IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 7, 2020, accepted August 17, 2020, date of publication August 20, 2020, date of current version September 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3018225

Efficient Ego-Motion Estimation for Multi-Camera Systems With Decoupled Rotation and Translation

MIAO TIAN[1,2], BANGLEI GUAN[1,2], ZHIBIN XING[3], AND FRIEDRICH FRAUNDORFER[4,5]

[1]College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China
[2]Hunan Provincial Key Laboratory of Image Measurement and Vision Navigation, National University of Defense Technology, Changsha 410073, China
[3]School of Non-Commissioned Officer, Space Engineering University, Beijing 102200, China
[4]Institute of Computer Graphics and Vision, Graz University of Technology, 8010 Graz, Austria
[5]Remote Sensing Technology Institute, German Aerospace Center, 82234 Weßling, Germany

Corresponding author: Banglei Guan (banglei.guan@hotmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 11902349.

ABSTRACT In this article, we present novel solutions to estimate the ego-motion of a multi-camera system with a known vertical direction (e.g., from the inertial measurement unit). By assuming small camera motion between successive video frames, we demonstrate that rotation and translation estimation can be decoupled. This makes our methods require fewer correspondences to estimate the ego-motion and have a good accuracy. Accordingly, we estimate the ego-motion with two steps. First, we propose a 1-point method to estimate rotation with only a single correspondence which produces up to two solutions. Then, we adopt a 3-point linear method and a 2-point sampling method to solve translation which produce a single solution. We compared our algorithms with state-of-the-art algorithms on synthetic and real datasets. The experiments demonstrate that our algorithms are accurate and efficient in road driving scenarios. We also demonstrate that our proposed methods can efficiently find an optimal inlier set using histogram voting or exhaustive search instead of RANSAC.

INDEX TERMS Generalized epipolar constraint, multi-camera system, relative pose estimation.

I. INTRODUCTION

The relative pose estimation problem is classical and fundamental in computer vision applications, such as robotics, automotive industry, augmented reality, and visual simultaneous localization and mapping. This problem refers to computing the pose of the current frame with respect to the coordinate system related to the previous frame [1]. Different camera configurations are used to solve the problem, such as monocular, stereo, and multi-camera system. Monocular attracted wide attention from researchers and a large number of algorithms have arisen in prior work. The classical and basic solvers are the normalized 8-point algorithm [2] and the 5-point minimal algorithm [3].

However, recently, the multi-camera system has been extensively used for many emerging applications, such as autonomous driving with drones and vehicles because it covers a potentially large field-of-view [4], [5]. The larger the field-of-view (FoV), the more information we obtain around the environment. This allows us to detect and track objects robustly, particularly in environments with little texture. Our work focuses on the relative pose estimation for a multi-camera system. A multi-camera system can be modeled

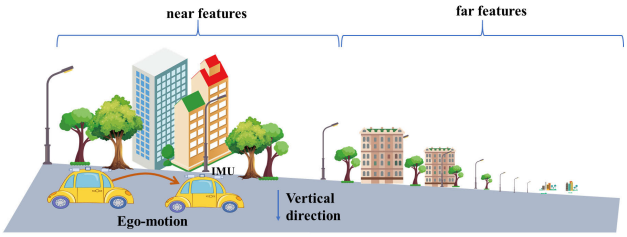


FIGURE 1. Example of a multi-camera system configuration mounted on a car. Given the known vertical direction, we estimate the relative rotation using the far features, and estimate the relative translation using the near features.

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan.

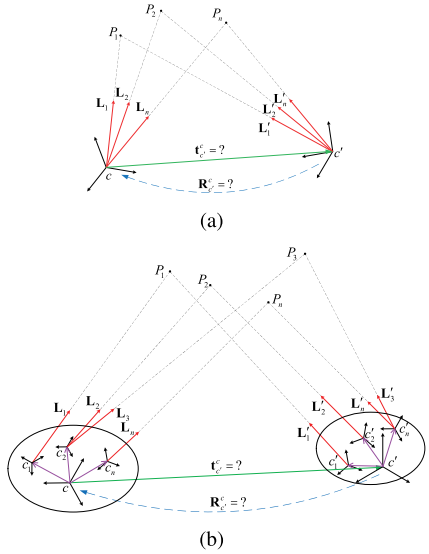


FIGURE 2. The ego-motion estimation problem of camera systems, (a) central camera system; (b) generalized multi-camera system.

as a generalized camera, which was proposed by [6], [7]. If the light rays passing through the three-dimensional (3D) world points and image points intersect at a single center of projection, the camera system is modeled as a central perspective projection model; otherwise, the camera system is modeled as a generalized camera model; that is, the difference between the central perspective projection model and generalized camera model is that the latter does not have a single center of projection, as shown in Fig. 2. The ego-motion of the multi-camera system can be obtained linearly using 17 points [6]–[8] and minimally using six points [9], [10].

To deal with the outlier matches, the ego-motion estimation algorithms are applied in a robust framework, such as random sample consensus (RANSAC) [11]. To reduce the computation cost and improve the robustness of RANSAC, reducing the number of points required for estimating a motion model is an efficient strategy [12]. Thus, it is necessary and important to study minimal solvers for ego-motion estimation. Researchers use additional information to reduce the number of points required. One approach is to use some motion constraints to simplify the problem of ego-motion estimation. For example, planar motion [13], [14] or Ackermann steering motion [15], [16] in road driving scenarios. Another approach is to obtain additional information from other sensors, for example, the inertial measurement unit (IMU). At the present time, as the IMU has become cheaper and prevalent, it is increasingly often fixed on multi-camera systems. As the accuracy of the yaw angle from the IMU sensor is not as good as those of the roll and pitch angles, we use the roll and pitch angles to determine the vertical direction, which reduces the degrees of freedom (DOFs) in the relative pose by two. Thus, this makes the ego-motion estimation process simpler and faster. This ideal has been applied in the monocular camera system [13], [17]–[19] and the multi-camera system [20]–[22].

Our work aims at solving the ego-motion estimation problem for a multi-camera system when the vertical direction in the multi-camera coordinate frame is provided by the IMU [13], [20]. The knowledge of the vertical direction can reduce the DOFs in the relative pose by two, and then the unknown translation and unknown yaw angle are left for us to solve. Additionally, we use the fact that for points that are far away, the parallax-shift (induced by translation) between two views is hardly noticeable [13]. Hence, we classify all points into two sets: "far points," which are far away, and "near points," which are nearby in the scene. In terms of the far points, the translation between consecutive frames is negligible while the ego-motion is small. Thus, we can decouple rotation and translation estimation if there are some far points in the scene. Accordingly, we estimate the ego-motion with two steps. First, we propose a 1-point algorithm to estimate the rotation. This allows us to solve the rotation of the multi-camera rig with a minimal set of one correspondence. To the end of accurate and robust results, 1-point algorithm is finally embedded into histogram voting and RANSAC loop. Then, we propose two methods to estimate translation, a linear method using three near points and a sampling method using two near points.

The main contributions of this article are as follows:

- Our work decouples rotation and translation estimation for multi-camera systems. We estimate rotation using the far points, and then estimate translation using the near points.
- We propose a 1-point method to estimate rotation for multi-camera systems on the condition of knowing the vertical direction. The method requires only a single point and produces up to two candidate solutions, thus improving the efficiency of our method in RANSAC.
- We propose two methods to estimate translation for multi-camera systems, 3-point linear method and 2-point sampling method, which have high accuracy.

