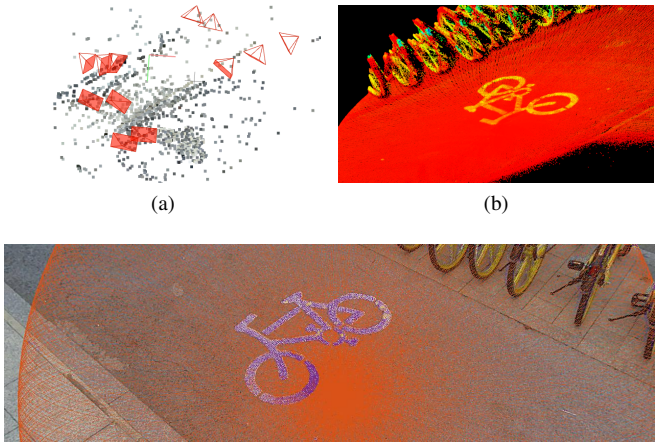
IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

PBACalib: Targetless Extrinsic Calibration for High-Resolution LiDAR-Camera System Based on Plane-Constrained Bundle Adjustment

Feiyi Chen[1], Liang Li[2], Shuyang Zhang[1], Jin Wu[1] and Lujia Wang[1]

***Abstract*—The strategy of fusing multi-model data, especially from cameras, light detection and ranging sensors (LiDAR), is frequently considered in robotics to enhance the performance of the perception and navigation tasks. Extrinsic calibration, which spatially aligns different sources into a unified coordinate representation, directly determines the performance of the combined data. In this paper, we propose *PBACalib*, a novel targetless extrinsic calibration algorithm aiming at the dense LiDAR-camera system based on the plane-constrained bundle adjustment (PBA). The proposed method utilizes the feature points derived from a prominent plane in the scene and iteratively minimizes the reprojection error. A maximum likelihood estimator (MLE) is designed by considering the uncertainty information of the measurements. Furthermore, we explore the distribution of collected data and characterize the robustness and solvability of the extrinsic estimates using a confidence factor. Simulation and real-world experiments both qualitatively and quantitatively demonstrate the robustness and accuracy of our method. The comparison experiments show that the proposed method outperforms another targetless method. To benefit the community, Matlab code has been publicly released on Github.**

***Index Terms*—Targetless extrinsic calibration, High-resolution LiDAR, LiDAR-Camera calibration, bundle adjustment.**



(a) (b) (c)

Fig. 1. Data representation in different sensors and calibration result. (a) 3D structure and camera poses recovered from SfM. (b) Point cloud from Livox (c) Reprojection result after calibration. Different colors represent different intensity values acquired from point cloud

I. Introduction

In robotic systems, LiDARs and cameras, as the most commonly used sensors, compensate each other by providing rich texture information and 3D measurements of the environment. Extrinsic calibration becomes essential to fuse these two types of data into the same coordinate system, especially for conducting automated tasks in a complicated environment, like tightly coupled SLAM, colorization, etc. In this paper, our work explores the extrinsic calibration problem in dense LiDAR-camera system.

In the autonomous driving industry, the resolution of the LiDAR equipped on the vehicle grows rapidly with reduced cost and the release of new solid-state LiDAR (e.g., Livox). However, most existing works focus on mechanical LiDAR (e.g., Velodyne) and rely on prepared artificial targets, such as checkerboard [1], circle [2], and sphere [3], which are sometimes unavailable. Besides, it is challenging to implement some old methods on dense LiDAR because of different data structures. For instance, the large number of bleeding points that exist around depth-discontinuous edges in dense LiDAR, as explained in [4], degrade the performances of some edge extraction algorithms, like [1] [5]. Moreover, zero-valued and multi-valued mapping problems [4] also make the mutual information-based method [6] unstable. On the other hand, the mounting position and orientation of sensors depend on the actual needs, and it fails some calibration methods. For example, the targetless methods, like [4], resort to the depth-continuous edges of the environment. In this case, LiDARs need to be mounted upwards to observe enough edges of the buildings for the calibration, which is not practical in some cases, as shown in Fig. 2, with LiDAR installed toward the ground.

Considering the above challenges, we propose *PBACalib*, which captures several pairs of images and point clouds around a plane with arbitrary texture to calibrate. Three different features are utilized for the extrinsic calibration, which contains plane coefficients in the LiDAR frame, pixel feature points

Manuscript received: July 5, 2022; Revised: September 30, 2022; Accepted: November 3, 2022.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by Guangdong Basic and Applied Basic Research Foundation, under project 2021B1515120032 (*Corresponding author: Lujia Wang*).

[1]Feiyi Chen, Shuyang Zhang, Jin Wu and Lujia Wang are with the Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China and the Clear Water Bay Institute of Autonomous Driving, Nanshan, Shenzhen. The authors are also with Unity Drive Inc. (email: eewanglj@ust.hk, fchenak@connect.ust.hk).

[2]Liang Li is with the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong SAR, China

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

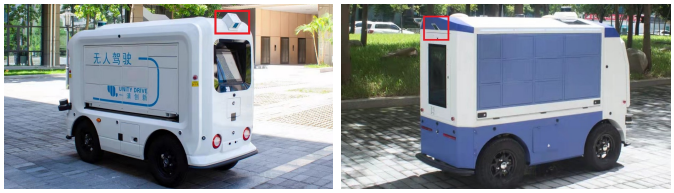


Fig. 2. LiDARs are installed towards ground

and restored 3D points in the camera frame. To ensure convergence, the extrinsics are found in a coarse-to-fine manner. Camera poses and the extrinsics are optimized simultaneously by iteratively minimizing the reprojection error. Specifically, our contributions can be summarized as follows:

- A novel targetless extrinsic calibration method for high-resolution LiDAR and camera based on plane-constrained bundle adjustment. It only needs a common textured ground, wall, or other planes to accomplish the calibration.
- Validity analysis on the collected dataset. We theoretically analyze the distribution of the collected data and introduce a *confidence factor* to determine whether the input data is sufficient for calibration. Specific requirements are listed to guide users to stabilize the calibration result, which are: *1)* we need at least four poses; *2)* the target planes do not intersect at the same point; *3)* at least three normal vectors are non-coplanar.
- Evaluated with various simulation, real-world and comparison experiments, which reveal that the proposed method is accurate and robust. To benefit the community, we publicly release the source code on Github[1].

II. Related Work

In the literature, the extrinsic calibration between camera and LiDAR can be summarized into two categories: the target-based and targetless methods.

