

Efficient Minimal Solvers for Visual-Inertial Relative Pose Estimation in Multi-Camera Systems

Tao Li¹, Zhenbao Yu², Banglei Guan^{2*}, Jianli Han¹ and Weimin Lv¹

Abstract—Estimating the relative poses of multi-camera systems is a fundamental problem in computer vision, with critical applications in autonomous vehicles, mobile devices, and unmanned aerial vehicles (UAVs). However, existing solutions often suffer from high computational complexity or rely on an excessive number of point correspondences, limiting their real-world applicability. To address these limitations, we propose two efficient minimal solvers for estimating the relative poses of multi-camera systems using a novel parameterization. The first solver leverages the vertical direction prior provided by Inertial Measurement Units (IMUs), while the second utilizes the rotation axis direction prior from IMUs. Our methods require only four point correspondences and reduce the problem of multi-camera relative pose estimation to solving a univariate 6th-degree polynomial—a significant improvement over existing approaches, which typically involve 8th-degree polynomials. This reduction in computational complexity and correspondence requirements makes our solvers particularly effective when integrated into RANSAC frameworks, demonstrating strong potential for visual odometry applications. Through rigorous evaluations on synthetic data and the KITTI benchmark, our methods achieved superior computational efficiency and competitive accuracy compared to state-of-the-art algorithms.

I. INTRODUCTION

Relative pose estimation between two camera views is a core problem in computer vision, playing a key role in applications such as autonomous navigation, industrial inspection, and augmented reality [1]–[5]. This problem has been extensively studied, with classical solutions including the normalized 8-point algorithm [6] and the 5-point algorithm [7]. The accuracy, efficiency, and robustness of these algorithms are crucial, as they directly determine both the precision of visual localization and the quality of subsequent 3D reconstruction, ultimately affecting their real-world applicability.

Based on the number of cameras used, relative pose estimation systems can be generally categorized into monocular [8]–[10] and multi-camera configurations [11]–[14]. Monocular systems utilize a single camera, while multi-camera systems employ multiple cameras rigidly mounted on a common platform. Compared to monocular systems, multi-camera systems offer several distinct advantages. Primarily, they provide a wider field of view and richer envi-

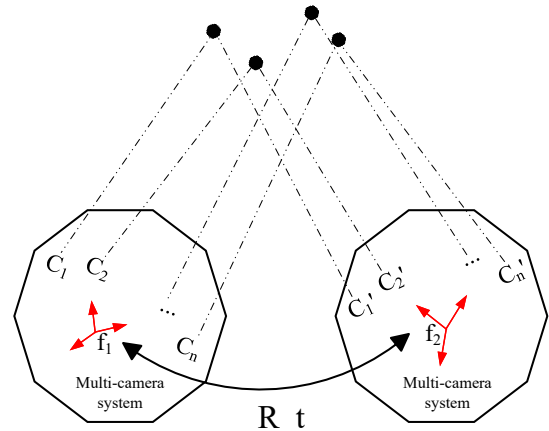


Fig. 1. The relative pose estimation for multi-camera systems.

ronmental information, enabling synchronized observations from different viewpoints. Furthermore, by leveraging pre-calibrated baseline distances between cameras, multi-camera systems can directly recover absolute scene scale, thereby resolving the inherent scale ambiguity problem in monocular approaches. Moreover, multi-camera systems exhibit remarkable flexibility, as the number and orientation of cameras can be precisely tailored to suit specific scenes and tasks, enabling better adaptation to diverse challenging scenarios.

While multi-camera systems offer numerous advantages, they also introduce significant challenges. Unlike monocular systems, where light rays converge at a single center of projection, multi-camera configurations exhibit more complex geometric constraints due to their multiple projection centers [15], see Fig. 1. This inherent complexity makes multi-camera relative pose estimation considerably more intricate and challenging. As a result, algorithms for multi-camera pose estimation typically require more feature correspondences and exhibit greater computational complexity. For instance, the linear method for multi-camera systems necessitates at least 17 point correspondences [16], a substantial increase compared to the 8 point correspondences required for monocular systems. When integrated with the RANSAC framework [17]–[19], this increased point correspondence requirement would further amplify the iteration count, thereby degrading computational efficiency and making it challenging to meet real-time requirements. Therefore, it is desirable to find more effective multi-camera solvers that can guarantee both high solution accuracy and computational efficiency.

Corresponding author: Banglei Guan.

¹Tao Li, Jianli Han, and Weimin Lv are with the College of Aerospace Science and Engineering, Naval Aviation University, Yantai 264000, China litao0931@alumni.nudt.edu.cn, jianlihan1585@163.com, 2016150315@jou.edu.cn

²Zhenbao Yu, and Banglei Guan is with the College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410000, China zhenbaoyu@whu.edu.cn, guanbanglei12@nudt.edu.cn

In this paper, we introduce two novel minimal solutions for this problem. The main contributions of this work are as follows:

- Using a novel depth-based parameterization for multi-camera system translation, we reformulate the new generalized epipolar constraint.
- We introduce two novel minimal solvers using four point correspondences for multi-camera relative pose estimation. The first exploits the IMU vertical direction prior, while the second uses the IMU rotation axis direction prior. Both approaches reduce the problem to solving a 6th-degree univariate polynomial, offering significant computational advantages over existing 8th-degree solvers [20]–[22].
- Extensive experiments demonstrate that our proposed solvers achieved competitive computational efficiency and accuracy compared to state-of-the-art methods.

II. RELATED WORK

For generalized camera pose estimation, Stewénus et al. introduced the 6-point minimal solver based on Gröbner-basis theory, which generates up to 64 potential solutions, resulting in significant computational demands [23]. Guan et al. proposed a different set of 6-point minimal solvers, including both a general solver and specialized versions tailored for two-camera rig configurations. A key advancement lies in the incorporation of a ray bundle constraint, which significantly reduces the potential solution space [24]. In addition to the minimum solvers based on 6-point correspondences, Li et al. advanced the linear 17-point method by incorporating camera configuration effects on algorithm performance, resulting in enhanced capability to handle degenerate scenarios such as locally central and axial configurations [16]. Kim et al. systematically analyzed the degradation of the 17-point method through the decomposition of the measurement matrix into ray direction and projection center components [25]. Chen et al. further demonstrated how visual overlap significantly impacts both the solvability and precision of the 17-point method, providing key insights for practical implementation [26]. Complementary to minimal solvers depending on point correspondences, some approaches leveraging affine correspondences have also shown promising experimental performance [27], [28].