The remainder of this article is organized as follows: in Section 2, we present an overview of related work. We briefly establish notation and introduce the generalized epipolar constraint (GEC) in Section 3. In Section 4, we describe our methods in detail. We conduct experiments on simulation and real datasets in Section 5, where our methods are compared with state-of-the-art methods.

II. RELATED WORK

Pless *et al.* [6] formulated the GEC, which is a mechanism that makes a net of cameras a single camera. The linear solution of ego-motion for generalized cameras requires 17 corresponding image rays because there are 18 unknowns in the constraint. Similarly, Sturm *et al.* [7] provided epipolar geometry for generalized cameras and suggested 17 correspondences to solve the relative motion linearly. The above methods have their merits, but do not work on real data. Li *et al.* [8] found that the 17-point approach is not applicable to certain special generalized

camera configurations, hence, provided an extension to 17-point approach and proposed linear 16-point and 14-point approaches in certain special configurations such as multi-camera systems where the camera centers are aligned.

A minimal solution was first proposed by Stewénius *et al.* [9]. The algorithm requires only six correspondence pairs to determine relative motion using the Gröbner basis technique and provides up to 64 solutions. However, it is unsuitable for a real-time system because of the high computational complexity of the solver. Kneip *et al.* proposed a nonlinear optimization algorithm over relative rotation only based on an efficient eigenvalue minimization strategy [23], [24]. A single solution is sought by a closed-form function using seven or more correspondences and is susceptible to obtaining a local optimal result. Ventura *et al.* [10] proposed a solver that uses the first-order approximation to the relative pose. The approximation motion model is appropriate under the assumption of small motion between two images, so it is applied in continuous motion. Although the model simplifies the relative pose problem, the method yields up to 20 solutions, so it is unsuitable for inclusion in a RANSAC scheme. Moreover, to solve a 20th-degree polynomial makes the method sensitive to noise.

To reduce the DOFs of relative motion, researchers have exploited extra information and/or assumptions from motion models. Consequently, fewer correspondences are required to solve the problem. By constraining the camera motion to the Ackermann motion model, Lee *et al.* [15] proposed a 2-point method that yields up to six candidate solutions. However, the model applies the case in which a car undergoes on a planar, which is a very strict assumption in practice. For a practical application, Lee *et al.* [20] used the information of the vertical direction to reduce the DOFs of relative motion by two and then proposed minimal 4-point and linear 8-point algorithms. The minimal 4-point algorithm provides up to eight possible solutions via the hidden variable resultant method and the linear 8-point algorithm yields up to one solution using the standard SVD method. Sweeney *et al.* [21] derived relative motion problems as quadratic eigenvalue problems with a known axis of rotation. Similar to the algorithm of Lee *et al.* [20], it provides an eight-degree polynomial. Unlike the algorithm of Lee *et al.* [20], however, it yields up to six solutions using four correspondences. Liu *et al.* [22] used a first-order approximation motion model and an IMU sensor to determine the unknown yaw angle from the roots of a four-degree polynomial.

III. GENERALIZED EPIPOLAR CONSTRAINT

In this section, we introduce the generalized epipolar constraint briefly. A multi-camera rig can be described as a generalized camera that captures a set of light rays [7], [8]. We use the Plücker vector to express a light ray. The Plücker vector is composed of a pair of 3-vectors, $\mathbf{u}$ and $\mathbf{q}$, which are the direction vector and moment vector, respectively. We choose a reference frame V arbitrarily, and then the extrinsic matrix of the ith camera C_i in V is denoted by $[\mathbf{R}_{Ci}, \mathbf{T}_{Ci}]$ and the intrinsic matrix is denoted by $\mathbf{K}_{Ci}$. The Plücker coordinate of the light ray from the optical center of the camera C_i to the normalized image point $\tilde{\mathbf{x}}_{ij} = \mathbf{K}_{Ci}^{-1}\mathbf{x}_{ij}$ is given by

$$\mathbf{L}_{ij} = \begin{bmatrix} \mathbf{u}_{ij}^T & \mathbf{q}_{ij}^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{u}_{ij}^T & \left(\mathbf{T}_{Ci} \times \mathbf{u}_{ij}\right)^T \end{bmatrix}^T, \quad (1)$$

where $\mathbf{u}_{ij} = \mathbf{R}_{Ci}\tilde{\mathbf{x}}_{ij}$ is the unit direction of the light ray in the reference coordinate system. The transformation from k frame to $k+1$ frame is denoted by rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$. Suppose $\begin{pmatrix} \mathbf{L}_{ij,k} & \mathbf{L}_{ij,k+1} \end{pmatrix}$ is a pair of Plücker line correspondences in two views. Then the Plücker coordinate of $\mathbf{L}_{ij,k}$ in the $k+1$ frame is expressed as

$$\mathbf{L'}_{ij,k} = \begin{bmatrix} \mathbf{R}\mathbf{u}_{ij,k} \\ \mathbf{R}\mathbf{q}_{ij,k} + [\mathbf{t}]_\times \mathbf{R}\mathbf{u}_{ij,k} \end{bmatrix}, \quad (2)$$

where $[\mathbf{t}]_\times$ is a skew-symmetric matrix made up of translation vector $\mathbf{t}$. $\mathbf{L'}_{ij,k}$ and $\mathbf{L}_{ij,k+1}$ intersect in space if and only if

$$\mathbf{q}_{ij,k+1}^T \left(\mathbf{R}\mathbf{u}_{ij,k}\right) + \mathbf{u}_{ij,k+1}^T \left(\mathbf{R}\mathbf{q}_{ij,k} + [\mathbf{t}]_\times \mathbf{R}\mathbf{u}_{ij,k}\right) = 0. \quad (3)$$

As a result, the generalized epipolar constraint (GEC) can be written as

$$\mathbf{L}_{ij,k+1}^T \begin{bmatrix} [\mathbf{t}]_\times \mathbf{R} & \mathbf{R} \\ \mathbf{R} & \mathbf{0} \end{bmatrix} \mathbf{L}_{ij,k} = 0. \quad (4)$$

IV. METHODS

We solve the ego-motion estimation problem of multi-camera system using three steps. First, we use the roll and pitch angle from the IMU sensor to transform the Plücker line correspondences, thereby aligning the vertical direction of the multi-camera system. This step reduces the DOFs of rotation from three to one so that we have a single unknown in rotation to solve. Second, because the multi-camera system is greater than or equal to 10 Hz for road vehicle application, a car cannot run much farther within a 0.1-second time interval or fewer time interval. It is a reasonable assumption that the ego-motion between two successive frames is small [10], [22]. In practice, in the case of small ego-motion, we find that the change of the near point's image coordinate contains rotation and translation information, and the change of the far point's image coordinate only contains rotation information. Accordingly, we propose a 1-point method to estimate the rotation. Finally, according to the given rotation, we estimate translation with a 3-point linear method and a 2-point sampling method.