Target-based methods calibrate the extrinsics using artificial objects, like a checkerboard [1], calibration room [7], custom-built calibration pattern [8], etc. Compared with the targetless method, the estimation of the extrinsics is usually more accurate and robust for knowing the prior information of the targets, like the geometric size, shape, and texture. However, the calibration targets need to be prepared in advance, which is not practical sometimes in real applications. Specifically, Zhou *et al.* [1] extracted the geometric boundary of the checkerboard in both camera and LiDAR data, and the extrinsics are refined by iteratively minimizing the points-to-line and points-to-plane distance. Xie *et al.* [7] built a room pasted with fiducial markers (Apriltag) to calibrate multiple cameras and LiDARs even without common field of view (FoV). The poses of cameras and LiDARs are calculated precisely by leveraging fiducial markers and the geometry of this room respectively, while the cost of building this room is not affordable for most users.

Targetless methods find the natural features around the environment, like objects [9], edges [4] and data intensity [6], to align cameras and LiDARs. They can also be generally characterized into two categories: registration-based and motion-based methods. For the registration-based methods, the measurements are aligned by extracting the common features from multi-model sensors in the common FoV. For example, Pandey *et al.* [6] found the distribution of reflective intensity and pixel intensity are similar, and the mutual information reaches a maximum when the extrinsics are correct. However, it suffers from illumination problems in images and sparser LiDAR, like 16-beam LiDAR is not applicable. In contrast, the motion-based methods first estimate the sensors' ego poses separately from multiple frames of data and find the extrinsics using hand-eye calibration [10]. For instance, Horn *et al.* [11] utilized dual quaternion to find the global optimal solution even in planar motion cases. Taylor *et al.* [12] introduced a measurement noise model to drive a more stable solution in an unconstrained environment. Recently, Wu *et al.* [13] summarized the pose estimation problem into general QPEPs in a unified quaternion framework, which contains many algorithms used in target-based and targetless calibration, like PnP, hand-eye calibration, point-to-plane registration, etc.

For dense LiDAR, more research [4] [14] [15] have come out in recent years. As stated in [14], the data distribution and non-repetitive scanning style make Livox different from mechanical LiDAR, and it is challenging to generalize previous methods on Livox. By considering these differences, they proposed their method on data processing and feature extraction, followed by RANSAC PnP to align 2D-3D interior pattern corners on the checkerboard. On the other hand, Yuan *et al.* [4] fully utilized the dense representation of the point cloud to extract depth-continuous edges from the structural environment. The extrinsics were iteratively optimized by aligning 2D-3D edges. In addition, some calibration techniques used in other active depth sensing devices are also applicable to LiDARs. For example, Zeisl *et al.* [16] used the sparse map recovered by SfM as a geometric prior and minimized the alignment error iteratively to find the intrinsic and extrinsic parameters simultaneously in RGB-D camera.

Compared with the previous methods [1] [14], our method achieves accurate results without using any prepared targets. Compared with [11] [17], we need fewer poses. Specifically, four frames are enough. Moreover, compared with [4], the calibration scene in our method is easier to find, and our method is less sensitive to the initial parameters. Our method can still work using a inclined plane near the ground when the sensor suite is mounted on the vehicle toward the ground.

III. Problem Statement

Our PBACalib chooses a plane with arbitrary texture to calibrate and collects several frames of data. The plane on both sensors is extracted using plane-RANSAC fitting. As the images are 2D measurements, the structure from motion (SfM) and PBA is implemented separately to calculate the camera poses. Finally, the extrinsics are initialized in a closed-form solution and also optimized by PBA. By introducing the measurement noise model, an MLE estimator is designed to iteratively minimize the residual function. And the PBA and MLE are further explained as follows.

[1]https://github.com/chenfeiyi/PBACalib

TABLE I
NOMENCLATURE

Notation	Explanation
$()^w, ()^{c_i}, ()^{l_i}$	Frame of the world, i-th camera, and i-th frame LiDAR
$\mathbf{T}$	Transformation in the Lie group $SE(3)$
$\mathbf{R}$	Rotation in the Lie group $SO(3)$
$\mathbf{t}$	Translation in $\mathbb{R}^3$
s	Global scale factor in camera poses
$\mathbf{x}$	State vector
$\mathbf{\Pi}$	Plane coefficients in $\mathbb{R}^6$
$\mathbf{n}$	Plane normal vector in $\mathbb{R}^3$
$\bar{\mathbf{p}}$	A 3D point on the plane
$\mathbf{u}$	2D pixel point in image
$\mathbf{K}$	Camera intrinsic matrix
$\mathbf{w}$	Zero-mean Gaussian noise
$\mathbf{\Xi}$	Covariance matrix of the state vectors
$\mathbf{\Sigma}$	Covariance matrix of the extracted features
$\mathbf{\Lambda}$	The information matrix of the state vectors
$\mathbf{J}$	Jacobian matrix of the state vectors

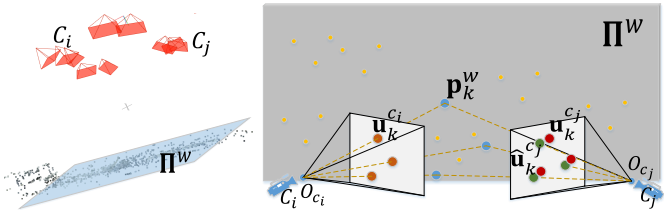
A. Plane-Constrained Bundle Adjustment

The conventional SfM algorithm recovers the camera poses and 3D structure simultaneously by minimizing the reprojection error, which is proved to be very effective. In special cases, the prior information can be integrated into optimization to further increase the accuracy. For example, the plane information we used in our calibration. After the initialization of SfM, the camera poses $\mathcal{C} = \{\mathbf{T}^{c_1}, \mathbf{T}^{c_2}, ..., \mathbf{T}^{c_N}\}$, 2D feature points in i-th frame $\mathcal{U}^{c_i} = \{\mathbf{u}_1^{c_i}, ..., \mathbf{u}_{M_i}^{c_i}\}$, and the 3D structure points $\mathcal{P}^w = \{\mathbf{p}_1^w, \mathbf{p}_2^w, ..., \mathbf{p}_M^w\}$ in the world coordinate system are recovered. $\mathbf{T}^{c_i}$ denotes the transformation from world coordinate system to i-th frame camera coordinate system. By introducing the plane information of the environment, the camera poses can be further refined using the PBA [18]. The geometric relation is shown in Fig. 3. Generally, PBA in our calibration is summarized into 4 steps: *1)* Extract the planes from 3D points $\mathcal{P}$ using plane-RANSAC fitting iteratively. Only the plane $\mathbf{\Pi}^w = [\mathbf{n}^{w\top}, (\bar{\mathbf{p}}^w)^\top]^\top \in \mathbb{R}^{6\times 1}$ with the most 3D points is kept; *2)* Find the new 3D plane structure points $\mathcal{P}^{c_i} = \{\mathbf{p}_1^{c_i}, ..., \mathbf{p}_k^{c_i}, ...\}$ in i-th frame camera coordinate system based on $\mathcal{U}^{c_i}$ and $\mathbf{\Pi}^w$; *3)* Reproject the 3D points $\mathcal{P}^{c_i}$ into j-th frame image plane, and the projected 2D points are represented by $\hat{\mathcal{U}}^{c_j} = \{\hat{\mathbf{u}}_1^{c_j}, ..., \hat{\mathbf{u}}_k^{c_j}, ...\}$; *4)* Refine camera poses by minimizing the reprojection error between the projected points $\hat{\mathcal{U}}^{c_j}$ and the detected 2D feature points $\mathcal{U}^{c_j}$ in j-th frame. Specifically, the new 3D plane structure points $\mathcal{P}^{c_i}$ are the intersection between the plane $\mathbf{\Pi}^w$ and the light ray across the 2D feature points $\mathcal{U}^{c_i}$ and the camera origin, which are derived by

