Minimal Cases for Computing the Generalized Relative Pose using Affine Correspondences

Banglei Guan¹, Ji Zhao*, Daniel Barath² and Friedrich Fraundorfer³,⁴

¹College of Aerospace Science and Engineering, National University of Defense Technology, China
²Computer Vision and Geometry Group, Department of Computer Science, ETH Zürich
³Institute for Computer Graphics and Vision, Graz University of Technology, Austria
⁴Remote Sensing Technology Institute, German Aerospace Center, Germany

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 assumed. 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 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 real-world scenes from the KITTI odometry 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 [22, 10, 45, 46, 51], 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 [30, 50, 1, 16, 6, 12, 31]. 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.

Since a multi-camera system contains multiple individ-

*Corresponding author.
tions for relative pose estimation of multi-camera systems ranges back to the work of Stewénius et al. with the 6-point method [24]. Then other classical works have been subsequently proposed, such as the 17-point linear method [32] and techniques based on iterative optimization [27]. The minimal number of necessary points can be further reduced by taking additional motion constraints into account [29] or exploiting the measurements from other sensors, like an inertial measurement unit (IMU) [30, 48, 49, 34, 36]. Typically, the assumption of planar motion or considering known vertical direction are common for self-driving cars and ground robots [9, 20, 18, 44, 31], 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 [35] or SURF [7] detectors. Due to containing more information about the underlying surface geometry than PCs, ACs enable to estimate the pose from fewer correspondences. In this paper, we focus on the relative pose estimation of a multi-camera system from ACs, instead of PCs. 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 easily generalized to 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 [29] 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 [30, 48].

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 as Plücker lines and derived the generalized camera model which has become a standard representation for the multi-camera systems [41]. 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 [24]. Li et al. provided several linear solvers to compute the relative pose, among which the most commonly used one requires 17 PCs [32]. Kneip et al. proposed an iterative approach for the relative pose estimation based on eigenvalue minimization [27]. Ventura et al. used first-order approximation of the rotation to simplify the problem and estimated the relative pose from 6 PCs [50].

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, which constrains the relative motion by the Ackermann motion model [29]. 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 minimum of 4 PCs [30, 48, 34].

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 [8]. Raposo et al. computed homography and essential matrix using two ACs [43]. Barath et al. derived the constraints between the local affine transformation and the essential matrix and recovered the essential matrix from two ACs [4]. Hajder et al. [20] and Guan et al. [18, 19] 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. For multi-camera systems, Alyousefi et al. recently proposed a linear solver to estimate the relative pose using 6 ACs [2]. Guan et al. estimated the relative pose from 2 ACs by utilizing a first-order rotation approximation [17]. 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 solvers, including the DOF of the motion, feature types and number of points required.

<table>
<thead>
<tr>
<th>Solver</th>
<th>6DOF</th>
<th>4DOF</th>
<th>3DOF</th>
<th>1AC plane</th>
<th>2AC plane</th>
<th>2AC vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>PCs</td>
<td>ACs</td>
<td>PCs</td>
<td>ACs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point #</td>
<td>17</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

3. Geometric Constraints from ACs

A multi-camera system is made up of individual cameras denoted by $C_i$, as shown in Fig. 1. Take an AC seen by same camera for an example. The geometric constraints can be easily generalized to the case that the AC is seen by different cameras. The extrinsic parameters of camera $C_i$ expressed in a multi-camera reference frame are represented as $(R_i, t_i)$. For general motion, there is a 3DOF rotation and a 3DOF translation between two reference frames at time $k$ and $k+1$. Rotation $R$ using Cayley parameterization
and translation \( \mathbf{t} \) can be written as:

\[
\mathbf{R} = \frac{1}{1 + q_z^2 - q_y^2 - q_x^2} \left[ \begin{array}{ccc} 1 + q_x^2 - q_y^2 - q_z^2 & 2q_xq_y - 2q_z & 2q_xq_z + 2q_yq_z \\ 2q_xq_y + 2q_z & 1 - q_x^2 - q_y^2 - q_z^2 & 2q_yq_z - 2q_xq_z \\ 2q_xq_z - 2q_y & 2q_yq_z + 2q_xq_z & 1 - q_x^2 - q_y^2 + q_z^2 \end{array} \right].
\]

\[ \mathbf{t} = [t_x, t_y, t_z]^T, \quad (1) \]

