YOCO: You Only Calibrate Once for Accurate Extrinsic Parameter in LiDAR-Camera Systems

Tianle Zeng, Dengke He, Feifan Yan, Meixi He

***Abstract*—In a multi-sensor fusion system composed of cameras and LiDAR, precise extrinsic calibration contributes to the system's long-term stability and accurate perception of the environment. However, methods based on extracting and registering corresponding points still face challenges in terms of automation and precision. This paper proposes a novel fully automatic extrinsic calibration method for LiDAR-camera systems that circumvents the need for corresponding point registration. In our approach, a novel algorithm to extract required LiDAR correspondence point is proposed. This method can effectively filter out irrelevant points by computing the orientation of plane point clouds and extracting points by applying distance- and density-based thresholds. We avoid the need for corresponding point registration by introducing extrinsic parameters between the LiDAR and camera into the projection of extracted points and constructing co-planar constraints. These parameters are then optimized to solve for the extrinsic. We validated our method across multiple sets of LiDAR-camera systems. In synthetic experiments, our method demonstrates superior performance compared to current calibration techniques. Real-world data experiments further confirm the precision and robustness of the proposed algorithm, with average rotation and translation calibration errors between LiDAR and camera of less than 0.05° and 0.015m, respectively. This method enables automatic and accurate extrinsic calibration in a single one step, emphasizing the potential of calibration algorithms beyond using corresponding point registration to enhance the automation and precision of LiDAR-camera system calibration.**

***Index Terms*—Sensor Fusion, Extrinsic Calibration, Point Cloud, 3-D Light Detection and Ranging(LiDAR).**

I. Introduction

SENSOR fusion has been widely discussed in the robotics and computer vision fields. The complementary characteristics between sensors can capitalize on the advantages of different sensors and compensate for the shortcomings, which makes the multi-sensor system have higher accuracy and robustness than the single-sensor system. Cameras provide a cost-effective solution for perception systems, offering high-resolution color images with deep-learning capabilities for tasks like object detection and semantic segmentation [1], [2]. However, they lack direct depth measurement and are less efficient in low-light environments. 3D Light Detection And Ranging(LiDAR) sensors generate accurate point clouds independently of ambient light, overcoming these limitations. Although LiDAR point clouds can be sparse with lower refresh rates, combining LiDAR and camera measurements compensates for their respective shortcomings. The success of sensor fusion critically depends on the precise calibration of extrinsic parameters. These parameters are primarily computed through the extraction and registration of corresponding points between the camera and LiDAR sensor. Therefore, the essence of LiDAR-camera system calibration lies in establishing and registering corresponding relationships among co-observable points.

However, both establishing corresponding relationships and registering correspondence points are highly challenging tasks. In establishing corresponding relationships, main approaches involve identifying corresponding features between cameras and LiDAR sensors. Previous studies have concentrated on geometric features as well as artificial features [3]. Typically, detecting features in 2D images is straightforward, but identifying and extracting corresponding 3D feature points in the LiDAR point cloud presents a more complex challenge. This is because 2D images contain rich semantic and color information, while point clouds often only consist of coordinates in 3D space. These sparsely distributed point clouds make it difficult to find suitable features. To solve this problem, a general approach is to select corresponding points manually [4]. Once feature points, such as corner points, are detected in images, manual selection and extraction of matching points in LiDAR point clouds are performed. For manually selected correspondence points, point cloud registration is relatively straightforward, with common methods including point-to-point, point-to-line, line-to-line, and line-to-plane registration [5], [6], [7], [8]. Due to the relative accuracy of manually selected correspondence points, high-precision extrinsic parameters can be obtained through point cloud registration. However, manual selection of correspondence points entails significant human intervention, leading to a complex, cumbersome, and time-consuming calibration process.

In recent years, automated methods for selecting and extracting corresponding points have gained prominence in the

Tianle Zeng is with the State Key Labortaory for Fine Exploration and Intelligent Development of Coal Resources, China University of Mining and Technology (Beijing), Beijing 100083, China, and also with the College of Geoscience and Surveying Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China (e-mail: zqt2200202081@student.cumtb.edu.cn).
Dengke He is with the State Key Labortaory for Fine Exploration and Intelligent Development of Coal Resources, China University of Mining and Technology (Beijing), Beijing 100083, China, and also with the College of Geoscience and Surveying Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China (e-mail: he_dengke@126.com).
Feifan Yan is with the State Key Labortaory for Fine Exploration and Intelligent Development of Coal Resources, China University of Mining and Technology (Beijing), Beijing 100083, China, and also with the College of Geoscience and Surveying Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China (e-mail: yabiyff@163.com).
Meixi He is with the State Key Labortaory for Fine Exploration and Intelligent Development of Coal Resources, China University of Mining and Technology (Beijing), Beijing 100083, China, and also with the College of Geoscience and Surveying Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China (e-mail: zqt2200202050@student.cumtb.edu.cn).