A. ALIGN THE VERTICAL DIRECTION

The vertical direction refers to the direction of gravity, that is, the "up" direction of the multi-camera system. The knowledge of the vertical direction can be provided by vanishing points [25] or the IMU measurements. In this study, as the accuracy of the yaw angle from the IMU sensor is not as good as those of the roll and pitch angles, we use the roll and pitch angles from IMU to align the vertical direction. We can obtain the pitch angle (rotation around the X-axis), roll angle (rotation around the Y-axis), and yaw angle (rotation around the Z-axis) from the IMU with respect to the reference frame V,

where the XY plane is parallel to the ground, and the Z-axis points down. The rotation matrices from the yaw, pitch, and roll angles between the two consecutive generalized camera frames are denoted by $(\mathbf{R}_y, \mathbf{R}_p, \mathbf{R}_r) \leftrightarrow (\mathbf{R}'_y, \mathbf{R}'_p, \mathbf{R}'_r)$. Hence, the relative rotation matrix $\mathbf{R}$ is written as

$$\mathbf{R} = \mathbf{R}_r^T \mathbf{R}_p^T \mathbf{R}_y^T \mathbf{R}'_y \mathbf{R}'_p \mathbf{R}'_r. \tag{5}$$

Coincidentally, $\mathbf{R}_y^T \mathbf{R}'_y$ is the relative yaw rotation matrix, so we denote it by $\Delta\mathbf{R}_y$. As observed from (5), only a single unknown $\Delta\mathbf{R}_y$ in the relative rotation remains to be solved. We substitute (5) into the GEC in (4), and eliminate the Plücker line and camera indices ij for brevity:

$$\mathbf{L}^T \begin{bmatrix} [\mathbf{t}]_\times \mathbf{R}_r^T \mathbf{R}_p^T \Delta\mathbf{R}_y \mathbf{R}'_p \mathbf{R}'_r & \mathbf{R}_r^T \mathbf{R}_p^T \Delta\mathbf{R}_y \mathbf{R}'_p \mathbf{R}'_r \\ \mathbf{R}_r^T \mathbf{R}_p^T \Delta\mathbf{R}_y \mathbf{R}'_p \mathbf{R}'_r & \mathbf{0} \end{bmatrix} \mathbf{L}' = 0. \tag{6}$$

To simplify (6), we first factor out $\mathbf{R}'_p \mathbf{R}'_r$ to obtain

$$\mathbf{L}^T \begin{bmatrix} [\mathbf{t}]_\times \mathbf{R}_r^T \mathbf{R}_p^T \Delta\mathbf{R}_y & \mathbf{R}_r^T \mathbf{R}_p^T \Delta\mathbf{R}_y \\ \mathbf{R}_r^T \mathbf{R}_p^T \Delta\mathbf{R}_y & \mathbf{0} \end{bmatrix} \hat{\mathbf{L}}' = 0, \tag{7}$$

where $\hat{\mathbf{L}}' = \begin{bmatrix} \mathbf{R}'_p \mathbf{R}'_r & \mathbf{0} \\ \mathbf{0} & \mathbf{R}'_p \mathbf{R}'_r \end{bmatrix} \mathbf{L}'$. To factor out $\mathbf{R}_r^T \mathbf{R}_p^T$, $\mathbf{t}$ is denoted by

$$\mathbf{t} = \mathbf{R}_r^T \mathbf{R}_p^T \hat{\mathbf{t}}. \tag{8}$$

Hence, $[\mathbf{t}]_\times \mathbf{R}_r^T \mathbf{R}_p^T$ can be replaced by $\mathbf{R}_r^T \mathbf{R}_p^T \left[\hat{\mathbf{t}}\right]_\times$. Then factor out $\mathbf{R}_r^T \mathbf{R}_p^T$ to obtain

$$\mathbf{L}^T \begin{bmatrix} \mathbf{R}_r^T \mathbf{R}_p^T & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_r^T \mathbf{R}_p^T \end{bmatrix} \begin{bmatrix} \left[\hat{\mathbf{t}}\right]_\times \Delta\mathbf{R}_y & \Delta\mathbf{R}_y \\ \Delta\mathbf{R}_y & \mathbf{0} \end{bmatrix} \hat{\mathbf{L}}' = 0. \tag{9}$$

From (9), we can obtain a simplified GEC:

$$\hat{\mathbf{L}}^T \begin{bmatrix} \left[\hat{\mathbf{t}}\right]_\times \Delta\mathbf{R}_y & \Delta\mathbf{R}_y \\ \Delta\mathbf{R}_y & \mathbf{0} \end{bmatrix} \hat{\mathbf{L}}' = 0, \tag{10}$$

where $\hat{\mathbf{L}} = \begin{bmatrix} \mathbf{R}_p \mathbf{R}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_p \mathbf{R}_r \end{bmatrix} \mathbf{L}$. Thus, after aligning the vertical direction, we have two unknowns to solve: $\hat{\mathbf{t}}$ and $\Delta\mathbf{R}_y$.

B. ROTATION ESTIMATION METHODS

We assume that the change of the far point's image coordinate is only affected by the relative rotation, that is, the relative translation is close to zero when we only use the far points [13]. Therefore, if the Plücker lines are both formed by the far points, a new GEC can be obtained:

$$\hat{\mathbf{L}}^T \begin{bmatrix} \mathbf{0} & \Delta\mathbf{R}_y \\ \Delta\mathbf{R}_y & \mathbf{0} \end{bmatrix} \hat{\mathbf{L}}' = 0. \tag{11}$$

We rewrite $\Delta\mathbf{R}_y$ by applying the tangent half-angle substitution given by $\cos\alpha = (1-q^2)/(1+q^2)$ and $\sin\alpha = (2q)/(1+q^2)$, where α is the relative yaw angle that makes up $\Delta\mathbf{R}_y$:

$$\Delta\mathbf{R}_y = \frac{1}{1+q^2} \begin{bmatrix} 1-q^2 & -2q & 0 \\ 2q & 1-q^2 & 0 \\ 0 & 0 & 1+q^2 \end{bmatrix}. \tag{12}$$

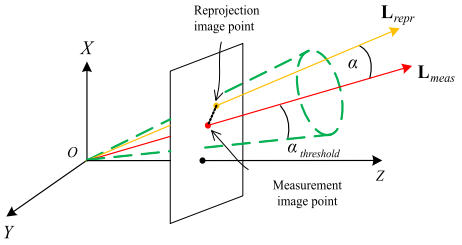


FIGURE 3. Geometric meaning of the reprojection errors described by the angle α between the measured Plücker line $\mathbf{L}_{meas}$ and the reprojected Plücker line $\mathbf{L}_{repr}$.

We substitute the yaw rotation matrix $\Delta\mathbf{R}_y$ into the generalized epipolar constraint in (11) to obtain

$$Aq^2 + Bq + C = 0, \tag{13}$$

where the coefficients A, B, and C are formed by the elements of the Plücker line correspondence $\hat{\mathbf{L}}(\hat{l}_1, \hat{l}_2, \hat{l}_3, \hat{l}_4, \hat{l}_5, \hat{l}_6) \leftrightarrow \hat{\mathbf{L}}'(\hat{l}'_1, \hat{l}'_2, \hat{l}'_3, \hat{l}'_4, \hat{l}'_5, \hat{l}'_6)$:

$$\begin{cases} A = \hat{l}_3\hat{l}'_6 - \hat{l}_4\hat{l}'_1 - \hat{l}_2\hat{l}'_5 - \hat{l}_5\hat{l}'_2 - \hat{l}_1\hat{l}'_4 + \hat{l}_6\hat{l}'_3 \\ B = 2\hat{l}_2\hat{l}'_4 - 2\hat{l}_1\hat{l}'_5 - 2\hat{l}_4\hat{l}'_2 + 2\hat{l}_5\hat{l}'_1 \\ C = \hat{l}_1\hat{l}'_4 + \hat{l}_4\hat{l}'_1 + \hat{l}_2\hat{l}'_5 + \hat{l}_5\hat{l}'_2 + \hat{l}_3\hat{l}'_6 + \hat{l}_6\hat{l}'_3. \end{cases} \tag{14}$$

We obtain q by solving (13). A single Plücker line correspondence provides up to two possible solutions for q. We use two methods to generate an optimal solution in this article 1-point RANSAC and histogram voting.