$$\mathbf{p}_k^{c_i} = f(\mathbf{T}^{c_i}, \mathbf{\Pi}^w) = \frac{\mathbf{t}^{c_i\top}\mathbf{R}^{c_i}\mathbf{n}^w + \mathbf{n}^{w\top}\bar{\mathbf{p}}^w}{\mathbf{q}_k^{c_i\top}\mathbf{R}^{c_i}\mathbf{n}^w}\mathbf{q}_k^{c_i}, \quad (1)$$

where $\mathbf{q}_k^{c_i} = \mathbf{K}^{-1}[\mathbf{u}_k^{c_i\top}, 1]^\top$ denotes the unit depth back projection. The plane normal $\mathbf{n}^w$ is the eigenvector associated with the minimum eigenvalue of the points covariance matrix

$$\mathbf{S} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{p}_i^w - \bar{\mathbf{p}}^w)(\mathbf{p}_i^w - \bar{\mathbf{p}}^w)^\top, \bar{\mathbf{p}}^w = \frac{1}{M}\sum_{i=1}^{M}\mathbf{p}_i^w. \quad (2)$$



(a) SfM (b) PBA

Fig. 3. The SfM and PBA. The orange points denote the 3D points recovered from SfM. The blue points are the intersection between the plane and the light ray. The notation refers to Table. I

The reprojection error between $\hat{\mathcal{U}}^{c_j}$ and $\mathcal{U}^{c_j}$ is formulated as follows,

$$\begin{aligned}\mathbf{e}_k &= \frac{1}{z_k}\mathbf{K}(\mathbf{R}_{c_i}^{c_j}\mathbf{p}_k^{c_i} + \mathbf{t}_{c_i}^{c_j}) - \mathbf{u}_k^{c_j} \\ &= \hat{\mathbf{u}}_k^{c_j} - \mathbf{u}_k^{c_j}, \quad (3)\\ [x_k, y_k, z_k]^\top &= \mathbf{K}(\mathbf{R}_{c_i}^{c_j}\mathbf{p}_k^{c_i} + \mathbf{t}_{c_i}^{c_j}), \quad (4)\\ \mathbf{R}_{c_i}^{c_j} &= \mathbf{R}^{c_j}\mathbf{R}^{c_i\top}, \mathbf{t}_{c_i}^{c_j} = \mathbf{t}^{c_j} - \mathbf{R}^{c_j}\mathbf{R}^{c_i\top}\mathbf{t}^{c_i}, \quad (5)\end{aligned}$$

where $\mathbf{K}$ denotes the camera intrinsic matrix.

B. Maximum Likelihood Estimation

We introduce the measurement noise into optimization and formulate our calibration as a MLE problem. In each iteration, the residual function is approximated by the first-order expansion

$$\mathbf{e}_k(\mathbf{x}) \approx \mathbf{r}_k + \mathbf{J}_{k,\mathbf{x}}\delta\mathbf{x} + \mathbf{J}_{\mathbf{w}_k}\mathbf{w}_k = 0, \quad (6)$$

where $\mathbf{x}$ is our estimated variables, and $\mathbf{w}_k$ is the measurement noise which is subjected to zero-mean Gaussian noise $\mathcal{N}(0, \mathbf{\Sigma}_k)$. It implies that $\mathbf{r}_k + \mathbf{J}_{k,\mathbf{x}}\delta\mathbf{x} \sim \mathcal{N}(0, \mathbf{\Sigma}_k)$, and the maximum likelihood estimator is formulated as

$$\begin{aligned}\mathbf{x} &= \arg\max_{\mathbf{x}} \log(\prod_k p(\mathcal{F}_k \mid \mathbf{x})) \\ &= \arg\min_{\mathbf{x}} \sum_k \|(\mathbf{r}_k + \mathbf{J}_{k,\mathbf{x}}\delta\mathbf{x})\|^2_{(\mathbf{J}_{\mathbf{w}_k}\mathbf{\Sigma}_k\mathbf{J}_{\mathbf{w}_k}^\top)}, \quad (7)\end{aligned}$$

where $\|\mathbf{a}\|^2_{\mathbf{\Sigma}} = \mathbf{a}^\top\mathbf{\Sigma}^{-1}\mathbf{a}$, and $\mathcal{F}_k$ contains all the measurements. Leveberg-Marquardt [19] method is used to iteratively update the state vector $\mathbf{x}$ and solve this equation until convergence. In the last iteration, the state covariance is calculated as $\mathbf{\Xi}_{\mathbf{xx}} = \mathbf{\Lambda}^{-1}$, where $\mathbf{\Lambda}$ is called *information matrix* [20], and $\mathbf{\Lambda} = \sum_k \mathbf{J}_{k,\mathbf{x}}(\mathbf{J}_{\mathbf{w}_k}^\top\mathbf{\Sigma}_k\mathbf{J}_{\mathbf{w}_k})^{-1}\mathbf{J}_{k,\mathbf{x}}^\top$.