Recent advances in generalized camera pose estimation have demonstrated the effectiveness of incorporating motion constraints [11] or inertial sensors [20]. Lee et al. introduced a minimal 4-point algorithm for estimating the relative pose of multi-camera systems with known vertical direction [20]. Utilizing the hidden variable resultant method, their approach reduces the problem to solving an 8th-degree univariate polynomial, which can yield up to 8 real solutions. Beyond utilizing vertical direction information, Liu et al.’s 4-point algorithm using first-order rotation approximation achieves remarkable processing efficiency, making it highly suitable for real-time applications in dense urban driving conditions [21]. Given the known rotation axis direction, Sweeney et al. developed an additional 4-point algorithm for relative pose

estimation between generalized cameras [22]. More recently, Martyushev et al. introduced a minimal 5-point algorithm that leverages the known relative rotation angle [29]. These IMU-constrained approaches are particularly relevant to our work as they demonstrate how prior orientation information can significantly simplify the multi-camera problem while maintaining real-time performance.

III. GENERALIZED CAMERA CONSTRAINTS UNDER GENERAL MOTION

A. Parameterization

Consider a motion platform equipped with multiple cameras, also known as a multi-camera system, that moves from time t_1 to t_2 . A 3D point \mathbf{P} is captured by cameras c_i and c_j at different time instances. The motion platform’s coordinate systems at these two different instances are denoted as \mathbf{f}_1 and \mathbf{f}_2 , respectively. Following the recent work of [27], the world coordinate system \mathbf{W} is defined with its origin at point \mathbf{P} and its orientation aligned with the platform’s initial coordinate frame \mathbf{f}_1 . Throughout this paper, \mathbf{T}_{ba} represents the transformation matrix from frame \mathbf{a} to frame \mathbf{b} . Specifically, the transformation matrix from frame c_i to frame \mathbf{f}_1 is $\mathbf{T}_{\mathbf{f}_1 c_i} = [\mathbf{Q}_i \ s_i]$, and the transformation matrix from frame \mathbf{f}_1 to frame \mathbf{W} is $\mathbf{T}_{\mathbf{W} \mathbf{f}_1} = [\mathbf{R}_1 \ t_1]$. Similarly, the transformation matrix from frame c_j to frame \mathbf{f}_2 is $\mathbf{T}_{\mathbf{f}_2 c_j} = [\mathbf{Q}_j \ s_j]$, and the transformation matrix from frame \mathbf{f}_2 to frame \mathbf{W} is $\mathbf{T}_{\mathbf{W} \mathbf{f}_2} = [\mathbf{R}_2 \ t_2]$. Since the orientation of the world coordinate system \mathbf{W} aligns with the coordinate system \mathbf{f}_1 , the rotation matrix $\mathbf{R}_1 = \mathbf{I}$. The transformation matrix from frame \mathbf{f}_1 to frame \mathbf{f}_2 is denoted as $\mathbf{T}_{\mathbf{f}_2 \mathbf{f}_1} = [\mathbf{R} \ t]$, where \mathbf{R} denotes the rotation matrix and t represents the translation vector. The specific meanings of each coordinate system, along with their corresponding matrix and vector representations, are illustrated in Fig. 2.

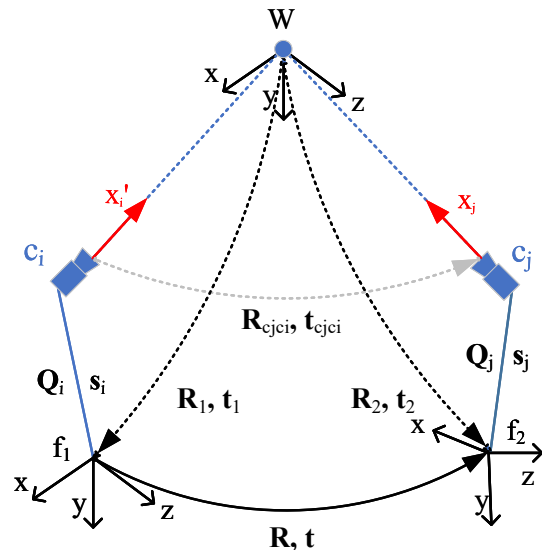


Fig. 2. The definition of coordinate systems in multi-camera systems.

Based on the coordinate system definitions above, the following transformation relationship holds:

$$\begin{aligned}\bar{\mathbf{T}}_{\mathbf{f}_2\mathbf{f}_1} &= \bar{\mathbf{T}}_{\mathbf{f}_2\mathbf{W}}\bar{\mathbf{T}}_{\mathbf{W}\mathbf{f}_1} = \begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{t}_1 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_2 & -\mathbf{R}_2\mathbf{t}_1 + \mathbf{t}_2 \\ \mathbf{0} & 1 \end{bmatrix}.\end{aligned}\quad (1)$$

Here, $\bar{\mathbf{T}}$ denotes the homogeneous augmentation of the 3×4 matrix \mathbf{T} , obtained by appending the row vector $[0, 0, 0, 1]$. From Eq. (1), we obtain:

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_2 \\ \mathbf{t} &= -\mathbf{R}_2\mathbf{t}_1 + \mathbf{t}_2.\end{aligned}\quad (2)$$

In the generalized camera model, the point correspondence $\mathbf{u}_i \leftrightarrow \mathbf{u}_j$ of the 3D point \mathbf{P} can be viewed as the intersection of two Plücker lines $\mathbf{L}_i = [\mathbf{x}_i^T \quad \mathbf{q}_i^T]^T$ and $\mathbf{L}_j = [\mathbf{x}_j^T \quad \mathbf{q}_j^T]^T$. In the Plücker line representation, \mathbf{x}_* denotes the unit direction vector, given by $\mathbf{x}_* = (\mathbf{Q}_* * \mathbf{u}_*) / \|\mathbf{Q}_* * \mathbf{u}_*\|$, and $\mathbf{q}_* = \mathbf{s}_* \times \mathbf{x}_*$ denotes the moment vector. According to Plücker line geometry, any point $\mathbf{U}(\lambda)$ on the line can be parameterized as:

$$\mathbf{U}(\lambda) = \mathbf{x} \times \mathbf{q} + \lambda \mathbf{x}, \lambda \in \mathbb{R}.\quad (3)$$