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) [41]. Let us denote an AC 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 \( 2 \times 2 \) local affine transformation. Indices \( i \) and \( j \) are the camera and point index, respectively. The related local affine transformation \( \mathbf{A} \) is a \( 2 \times 2 \) linear transformation which relates the infinitesimal patches around \( \mathbf{x}_{ij} \) and \( \mathbf{x}_{ij}' \) [3]. The normalized homogeneous image coordinates \((\mathbf{p}_{ij}, \mathbf{p}_{ij}')\) expressed in the multi-camera reference frame are given as

\[
\mathbf{p}_{ij} = \mathbf{R}_i \mathbf{x}_{ij}, \quad \mathbf{p}_{ij}' = \mathbf{R}_i \mathbf{x}_{ij}'.
\]

(3)

The unit direction of rays \((\mathbf{u}_{ij}, \mathbf{u}_{ij}')\) expressed in the multi-camera reference frame are given as: \( \mathbf{u}_{ij} = \mathbf{p}_{ij}/\|\mathbf{p}_{ij}\|, \quad \mathbf{u}_{ij}' = \mathbf{p}_{ij}'/\|\mathbf{p}_{ij}'\| \). The 6-dimensional vector Plücker lines corresponding to the rays are denoted as \( \mathbf{l}_{ij} = [\mathbf{u}_{ij}^T, (\mathbf{t}_i \times \mathbf{u}_{ij})^T]^T, \quad \mathbf{l}_{ij}' = [\mathbf{u}_{ij}'^T, (\mathbf{t}_i \times \mathbf{u}_{ij}')^T]^T \). The generalised epipolar constraint is written as [41]

\[
\mathbf{l}_{ij}^T \left[ \begin{array}{c} [\mathbf{t}_i \mathbf{R}_i] \\ \mathbf{R}_i \end{array} \right] \mathbf{l}_{ij} = 0, \quad (4)
\]

where \( \mathbf{l}_{ij}^T \) 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 \((\mathbf{R}_{C_1}, \mathbf{t}_{C_1})\), which is represented as:

\[
\begin{bmatrix} \mathbf{R}_{C_i} & \mathbf{t}_{C_i} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i' \mathbf{R} \mathbf{R}_i & \mathbf{R}_i' \mathbf{R} \mathbf{t} + \mathbf{R}_i \mathbf{t}_i - \mathbf{R}_i \mathbf{t}_i \end{bmatrix}. \quad (5)
\]

Essential matrix \( \mathbf{E} \) of two frames of camera \( C_i \) is given as:

\[
\mathbf{E} = [\mathbf{t}_{C_1} \times \mathbf{R}_{C_1} = \mathbf{R}_i \mathbf{t}_{C_1}] \mathbf{x} \mathbf{R}_i, \quad (6)
\]

where \( [\mathbf{R}_i \mathbf{t}_{C_1}] \times \mathbf{R}_{C_1} = \mathbf{R}_i T [\mathbf{R}_i \mathbf{t}_{C_1}] \mathbf{x} \mathbf{R}_i \), and the translation \( \mathbf{t} \)

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

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

\[
(\mathbf{R}_i^T \hat{\mathbf{R}}^T [\mathbf{R}_i \mathbf{t}_{C_1}] \mathbf{x} \mathbf{R}_i \mathbf{x}_{ij})'(1:2) = -(\hat{\mathbf{A}}^T \hat{\mathbf{R}}^T [\mathbf{R}_i \mathbf{t}_{C_1}] \mathbf{x} \mathbf{R}_i \mathbf{x}_{ij})'(1:2). \quad (8)
\]

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

\[
\begin{align*}
(\mathbf{R}_i^T (\mathbf{t}_i \times \mathbf{R}) + \mathbf{R}^T \mathbf{t}) \times -\mathbf{R}^T \mathbf{t}_i \mathbf{p}_{ij}(1:2) &= (\hat{\mathbf{A}}^T \mathbf{R}^T [\mathbf{R}_i \mathbf{t}_i] \times + \mathbf{t}_i \times \mathbf{R} - [\mathbf{t}_i] \mathbf{R}) \mathbf{p}_{ij}(1:2). \quad (9)
\end{align*}
\]

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

For each AC \((\mathbf{x}_{ij}, \mathbf{x}_{ij}', \mathbf{A})\), we get three polynomials based on Eqs. (4) and (9), see supplementary material for details. 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 Fig. 2. We show that two special cases can be efficiently solved with ACs.