IV. METHODOLOGY

A. Camera Poses Estimation

Several frames of camera and LiDAR data around the plane scene are collected in our calibration. As stated in section III-A, camera poses and the 3D scene are recovered by SfM and refined by PBA. In practice, we use the open-source tool COLMAP [21] to perform sparse SfM. Specifically, the objective function of PBA is formulated as

$$\mathcal{C} = \arg\min_{\mathcal{C}} \sum_{i=1}^{N}\sum_{j\neq i}^{N}\sum_{k=1}^{N_{i,j}} \|\mathbf{e}_k\|^2. \quad (8)$$

The MLE estimator is used to iteratively find the optimal solution. The measurement noise comes from the 2D feature detection. The detected pixel is denoted by $\mathbf{u}_k^{c_i} = \mathbf{u}_{k,gt}^{c_i} + \mathbf{w}_k^{c_i}$, where $\mathbf{u}_{k,gt}^{c_i}$ represents the ground truth of the 2D feature point, and $\mathbf{w}_k^{c_i}$ subjects to zero-mean Gaussian noise $\mathcal{N}(0, \mathbf{\Sigma}_k^{c_i})$. Therefore, $\mathbf{w}_k = [\mathbf{w}_k^{c_1\top}, \mathbf{w}_k^{c_2\top}, ..., \mathbf{w}_k^{c_N\top}]^\top \sim \mathcal{N}(0, \mathrm{diag}(\mathbf{\Sigma}_k^{c_1}, \mathbf{\Sigma}_k^{c_2}, ..., \mathbf{\Sigma}_k^{c_N}))$. In practice, we set $\mathbf{\Sigma}_k^{c_i} = \sigma_c \mathbf{I}_{2\times 2}$ for all feature points, and $\sigma_c = 1.5$.

B. Plane Matching

Plane-constrained bundle adjustment is also used in the extrinsic calibration, and the corresponding plane in the LiDAR frame needs to be extracted. We assume that the sensor suite shares common FoV, and the rough initial extrinsics are given manually. Based on the previous section on which the plane is targeted, we project the point cloud into the image and extract the projected points near the feature pixels extracted in SfM. As we know which feature pixels lie on the target plane, the 3D points in LiDAR frame that lie on the target plane can be clustered. Then the plane coefficient is estimated by the RANSAC plane-fitting algorithm. To reduce the influence of measurement noise, we find all points close to the estimated plane from raw data and repeatedly use plane RANSAC to derive the final corresponding plane $\mathbf{\Pi}^{l_i} = [\mathbf{n}^{l_i\top}, (\bar{\mathbf{p}}^{l_i})^\top]^\top$.

When the sensors share no common FoV, the plane correspondences can also be established with other tricks. The first method is to simplify our calibration scene and let both sensors toward a single plane, for instance, the ground. If in a complicated scenario, we can estimate LiDAR trajectory first [22]. Then the extrinsics can be initialized by conducting hand-eye calibration [10] or given manually, and the correspondences can be established using the nearest neighbor search (NNS) strategy. However, this complicated situation is not discussed in this paper.

C. Extrinsic Calibration

1) Initialization: For the initial extrinsic parameters may be inaccurate and the optimization is non-convex, a coarse-to-fine fashion is designed for the extrinsics estimation. In the camera frame, as stated before, the plane $\mathbf{\Pi}^w$ is extracted from 3D environment using RANSAC in world coordinate system. Since we know the camera poses, the same plane in the local coordinate system for the i-th frame can be calculated and represented by $\mathbf{\Pi}^{c_i} = [\mathbf{n}^{c_i\top}, (\bar{\mathbf{p}}^{c_i})^\top]^\top$. Likewise, the plane in the LiDAR frame is represented by $\mathbf{\Pi}^{l_i} = [\mathbf{n}^{l_i\top}, (\bar{\mathbf{p}}^{l_i})^\top]^\top$. The geometric relations of multiple planes constrain the extrinsic parameters uniquely. Specifically, the plane normal vectors from both sensors should be equal after transformation. Additionally, the plane's center point $\bar{\mathbf{p}}^{l_i}$ should lie on the plane $\mathbf{\Pi}^{c_i}$ after transformation, which can be formulated as

$$\mathbf{R}_l^c \mathbf{n}^{l_i} = \mathbf{n}^{c_i}, \tag{9}$$

$$\mathbf{n}^{c_i\top}((\mathbf{R}_l^c \bar{\mathbf{p}}^{l_i} + \mathbf{t}_l^c) - s\bar{\mathbf{p}}^{c_i}) = 0, \tag{10}$$

where s denotes the global scale factor, and $\mathbf{T}_l^c = (\mathbf{R}_l^c, \mathbf{t}_l^c) \in SO(3) \times \mathbb{R}^3$ is the transformation from LiDAR to camera coordinate system. The rotation matrix in equation (9) has a closed-form solution [23] based on SVD. By taking $\mathbf{R}_l^c$ into equation (10), translation and scale can be derived as

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \begin{bmatrix} s \\ \mathbf{t}_l^c \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}, \tag{11}$$

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{n}^{c_i\top}\mathbf{p}^{c_i} & -\mathbf{n}^{c_i\top} \end{bmatrix}, b_i = \mathbf{n}^{c_i\top}\mathbf{R}_l^c \mathbf{p}^{l_i}. \tag{12}$$

Therefore, the translation and scale factor are derived by $[\widetilde{s}, \widetilde{\mathbf{t}_l^c}^\top]^\top = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$.

2) Refinement: Similar to the camera poses' refinement, the extrinsics are also optimized by PBA. The plane we used in camera poses estimation is extracted from $\mathcal{P}^w$ in camera frame, while the plane used in extrinsic calibration is the plane extracted from the corresponding point cloud in the LiDAR frame. Therefore, the 3D cross-point $\mathbf{p}_k^{c_i}$ associated with the 2D feature point $\mathbf{u}_{c_i,k}$ is defined by the following equation

$$\mathbf{p}_k^{c_i} = f(\mathbf{T}_l^c, \mathbf{\Pi}^{l_i}), \tag{13}$$

All parameters containing camera poses $\mathcal{C}$ and the extrinsics $\mathbf{T}_l^c$ are simultaneously optimized with the following objective function

$$\hat{\mathbf{T}}_l^c, \hat{\mathcal{C}} = \arg\min_{\mathbf{T}_l^c, \mathcal{C}} \sum_{i=1}^{N}\sum_{j\neq i}^{N}\sum_{k=1}^{N_{i,j}} \|\epsilon_k\|^2, \tag{14}$$

$$\epsilon_k = \frac{1}{z_k}\mathbf{K}(\mathbf{R}_{c_i}^{c_j}\mathbf{p}_k^{c_i} + \widetilde{s}\mathbf{t}_{c_i}^{c_j}) - \mathbf{u}_k^{c_j}. \tag{15}$$

This loss function is solved iteratively, and our MLE estimator is applied. The measurement noise comes from the 2D feature detection in the camera frame and the plane extraction in the LiDAR frame. Benefiting from [24], we can represent and calculate the covariance of plane coefficients by $\mathbf{\Sigma}_{\mathbf{n},\bar{\mathbf{p}}}^{l_i} \in \mathbb{R}^{6\times 6}$. Then $\mathbf{w_k} = [\mathbf{w}_{\mathbf{n},\bar{\mathbf{p}}}^{l_i\top}, \mathbf{w}_k^{c_1\top}, ..., \mathbf{w}_k^{c_N\top}]^\top \in \mathbb{R}^{(6+2N)\times 1}$, and it subjects to $\mathcal{N}(0, \mathrm{diag}(\mathbf{\Sigma}_{\mathbf{n},\bar{\mathbf{p}}}^{l_i}, \mathbf{\Sigma}_k^{c_1}, ..., \mathbf{\Sigma}_k^{c_N}))$. The final state covariance $\mathbf{\Xi_{xx}} = \mathbf{\Lambda_{xx}}^{-1}$.