Since the 3D point \mathbf{P} lies at the intersection of two Plücker lines, its coordinates can be parameterized by these two lines simultaneously. The corresponding coordinates of point \mathbf{P} along these two lines are respectively given by $\mathbf{U}(\lambda_1)$ and $\mathbf{U}(\lambda_2)$. Specifically, $\mathbf{U}(\lambda_1)$ and $\mathbf{U}(\lambda_2)$ represent the coordinates of \mathbf{P} within the motion platform's coordinate systems \mathbf{f}_1 and \mathbf{f}_2 , respectively, where \mathbf{P} corresponds to the origin $[0, 0, 0]^T$ of the world coordinate system \mathbf{W} . Therefore, the following relationships hold:

$$\begin{aligned}\mathbf{U}(\lambda_1) &= \mathbf{x}_1 \times \mathbf{q}_1 + \lambda_1 \mathbf{x}_1 = \mathbf{R}_1[0, 0, 0]^T + \mathbf{t}_1 \\ \mathbf{U}(\lambda_2) &= \mathbf{x}_2 \times \mathbf{q}_2 + \lambda_2 \mathbf{x}_2 = \mathbf{R}_2[0, 0, 0]^T + \mathbf{t}_2.\end{aligned}\quad (4)$$

Then, the expressions of \mathbf{t}_1 and \mathbf{t}_2 can be derived using the depths λ_1 and λ_2 .

$$\begin{aligned}\mathbf{t}_1 &= \mathbf{x}_1 \times \mathbf{q}_1 + \lambda_1 \mathbf{x}_1 \\ \mathbf{t}_2 &= \mathbf{x}_2 \times \mathbf{q}_2 + \lambda_2 \mathbf{x}_2.\end{aligned}\quad (5)$$

This shows that the translation vector \mathbf{t} can be expressed using two depth parameters λ_1 and λ_2 , rather than adopting the conventional $[t_x, t_y, t_z]^T$ representation. Utilizing this novel parameterization, we can derive new formulations for the generalized epipolar constraint (GEC).

B. Generalized Epipolar Constraint

Since \mathbf{L}_i and \mathbf{L}_j represent Plücker lines in two distinct coordinate frames \mathbf{f}_1 and \mathbf{f}_2 respectively, they need to be transformed into a common coordinate system. When transforming the Plücker line \mathbf{L}_i from frame \mathbf{f}_1 to frame \mathbf{f}_2 , the resulting coordinate are given by $\mathbf{L}'_i = [\mathbf{x}'_i{}^T \quad \mathbf{q}'_i{}^T]^T$. The relationship between \mathbf{L}'_i and \mathbf{L}_i is described by the following transformation:

$$\mathbf{L}'_i = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ [\mathbf{t}]_{\times} \mathbf{R} & \mathbf{R} \end{bmatrix} \mathbf{L}_i = \begin{bmatrix} \mathbf{R}\mathbf{x}_i \\ \mathbf{R}\mathbf{q}_i + [\mathbf{t}]_{\times} \mathbf{R}\mathbf{x}_i \end{bmatrix}.\quad (6)$$

According to [15], the necessary and sufficient condition for two straight lines \mathbf{L}'_i and \mathbf{L}_j to intersect within the same coordinate system \mathbf{f}_2 is given by:

$$\mathbf{x}'_i{}^T \mathbf{q}_j + \mathbf{x}_j^T \mathbf{q}'_i = 0.\quad (7)$$

By substituting (6) into (7), the generalized epipolar constraint can be derived as follows:

$$(\mathbf{R}\mathbf{x}_i)^T \mathbf{q}_j + \mathbf{x}_j^T \mathbf{R}\mathbf{q}_i + \mathbf{x}_j^T [\mathbf{t}]_{\times} \mathbf{R}\mathbf{x}_i = 0.\quad (8)$$

Substituting (2) into (8) yields:

$$\mathbf{x}_i^T \mathbf{R}^T \mathbf{q}_j + \mathbf{x}_j^T \mathbf{R}\mathbf{q}_i + \mathbf{x}_j^T ([\mathbf{t}_2]_{\times} \mathbf{R} - \mathbf{R}[\mathbf{t}_1]_{\times}) \mathbf{x}_i = 0.\quad (9)$$

Finally, by substituting (5) into (9), the generalized epipolar constraints formulated in terms of depths λ_1 and λ_2 can be derived:

$$\begin{aligned}-\lambda_1 \mathbf{x}_j^T \mathbf{R}[\mathbf{x}_1]_{\times} \mathbf{x}_i + \lambda_2 \mathbf{x}_j^T ([\mathbf{x}_2]_{\times} \mathbf{R}) \mathbf{x}_i + \mathbf{x}_j^T \mathbf{R}\mathbf{q}_i + \\ \mathbf{x}_i^T \mathbf{R}^T \mathbf{q}_j + \mathbf{x}_j^T ([\mathbf{x}_2 \times \mathbf{q}_2]_{\times} \mathbf{R} - \mathbf{R}[\mathbf{x}_1 \times \mathbf{q}_1]_{\times}) \mathbf{x}_i = 0.\end{aligned}\quad (10)$$

IV. 4-POINT ALGORITHM FOR MULTI-CAMERA SYSTEMS

A. 4-Point algorithm with Known Vertical Direction

Through IMU alignment, the platform's y-axis can be precisely aligned with gravitational direction, ensuring orthogonality to the ground plane. Let \mathbf{R}_{imu1} and \mathbf{R}_{imu2} denote the alignment matrices derived from IMU measurements at timestamps t_1 and t_2 , respectively. The relationship between the original rotation matrix \mathbf{R} and its aligned counterpart \mathbf{R}_{v2v1} is given by:

$$\mathbf{R} = \mathbf{R}_{imu2}^T \mathbf{R}_{v2v1} \mathbf{R}_{imu1}.\quad (11)$$

Then, the problem can be reduced to solving for \mathbf{R}_{v2v1} , which depends solely on the yaw angle. Using Cayley parameterization, the aligned rotation matrix \mathbf{R}_{v2v1} takes the form:

$$\mathbf{R}_{v2v1} = \frac{1}{1+s^2} \begin{bmatrix} 1-s^2 & 0 & 2s \\ 0 & 1+s^2 & 0 \\ -2s & 0 & 1-s^2 \end{bmatrix},\quad (12)$$

where θ_y denotes the yaw angle and $s = \tan(\theta_y/2)$.