### 4. Relative Pose under Planar Motion

![Figure 2](image-url)

Figure 2. Special cases of multi-camera motion: (a) Planar motion in top-view. There are three unknowns: yaw angle \( \theta \), translation direction \( \phi \) and translation distance \( \rho \). (b) Motion with known vertical direction. There are four unknowns: a Y-axis rotation \( \mathbf{R}_y \) and 3D translation \( \mathbf{t} = [t_x, t_y, t_z]^T \).

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:

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

$$
t = [t_x \ 0 \ t_z]^T.
$$

where $q_y = \tan(\frac{\theta}{2})$, $t_x = \rho \sin(\phi)$, $t_z = -\rho \cos(\phi)$, $\rho$ 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 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 an equation system:

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

(11)

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

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

(12)

where $\text{quot}(a, b)$ means calculating the quotient of $a$ divided by $b$, $w_0, \ldots, w_q$ are formed by a Plücker line correspondence and 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. (12) 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 $M(q_y)$ and solving the linear system via calculating its null vector. Finally, the rotation matrix $R_y$ is recovered from Eq. (10).

However, we proved that the solver using a single AC has a degenerate case, i.e., the distances between the motion plane and optical centers of the cameras being equal, see supplementary material for details. This degenerate case might happen in the self-driving scenario, which would lead to that both the translation direction and the translation scale cannot be calculated using one AC. To overcome this issue, two ACs are used to estimate the relative pose. For example, the first and second constraints of the first AC, and the first constraint of the second AC are also stacked into three equations in three unknowns, just as Eq. (11). The solution procedure remains the same, except that the code for constructing the coefficient matrix $M(q_y)$ is replaced.

### 5. Relative Pose with Known Vertical Direction

In this section a minimal solution using two ACs 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 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. Rotation $R_{imu}$ for aligning the reference frame to the aligned reference frame is written as:

$$
R_{imu} = R_p R_r
$$

$$
= \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\theta_r) & \sin(\theta_r) \\
0 & -\sin(\theta_r) & \cos(\theta_r)
\end{bmatrix}
\begin{bmatrix}
\cos(\theta_p) & \sin(\theta_p) & 0 \\
-\sin(\theta_p) & \cos(\theta_p) & 0 \\
0 & 0 & 1
\end{bmatrix},
$$

where $\theta_r$ and $\theta_p$ are roll and pitch angles provided by the coupled IMU, respectively. Thus, there are only a Y-axis rotation $R_y$ and 3D translation $\hat{t} = [\hat{t}_x, \hat{t}_y, \hat{t}_z]^T$ to be estimated between the aligned multi-camera reference frames at time $k$ and $k+1$. In this section, we show that the geometric constraints in Section 3 can be generalized to the multi-camera motion with known vertical direction.

### 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 $R_{imu}$ and $R'_{imu}$. The relative rotation between two reference frames is

$$
R = (R'_{imu})^T R_y R_{imu}.
$$

(13)

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

$$
\begin{bmatrix}
R_{imu} & 0 \\
0 & R_{imu}
\end{bmatrix}
\begin{bmatrix}
I_{ij} \\
I'_{ij}
\end{bmatrix}
\begin{bmatrix}
R_y \\
R_y
\end{bmatrix}
= 0,
$$

(14)

where $I_{ij}$ and $I'_{ij}$ 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 follows:

$$
\begin{bmatrix}
R_{C_i} & t_{C_i} \\
0 & 1
\end{bmatrix} = \left( \begin{bmatrix} R_{imu} & 0 \\
0 & 1 \end{bmatrix} \begin{bmatrix} R_i \\
0 & 1 \\
0 & 1 \\
0 & 1 \end{bmatrix} \right)^{-1}
\begin{bmatrix}
R_{imu} \\
0 & 1
\end{bmatrix} \begin{bmatrix} R_i \\
0 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1 \end{bmatrix}.
$$