For the camera poses are not our interest, parameter reduction is required to find the uncertainty of the extrinsics. Specifically, the overall information matrix is represented by

$$\mathbf{\Lambda_{xx}} = \begin{bmatrix} \mathbf{\Lambda_{TT}} & \mathbf{\Lambda_{TC}} \\ \mathbf{\Lambda_{CT}} & \mathbf{\Lambda_{CC}} \end{bmatrix}, \tag{16}$$

where $\mathbf{\Lambda_{CT}} = \sum_k \mathbf{J}_{k,\mathcal{C}}(\mathbf{J}_{\mathbf{w}_k}^\top \mathbf{\Sigma}_k \mathbf{J}_{\mathbf{w}_k})^{-1}\mathbf{J}_{k,\mathbf{T}}^\top$. Using the tool in [20], the reduced information matrix of the extrinsics is $\overline{\mathbf{\Lambda}}_{\mathbf{TT}} = \mathbf{\Lambda_{TT}} - \mathbf{\Lambda_{TC}}\mathbf{\Lambda_{CC}}^{-1}\mathbf{\Lambda_{CT}}$. Therefore, the covariance matrix of the extrinsics is represented by $\mathbf{\Xi_{TT}} = \overline{\mathbf{\Lambda}}_{\mathbf{TT}}^{-1}$.

In the real application, the plane extraction in LiDAR may be inaccurate, and the error may be propagated into camera poses refinement. In this case, we fix the camera pose and only update the extrinsics and scale factor. The residual function is reformulated as

$$\hat{\mathbf{T}}_l^c, \hat{s} = \arg\min_{\mathbf{T}_l^c, s} \sum_{i=1}^{N}\sum_{j\neq i}^{N}\sum_{k=1}^{N_{i,j}} \|\epsilon_k\|^2. \tag{17}$$

D. Validity Analysis

The calibration problem will degenerate with improper data distribution. Validity analysis is needed before calibration. In the coarse stage, the extrinsics are initialized with a closed-form solution. The rotation matrix contains 3 degrees of freedom (DoF), which means at least three independent constraints are required to find a unique solution. In other words, three non-coplanar normal vectors are required. For translation and scale factor solved in equation (11), the matrix $\mathbf{A} \in \mathbb{R}^{N\times 4}$ needs full rank to find unique parameters with 4 DoF, and at least four poses are needed. Considering the minimum requirement with four poses collected, we suppose that the first three poses are non-coplanar, and they intersect at the point $\bar{\mathbf{p}}_0$. The first three planes are represented by $[\mathbf{n}^{c_1\top}, \bar{\mathbf{p}}_0^\top]^\top, [\mathbf{n}^{c_2\top}, \bar{\mathbf{p}}_0^\top]^\top, [\mathbf{n}^{c_3\top}, \bar{\mathbf{p}}_0^\top]^\top$, and the fourth plane is denoted by $[\mathbf{n}^{c_4\top}, (\bar{\mathbf{p}}^{c_4})^\top]^\top$. If $\mathbf{A}$ is rank-deficient, the column vectors are dependent, and we can get the following combination

$$\mathbf{A}_1 = \alpha\mathbf{A}_2 + \beta\mathbf{A}_3 + \gamma\mathbf{A}_4, \tag{18}$$

which can be expanded as

$$\mathbf{n}^{c1\top}\bar{\mathbf{p}}_0 = \alpha\mathbf{n}^{c2\top}\bar{\mathbf{p}}_0 + \beta\mathbf{n}^{c3\top}\bar{\mathbf{p}}_0 + \gamma\mathbf{n}^{c4\top}\bar{\mathbf{p}}^{c_4}, \tag{19}$$

$$\mathbf{n}^{c_1} = \alpha\mathbf{n}^{c_2} + \beta\mathbf{n}^{c_3} + \gamma\mathbf{n}^{c_4}. \tag{20}$$

By transposing the vector in equation (20) on both sides and multiplying $\bar{\mathbf{p}}_0$, we have

$$\mathbf{n}^{c_1\top}\bar{\mathbf{p}}_0 = \alpha\mathbf{n}^{c_2\top}\bar{\mathbf{p}}_0 + \beta\mathbf{n}^{c_3\top}\bar{\mathbf{p}}_0 + \gamma\mathbf{n}^{c_4\top}\bar{\mathbf{p}}_0. \tag{21}$$

Subtracting equation (21) from equation (19) on both sides, we get

$$\mathbf{n}^{c_4\top}\bar{\mathbf{p}}_0 = \mathbf{n}^{c_4\top}\bar{\mathbf{p}}^{c_4} \Leftrightarrow \mathbf{n}^{c_4\top}(\bar{\mathbf{p}}_0 - \bar{\mathbf{p}}^{c_4}) = 0, \tag{22}$$

which indicates $\mathbf{p}_0$ also lies on the fourth plane. Therefore, in degenerated cases, all four planes intersect at one point. As a result, we need *1)* at least four poses; *2)* they do not intersect at the same point; *3)* at least three normal vectors are non-coplanar. In practice, we introduce a *confidence factor* τ to determine whether the collected data is sufficient to find a unique solution. Only when $\tau = \frac{\lambda_4}{\lambda_1} > 4\times 10^{-5}$, the dataset is considered as a valid collection, where $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ are the eigenvalues of matrix $\mathbf{A}^\top\mathbf{A}$ and $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$.