Substituting (11) into the generalized epipolar constraint (10), we can obtain:

$$\begin{aligned}-\lambda_1 \tilde{\mathbf{x}}_j^T \mathbf{R}_{v2v1} [\tilde{\mathbf{x}}_1]_{\times} \tilde{\mathbf{x}}_i + \lambda_2 \tilde{\mathbf{x}}_j^T ([\tilde{\mathbf{x}}_2]_{\times} \mathbf{R}_{v2v1}) \tilde{\mathbf{x}}_i \\ + \tilde{\mathbf{x}}_j^T \mathbf{R}_{v2v1} \tilde{\mathbf{q}}_i + \tilde{\mathbf{x}}_i^T \mathbf{R}_{v2v1}^T \tilde{\mathbf{q}}_j \\ + \tilde{\mathbf{x}}_j^T ([\tilde{\mathbf{x}}_2 \times \tilde{\mathbf{q}}_2]_{\times} \mathbf{R}_{v2v1} - \mathbf{R}_{v2v1} [\tilde{\mathbf{x}}_1 \times \tilde{\mathbf{q}}_1]_{\times}) \tilde{\mathbf{x}}_i = 0\end{aligned}\quad (13)$$

Comparing (13) with (10) reveals their structural similarity, with the key distinction being that the variables in (13) have been adjusted by alignment matrices. The detailed adjustments include: $\tilde{\mathbf{x}}_i = \mathbf{R}_{imu1} \mathbf{x}_i$, $\tilde{\mathbf{x}}_j = \mathbf{R}_{imu2} \mathbf{x}_j$, $\tilde{\mathbf{q}}_i = \mathbf{R}_{imu1} \mathbf{q}_i$, $\tilde{\mathbf{q}}_j = \mathbf{R}_{imu2} \mathbf{q}_j$.

Using three non-origin point correspondences, we derive three equations from (13). We specifically exclude the point correspondence corresponding to the world origin \mathbf{P} because its constraint equation reduces to 0, providing no valuable

information for solving the unknown parameters. The three remaining equations can be organized in matrix form as:

$$\mathbf{F}_{3 \times 3}(s) \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{bmatrix} = \mathbf{0}_{3 \times 1}. \quad (14)$$

Since the system of equations (14) has a non-trivial solution, the matrix $\mathbf{F}(s)$ must satisfy $\text{rank}(\mathbf{F}(s)) \leq 2$, which implies:

$$\det(\mathbf{F}(s)) = 0. \quad (15)$$

Based on (15), we can obtain a 6th-degree univariate polynomial in the variable s , providing up to 6 real solutions.

Once the solutions for s are obtained, they can be substituted back into (12) and subsequently into (11) to recover the complete rotation matrix \mathbf{R} .

B. 4-Point algorithm with Known Rotation Axis Direction

Instead of using the Cayley parameterization or Euler angles to represent rotations, this section employs the quaternion representation. The rotation quaternion is denoted as $[q_1 \ q_2 \ q_3 \ \omega]^T$, which can also be compactly written as $[\mathbf{v}^T \ \omega]^T$, where $\mathbf{v} = [q_1, q_2, q_3]^T$ is the vector part and ω is the scalar part. For a unit quaternion, the vector component \mathbf{v} defines the rotation axis direction, and the scalar component ω gives the cosine of half the rotation angle. The relationship between a unit quaternion and its corresponding rotation matrix is given by:

$$\mathbf{R} = \mathbf{v}\mathbf{v}^T + \omega^2\mathbf{I} + 2\omega[\mathbf{v}]_{\times} + [\mathbf{v}]_{\times}^2, \quad (16)$$

where $[\mathbf{v}]_{\times}$ denotes the skew-symmetric matrix of vector \mathbf{v} .

When the quaternion representing \mathbf{R} is not necessarily normalized, its vector and scalar components are denoted as $\tilde{\mathbf{v}}$ and $\tilde{\omega}$ respectively. Here, $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$ is the unit vector along the rotation axis, and $\tilde{\omega} = \omega/\|\mathbf{v}\|$. The rotation matrix $\tilde{\mathbf{R}}$ constructed using these non-normalized components follows the same formula structure and is proportional to the true rotation matrix \mathbf{R} :

$$\mathbf{R} \sim \tilde{\mathbf{R}} = \tilde{\mathbf{v}}\tilde{\mathbf{v}}^T + \tilde{\omega}^2\mathbf{I} + 2\tilde{\omega}[\tilde{\mathbf{v}}]_{\times} + [\tilde{\mathbf{v}}]_{\times}^2. \quad (17)$$

Note that $\tilde{\mathbf{R}}$ and \mathbf{R} differ only by a scale factor, which does not affect the solution validity since the generalized epipolar constraint is a homogeneous equation.

Substituting the quaternion-based rotation representation from (17) into the generalized epipolar constraint (10), we can establish a system of three equations:

$$\mathbf{F}_{3 \times 3}(\tilde{\omega}) \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{bmatrix} = \mathbf{0}_{3 \times 1}. \quad (18)$$

Analogous to the derivation of (15), this system leads to a similar determinant condition:

$$\det(\mathbf{F}(\tilde{\omega})) = 0. \quad (19)$$

Consistent with the previous finding for the variable s , expanding (19) yields a 6th-degree univariate polynomial in the variable $\tilde{\omega}$, which provides up to 6 real solutions.

The obtained solutions for $\tilde{\omega}$, combined with the assumed known rotation axis $\tilde{\mathbf{v}}$, define an unnormalized quaternion. By normalizing this quaternion, we recover the unit vector \mathbf{v} and scalar ω components of the unit quaternion. Subsequently, substituting these normalized components into equation (16) enables the complete recovery of the rotation matrix \mathbf{R} .

C. Translation Vector Estimation

Once the values for s or $\tilde{\omega}$ are determined from the 6th-degree polynomial, they are substituted into the corresponding matrix $\mathbf{F}(\cdot)$ defined in (14) or (18). We can obtain a homogeneous linear equation system in terms of the unknown depth parameters λ_1 and λ_2 . Using Singular Value Decomposition (SVD), the depth parameters λ_1 and λ_2 are obtained from the null space of the matrix $\mathbf{F}(\cdot)$.

Subsequently, by substituting these calculated depths λ_1 and λ_2 back into the definitions of \mathbf{t}_1 and \mathbf{t}_2 as provided in (5), we can compute the specific values for \mathbf{t}_1 and \mathbf{t}_2 . Finally, the overall translation vector \mathbf{t} can be recovered using the following formula:

$$\mathbf{t} = -\mathbf{R}\mathbf{t}_1 + \mathbf{t}_2, \quad (20)$$

where \mathbf{R} corresponds to the rotation matrix determined in the previous steps through either the Cayley parameterization or quaternion representation.