(15)

we denote that

$$
\begin{bmatrix}
\hat{R}_{imu} & \hat{t}_{imu} \\
0 & 1
\end{bmatrix} = \left( \begin{bmatrix} R_{imu} & 0 \\
0 & 1 \end{bmatrix} \begin{bmatrix} R_i \\
0 & 1 \\
0 & 1 \\
0 & 1 \end{bmatrix} \right)^{-1}
\begin{bmatrix}
R_{imu} \\
0 & 1
\end{bmatrix} \begin{bmatrix} R_i \\
0 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1 \end{bmatrix}.
$$

(16)

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

$$
\begin{bmatrix}
R_{C_i} & t_{C_i} \\
0 & 1
\end{bmatrix} = \left( \begin{bmatrix} \hat{R}_{imu} & \hat{t}_{imu} \\
0 & 1
\end{bmatrix} \begin{bmatrix} \hat{R}_{imu} & \hat{t}_{imu} \\
0 & 1
\end{bmatrix} \begin{bmatrix} R_{imu} & 0 \\
0 & 1 \end{bmatrix} \begin{bmatrix} R_i \\
0 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1 \end{bmatrix} \right)^{-1}
\begin{bmatrix}
\hat{R}_{imu} \\
0 & 1
\end{bmatrix} \begin{bmatrix} \hat{R}_{imu} & \hat{t}_{imu} \\
0 & 1
\end{bmatrix} \begin{bmatrix} R_{imu} & 0 \\
0 & 1 \end{bmatrix} \begin{bmatrix} R_i \\
0 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1 \end{bmatrix}.
$$

(17)

Essential matrix $E$ between the two frames is given as

$$
E = [t_{C_i}] \times R_{C_i} = (\hat{R}_{imu}^T) R_{imu} t_{C_i} \times R_{imu} \hat{R}_{imu}.
$$

(18)

where $[\hat{R}_{imu} t_{C_i}] \times = R_y [\hat{t}_{imu}] x R_y^T + [\hat{t}] x - [\hat{t}]$. By substituting Eq. (18) into Eq. (7), we obtain

$$
(\hat{R}_{imu}^T R_y [\hat{R}_{imu} t_{C_i}] \times \hat{R}_{imu} x_{ij})_{1:2} = -(A^T (\hat{R}_{imu}^T [\hat{R}_{imu} t_{C_i}] \times R_y \hat{R}_{imu} x_{ij})_{1:2}.
$$

(19)

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

$$
\hat{p}_{ij} = \hat{R}_{imu} x_{ij}, \quad \hat{p'}_{ij} = \hat{R}_{imu} x'_{ij}.
$$

(20)

Based on the above equation, Eq. (19) is rewritten as

$$
(\hat{R}_{imu}^T [\hat{t}_{imu}] x R_y^T + R_y^T [\hat{t}] x - R_y^T [\hat{t}_{imu}] x) \hat{p'}_{ij})_{1:2} =
(\hat{A}^T (\hat{R}_{imu}^T R_y [\hat{t}_{imu}] x + [\hat{t}] x R_y - [\hat{t}_{imu}] x R_y) \hat{p'}_{ij})_{1:2}.
$$

(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 four unknowns $q_y$, $\hat{t}_x$, $\hat{t}_y$ and $\hat{t}_z$. 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. (14) and (21), for example, three constraints of the first AC, and the first constraint of the second AC are stacked into 4 equations in 4 unknowns:

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

(22)

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

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

(23)

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

This univariate polynomial leads to a maximum of 6 solutions. Equation (23) can be efficiently solved by the companion matrix method [11] or Sturm bracketing method [39]. Once $q_y$ has been obtained, the rotation matrix $R_y$ is recovered from Eq. (10). For the relative pose between two multi-camera reference frames at time $k$ and $k+1$, the rotation matrix $R$ is recovered from Eq. (13) and the translation is computed by $t = (\hat{R}_{imu}^T \hat{t})$.

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 [32], 8pt-Kneip [27], 6pt-Stewénius [24] and 6AC-Ventura [2].

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 [32], 8pt-Kneip [27], 6pt-Stewénius [24], 4pt-Lee [30], 4pt-Sweeney [48], 4pt-Liu [34] and 6AC-Ventura [2].