V. Experiments

Our proposed algorithm is verified through simulation, real-world, and comparison experiments. The simulation (as shown in Fig. 4b) is built on Gazebo [25] and mimics the real-world sensor suite configuration, containing one dense LiDAR (Livox mid-70) and one frame camera. A traffic mat is placed on the ground to supply enough texture information. In real-world experiments, we collect four different scenes to evaluate the performance of our algorithm, and the sensor suite (as shown in Fig. 4a) consists of a dense LiDAR (Livox mid-70) and a camera (SENSING-GSML-0143-H090, with the resolution of 1280×720, and the horizontal FoV of 90°). During data acquisition, the camera and LiDAR are placed statically in a position. Image and point cloud are captured simultaneously. To fully utilize the advantage of non-repetitive scanning pattern of Livox, point cloud is accumulated for 5 seconds to get the dense representation of the environment.

A. Simulation

In the simulated environment with the ground truth transformation $\mathbf{T}_{gt}$, we compute the transformation error with our estimated transformation $\hat{\mathbf{T}}$ as our evaluation metric. The transformation error contains rotation and translation error

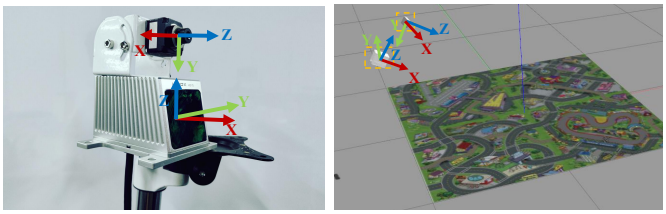
$$e_{\mathbf{R}} = \|(\log(\mathbf{R}_{gt}^\top\hat{\mathbf{R}}))^\vee\| = \|(\log\delta\mathbf{R})^\vee\|, \tag{23}$$

$$e_{\mathbf{t}} = \|\mathbf{t}_{gt} - \hat{\mathbf{t}}\| = \|\delta\mathbf{t}\|, \tag{24}$$

where $()^\vee$ converts a skew-symmetric matrix to a 3×1 vector. Three different levels of zero-mean Gaussian noise $\mathcal{N}(0, k\sigma^2)$ are added to the LiDAR range measurements and camera data, where $k \in \{1, 2, 3\}$. Specifically, the standard deviation of noise in image is set to $\sigma_c = 0.007$, and $\sigma_l = 0.01$m in LiDAR range measurements.

The proposed algorithm only takes the initial extrinsics to find the corresponding planes. No matter how we change the initial extrinsic parameters, as long as the correct correspondences are built, the extrinsics after initialization would stay close and cause few effects in the final optimization. Therefore, instead of manipulating the given initial extrinsics, we randomly selected $N \in \{4, 5, ..., 10\}$ frames from the collected 12 pairs of images and point clouds to evaluate the performance with different data distributions. The calibration is repeated 50 times.

Fig. 5 shows the error distribution. With more data involved, the calibration performs better in terms of accuracy and stability, and it reaches $[0.13, 0.15, 0.2]$ degree in rotation and $[0.5, 0.9, 0.8]$ cm in translation in different noise levels.



(a) (b)

Fig. 4. The sensor suite in real world and simulation. In simulation, we bind these two sensors as a sensor suite and take Livox as the reference coordinate system. Specifically, the extrinsic parameters' setting is $[x, y, z] = [0.1, 0.3, 0.2]$m, and $[roll, pitch, yaw] = [0, \pi/40, \pi/20]$.

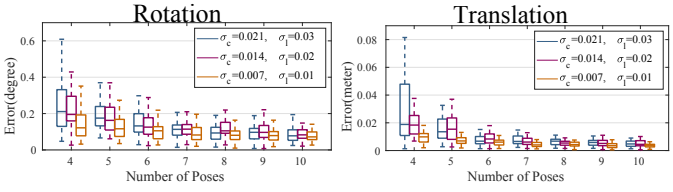


Fig. 5. The quantitative results in simulation. Different levels of Gaussian noise $\mathcal{N}(0, k\sigma)$ are added into images and point clouds, where $k \in \{1, 2, 3\}$. Specifically, standard deviation $\sigma_c = 0.007$ is set in image and $\sigma_l = 0.01$m in LiDAR range measurements.

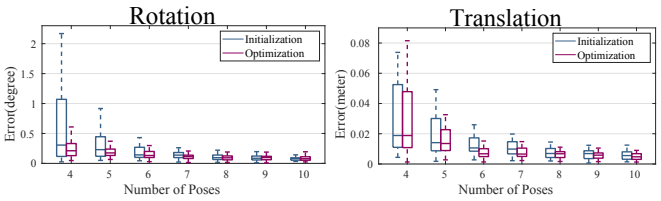


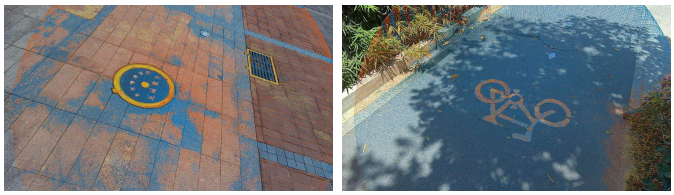
Fig. 6. Ablation study. The calibration results after initialization and optimization using the simulated data. The zero-mean Gaussian noise is added into images and point clouds with standard deviation $\sigma_c = 0.021$, $\sigma_l = 0.03$m, respectively.

To evaluate the effect of PBA, the ablation study is performed. The calibration results after initialization and optimization are shown in Fig. 6. It is seen that the PBA greatly reduces the variance and increases the accuracy in rotation, especially when the number of poses is less than 6.

B. Real-world Experiments

1) Qualitative and quantitative results: Although we add noise to raw data in the previous simulation environment, many differences still exist with the real world. Specifically, the ground may not be strictly flat, and the texture is not rich, as supported in the simulation. Moreover, the bleeding points (as described in [4]) and connected planes reduce the stability of extracting plane coefficients. Therefore, real-world experiments are required to validate the proposed algorithm further.

Based on the previous simulation result, the accuracy curve becomes smooth when the number of poses reaches around 7. Therefore, we randomly choose 7 frames from the collected 12 pairs of real-world data in each scene. Fig. 7 shows the qualitative results in 4 different scenes. Two scenes (scene 1 and scene 2) capture the texture on the ground, and two scenes (scene 3 and scene 4) capture the texture on the vertical



(a) Scene 1: the manhole cover on the road

(b) Scene 2: the bicycle sign on the ground

(c) Scene 3: the whiteboard with graffiti

(d) Scene 4: the banner pasted on a kiosk

Fig. 7. 4 valid calibration scenes, and the qualitative results in real world. The projected points in (a)(b) are colorized by points' intensity value. In (c)(d), different colors represent different planes, which are iteratively extracted by plane RANSAC.