D. Degeneracy Analysis

Prior work [20] has identified that degeneracy occurs when a multi-camera system undergoes pure translation while relying solely on intra-camera correspondences. Although our approach employs the depth-based parameterization for translation representation rather than the conventional $[t_x, t_y, t_z]^T$ formulation, the multi-camera system still exhibits degeneracy under pure translational motion with intra-camera correspondences, resulting in unrecoverable scale factors [27]. Consider the case of pure translation where the relative rotation $\mathbf{R} = \mathbf{I}$, combined with intra-camera correspondences where $\mathbf{s}_i = \mathbf{s}_j$. Under these conditions, the cross-term in the generalized epipolar constraint vanishes:

$$\mathbf{x}_i^T \mathbf{R}^T \mathbf{q}_j + \mathbf{x}_j^T \mathbf{R} \mathbf{q}_i = \mathbf{x}_i^T \mathbf{s}_i \times \mathbf{x}_j + \mathbf{x}_j^T \mathbf{s}_i \times \mathbf{x}_i = \mathbf{0}. \quad (21)$$

Consequently, the generalized epipolar constraint (9) reduces to the following form:

$$\mathbf{x}_j^T ([\mathbf{t}_2 - \mathbf{t}_1]_{\times}) \mathbf{x}_i = 0. \quad (22)$$

In this degenerate case, the constraint matrix becomes homogeneous with respect to the relative translation $\mathbf{t}_2 - \mathbf{t}_1$. Consequently, introducing a free parameter κ reveals that $\kappa(\mathbf{t}_2 - \mathbf{t}_1)$ always satisfies equation (22), indicating the loss of scale information.

V. EXPERIMENT

In the experiments, the solver presented in Section IV-A is denoted as `4pt-Our`, while the solver from Section IV-B is referred to as `4pt-Our-Axis`. For comprehensive comparison, we benchmark our solvers against both state-of-the-art

4-point methods, including 4pt-Lee [20], 4pt-Liu [21], and 4pt-Sweeney [22], as well as the recently proposed 5-point method 5pt-Martyushev [29]. In the experiments, we evaluate rotation and translation errors using the following metrics [27]:

$$\begin{aligned}\varepsilon_{\mathbf{R}} &= \arccos((\text{trace}(\mathbf{R}_{gt}\mathbf{R}^T) - 1)/2) \\ \varepsilon_{\mathbf{t}} &= \arccos((\mathbf{t}_{gt}^T \mathbf{t}) / (\|\mathbf{t}_{gt}\| \cdot \|\mathbf{t}\|))\end{aligned}\quad (23)$$

Here, \mathbf{R}_{gt} and \mathbf{t}_{gt} represent the ground truth values, while \mathbf{R} and \mathbf{t} represent the estimated values. $\varepsilon_{\mathbf{R}}$ represents the angular error of the rotation, $\varepsilon_{\mathbf{t}}$ indicates the angular error of the translation.

A. Efficiency Comparison and Numerical Stability

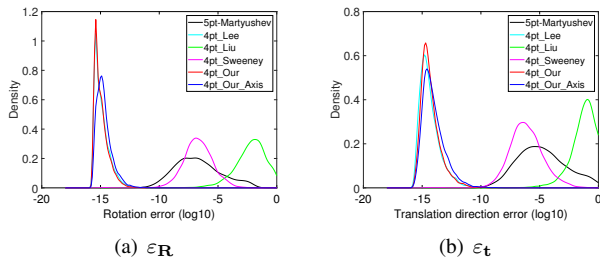


Fig. 3. Numerical stability comparison of relative pose estimation solvers.

To ensure a fair comparison, all solvers were evaluated on an AMD R9-7945HX 2.50 GHz processor using C++ implementations. Table I presents the average computation times across 10,000 independent runs. The experimental results indicate that all 4-point solvers are substantially more computationally efficient than the 5-point solver 5pt-Martyushev [29]. Among them, 4pt-Liu [21] achieves the fastest runtime due to its use of a first-order rotation approximation to simplify the computation. Among solvers that do not rely on such approximations, our proposed 4pt-Our and 4pt-Our-Axis methods run more efficiently than 4pt-Lee [20] and 4pt-Sweeney [22]. This efficiency gain stems from our derivation of more compact polynomial equations with fewer terms, which reduces the computational burden of polynomial expansion and root-solving. Overall, these results demonstrate the strong computational efficiency of our proposed solvers.

Fig. 3 presents a comparative analysis of solver numerical stability under noise-free conditions. Each solver was executed 10,000 independent runs, with the probability density functions plotted against the \log_{10} of the rotation errors and translation errors. A higher and more left-shifted peak in these distributions indicates superior numerical stability, whereas a broader and right-shifted distribution suggests poorer stability. Among solvers utilizing IMU vertical angle priors, our proposed 4pt-Our achieves exceptional numerical stability, with errors concentrated around 10^{-15} . The 4pt-Lee [20] solver shows nearly comparable performance, whereas 4pt-Liu [21] exhibits degraded stability due to its reliance on first-order rotation approximation. Similarly, for solvers using IMU rotation axis priors, our

4pt-Our-Axis method shows better performance, with errors concentrated around 10^{-14} .

B. Experiments on Synthetic Data

For synthetic evaluation, we adopted a multi-camera system consisting of two simulated cameras with a fixed baseline of 1 meter. The system’s position and orientation varied randomly over time, with translational displacements limited to a maximum of 3 meters and rotational variations confined to a range of -10° to 10° . The camera intrinsics were configured with a focal length of 400 pixels and principal point coordinates centered at (320, 240). Spatially distributed 3D points were randomly generated and designed to be simultaneously observable by both cameras. Point correspondences were exclusively intra-camera correspondences. The evaluation comprised 1,000 independent trials, with pose estimation accuracy quantified using median rotation and translation errors.

1) *Accuracy with Image Noise:* We evaluate the accuracy of the solvers in the presence of image noise by introducing Gaussian noise with standard deviations ranging from 0 to 1.0 pixels. The multi-camera system motion patterns included forward, random, and sideways motions. As shown in Fig. 4, the 4-point solvers generally demonstrate greater robustness to image noise compared to the 5-point solver 5pt-Martyushev [29], except in the case of sideways motion. For the 4-point solvers leveraging IMU rotation axis priors, both our proposed 4pt-Our-Axis and 4pt-Sweeney [22] achieve comparable accuracy and outperform all other evaluated methods in most experimental scenarios. For the 4-point solvers utilizing IMU vertical angle priors, our 4pt-Our and 4pt-Lee [20] exhibited similar accuracy levels, while both showed significant accuracy improvements over the 4pt-Liu [21] method. This performance disparity primarily stems from 4pt-Liu’s reliance on first-order rotation approximation, which is only effective for small-angle rotations and becomes insufficient when handling larger angular motions.