field, aimed at reducing manual intervention and enhancing calibration efficiency [9]. These methods primarily involve identifying common geometric features and fitting various geometric shapes of different dimensions, such as 1-D lines, 2-D planes, and 3-D spheres, in both images and LiDAR point clouds [10], [11]. However, while the automated extraction method of correspondence points minimizes human intervention, it brings challenges to subsequent point cloud registration, particularly in terms of time and precision. The time-related challenge arises due to the limited availability of point clouds meeting the feature point extraction requirements, necessitating the collection of a large amount of input data. Consequently, processing this large volume of data prolongs the algorithm execution times. Moreover, learning-based methods require extensive data collection and prolonged training time to achieve satisfactory results. Additionally, the precision-related challenge stems from the fact that automatically extracted corresponding feature points may not guarantee complete accuracy. This can lead to misalignment issues that affect calibration accuracy. Furthermore, the extensive demand for input data in the point cloud registration process further leads to more misaligned points being used for registration, ultimately impacting accuracy. Although several methods [12], [13] have been proposed to optimize the processes of point cloud extraction and registration, they still cannot fundamentally resolve the issue.

Motivated by these challenges, this paper introduces a novel calibration method that automatically extracts corresponding points and circumvents the point cloud registration process through innovative extrinsic calculation, addressing challenges in time and precision. This approach minimizes errors caused by mismatched corresponding points, maximizing point cloud information utilization, reducing calibration input data requirements, and enhancing precision and speed. The proposed method is straightforward to implement, requiring minimal data for high-precision calibration without manual intervention. Extensive simulations and real-world experiments on various LiDAR-camera setups, including publicly available datasets, demonstrate the versatility and effectiveness of our approach. The main contributions of this study are summarized as follows:

1)We propose a novel method for extrinsic parameter calculation that eliminates the necessity of registering correspondence points between cameras and LiDAR sensors, this offers a fresh perspective and technical solution to the conventional calibration process.

2)We introduce a novel plane extraction method for detecting and extracting correspondence plane point clouds, which provides valuable insight for plane extraction scenarios involving prior knowledge of spatial geometry.

3)We present a fully automatic calibration pipeline between LiDAR and camera systems. This process has a simple requirement for calibration data collection and can be applied for the offline calibration of various LiDAR and camera configurations, The source code will be available at https://github.com/louiszengCN/lidar_camera_auto_calibration.

The remainder of this paper is organized as follows. We briefly survey the field of LiDAR-camera system calibration in Section II. An overview of the proposed method is given in Section III. After that, the proposed methodology is presented in Sections IV. In Section V we evaluate our method with simulation and real-world experiments. Finally, we conclude our work in Section VI.

II. Relative work

Extrinsic calibration techniques are commonly classified into two main categories: target-based methods and targetless methods. Each of these approaches has been thoroughly researched and implemented in practice, with a detailed examination of their respective strengths and drawbacks.

A. Target-Based Methods

The target-based method has received extensivee research attention, covering manual, semi-automatic, fully automatic methodologies. It is extensively utilized in the calibration of cameras and LiDAR systems. Zhang et al. [14] use a checkerboard as a planar geometry feature and then extract this feature both in camera and laser point to calibrate the lidar and camera. Zhao et al. [15] project points in laser and cameras to a special coordinate system to calculate the ridge body transformation matrix between laser and camera. Gong et al. [16] extract trihedron corner points from laser and camera data and calibrate the extrinsic parameter by matching the corner points. However, these approaches necessitate manual extraction of feature points in the LiDAR point cloud. Methods requiring manual involvement have a high level of calibration precision, but it often has specific data acquisition requirements, limited applicability, and complexities in the calibration process, demanding significant time and cost investments.