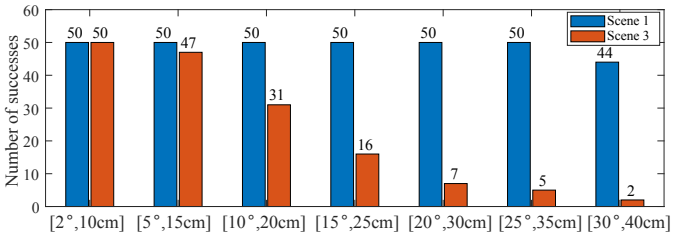


Fig. 8. The success rate. We take the ground truth extrinsics from ACSC method (24 poses, see Table II). 7 different levels of transformation error are added into the ground truth to see the effect of the manually given extrinsics. Specifically, [2°,10cm] means 2 degrees error in rotation and 10cm error in translation. In each setup, 50 test runs are conducted.

plane. Fig. 9 shows the distribution of the extrinsics in each axis. It is seen that the median values of the extrinsics among different scenes are converged into almost the same value, which suggests the consistency of our algorithm. However, compared with simulation, the variance in each axis is larger because of sparser feature points, connected planes, and the restricted orientation of the sensor suite.

In our method, we find the extrinsics via plane features, which are the principal component of the point cloud. In this way, our calibration is less sensitive to the initial parameters. To illustrate this, an experiment is conducted to see the success rate of our calibration in different scenes, shown in Fig. 8. Seven different levels of manually given extrinsic error are added, which are [2°,10cm], [5°,15cm], [10°,20cm], [15°,25cm], [20°,30cm], [25°,35cm], [30°,40cm]. Specifically, we take the extrinsics from ACSC method (24 poses, see Table II) as the ground truth, and [2°,10cm] means 2 degrees error in rotation and 10cm error in translation. The rotation axis and translation error direction are sampled uniformly on the surface of a sphere. In each setup, 50 test runs are conducted. The calibration will be considered successful when rotation and translation error is less than 0.5°and 5cm, respectively. As we can see in Fig. 8, the tolerance in scene 1 could be [30°, 40cm] or even higher.

2) Bad scenes: As detailed in the previous sections, the camera poses are restored using PBA, which requires the textured plane. Moreover, in the matching stage, the target plane is located by plane RANSAC, and this plane contains most feature points. For these requirements, the calibration may fail in some calibration scenes, like the scenes shown in Fig. 10. The background in scene 5 is complicated. The arrows and zebra crossings on the ground can also be recovered by SfM, which will affect the target plane selection. If the ground is selected as the target plane, the planes' distribution

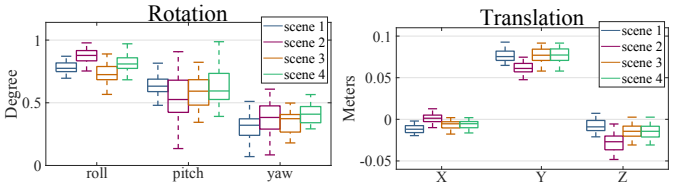
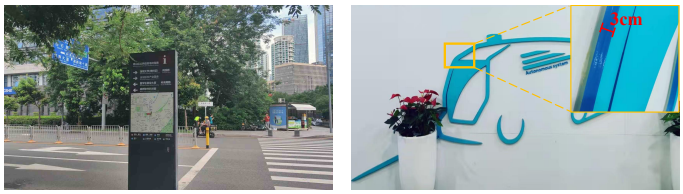
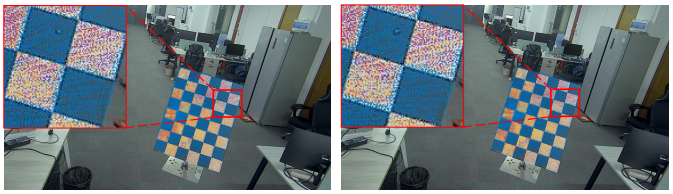


Fig. 9. The distribution of the extrinsic parameters in four different scenes. The nominal part has been removed.



(a) Bad scene 5 (b) Bad scene 6.

Fig. 10. Bad calibration scenes. The zoom-in picture in (b) is taken from the side view.



(a) ACSC [14] (b) Ours

Fig. 12. The projection results of both methods, which shows that our method achieves comparable accuracy with ASCS.

may not be good enough. Meanwhile, many cars pass on the road, which causes the accumulation of 3D points of cars into the final dense point cloud, and further makes the plane matching inaccurate. Note that the background of scene 4 is also complicated, but the feature points on the target plane are much denser. In scene 6, the text is above the plane, which causes bias in translation, but we can manually eliminate it.

C. Comparison Experiments

We compare our method with [14], [4]. ACSC [14] uses standard checkerboard to calibrate the extrinsics by aligning inner pattern corners and solves the PnP problem. Delicate data preprocessing is designed, and the intensity channel in the point cloud is utilized to extract inner pattern corners. As the checkerboard is also a plane with texture, we modify our camera pose estimation algorithm and use the checkerboard to calibrate. Based on the guideline of the ASCS open source code, 12 pairs of data are required to ensure robustness. Similarly, we collect 24 pairs of data and randomly select 12 frames to repeatedly conduct calibration on both methods. Fig. 11 shows the distribution of the extrinsics and the scale factor after repeating 20 times. ACSC performs a little better in terms of stability, but the variance of both methods is small enough to get accurate extrinsic parameters. On the other hand, we do not need the pattern size of the checkerboard, and the scale factor is estimated accurately, as shown in Fig. 11. The projection result is shown in Fig. 12 using both methods. The extrinsics are chosen from the median value on each axis. We can barely find the difference between each other.

Yuan's [4] method utilizes the environment depth-continuous edges as the feature to calibrate, which is quite different from ours. As the perfect ground truth of transformation is unavailable in the real world, we take the extrinsics calibrated from ACSC as a comparison and use all poses (24 poses). To fairly compare with [4], we use [4] to calibrate

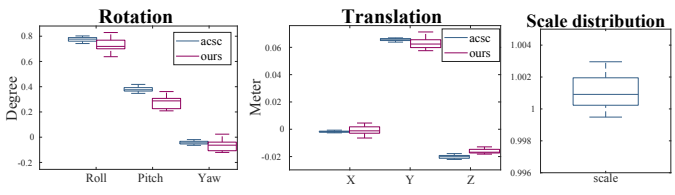
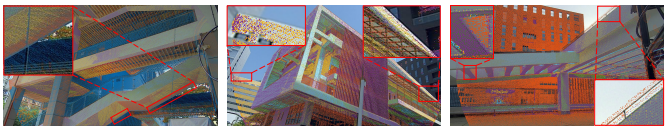


Fig. 11. The distribution of the extrinsic parameters and scale factor using our method and ACSC. In contrast, our method does not require priori information of the checkerboard size.