A single run of the proposed solvers 1AC plane, 2AC plane and 2AC vertical take 3.6, 3.6 and 17.8 µs in C++, respectively. Due to space limitations, the
efficiency comparison and stability study are provided in the supplementary material. 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° following the definition in OpenGV [26]. We also show the feasibility of our methods on the KITTI dataset [15]. 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 altitude. 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 × 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. 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 plane is reprojected onto two cameras to get the image point pair. The corresponding affine transformation is obtained by the following procedure. First, four points are chosen as the vertices of a square in view 1, where the center of the square is the point coordinates of ACs. The side length of the square is set as 20 or 40 pixels. A larger side length causes smaller noise of affine transformation. Second, the four corresponding points in view 2 are calculated by the ground truth homography. Third, the 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. This procedure enables an indirect but geometrically interpretable way of adding noise to the affine transformation [5].

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 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: \( \varepsilon_R = \arccos(\text{trace}(R_{gt}R^T) - 1)/2 \), where \( R_{gt} \) and \( R \) are the ground truth and estimated rotation matrices. Following the definition in [42, 30], the translation error is defined as: \( \varepsilon_t = \frac{\| (t_{gt} - t) \|}{\| t_{gt} \| + \| t \|} \), where \( t_{gt} \) and \( t \) are the ground truth and estimated translations.

#### 6.1.1 Planar Motion Estimation

In this scenario, the planar motion of the multi-camera system is described by \((\theta, \phi)\), see Fig. 2(a). The magnitudes of both angles ranges from \(-10^\circ\) to \(10^\circ\). We use Gaussian image noise with a standard deviation ranging from 0 to 1 pixel. Fig. 3(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 translation direction estimation, but has worse performance in translation estimation. As shown in Fig. 3(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.

![Figure 3. Rotation and translation error under planar motion. (a–c): varying image noise under perfect planar motion. (d–f): varying planar motion noise and fixed 1.0 pixel std. image noise.](image-url)
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 [9] 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 3(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. 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. 3(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.

Both the 1AC plane and 2AC plane methods have a significant computational advantage over comparative methods, because the efficient solver for 4-degree polynomial equation takes only about 3.6 μs. Moreover, since only a single AC is required, 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. See supplementary material for details.

### 6.1.2 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 frame is aligned with the vertical direction. Due to space limitations, we only show the results for random motion. Other results are in the supplementary material. Figs. 4(a) and (d) show the performance of 2AC vertical against image noise with perfect IMU data. The proposed method is robust to image noise and performs better than the other methods.

Figs. 4(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 and 6AC-Ventura 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°. The results under forward and sideways motion also demonstrate that the 2AC vertical method performs basically better than all comparative methods against image noise and provides comparable accuracy for increasing IMU noise.