2) *Accuracy with IMU Noise:* In practical applications, high-end IMUs can achieve angle measurement accuracy as precise as 0.02° , while even low-cost IMUs typically offer errors below 0.5° [30]. To thoroughly evaluate our algorithm’s robustness against IMU measurement errors, we conducted simulations with angular noise spanning 0° to 1° —a range that encompasses the performance of both commercial and industrial-grade IMUs. The evaluated methods differ in their use of IMU priors. 4pt-Lee [20], 4pt-Liu [21], and 4pt-Our rely on IMU vertical angle priors, while 4pt-Our-Axis and 4pt-Sweeney [22] leverage IMU rotation axis priors. To ensure consistency in our simulations, we generated noisy data based on IMU vertical angle noise, as this can be directly transformed into equivalent IMU rotation axis noise. Specifically, noise was introduced by perturbing the pitch and roll angles, while the image noise was kept constant at 0.5 pixels. Since the 5pt-Martyushev [29] method depends on the rotation angle rather than the vertical angle, the comparison of the

TABLE I
 RUNTIME COMPARISON OF RELATIVE POSE ESTIMATION SOLVERS (UNIT: μs).

Methods	5pt-Martyushev [29]	4pt-Lee [20]	4pt-Liu [21]	4pt-Sweeney [22]	4pt-Our	4pt-Our-Axis
Runtime	591.356	28.024	8.268	22.186	21.343	14.334

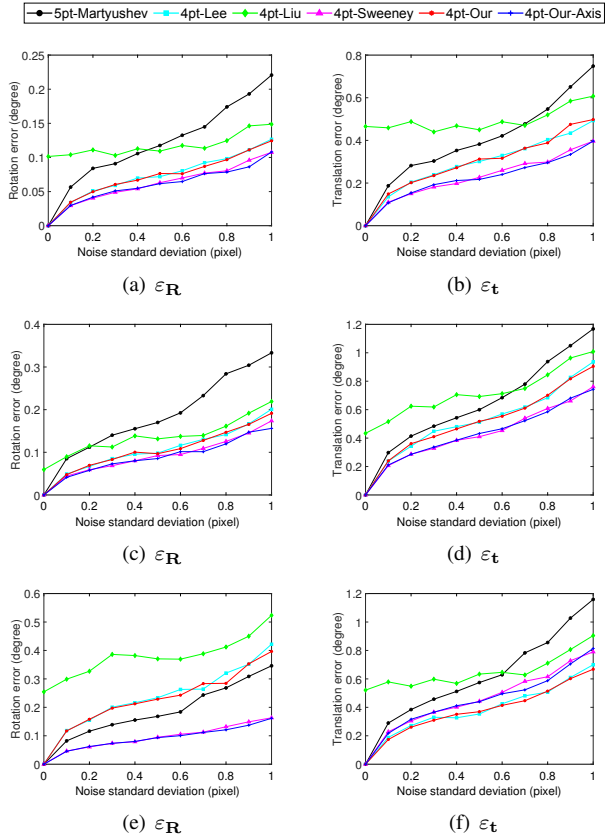


Fig. 4. Rotation and translation errors for multi-camera systems with increasing image noise. The top, middle, and bottom rows present the performance of different solvers under forward, random, and sideways motions, respectively.

impact of pitch and roll angle errors is restricted to the 4pt-Lee [20], 4pt-Liu [21], and 4pt-Sweeney [22] methods.

Fig. 5 illustrates the performance of the proposed solvers under increasing pitch angle noise across forward, random, and sideways motion patterns. As the pitch angle noise increases, the accuracy of all 4-point solvers gradually degrades. Among the evaluated methods, 4pt-Our-Axis and 4pt-Sweeney [22] exhibit comparable accuracy and consistently outperform other 4-point solvers. This enhanced performance can be attributed to their rotation representation. Since IMU rotation axis direction noise is generally non-orthogonal to IMU vertical angle noise, this representation appears to effectively mitigate the impact of disturbances caused by IMU vertical angle noise. Meanwhile, 4pt-Our and 4pt-Lee [20] exhibit similar accuracy and perform better than 4pt-Liu [21].

Fig. 6 illustrates the performance of the proposed solvers

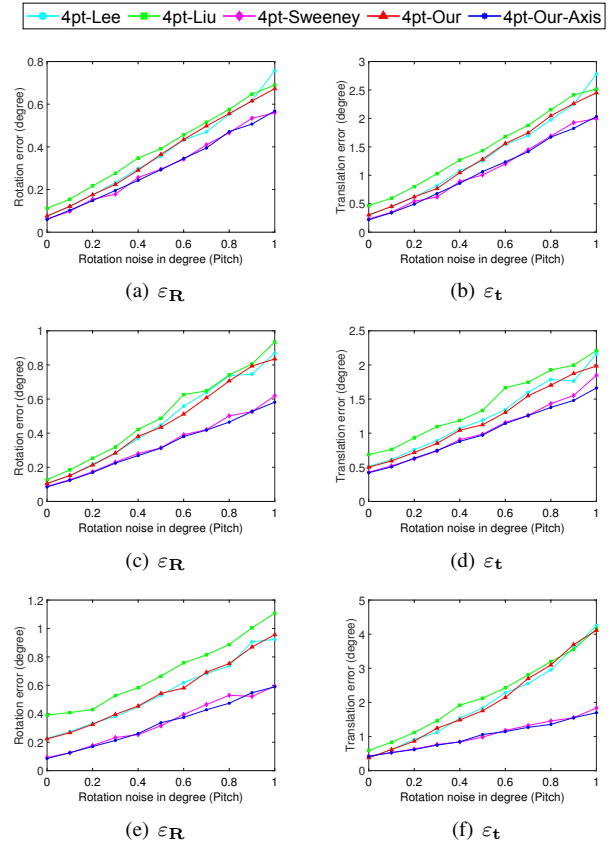


Fig. 5. Rotation and translation errors for multi-camera systems with increasing IMU pitch angle noise. The top, middle, and bottom rows present the performance of different solvers under forward, random, and sideways motions, respectively.

under increasing roll angle noise across forward, random, and sideways motion patterns. As the roll angle noise increases, 4pt-Our-Axis and 4pt-Sweeney [22] exhibit comparable performance and achieve better rotation accuracy than the other three methods. Regarding translation estimation, 4pt-Our and 4pt-Lee [20] demonstrate similar performance, and both outperform 4pt-Liu [21] in forward motion and sideways motion. In random motion, the translation errors of 4pt-Our and 4pt-Our-Axis are also smaller than those of 4pt-Liu [21]. These results reinforce the robustness of the evaluated methods under diverse angular noise conditions.