1) 1-POINT RANSAC

RANSAC is the standard process for estimating the model to deal with the outlier matches. RANSAC randomly samples minimal data sets to generate model hypothesis. Then, the model hypothesis are tested on the whole data set to identify and remove outliers. Finally, only inliers are used to estimate the model. As illustrated in Fig. 3, if the measured transformation is perfect, the difference between the reprojected Plücker line $\mathbf{L}_{repr}$ and the measured Plücker line $\mathbf{L}_{meas}$ is negligible. Hence, we compute the reprojection error as $1 - \mathbf{L}_{meas}^T \mathbf{L}_{repr}$ in this article. It should be noted that we consider a corresponding point pair as an inlier if the angle α between $\mathbf{L}_{meas}$ and $\mathbf{L}_{repr}$ is lower than the threshold $\alpha_{threshold}$ given by $\arctan(t/f)$, where f is the focal length and t is the threshold of the classical reprojection error in pixels [26].

2) HISTOGRAM VOTING

The possible solutions for q only use a single Plücker line correspondence; hence, a straightforward approach that requires no iteration is based on histogram voting method [16]. The method is more efficient than RANSAC because the histogram voting method avoids computing the inliers and outliers for each possible solution. A Plücker line correspondence is used to compute a hypothesis of q. Due to our assumption that the ego-motion is small, thus leading to $q \in (-1, 1)$. Then we use these hypotheses of q to generate histogram statistics in discrete bins (e.g., a bin size of 0.01).

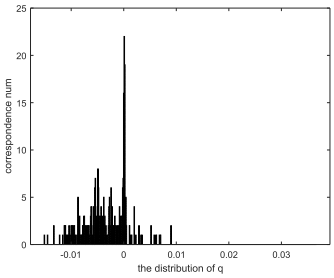


FIGURE 4. An example histogram generated using real data.

According to the number of elements in each container in the histogram, the center of the bin corresponding to the maximum number is considered as the best solution for q. Finally, we substitute q into (12) to obtain the relative yaw angle matrix. Therefore, we can obtain the relative rotation according to (5). Fig. 4 shows an example histogram generated using real data.

C. TRANSLATIOIN ESTIMATION MENTHODS

After estimating rotation, it is easy to perform translation estimation using near points. According to the generalized epipolar constraint in (10), after aligning the vertical direction, we factor out $\Delta\mathbf{R}_y$ to rewrite (10) as:

$$\hat{\mathbf{L}}^T \begin{bmatrix} [\hat{\mathbf{t}}]_\times & \mathbf{E} \\ \mathbf{E} & \mathbf{0} \end{bmatrix} \hat{\mathbf{L}}'' = 0, \tag{15}$$

where $\hat{\mathbf{L}}'' = \begin{bmatrix} \Delta\mathbf{R}_y & \mathbf{0} \\ \mathbf{0} & \Delta\mathbf{R}_y \end{bmatrix} \hat{\mathbf{L}}'$.

1) 3-POINT LINEAR METHOD

In this subsection, we present a linear method to solve the remaining translation parameters. We parameterize $\hat{\mathbf{L}}''$ as $(\hat{l}''_1, \hat{l}''_2, \hat{l}''_3, \hat{l}''_4, \hat{l}''_5, \hat{l}''_6)$ and expand (15) to obtain

$$\mathbf{M}\hat{\mathbf{t}} = n, \tag{16}$$

where $\mathbf{M} = \begin{bmatrix} l_2 l''_3 - l_3 l''_2 & -l_1 l''_3 + l_3 l''_1 & l_1 l''_2 - l_2 l''_1 \end{bmatrix}$, $n = l_4 l''_1 + l_5 l''_2 + l_6 l''_3 + l_1 l''_4 + l_2 l''_5 + l_3 l''_6$, and $\hat{\mathbf{t}} = \begin{bmatrix} \hat{t}_x & \hat{t}_y & \hat{t}_z \end{bmatrix}^T$. The constraints from the three Plücker line correspondences can be stacked into a linear equation system. We can solve the linear equation system to obtain $\hat{\mathbf{t}}$. Then the estimated translation $\mathbf{t}$ is recovered using (8).

2) 2-POINT METHOD USING DISCRETE SAMPLING FOR THE X-Y TRANSLATION DIRECTION

The 3-point linear method in the previous subsection requires three Plücker line correspondences. Despite this, in this subsection, we adopt a sampling method so that we can reduce the correspondences required. First, we choose an appropriate parameter to perform discrete sampling within a suitable bounded range. Then we search for a global optimality using an exhaustive search method. Because the direction of the translation in the x-y plane has the obvious value of the discrete step bound, we sample the direction in this article. The direction of the translation can be described as θ and can be sampled in steps of 1° from 0° to 360°. Consequently, we can rewrite (16) as

$$\begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} \sqrt{\hat{t}_x^2 + \hat{t}_y^2}\cos\theta \\ \sqrt{\hat{t}_x^2 + \hat{t}_y^2}\sin\theta \\ \hat{t}_z \end{bmatrix} = n. \tag{17}$$

Equation (17) has two unknowns, $\sqrt{\hat{t}_x^2 + \hat{t}_y^2}$ and $\hat{t}_z$, which leads to a 2-point method for estimating translation. We denote $\sqrt{\hat{t}_x^2 + \hat{t}_y^2}$ by r. The constraints from two correspondences can be stacked into an equation system, which finally leads to

$$\begin{cases} (m_1\cos\theta + m_2\sin\theta)r + m_3\hat{t}_z = n \\ (m'_1\cos\theta + m'_2\sin\theta)r + m'_3\hat{t}_z = n'. \end{cases} \tag{18}$$

For a given angle θ, the two unknowns r and $\hat{t}_z$ can be obtained as:

$$r = \frac{m'_3 n - m_3 n'}{(m_1\cos\theta + m_2\sin\theta)m'_3 - (m'_1\cos\theta + m'_2\sin\theta)m_3}, \tag{19}$$

$$\hat{t}_z = \frac{(m_1\cos\theta + m_2\sin\theta)n' - (m'_1\cos\theta + m'_2\sin\theta)n}{(m_1\cos\theta + m_2\sin\theta)m'_3 - (m'_1\cos\theta + m'_2\sin\theta)m_3}. \tag{20}$$

According to the values of r and $\hat{t}_z$, we can recover $\hat{\mathbf{t}}$. Then, we substitute $\hat{\mathbf{t}}$ into (8) to obtain translation $\mathbf{t}$.

V. EXPERIMENTS

We performed experiments on both synthetic and real scene data to validate the performance of the proposed methods. The tests on the synthetic scene were used to demonstrate the accuracy and robustness of our methods with respect to pixel noise and IMU noise. The tests on the real scene were used to demonstrate the feasibility of our solvers in practical autonomous driving scenarios. We compared the root mean square errors of the rotation and translation direction with those of state-of-the-art solvers. The errors were defined as follows:

$$E_{\mathbf{R}} = \arccos((trace(\mathbf{R}_{gt}^T\mathbf{R}_{est}) - 1)/2), \tag{21}$$

$$E_{\mathbf{t}} = \arccos((\mathbf{t}_{gt}^T\mathbf{t}_{est})/(\|\mathbf{t}_{gt}\|\,\|\mathbf{t}_{est}\|)), \tag{22}$$

where $\mathbf{R}_{gt}$ denotes the ground truth rotation, $\mathbf{R}_{est}$ denotes the corresponding estimated rotation, $\mathbf{t}_{gt}$ denotes the ground truth translation, $\mathbf{t}_{est}$ denotes the corresponding estimated translation. The abbreviations of the solvers for comparison are as follows:

17pt-Li: linear solver of Li *et al.* to determine the relative pose problem of a multi-camera system with 17 correspondences [8].