There are methods that do not require manual intervention and can automatically extract LiDAR point cloud data for calibration. Geiger et al. [17] employ Principal Component Analysis (PCA) to compute normal vectors for each data point within the point cloud. Subsequently, they utilize a greedy region-growing process to segment the chessboard plane. Pandey et al. [18] use a checkerboard pattern as the co-observable features in both camera and laser data and apply Random Sampling Consensus (RANSAC) plane fitting algorithm to extract point clouds of plane features in LiDAR data. Toth et al. [19] use Spheres as a calibration target, employing the RANSAC algorithm to detect the inlier points and formulating a least-squares optimization problem based on the radius constraint to extract the sphere point cloud from the LiDAR data.

Relative to targetless methods, target-based calibration is typically characterized by higher precision and robustness, with the added advantage of traceable calibration errors. This makes it particularly well-suited for achieving precise calibration within controlled settings. Consequently, the method proposed in this paper employs a target-based approach, utilizing a printed checkerboard as the calibration target.

B. Targetless-Based Methods

The targetless method [20] endeavors to identify natural patterns, predominantly lines or orthogonal features, within

the scene. Subsequently, it formulates these patterns using geometric constraints to deduce the extrinsic parameters. This methodology encompasses two primary approaches: the static-based method and motion-based methods.

For the static-based methodologies, it is assumed that the camera remains stationary or moves slowly relative to the scene. Key features such as lines, corners, or other discernible patterns are detected within the scene, and their geometric attributes are leveraged to estimate the camera's pose. Techniques such as structure-from-motion (SfM) or simultaneous localization and mapping (SLAM) can be applied within this framework. Nie et al. [21] presented a novel approach for dual LiDAR calibration based on adaptive surface normal estimation. Zhang et al. [22] introduced a method that eliminates the need for overlapping field of view (FOV) between LiDAR and camera. Instead, external parameters are obtained from odometry estimation conducted independently by the sensor. Additionally, recent advancements in deep learning techniques have been utilized for targetless multisensor calibration. Ye et al. [23] proposed a network for 2D-3D pose estimation based on keypoints. This approach embeds an optimizer based on geometric constraints into an end-to-end network and employs a trainable point weighting layer to extract sparse corresponding points for calibration. Zhu et al. [24] trained a cross-modal graph neural network (GNN) for the purpose of calibrating a LiDAR-Camera system. They utilize PointNet and PointNet++ models, which are integral for extracting feature points from point cloud data. Motion-based methods exploit the motion of the camera or sensor to estimate its pose. By analyzing the apparent motion of features in the scene over time, the camera's trajectory and orientation can be inferred. Optical flow, visual odometry, or feature tracking algorithms are commonly used in motion-based methods.

Nevertheless, these methods encounter specific challenges. Firstly, ensuring the generalization ability of calibration results across various scenarios proves difficult due to the variability and diversity of data within different scenes, which can influence calibration outcomes. Secondly, a deficiency exists in real-time evaluation techniques to adequately assess calibration results. This deficiency is crucial for ensuring the quality and reliability of sensor fusion systems in practical field applications.

III. OVERVIEW

As depicted in Fig.1, this method utilizes a checkerboard as the calibration target. A LiDAR and a camera are mounted on a carrier. Their coordinate systems are denoted as O_{L} and O_{C} respectively. We define the xoy plane of the world coordinate system as O_{w}, which is aligned with the checkerboard plane. The objective is to determine the rigid transformation relationship between the LiDAR and the camera, expressed as:

$$P_{\mathrm{C}}^{i} = R_{\mathrm{CL}} \cdot P_{\mathrm{L}}^{i} + t_{\mathrm{CL}} \tag{1}$$

Here, P_{C}^{i} and P_{L}^{i} represent a pair of point clouds captured by the LiDAR and the camera, respectively. R_{CL} denotes the rotation matrix, and t_{CL} is the translation vector. R_{CL} and t_{CL} constitute the final output of the proposed calibration method.

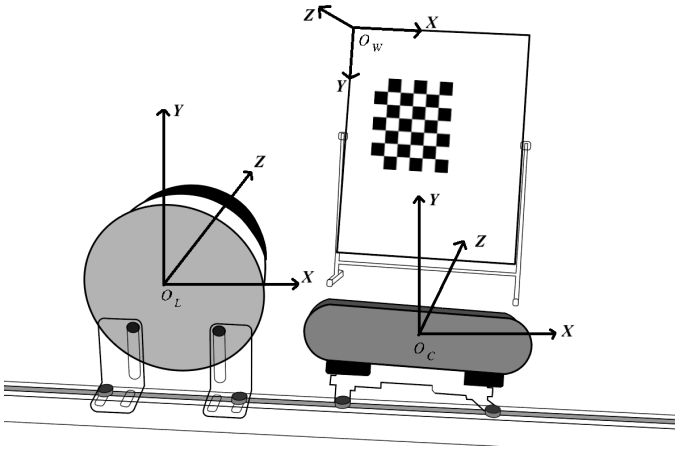

Fig. 1: A LiDAR-camera system, O_{L} and O_{C} denotes the coordinate system of LiDAR and camera. The xoy plane of word coordinate system O_{W} is attached to the plane of checkerboard in this paper.