C. Experiments on Real Data

The proposed methods were evaluated on the KITTI dataset [31], a widely-used benchmark for evaluating autonomous driving vision algorithms. This dataset contains multiple sequences, with ground truth values for sequences

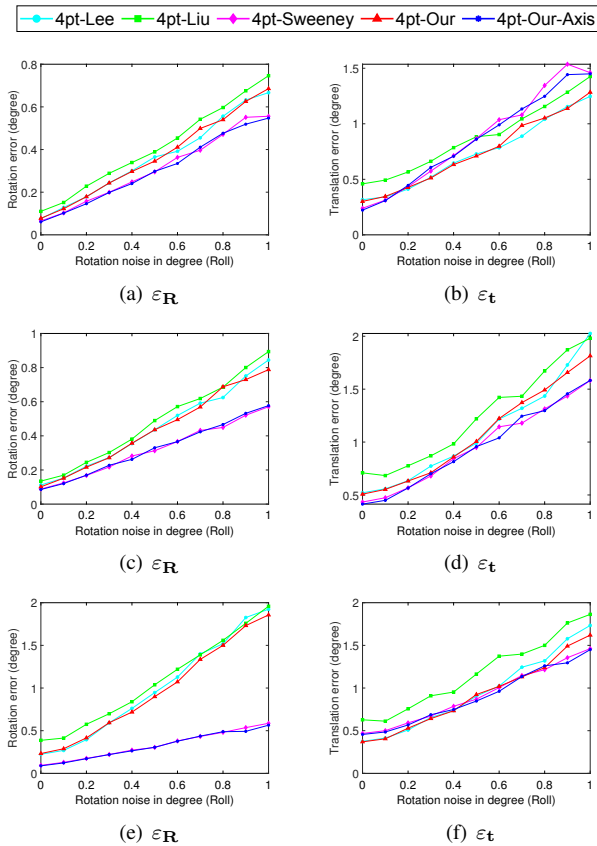


Fig. 6. Rotation and translation errors for multi-camera systems with increasing IMU roll angle noise. The top, middle, and bottom rows present the performance of different solvers under forward, random, and sideways motions, respectively.

00-10 provided by its onboard GPS/IMU units. We compared the performance across all 11 sequences, which collectively contain approximately 23,000 image pairs. To ensure robustness against outliers, all methods were embedded within an RANSAC framework. Notably, the final results correspond to the initial hypotheses that maximized the inlier count; no subsequent bundle adjustment or local optimization was performed. Detailed results are presented in Table II and Table III.

From Table II and Table III, it can be observed that the 4-point solvers consistently achieve higher rotation accuracy than the 5-point solver 5pt-Martyshev [29] across all 11 sequences. In terms of translation accuracy, the 4-point methods also outperform the 5-point solver in most sequences, with the exception of sequences 01, 04, and 06. Within the 4-point solvers, our proposed methods demonstrate a slight advantage. Notably, 4pt-Our-Axis achieves the best translation accuracy in all sequences except sequences 04 and 07. Since our proposed methods fall within the category of 4-point solvers, we provide a more intuitive comparison among these methods by visualizing the estimated camera trajectories in Fig. 7. The ground truth trajectory is shown in black, while the estimated trajectories are color-coded according to their absolute trajectory error

TABLE II
MEDIAN ROTATION ERRORS FOR KITTI SEQUENCES(UNIT: DEGREE)

Seq.	5pt-Mart. [29]	4pt-Lee [20]	4pt-Liu [21]	4pt-Sw. [22]	4pt-Our	4pt-Our -Axis
00	0.059	0.031	0.032	0.033	0.031	0.032
01	0.047	0.044	0.043	0.028	0.045	0.027
02	0.056	0.028	0.028	0.030	0.027	0.029
03	0.058	0.037	0.038	0.037	0.037	0.036
04	0.036	0.019	0.022	0.021	0.020	0.022
05	0.046	0.022	0.022	0.023	0.022	0.021
06	0.038	0.024	0.023	0.022	0.023	0.021
07	0.047	0.022	0.022	0.022	0.021	0.022
08	0.048	0.023	0.022	0.024	0.021	0.024
09	0.054	0.025	0.025	0.027	0.024	0.026
10	0.057	0.024	0.024	0.026	0.023	0.025

TABLE III
MEDIAN TRANSLATION DIRECTION ERRORS FOR KITTI SEQUENCES(UNIT: DEGREE)

Seq.	5pt-Mart. [29]	4pt-Lee [20]	4pt-Liu [21]	4pt-Sw. [22]	4pt-Our	4pt-Our -Axis
00	1.409	1.317	1.31	1.116	1.288	1.113
01	1.991	2.228	2.066	1.263	2.174	1.218
02	1.112	1.062	1.068	0.977	1.069	0.975
03	1.601	1.436	1.484	1.21	1.389	1.175
04	0.723	0.676	0.759	0.605	0.756	0.612
05	1.113	0.928	0.925	0.776	0.923	0.75
06	0.722	0.71	0.728	0.571	0.72	0.566
07	1.463	1.066	1.064	0.918	1.039	0.95
08	1.477	1.412	1.371	1.248	1.372	1.222
09	0.995	0.922	0.894	0.873	0.933	0.824
10	1.101	1.05	1.059	0.92	1.065	0.883

(ATE) magnitudes [32]. For conciseness, we present only the trajectory results for sequence 00. As shown in Fig. 7, the 4pt-Our-Axis and 4pt-Sweeney solver achieves the highest accuracy compared to other methods. Furthermore, among the 4-point solvers utilizing IMU vertical angle priors, 4pt-Our also exhibits smaller ATE values than 4pt-Lee and 4pt-Liu. These observations further validate the effectiveness of our proposed method in real-world settings.

VI. CONCLUSION

In this work, we present two efficient minimal solvers for estimating the relative pose of multi-camera systems using a minimal number of four point correspondences. The first solver leverages the vertical direction provided by IMUs, while the second utilizes the rotation axis direction prior. Both proposed solvers are built upon a novel depth-based parameterization of translation, as opposed to traditional translation representations. Building on this foundation, we reduce the 4-DOF multi-camera relative pose estimation problem to solving a univariate 6th-degree polynomial. Extensive simulations and real-world experiments demonstrate that our solvers achieve improved computational efficiency while maintaining competitive accuracy compared to state-of-the-art solvers for multi-camera ego-motion estimation.