8pt-Kneip: solver of Kneip *et al.* to determine the relative pose of a multi-camera system with an efficient eigenvalue minimization strategy [24].

6pt-Stewénius: minimal solver of Stewénius *et al.* to determine the relative pose of a multi-camera system with the Gröbner basis technique [9].

TABLE 1. Average computation time comparison of multi-camera ego-motion solvers (unit:μs).

Solvers	Timings
17pt-Li [8]	62.5
8pt-Kneip [24]	5126.8
6pt -Stewénius [9]	131.7
4pt-Liu [22]	5.4
4pt-Lee [20]	41.5
4pt-Our	0.2

4pt-Liu: minimal solver of Liu *et al.* to determine the relative pose of a multi-camera system using a first-order approximation motion model [22].

4pt-Lee: minimal solver of Lee *et al.* to determine the relative pose of a multi-camera system using the hidden variable resultant method [20].

4pt-Our: our minimal solver to determine the relative pose of a multi-camera system using 1-point RANSAC rotation estimation method and 3-point linear translation estimation method.

Histogram voting: our solver to determine the relative rotation of a multi-camera system using the histogram voting approach.

Histogram voting+3pt: our solver to determine the relative translation of a multi-camera system using the 3-point linear method after solving the relative rotation using the histogram voting approach.

Histogram voting+2pt: our solver to determine the relative translation of a multi-camera system by sampling for the x-y translation direction after solving the relative rotation using the histogram voting approach.

All codes were implemented in C++ and tested on a 2.81 GHz Intel Core i7 with 16 GB RAM. The implementations of 17pt-Li, 8pt-Kneip, and 6pt-Stewenius were provided in the OpenGV library [26]. We used 4pt-Liu publicly available implementations from GitHub. We implemented the solver 4pt-Lee. We tested each solver on 10,000 randomly generated problems to compute the average computation times shown in Table 1.

A. SYNTHETIC DATA EXPERIMENTS

In the simulations, the experimental setup was as follows: we generated two cameras. The baseline of the two cameras was set to 0.5 m and the focal length was set to 1,000 pixels. The two cameras had non-overlapping FoVs. For each trial, we created two sets of 3D points: the far points were randomly distributed in the cube $[-10, 10]$ m $\times$ $[5, 1000]$ m $\times$ $[-10, 10]$ m and the near points were randomly distributed in the cube $[-5, 5]$ m $\times$ $[0, 5]$ m $\times$ $[-5, 5]$ m. The two sets of 3D points each contained 100 points. We projected the 3D points onto the image plane of the multi-camera system to obtain the feature points. The motion between consecutive frames is small in automatic driving; therefore, the relative rotation angle rotated on each axis was set to a random angle in the range of $[-1°,1°]$ and the translation was set from 0 to 0.5 m randomly in the simulations. These conditions were chosen to reflect realistic conditions. Each solver was used within a RANSAC scheme.

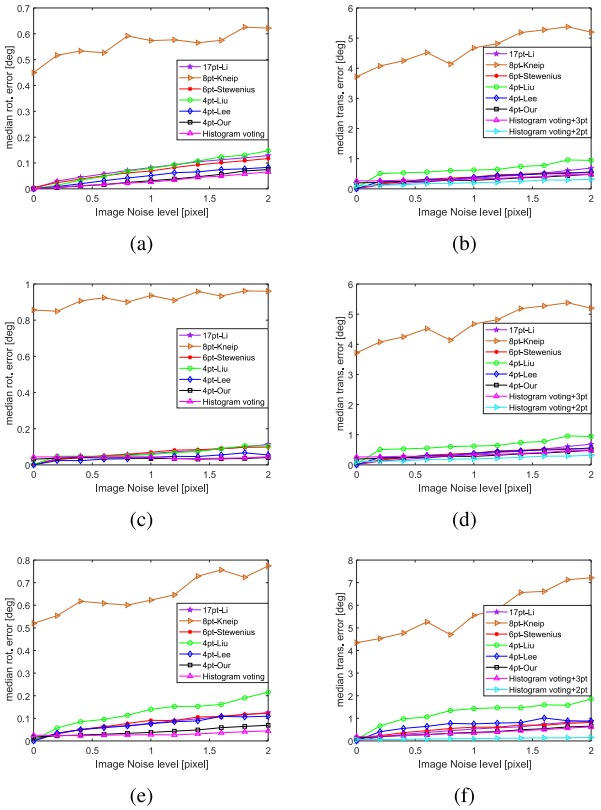


FIGURE 5. Median rotation (right) and translation (left) errors from 1,000 trials per image noise level with perfect IMU data. (a) and (b): forward motion, (c) and (d): sideways motion, (e) and (f): random motion.

1) IMAGE NOISE EXPERIMENT

We conducted experiments to validate how the accuracy of the image point coordinates affected the relative pose estimated by our solvers. Gaussian noise has been added to the image point coordinates ranging from 0 to 2 pixels of standard deviation at an interval of 0.2 pixels, while the IMU data has been kept perfect. We compared our solvers with four solvers (4pt-Liu [22], 4pt-Lee [20], 6pt-Stewénius [9], 8pt-Kneip [24], and 17pt-Li [8]). For each level of image noise, 1,000 random trials were generated with perfect IMU data, and then we used the median error as a measure of performance to evaluate the estimated transformation.

Fig. 5 shows the accuracy of rotation and translation computed using different solvers for three cases: forward motion, sideways motion, and random motion. As observed in Fig. 5(a) and (b), our two methods were close with each other, and slightly outperformed the other methods for forward motion. However, in the case of sideways motion shown in Fig. 5(c), there was no obvious tendency in terms of the rotational error using our methods with gradually increasing image noise. The results show that image noise had less effect on the accuracy of rotation estimation than sideways motion. For random motion, it is interesting to see that our methods were slightly worse in the absence of image noise, while our

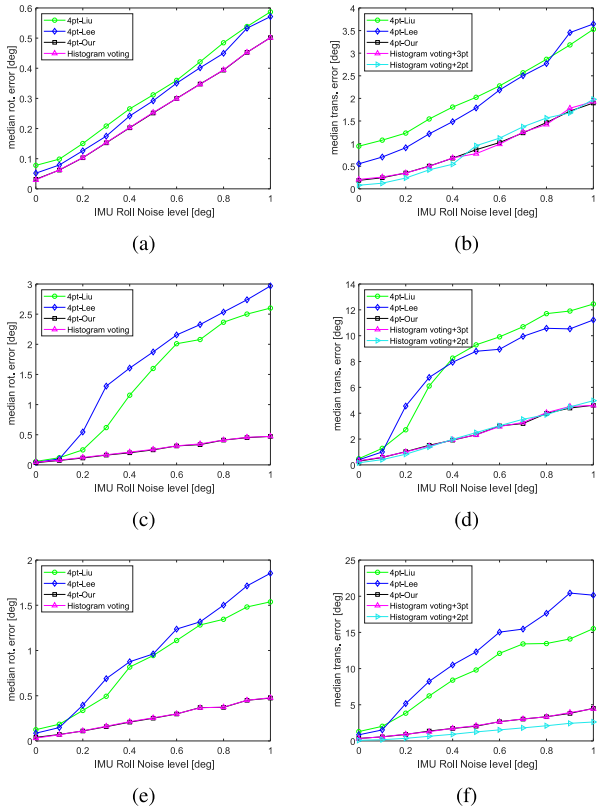


FIGURE 6. Median rotation (right) and translation (left) errors from 1,000 trials per roll angle noise level with image noise 1 pixel standard deviation. (a) and (b): forward motion, (c) and (d): sideways motion, (e) and (f): random motion.