### 6.2. Experiments on Real Data

We test the performance of our methods on the KITTI dataset [15] 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 [38] detector. The extraction of ACs can also be sped up by MSER [37], 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 [40]. All the solvers have been integrated into a RANSAC scheme.
Table 2. Rotation and translation error on KITTI sequences (unit: degree).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00 (4541 images)</td>
<td>0.139 2.412</td>
<td>0.130 2.400</td>
<td>0.229 4.007</td>
<td>0.065 2.469</td>
<td>0.050 2.190</td>
<td>0.066 2.519</td>
<td>0.142 2.499</td>
<td>0.280 2.243</td>
<td>0.031 1.738</td>
</tr>
<tr>
<td>01 (1101 images)</td>
<td>0.158 5.231</td>
<td>0.171 4.102</td>
<td>0.762 41.19</td>
<td>0.137 4.782</td>
<td>0.125 11.91</td>
<td>0.105 3.781</td>
<td>0.146 3.654</td>
<td>0.168 2.486</td>
<td>0.025 1.428</td>
</tr>
<tr>
<td>02 (4661 images)</td>
<td>0.123 1.740</td>
<td>0.126 1.739</td>
<td>0.186 2.508</td>
<td>0.057 1.825</td>
<td>0.044 1.579</td>
<td>0.057 1.821</td>
<td>0.121 1.702</td>
<td>0.213 1.975</td>
<td>0.030 1.558</td>
</tr>
<tr>
<td>03 (801 images)</td>
<td>0.115 2.744</td>
<td>0.108 2.805</td>
<td>0.265 6.191</td>
<td>0.064 3.116</td>
<td>0.069 3.712</td>
<td>0.062 3.258</td>
<td>0.113 2.731</td>
<td>0.238 1.849</td>
<td>0.037 1.888</td>
</tr>
<tr>
<td>04 (271 images)</td>
<td>0.099 1.560</td>
<td>0.116 1.746</td>
<td>0.202 3.619</td>
<td>0.050 1.564</td>
<td>0.051 1.708</td>
<td>0.045 1.635</td>
<td>0.100 1.725</td>
<td>0.116 1.768</td>
<td>0.020 1.228</td>
</tr>
<tr>
<td>05 (2761 images)</td>
<td>0.119 2.289</td>
<td>0.112 2.281</td>
<td>0.199 4.155</td>
<td>0.054 2.337</td>
<td>0.052 2.544</td>
<td>0.056 2.406</td>
<td>0.116 2.273</td>
<td>0.185 2.354</td>
<td>0.022 1.532</td>
</tr>
<tr>
<td>06 (1101 images)</td>
<td>0.116 2.071</td>
<td>0.118 1.862</td>
<td>0.168 2.739</td>
<td>0.053 1.757</td>
<td>0.092 2.721</td>
<td>0.096 2.170</td>
<td>0.115 1.956</td>
<td>0.137 2.247</td>
<td>0.023 1.303</td>
</tr>
<tr>
<td>07 (1101 images)</td>
<td>0.119 3.002</td>
<td>0.112 3.029</td>
<td>0.245 6.397</td>
<td>0.058 2.810</td>
<td>0.065 4.554</td>
<td>0.054 3.048</td>
<td>0.137 2.892</td>
<td>0.173 2.902</td>
<td>0.023 1.820</td>
</tr>
<tr>
<td>08 (4071 images)</td>
<td>0.116 2.386</td>
<td>0.111 2.349</td>
<td>0.196 3.899</td>
<td>0.051 2.433</td>
<td>0.046 2.422</td>
<td>0.053 2.457</td>
<td>0.108 2.344</td>
<td>0.203 2.569</td>
<td>0.024 1.911</td>
</tr>
<tr>
<td>09 (1591 images)</td>
<td>0.133 1.977</td>
<td>0.125 1.806</td>
<td>0.179 2.592</td>
<td>0.056 1.838</td>
<td>0.046 1.656</td>
<td>0.058 1.793</td>
<td>0.124 1.876</td>
<td>0.189 1.997</td>
<td>0.027 1.440</td>
</tr>
<tr>
<td>10 (1201 images)</td>
<td>0.127 1.889</td>
<td>0.115 1.893</td>
<td>0.201 2.781</td>
<td>0.052 1.932</td>
<td>0.040 1.658</td>
<td>0.058 1.888</td>
<td>0.203 2.057</td>
<td>0.223 2.296</td>
<td>0.025 1.586</td>
</tr>
</tbody>
</table>

Table 3. Runtime of RANSAC averaged over KITTI sequences combined with different solvers (unit: s).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean time</td>
<td>52.82</td>
<td>10.36</td>
<td>79.76</td>
<td>0.85</td>
<td>0.63</td>
<td>0.45</td>
<td>6.83</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>2.62</td>
<td>1.59</td>
<td>4.52</td>
<td>0.093</td>
<td>0.057</td>
<td>0.058</td>
<td>0.61</td>
<td>0.0071</td>
<td>0.0086</td>
</tr>
</tbody>
</table>

The proposed methods 2AC plane and 2AC vertical are compared against 17pt-Li [32], 8pt-Kneip [27], 6pt-Stewénius [24], 4pt-Lee [30], 4pt-Sweeney [48], 4pt-Liu [34] and 6AC-Ventura [2]. 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 [44, 18, 31]. To ensure the fairness of the experiment, the roll and pitch angles are provided as a prior. Table 2 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 3. 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 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 the 2AC plane method basically outperforms the 6pt-Stewénius method, even though the planar motion assumption does not fit the KITTI dataset well. To visualize the comparison results, the estimated trajectory for sequence 00 is plotted in the supplementary material. 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.

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 autonomous driving scenes. We evaluate the proposed solvers 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.

Acknowledgments

This work has been partially funded by the National Natural Science Foundation of China (11902349, 11727804).
References


[28] Laurent Kneip, Chris Sweeney, and Richard Hartley. The generalized relative pose and scale problem: View-graph fu-