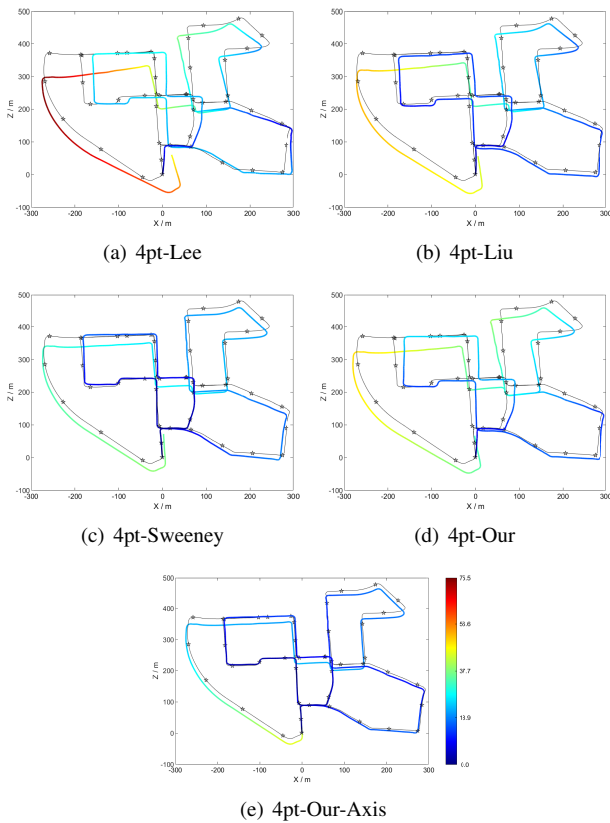


Fig. 7. Estimated visual odometry trajectories for KITTI sequences 00 (unit: meter). (a) Trajectory estimation with 4pt-Lee method, (b) Trajectory estimation with 4pt-Liu method, (c) Trajectory estimation with 4pt-Sweeney method, (d) Trajectory estimation with 4pt-Our method, (e) Trajectory estimation with 4pt-Our-Axis method.

REFERENCES

- [1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [2] Y. Ding, V. Kocur, V. Vávra, Z. B. Haladová, J. Yang, T. Sattler, and Z. Kukulova, “Reposed: Efficient relative pose estimation with known depth information,” in *IEEE International Conference on Computer Vision*, 2025, pp. 14 876–14 886.
- [3] J. Astermark, Y. Ding, V. Larsson, and A. Heyden, “Fast relative pose estimation using relative depth,” in *IEEE International Conference on 3D Vision*, 2024, pp. 873–881.
- [4] L. Taihang, G. Banglei, L. Minzu, L. Xiangyu, L. Jianbing, T. Jing, S. Yang, and Y. Qifeng, “Event-based multi-view photogrammetry for high-dynamic, high-velocity target measurement,” *Acta Mechanica Sinica*, 2025.
- [5] I. Eichhardt and D. Barath, “Relative pose from deep learned depth and a single affine correspondence,” in *European conference on computer vision*. Springer, 2020, pp. 627–644.
- [6] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [7] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [8] B. Guan, J. Zhao, Z. Li, F. Sun, and F. Fraundorfer, “Relative pose estimation with a single affine correspondence,” *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10 111–10 122, 2022.
- [9] Z. Liu, B. Guan, Y. Shang, Q. Yu, and L. Kneip, “Line-based 6-dof object pose estimation and tracking with an event camera,” *IEEE Transactions on Image Processing*, vol. 33, pp. 4765–4780, 2024.
- [10] T. Li, Z. Yu, B. Guan, J. Han, W. Lv, and F. Fraundorfer, “Trifocal tensor and relative pose estimation with known vertical direction,”

- IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1305–1312, 2025.
- [11] B. Guan, J. Zhao, D. Barath, and F. Fraundorfer, “Minimal solvers for relative pose estimation of multi-camera systems using affine correspondences,” *International Journal of Computer Vision*, vol. 131, no. 1, pp. 324–345, 2023.
- [12] B. Guan and J. Zhao, “Relative pose estimation for multi-camera systems from point correspondences with scale ratio,” in *ACM International Conference on Multimedia*, 2022, pp. 5036–5044.
- [13] B. Guan, J. Zhao, Z. Li, F. Sun, and F. Fraundorfer, “Minimal solutions for relative pose with a single affine correspondence,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1929–1938.
- [14] M. Tian, B. Guan, Z. Xing, and F. Fraundorfer, “Efficient ego-motion estimation for multi-camera systems with decoupled rotation and translation,” *IEEE Access*, vol. 8, pp. 153 804–153 814, 2020.
- [15] R. Pless, “Using many cameras as one,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. II–587.
- [16] H. Li, R. Hartley, and J.-h. Kim, “A linear approach to motion estimation using generalized camera models,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [17] O. Chum and J. Matas, “Matching with prosc-progressive sample consensus,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 220–226.
- [18] D. Barath, M. Ivashechkin, and J. Matas, “Progressive napsac: sampling from gradually growing neighborhoods,” *arXiv preprint arXiv:1906.02295*, 2019.
- [19] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, “Magsac++, a fast, reliable and accurate robust estimator,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1304–1312.
- [20] G. Hee Lee, M. Pollefeys, and F. Fraundorfer, “Relative pose estimation for a multi-camera system with known vertical direction,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 540–547.
- [21] L. Liu, H. Li, Y. Dai, and Q. Pan, “Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2432–2444, 2017.
- [22] C. Sweeney, J. Flynn, and M. Turk, “Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem,” in *IEEE International Conference on 3D Vision*, vol. 1, 2014, pp. 483–490.
- [23] M. HenrikStewénius, K. Aström, and D. Nistér, “Solutions to minimal generalized relative pose problems,” in *Workshop on Omnidirectional Vision in conjunction with ICCV*, 2005, pp. 1–8.
- [24] B. Guan, J. Zhao, S. Mitra, and L. Kneip, “Six-point method for multi-camera systems with reduced solution space,” *International Journal of Computer Vision*, vol. 133, no. 10, pp. 7270–7292, 2025.
- [25] J.-S. Kim and T. Kanade, “Degeneracy of the linear seventeen-point algorithm for generalized essential matrix,” *Journal of Mathematical Imaging and Vision*, vol. 37, pp. 40–48, 2010.
- [26] C. Xie, R. Xing, N. Hao, and F. He, “17-point algorithm revisited: Toward a more accurate way,” in *IEEE International Conference on Robotics and Automation*, 2024, pp. 10 208–10 214.
- [27] B. Guan and J. Zhao, “Affine correspondences between multi-camera systems for relative pose estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [28] B. Guan, J. Zhao, D. Barath, and F. Fraundorfer, “Minimal cases for computing the generalized relative pose using affine correspondences,” in *International Conference on Computer Vision*, 2021, pp. 6068–6077.
- [29] E. Martynushev and B. Li, “Efficient relative pose estimation for cameras and generalized cameras in case of known relative rotation angle,” *Journal of Mathematical Imaging and Vision*, vol. 62, no. 8, pp. 1076–1086, 2020.
- [30] Z. Kukulova, M. Bujnak, and T. Pajdla, “Closed-form solutions to minimal absolute pose problems with known vertical direction,” in *Asian Conference on Computer Vision*, 2010, pp. 216–229.
- [31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [32] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 573–580.