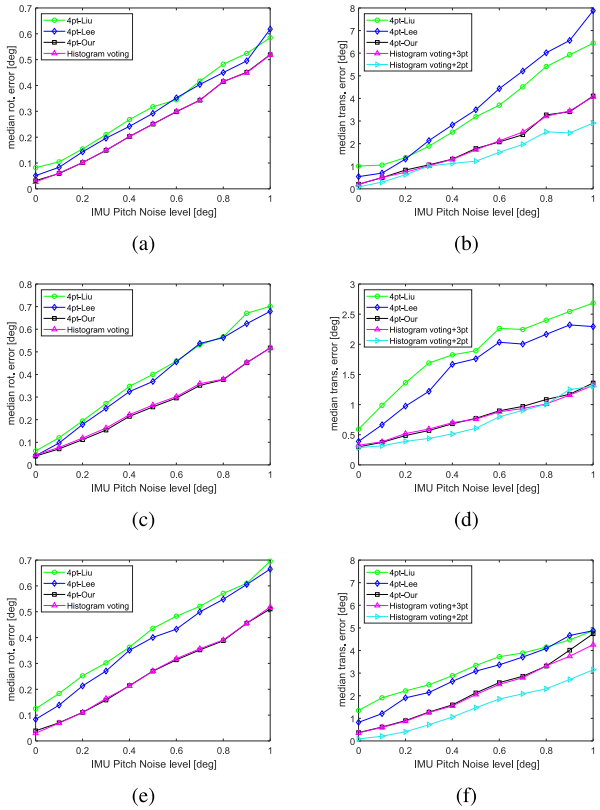


FIGURE 7. Median rotation (right) and translation (left) errors from 1,000 trials per pitch angle noise level with image noise 1 pixel standard deviation. (a) and (b): forward motion, (c) and (d): sideways motion, (e) and (f): random motion.

methods worked better for gradually increased image noise levels. It seems that our assumption that far points are only influenced by rotation led to certain errors. We observe that results of 8pt- Kneip [24] were poor. It seems that the method becomes numerically degenerate when the rotation matrix is close to identity.

2) IMU NOISE EXPERIMENT

We conducted experiments to validate how the accuracy of IMU affects the relative pose estimated by our solvers as our solvers rely on IMU measurements. Hence, we compared our solvers with 4pt-Liu [22] and 4pt-Lee [20], which work with the known vertical direction. Gaussian noise of standard deviation ranging from 0° to 1° was added to the IMU while assuming image noise with a standard deviation of 1 pixel. Figs. 6 and 7 show the median error of rotation and translation from 1,000 trials at each level of IMU noise using our methods compared with 4pt-Liu [22] and 4pt-Lee [20]. The figures demonstrate that our methods were close to each other in terms of the median error, and outperformed the other methods at all levels of IMU noise in terms of both rotation and translation estimation.

B. REAL-WORLD DATASET EXPERIMENTS

To determine the performance of our algorithms in a practical driving scene, we compared our methods with state-of-the-art methods on the KITTI autonomous driving benchmarking dataset [27]. We performed experiments on the first 11 sequences (00–10) in the visual odometry benchmark dataset, which provide the ground truth. These sequences provide left and right images, and contain approximately 46,000 images. In our experiments, we extracted feature matches from each camera individually using the SURF algorithm and we did not use any cross-camera matches. In the following sections we conducted 3 sets of experiments with the KITTI dataset. In the first experiment, we tested the effectiveness of our strategy which removes a big part of feature points for the rotation estimation. In the second experiment, we tested the performance of our algorithms compared to state-of-the-art methods. In the third experiment, we tested the quality of the inlier detection in comparison to state-of-the-art methods.

1) SELECTION OF THE INLIER USED TO ESTIMATE ROTATION ACCORDING TO THE Y-COORDINATE

In the study, we only use the far points to estimate rotation, thus we wish to preemptively discard those near points as early as possible. In practice, we observed that the y-coordinate of image coordinate of point faraway changes little in consecutive frames after aligning the vertical direction. As shown in Fig. 8, if the camera moves forward with a distance of d, the change of the y-coordinate of image point

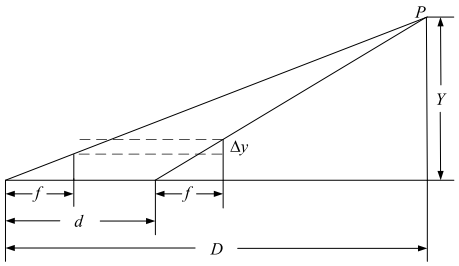


FIGURE 8. The schematic of the threshold of separation of far and near points.

can be computed as:

$$\Delta y = f\left(\frac{Y}{D-d} - \frac{Y}{D}\right), \tag{23}$$

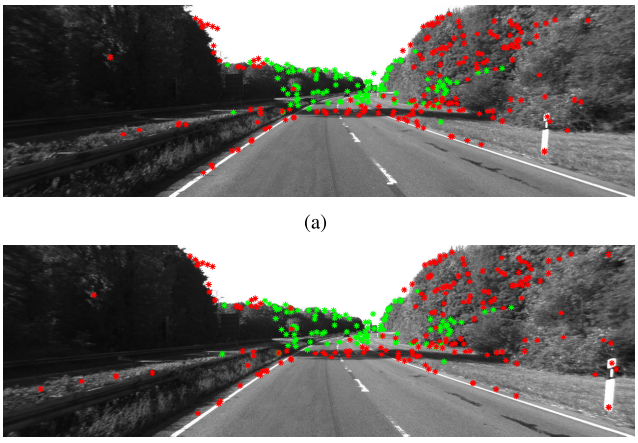
where f is the focal length, D is the distance from a 3D point $\mathbf{P}$ to the camera, Y is the distance from the 3D point to the optical axis. As observed in (23), the change of the y-coordinate of image point is approximate to 0 while the distance from a point $\mathbf{P}$ to the camera D is far enough. Consequently, for far points, the parallax-shift (induced by translation) between two views is hardly noticeable.The yaw rotation matrix influences the change of the x-coordinate of the image point, and does not influence the change of the y-coordinate of the image point. It is exactly based on this we can decouple the rotation and translation.

We separate far from near points in two steps. First, with the knowledge of the vertical direction, we pre-rotate the image point to make the camera plane vertical to the ground plane. Then, these points are partitioned to far points and near points according to whether the change of the y-coordinate with respect to the image coordinate system is less than 1 pixel. This threshold value was given on the based of simulation experiments. Fig. 9 shows an example of the separation of far and near points using this criterion, where the green points denote far points, and red points denote near points. As observed in Fig. 9, the far points were well separated from the near points using y-coordinate of image points. Table 2 shows the number of far points in each sequence based on the simple criterion. *NumberPoints* refers to the average number of points extracted using SURF from the right and left images in each sequence. *NumberFar* refers to the average number of far points chosen using the y-coordinate. $Ratio = NumberFar/NumberPoints$ refers to the average percentage of far points. As shown in Table 2, the criterion can reject outliers with a percentage of more than 50% for rotation estimation. This allows to significantly remove a big part of feature points for the rotation estimation, thus making it more efficient.