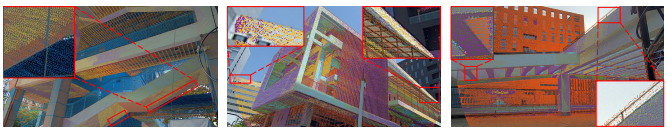
three times using their open-sourced code with $1, 4$ and 7 poses, respectively. Similarly, our method is conducted twice as well with 4 and 7 poses respectively. Table II shows the final result. It indicates that Yuan's method and our method are both accurate on rotation, but our method outperforms Yuan's method in translation. Fig 13 presents the projection result using both methods. We can find the mismatches marked with the red square in Fig. 13a. The edges in this area are not observable in Yuan's method because they are depth-discontinuous edges. In contrast, the projection in our method looks better. Table II summarizes the extrinsics parameters using different methods. As ACSC extracted inner pattern corners in both camera and LiDAR data, we use these features to calculate the mean projection error (MPE) based on the 2D-3D correspondences

$$\text{MPE} = \frac{1}{M}\sum_{i=1}^{M} \|\pi(\mathbf{p}_i^l) - \mathbf{u}_i^c\|. \tag{25}$$

It is seen that our method outperforms Yuan's method when both use 4 poses in terms of MPE, which is consistent with the projection result in Fig. 13.



(a) Yuan [4], 4 poses



(b) Ours, 4 poses

Fig. 13. The qualitative comparison using Yuan's approach and our method. The projected points are colorized by points' intensity value. In our method, we use scene 2 (Fig. 7b), scene 1 (Fig. 7a), scene 3 (Fig. 7c) to calibrate and take the recovered extrinsic to project the point cloud into images in (b) respectively.

VI. Conclusion

In this paper, we proposed *PBACalib*, a novel targetless extrinsic calibration algorithm for dense LiDAR-camera system based on plane-constrained bundle adjustment. By taking the uncertainty of the measurement into optimization, an MLE estimator is designed to iteratively find the optimal solution, and the covariance of the extrinsics is given. To ensure the success

TABLE II
THE CALIBRATION RESULT USING REAL DATA. WE TAKE THE RESULT OF ACSC METHOD WITH 24 POSES AS THE GROUND TRUTH, MARKED IN RED. THE BEST RESULT IS MARKED IN BOLD. ↓ INDICATES THAT THE LOWER THE VALUE, THE BETTER THE PERFORMANCE.

Methods	Rotation (degree)			Translation (cm)			
	Roll	Pitch	Yaw	X	Y	Z	MPE↓
ACSC [14] (24 poses)	88.77	0.58	88.46	-0.18	6.57	-2.02	0.4
Yuan [4] (1 pose)	88.80	0.57	88.35	-5.44	8.23	8.32	19
Yuan [4] (4 poses)	88.56	0.61	88.46	-1.15	3.52	3.86	7.3
Ours (4 poses)	88.70	0.62	88.41	-1.38	8.14	-1.06	6.9
Yuan [4] (7 poses)	88.89	0.52	88.52	-0.3	7.53	3.84	4.5
Ours (7 poses)	88.75	0.76	88.41	-0.69	6.66	-2.06	**1.3**

of the calibration, we analyzed the validity of the collected data and set a confidence factor to determine if it is sufficient to make the extrinsics unique. Finally, simulation, various real-world and comparison experiments indicated that the proposed method is robust and outperforms another targetless method and achieves similar accuracy compared with the target-based method.

ACKNOWLEDGMENT

The authors would like to thank Prof. Ming Liu and Dr. Jianhao Jiao from Hong Kong University of Science and Technology for insightful suggestions on this work. The authors would also like to thank Unity Drive Inc. for providing essential experimental devices and services for this work.

REFERENCES

[1] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5562–5569.

[2] S. A. Rodriguez F., V. Fremont, and P. Bonnifait, "Extrinsic calibration between a multi-layer lidar and a camera," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008, pp. 214–219.

[3] J. Kümmerle and T. Kühner, "Unified intrinsic and extrinsic camera and lidar calibration under uncertainties," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6028–6034.

[4] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517–7524, 2021.

[5] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: science and systems*, vol. 2, no. 7. Berlin, Germany, 2013.

[6] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[7] Y. Xie, R. Shao, P. Guli, B. Li, and L. Wang, "Infrastructure based calibration of a multi-camera and multi-lidar system using apriltags," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 605–610.

[8] J. Beltrán, C. Guindel, A. de Escalera la, and F. García, "Automatic extrinsic calibration method for lidar and camera sensor setups," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2022.

[9] B. Nagy, L. Kovács, and C. Benedek, "Online targetless end-to-end camera-lidar self-calibration," in *2019 16th International Conference on Machine Vision Applications (MVA)*, 2019, pp. 1–6.

[10] R. Horaud and F. Dornaika, "Hand-eye Calibration," *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, June 1995.

[11] M. Horn, T. Wodtko, M. Buchholz, and K. Dietmayer, "Online extrinsic calibration based on per-sensor ego-motion using dual quaternions," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 982–989, 2021.

[12] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1215–1229, 2016.

[13] J. Wu, Y. Zheng, Z. Gao, Y. Jiang, X. Hu, Y. Zhu, J. Jiao, and M. Liu, "Quadratic pose estimation problems: Globally optimal solutions, solvability/observability analysis, and uncertainty description," *IEEE Transactions on Robotics*, pp. 1–22, 2022.

[14] J. Cui, J. Niu, Z. Ouyang, Y. He, and D. Liu, "Acsc: Automatic calibration for non-repetitive scanning solid-state lidar and camera systems," 2020.

[15] Y. Zhu, C. Zheng, C. Yuan, X. Huang, and X. Hong, "Camvox: A low-cost and accurate lidar-assisted visual slam system," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5049–5055.

[16] B. Zeisl and M. Pollefeys, "Structure-based auto-calibration of rgb-d sensors," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5076–5083.

[17] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-lidar calibration: A targetless and structureless approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1556–1563, 2020.

[18] S. Kim and R. Manduchi, "Multi-planar monocular reconstruction of manhattan indoor scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[19] H. P. Gavin, "The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems," *Department of Civil and Environmental Engineering, Duke University*, vol. 19, 2019.

[20] W. Förstner and B. P. Wrobel, *Photogrammetric computer vision*. Springer, 2016.

[21] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] X. Liu, C. Yuan, and F. Zhang, "Targetless extrinsic calibration of multiple small fov lidars and cameras using adaptive voxelization," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[23] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[24] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8518–8525, 2022.

[25] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.