in the proposition. Our proof is based on the following observation: whether a case is degenerate is independent of the pose solvers. Based on this point, we construct a new minimal solver which is different from the proposed solver in the main text.

(i) Since the multi-camera system is rotated by y-axis, the camera C_i in frame k and camera C_j in frame $k+1$ are under motion with known rotation axis. Thus we can use the `1AC-method` for perspective cameras [14] to estimate the relative pose between C_i and C_j. This is a minimal solver since one AC provides 3 independent constraints and there are three unknowns (1 for rotation, 2 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. 6, we have

$$\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0} & 1\end{bmatrix} = \begin{bmatrix}\mathbf{R}_j & \mathbf{t}_j\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}' & \lambda\mathbf{t}'\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_i & \mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{R}_j\mathbf{R}'\mathbf{R}_i^T & \lambda\mathbf{R}_j\mathbf{t}' + \mathbf{t}_j - \mathbf{R}_j\mathbf{R}'\mathbf{R}_i^T\mathbf{t}_i\\ \mathbf{0} & 1\end{bmatrix}. \tag{29}$$

From Eq. (29), we have

$$\mathbf{R} = \mathbf{R}_j\mathbf{R}'\mathbf{R}_i^T, \tag{30}$$

$$\mathbf{t} = \lambda\mathbf{R}_j\mathbf{t}' + \mathbf{t}_j - \mathbf{R}_j\mathbf{R}'\mathbf{R}_i^T\mathbf{t}_i. \tag{31}$$

From Eq. (30), the rotation $\mathbf{R}$ between frame k and frame $k+1$ for the multi-camera system can be recovered. From Eq. (31), we have

$$\lambda(\mathbf{R}_j\mathbf{t}') - \mathbf{t} + (\mathbf{t}_j - \mathbf{R}\mathbf{t}_i) = \mathbf{0}. \tag{32}$$

In Eq. (32), note that $\mathbf{t} = [t_x, 0, t_z]^T$ due to planar motion. 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, it can be verified that λ becomes a free parameter. In other words, the scale cannot be determined and 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 frame $k+1$ is $\mathbf{R}_j\mathbf{t}'$. The second entry of $\mathbf{R}_j\mathbf{t}'$ is zero means that the endpoints of $\mathbf{v}$ have the same y-coordinate in frame $k+1$, which is the condition (3) in the proposition.

□

C. Relative Pose Estimation with Known Vertical Direction

Refer to Eq. (23) in the paper, four constraints obtained from two affine correspondences are stacked into 4 equations in 4 unknowns. The elements $\tilde{M}_{ij}$ $(i = 1, \ldots, 4; j = 1, \ldots, 4)$ of the coefficient matrix $\tilde{\mathbf{M}}(q_y)$ are formed by the polynomial coefficients and one unknown variable q_y, which can be described as:

$$\tilde{\mathbf{M}}(q_y) = \begin{bmatrix}[2] & [2] & [2] & [2]\\ [2] & [2] & [2] & [2]\\ [2] & [2] & [2] & [2]\\ [2] & [2] & [2] & [2]\end{bmatrix}, \tag{33}$$

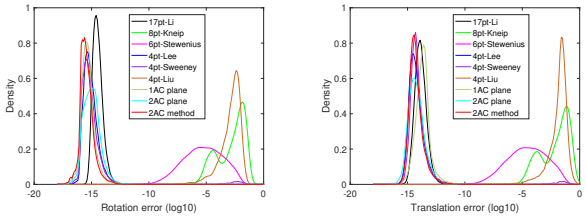
where $[N]$ denotes a polynomial of degree N in variable q_y.

D. Experiments

D.1. Efficiency comparison

The runtimes of our solvers and the comparative 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. We implemented solver `4pt-Lee`. For methods `4pt-Sweeney` and `4pt-Liu`, we used their publicly available implementations from GitHub. The average, over 10 000 runs, processing times of the solvers are shown in Table 3. The runtimes of the methods `1AC plane` , `2AC plane` and `4pt-Liu` are the lowest, because these methods solve the 4-degree polynomial equation. The `2AC method` which solves the 6-degree polynomial equation also requires low computation time.

D.2. Numerical stability



(a) (b)

Figure 7. 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. (a) reports the rotation error. (b) reports the translation error. The proposed `1AC plane` method, `2AC plane` method and `2AC method` are compared against `17pt-Li` [29], `8pt-Kneip` [24], `6pt-Stewénius` [19], `4pt-Lee` [28], `4pt-Sweeney` [44] and `4pt-Liu` [31].