2) COMPARISON OF ROTATION AND TRANSLATION ESTIMATION WITH THE GROUND TRUTH

We compared our algorithms with 17pt-Li [8], 8pt-Kneip [24], 6pt-Stewénius [9], 4pt-Liu [22], and 4pt-Lee [20]. To compare our algorithms fairly, we did not apply any nonlinear refinement, bundle adjustment, or loop closure. This means that we only computed the frame-to-frame visual odometry component. We ran all algorithms within a RANSAC framework. For all sequences, the number of RANSAC iterations was fixed at 100 and the inlier threshold was set to 1 pixel in the experiments.

We computed the median error of the rotation and translation estimates with respect to the ground truth. We report the accuracy results of the rotation and translation estimation on KITTI odometry sequences in Table 3 and Table 4. From the tables, we observe that the performances of our methods were comparable with or better than that of the other methods, particularly for translation estimation. The higher accuracy of the translation estimation was caused by the strategy that only used the near points to estimate the translation. According to Table 3, histogram voting was the best approach for rotation estimation on most sequences. Although, it should be noted that 4pt-Liu [22] and 4pt-Lee [20] were close to each other and 8pt-Kneip [24] was slightly poor, which coincides with the simulation experiments on image pixel noise. Table 4 shows that our three methods, 4pt-Our, Histogram voting+3pt, and Histogram voting+2pt, were all better than



(a)

(b)

FIGURE 9. The separation of far (green) and near (red) points using the change of y-coordinate of image point. (a) and (b) are consecutive frames from the KITTI visual odometry dataset seq-01.

TABLE 2. Effect of the y-coordinate test. Outliers with a percentage of more than 50% are rejected for rotation estimation.

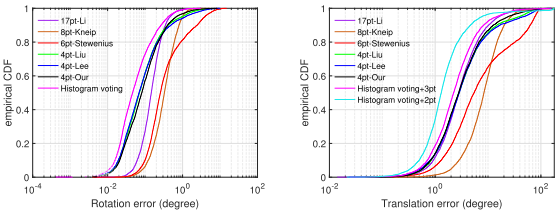
Seq.	$NumberPoints$	$NumberFar$	$Ratio$
00	762	280	36.75%
01	358	201	56.15%
02	770	213	26.67%
03	950	448	47.16%
04	647	249	38.49%
05	789	347	43.98%
06	527	219	41.56%
07	957	475	49.63%
08	815	312	38.28%
09	663	215	32.43%
10	774	321	41.47%

TABLE 3. Accuracy results of rotation estimation on KITTI odometry sequences 00–10 (unit: degrees). The median for each error measure is given.

Seq.	17pt-Li [8]	8pt-Kneip [24]	6pt -Stewénius [9]	4pt-Liu [22]	4pt-Lee [20]	4pt-Our	Histogram voting
00	0.146	0.327	0.256	0.071	0.069	0.082	**0.046**
01	0.189	0.295	1.192	0.085	0.098	0.076	**0.033**
02	0.132	0.347	0.196	0.057	0.059	0.076	**0.047**
03	0.116	0.279	0.287	**0.060**	0.068	0.084	0.065
04	0.105	0.290	0.205	0.037	0.038	0.040	**0.021**
05	0.126	0.289	0.231	0.059	0.057	0.065	**0.025**
06	0.145	0.306	0.175	0.071	0.066	0.084	**0.033**
07	0.134	0.262	0.271	0.063	0.066	0.079	**0.028**
08	0.124	0.296	0.212	0.056	0.055	0.065	**0.031**
09	0.147	0.334	0.204	0.062	**0.056**	0.087	0.072
10	0.124	0.316	0.233	0.056	0.055	0.072	**0.049**

TABLE 4. Accuracy results of translation estimation on KITTI odometry sequences 00–10 (unit: degrees). The median for each error measure is given.

Seq.	17pt-Li [8]	8pt-Kneip [24]	6pt -Stewénius [9]	4pt-Liu [22]	4pt-Lee [20]	4pt-Our	Histogram voting+3pt	Histogram voting+2pt
00	2.805	8.389	5.452	2.817	2.856	2.798	2.236	**1.308**
01	8.632	14.658	56.506	5.281	5.891	1.923	1.721	**0.951**
02	2.027	6.836	3.099	2.015	1.989	2.201	1.966	**1.205**
03	2.831	9.364	8.404	3.156	3.189	2.685	2.532	**1.358**
04	1.739	7.465	3.791	1.788	1.531	1.647	1.367	**0.716**
05	2.707	8.257	5.602	2.695	2.713	2.451	1.774	**1.055**
06	2.895	7.640	3.432	2.394	2.401	2.331	1.656	**0.938**
07	3.316	9.063	8.064	3.216	3.217	3.118	2.365	**1.347**
08	2.549	8.241	4.498	2.624	2.673	2.557	2.116	**1.420**
09	2.387	7.225	3.414	2.153	2.050	2.210	2.119	**1.044**
10	2.334	8.444	4.117	2.268	2.308	2.463	2.294	**1.200**

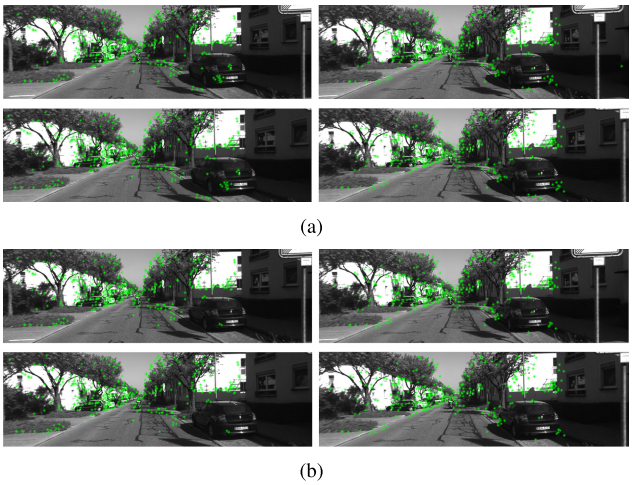


(a) (b)

FIGURE 10. Empirical cumulative error distributions for all frames in KITTI VO-seq-00: (a) rotation error, (b) translation error.

TABLE 5. Average RANSAC runtime comparison of our method with state-of-the-art multi-camera ego-motion methods over KITTI sequences (unit:s).

Methods	RANSAC time
17pt-Li [8]	57.32
8pt-Kneip [24]	11.61
6pt -Stewénius [9]	92.84
4pt-Liu [22]	0.45
4pt-Lee [20]	0.72
4pt-Our	0.03



(a)

(b)

FIGURE 11. Inlier set detection of the first two frames from KITTI seq-00. Green points represent inliers. Left: frames of the left camera. Right: frames of the right camera. (a) Ground truth inlier set (749 matches). Top row: previous frames. Bottom row: current frames. (b) Inlier detection results for Histogram voting+2pt method (705 matches). Top row: previous frame. Bottom row: current frame.

the other methods in terms of translation estimation; in particular, the accuracy of histogram voting+2pt was the highest.