The method proposed in this paper enables calibration in both simple and complex indoor and outdoor environments. While a clean environment facilitates algorithmic performance, the algorithm remains effective in complex settings (as discussed further in Section IV). The calibration process is straightforward: a large chessboard pattern is printed for use as the checkerboard, ensuring proper alignment with the camera during calibration to ensure the complete visibility of the pattern within the camera's field of view. These conditions are relatively easy to meet, especially for offline calibration. Following the simultaneous collection of checkerboard data with both LiDAR and camera systems, the algorithm automatically conducts extrinsic calibration, yielding the required R_{CL} and t_{CL} parameters without the need for manual intervention. The entire calibration process is efficient and straightforward, boasting high precision (as extensively discussed in Section V).

Fig.2 shows the pipeline of the calibration process. After inputting the data, first step is to calibrate the parameter of camera (Section IV-A), then using a proposed novel algorithm to filter point clouds (Section IV-B). Next is to extract the plane point cloud (Section IV-B). The last step is to establish the co-planar constraint and compute the extrinsic parameters by optimizing the cost function of co-planar constraint (Section IV-C).

IV. METHODOLOGY

The first step of our proposed method is camera calibration. Following the completion of camera calibration, conventional approaches typically proceed to identify correspondence points between LiDAR and camera, followed by point cloud registration to achieve extrinsic calibration. This process is often challenging and complex. In contrast to these methods, our approach offers superiority by exclusively processing LiDAR point clouds throughout the calibration process. There is no need to register for 2D-to-3D correspondence points between LiDAR and camera or perform point cloud registration. This