Figure 7 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 method`, `17pt-Li` [29], `4pt-Lee` [28] and `4pt-Sweeney` [44] are numerically stable. It can also be seen that the `4pt-Sweeney` method has a small peak, both in the rotation and translation error curves, around 10^{-2}. 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 `4pt-Liu` method inevitably has greater than zero error in the noise-free case.

Table 3. Run-time comparison of motion estimation algorithms (unit: μs).

Methods	17pt-Li [29]	8pt-Kneip [24]	6pt-St. [19]	4pt-Lee [28]	4pt-Sw. [44]	4pt-Liu [31]	**1AC plane**	**2AC plane**	**2AC method**
Timings	43.3	102.0	3275.4	26.5	22.2	3.7	**3.6**	**3.6**	17.8

D.3. Motion with known vertical direction

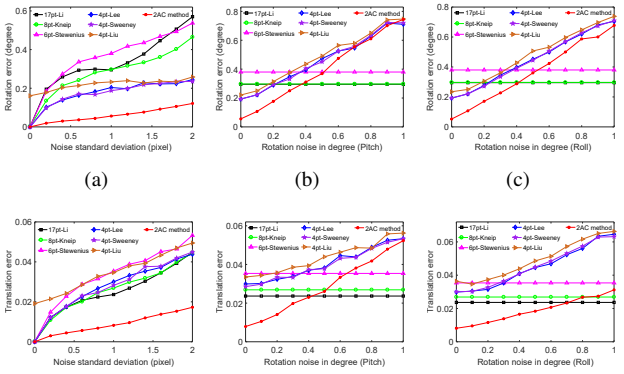


Figure 8. Rotation and translation error under forward motion. The upper row: rotation error, the bottom row: translation error. (a)(d): vary image noise. (b)(e) and (c)(f): vary IMU angle noise and fix the standard deviation of image noise as 1.0 pixel.

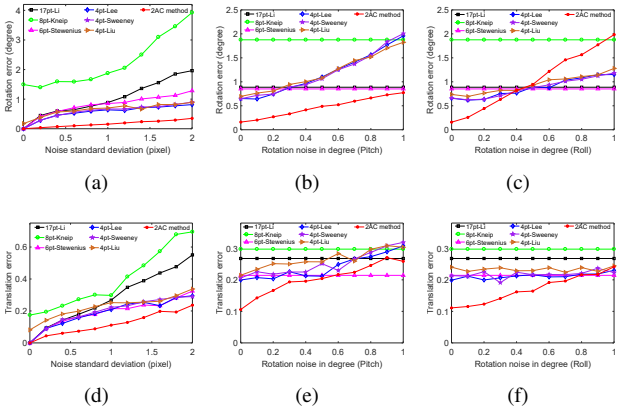


Figure 9. Rotation and translation error under sideways motion. The upper row: rotation error, the bottom row: translation error. (a)(d): vary image noise. (b)(e) and (c)(f): vary IMU angle noise and fix the standard deviation of image noise as 1.0 pixel.

In this section we show the performance of the proposed `2AC method` under forward and sideways motion. Figure 8 shows the performance of the proposed `2AC method` under forward motion. Figure 9 shows the performance of the proposed `2AC method` under sideways motion. The results demonstrate that the `2AC method` performs better than all compared methods against image noise and provides comparable accuracy for increasing IMU noise.

D.4. Cumulative errors distributions

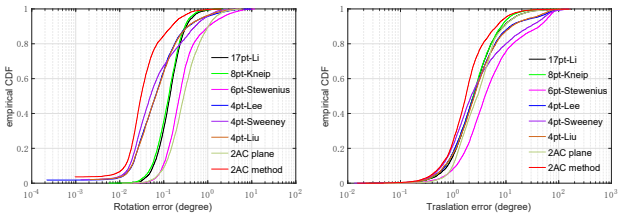


Figure 10. Empirical cumulative error distributions for KITTI sequence 00. (a) reports the rotation error. (b) reports the translation error. The proposed `1AC plane` method, `2AC plane` method and `2AC method` are compared against `17pt-Li` [29], `8pt-Kneip` [24], `6pt-Stewénius` [19], `4pt-Lee` [28], `4pt-Sweeney` [44] and `4pt-Liu` [31].

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 1 in the paper. Figure 10 shows the proposed `2AC method` offers the best overall performance in comparison to state-of-the-art methods.