The empirical cumulative error distributions of rotation and translation for KITTI VO-seq-00 are provided in Fig. 10. The proposed solvers (Histogram voting, Histogram voting+3pt, and Histogram voting+2pt) provided significantly better estimations than the state-of-the-art methods. 4pt-Our was in accordance with 4pt-Liu [22] and 4pt-Lee [20]. Average RANSAC runtime comparison of our method with state-of-the-art multi-camera ego-motion methods over KITTI sequences is shown in Table 5. The proposed method was more efficiently used within RANSAC for robust estimation in comparison to state-of-the-art methods with good accuracy of the ego-motion estimation.

The scenarios of KITTI odometry dataset are diverse, such as light changing or less of environment texture. It is interesting to see that our methods had outstanding performance on KITTI odometry datasets. Consequently, the real experiments

TABLE 6. The mean of inlier recovery rate on KITTI odometry sequences 00-10.

Seq.	17pt-Li [8]	8pt-Kneip [24]	6pt -Stewénius [9]	4pt-Liu [22]	4pt-Lee [20]	4pt-Our	Histogram voting+3pt	Histogram voting+2pt
all	81%	65%	28%	86%	86%	85%	**87**%	**87**%

demonstrate that a road driving scenario does fit our method very well, no matter light changing or less of environment texture.

3) COMPARISON OF THE INLIER RECOVERY RATE

Another extremely helpful application of our methods is selecting a correct inlier set required for the next step (e.g., accurate motion estimation and non-linear optimization). Therefore, we conducted an experiment that tested how many of the real inliers (calculated from the ground truth) can be found using our methods. Table 6 shows the mean of the inlier recovery rate on KITTI odometry sequences 00–10 using our methods and state-of-the-art methods. Histogram voting+3pt and Histogram voting+2pt were slightly better than the other methods. Fig. 11 shows the inlier set detection of the first two frames from KITTI seq-00 using Histogram voting+2pt.

VI. CONCLUSION

In this article, we proposed new methods to solve the problem of ego-motion estimation of a multi-camera system with decoupled rotation and translation estimation, while the vertical direction is known. We assumed that the far points were not affected by the translation on the condition of small motion, which proved to be correct in road driving scenes using experiments on KITTI datasets. According to the assumption, we proposed a minimal solver to estimate rotation with only a single far point. To estimate translation, we proposed a linear method with three near points and a sampling method with two near points. We verified the efficiency and robustness of our methods in a series of experiments on synthetic and real data. These experiments demonstrated that our methods applied very well to automatic driving scenarios which contain far features. In future work, we plan to try more reliable feature extraction to improve the accuracy of the ego-motion estimation.

REFERENCES

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[2] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997.

[3] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[4] E. Martyushev and B. Li, "Efficient relative pose estimation for cameras and generalized cameras in case of known relative rotation angle," *J. Math. Imag. Vis.*, pp. 1–11, May 2020, doi: 10.1007/s10851-020-00958-5.

[5] J. Zhao, W. Xu, and L. Kneip, "A certifiably globally optimal solution to generalized essential matrix estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12034–12043.

[6] R. Pless, "Using many cameras as one," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. 587–593.

[7] P. Sturm, "Multi-view geometry for general camera models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 206–212.

[8] H. Li, R. Hartley, and J.-h. Kim, "A linear approach to motion estimation using generalized camera models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[9] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström, "Solutions to minimal generalized relative pose problems," in *Proc. Workshop Omnidirectional Vis.*, Beijing, China, 2005.

[10] J. Ventura, C. Arth, and V. Lepetit, "An efficient minimal solution for multi-camera motion," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 747–755.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[12] Z. Kukelova, J. Kileel, B. Sturmfels, and T. Pajdla, "A clever elimination strategy for efficient minimal solvers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4912–4921.

[13] B. Guan, P. Vasseur, C. Demonceaux, and F. Fraundorfer, "Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2320–2327.

[14] O. Oreifej, N. Lobo, and M. Shah, "Horizon constraint for unambiguous UAV navigation in planar scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1159–1165.

[15] G. H. Lee, F. Faundorfer, and M. Pollefeys, "Motion estimation for self-driving cars with a generalized camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2746–2753.

[16] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4293–4299.

[17] O. Saurer, P. Vasseur, R. Boutteau, C. Demonceaux, M. Pollefeys, and F. Fraundorfer, "Homography based egomotion estimation with a common direction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 327–341, Feb. 2017.

[18] O. Naroditsky, X. S. Zhou, J. Gallier, S. I. Roumeliotis, and K. Daniilidis, "Two efficient solutions for visual odometry using directional correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 818–824, Apr. 2012.

[19] F. Fraundorfer, P. Tanskanen, and M. Pollefeys, "A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 269–282.

[20] G. H. Lee, M. Pollefeys, and F. Fraundorfer, "Relative pose estimation for a multi-camera system with known vertical direction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 540–547.

[21] C. Sweeney, J. Flynn, and M. Turk, "Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem," in *Proc. 2nd Int. Conf. 3D Vis.*, Dec. 2014, pp. 483–490.

[22] 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 Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2432–2444, Aug. 2018.

[23] L. Kneip and S. Lynen, "Direct optimization of Frame-to-Frame rotation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2352–2359.

[24] L. Kneip and H. Li, "Efficient computation of relative pose for multi-camera systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 446–453.

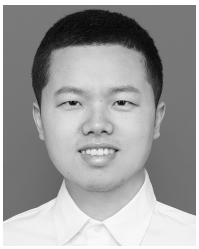
[25] J.-C. Bazin, H. li, S.-O. Kweon, C. Demonceaux, P. Vasseur, and K. Ikeuchi, "A branch-and-bound approach to correspondence and grouping problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 1565–1576, Dec. 2013.

[26] L. Kneip and P. Furgale, "OpenGV: A unified and generalized approach to real-time calibrated geometric vision," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1–8.

[27] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

MIAO TIAN received the B.S. degree in automation from Nanjing Normal University, Nanjing, China, in 2012, and the M.S. degree in guidance navigation and control from Information Engineering University, Zhengzhou, China, in 2016. She is currently pursuing the Ph.D. degree in aeronautical and astronautical science and technology from the National University of Defense Technology, Changsha, China. Her research interests include computer vision and visual navigation.

BANGLEI GUAN received the B.S. degree in geomatics engineering from Wuhan University, Wuhan, China, in 2012, and the Ph.D. degree in aeronautical and astronautical science and technology from the National University of Defense Technology, Changsha, China, in 2018. From 2016 to 2017, he was an Exchange Student with the Graz University of Technology, Graz, Austria. He is currently an Assistant Professor with the National University of Defense Technology. His research interests include computer vision and photogrammetry.



ZHIBIN XING received the B.S., M.S., and Ph.D. degrees from Information Engineering University, Zhengzhou, China, in 2013, 2016, and 2019, respectively. He is currently a Lecturer with Space Engineering University, Beijing, China. His research interest includes the area of physical geodesy.

FRIEDRICH FRAUNDORFER received the Ph.D. degree in computer science from the Graz University of Technology (TU Graz), Austria, in 2006. He had a postdoctoral stay at the University of Kentucky, at the University of North Carolina at Chapel Hill, at ETH Zürich. From 2012 to 2014, he was the Deputy Director of the Chair of Remote Sensing Technology, Technical University of Munich. He is currently an Associate Professor with the Institute of Computer Graphics and Vision, TU Graz. His main research interests include 3D computer vision and robot vision.