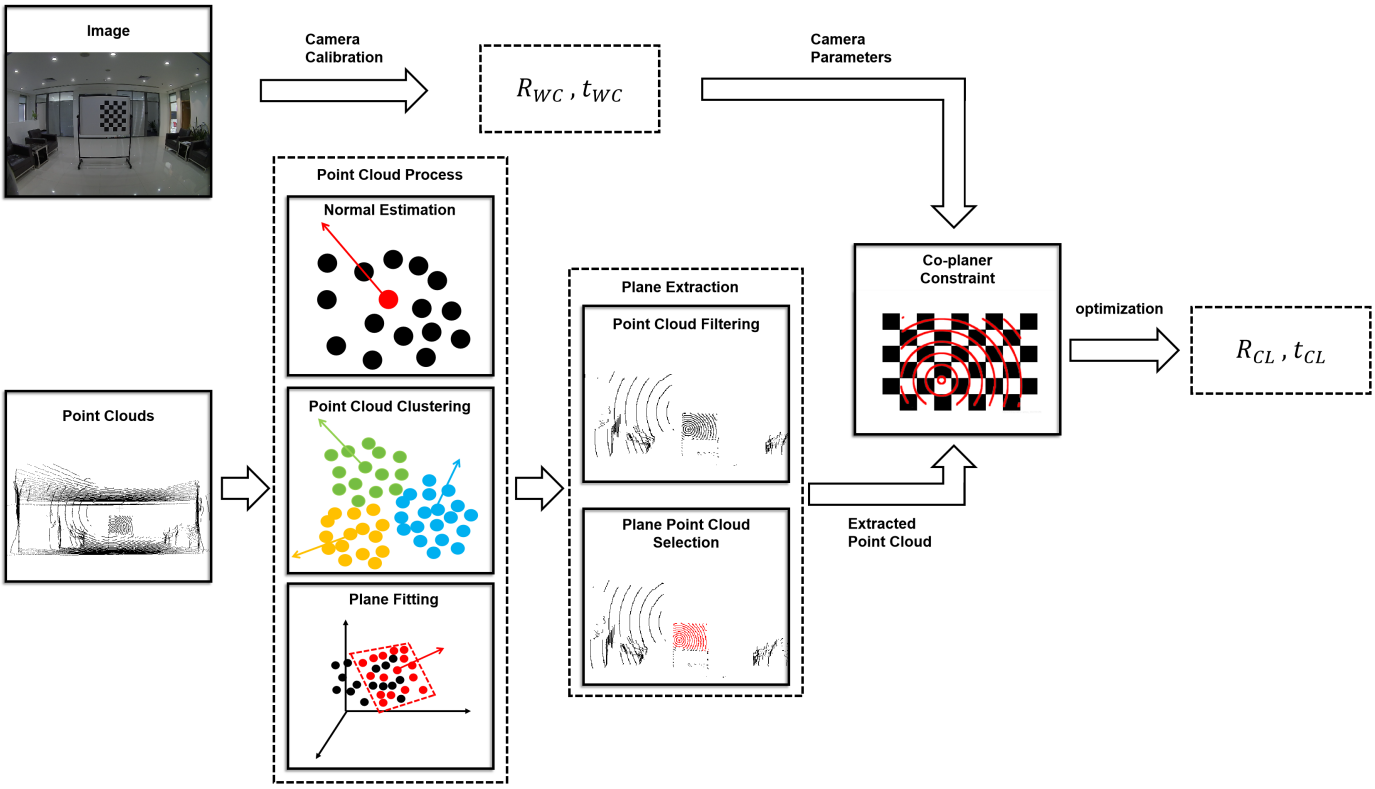

Fig. 2: Pipeline of proposed calibration method

process is realized through a novel plane point cloud extraction method and the co-planar constraint optimization based on this method.

A. Camera Calibration

The first step of the proposed method is to calibrate the camera using a Checkerboard. In the field of camera calibration, method using checkerboard as calibration target has been widely adopted. Methods like [14], [25], [26], [27], [28] are capable of calibrating the camera using only few images of a chessboard, obtaining the extrinsic parameters R_{WC} and t_{WC} from the camera coordinate system O_{C} to the checkerboard coordinate system O_{W}. For a point P_{C}^{i} in O_{C}, it can be projected onto O_{W} using the following formula:

$$P_{\mathrm{W}}^{i} = R_{\mathrm{WC}} \cdot P_{\mathrm{C}}^{i} + t_{\mathrm{WC}} \tag{2}$$

After the completion of the camera calibration process, an approximation of the distance between the checkerboard and the camera can be achieved through the employment of the pinhole camera model. As shown in Fig. 3, let f denote the focal length of the camera, l the distance from the checkerboard to the camera, w' the real-world distance between two distinct corner points on the checkerboard, and w the corresponding corner point distance within the image. According to the principles of similar triangles, this relationship can be succinctly represented as follows:

$$\frac{f}{w} = \frac{l}{w'} \tag{3}$$

From the camera calibration, we can obtain the value of

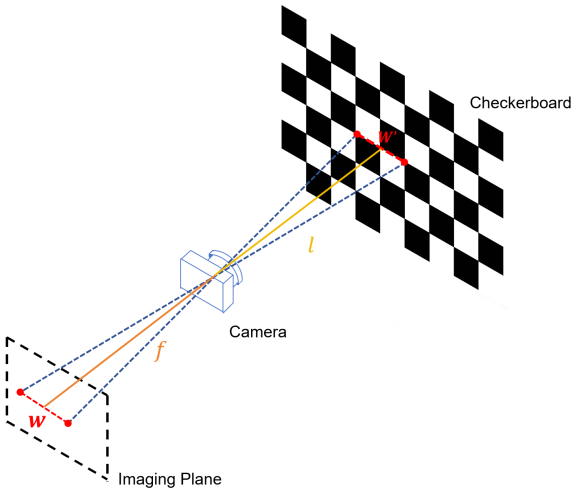

Fig. 3: Illustration of the camera pinhole model. The distance between the checkerboard and the camera can be estimated by employing the pinhole model of the camera and the camera parameters

f, while w can be ascertained by detecting and calculating the pixel distance between corner points during the camera calibration process. The correspondence w' can be obtained because we know the size of the checkerboard. Thus, we can obtain the distance from the checkerboard to the camera through the following formula:

$$l = \frac{w \cdot f}{w'} \tag{4}$$

Algorithm 1 Extrinsic Calibration

Require: Camera images, LiDAR Point cloud P_{L}^{i}
Output: R_{CL}, t_{CL}

i denotes i-th point, j denotes j-th point cluster
k denotes k-th plane, h denotes h-th filtered plane
From camera calibration get R_{WC}, t_{WC}
Calculate threshold l using Eq.4
for each P_{L}^{i} **do**
  Normal vector N_{L}^{i} = PCA(P_{L}^{i})
end for
$C[j]$ = DBSCAN(P_{L}^{i}, N_{L}^{i})
for each $C[j]$ **do**
  Calculate N_{C}^{j} using Eq.5
end for
$Plane[k]$ = RANSAC(C_j, N_{C}^{j})
Set threshold θ
for each $Plane[k]$ **do**
  Calculate α_k using Eq.6
  if $\alpha_k > \theta$ **then**
    Filter out $Plane[k]$
  end if
  if $\alpha_k \leq \theta$ **then**
    $Plane\_filtered[h]$ = $Plane[k]$
  end if
end for
for each $Plane\_filtered[h]$ **do**
  Calculate d_h using Eq.7
  if closest(d_h, l) && max(ρ_{plane}) **then**
    Extract P_{L}^{i} from $Plane\_filtered[h]$
  end if
end for
R_{CL}, t_{CL} = Optimization(P_{L}^{i}, R_{WC}, t_{WC}) using Eq.12
Return R_{CL}, t_{CL}

This distance is further utilized as a discriminant criterion in the selection process of point cloud extraction in Section IV-B.

B. LiDAR Point Cloud Processing

As mentioned in Section IV, we need to extract corresponding points of the checkerboard from the LiDAR point cloud to establish co-planar constraints, and then optimize to obtain the extrinsic parameters. To achieve this, we propose a novel point cloud extraction method based on clustering algorithms, which effectively extracts point clouds belonging to the Checkerboard in complex calibration environments, ensuring the algorithm's effectiveness across different scenarios. Additionally, the proposed approach solely relies on LiDAR point clouds, thereby circumventing the need for corresponding point registration between the camera and LiDAR. This method consists of three main steps: point cloud clustering, point cloud filtering, and plane correspondence point extraction.

Firstly, we perform point cloud clustering, aiming to categorize point clouds into different clusters based on their orientations, facilitating subsequent point cloud filtering. In this process, we initially compute the normals of the point cloud using Principal Component Analysis (PCA), a simple and efficient method commonly utilized for normal estimation in point clouds. Specifically, for a point P_{L}^{i} within the LiDAR coordinate system, its normal N_{L}^{i} is calculated.

Subsequently, the point cloud is clustered based on the normal directions. When the LiDAR scans objects, there are significant density variations between point clouds on different object surfaces [29]. Point clouds within a certain density range are more likely to have similar normal directions. Studies [30] have shown that the clustering algorithm Density-Based Spatial Clustering of Applications with Noise (DBSCAN), compared to other clustering methods, often achieves faster clustering speeds and better clustering results when clustering objects with density variations. Therefore, we adopt the DBSCAN method, known for its proficiency in clustering objects based on density differences, to cluster the point cloud.

The DBSCAN algorithm groups the point cloud P_{L}^{i} into clusters based on the normals N_{L}^{i}, grouping points with similar normal directions into the same cluster. After clustering, the space point cloud is segmented into different clusters $C_1, C_2, C_3, \ldots$ with similar orientations. For each cluster, the most representative normal N_{C}^{i} is identified by following equation:

$$N_{\mathrm{C}}^{i} = \frac{\sum_{i=1}^{n} W_i \cdot N_{\mathrm{L}}^{i}}{\| \sum_{i=1}^{n} W_i \cdot N_{\mathrm{L}}^{i} \|} \tag{5}$$

where n is the total number of point clouds in the cluster, W_i is the weight of the i-th point cloud determined based on the density of the point cloud, N_{L}^{i} is the normal of the i-th point cloud.

The second step involves point cloud filtering, where the Random Sample Consensus (RANSAC) method is utilized for plane fitting owing to its robustness in handling outliers and its efficacy in accurately estimating parameters from noisy data [31]. This choice is particularly suitable for identifying the maximum density plane within each cluster of point clouds. Subsequently, only the point clouds forming the maximum density plane are retained in each cluster. Then, the point cloud can be effectively filtered based on the orientation information of the point cloud planes. Considering that the plane of the checkerboard is likely to be parallel to or at a certain angle to the xoy plane—taking into account the possible tilted placement of the checkerboard—a distinctive characteristic of such planes is that their normal direction will be parallel to the z-axis or at a certain angle to the z-axis. Therefore, we have set a threshold θ to filter based on the normal direction of each point cloud cluster. We define the angle α_i as the angle between the normal N_{C}^{i} and the z-axis. This angle can be calculated using the formula:

$$\alpha_i = \cos^{-1}\left(\frac{N_{\mathrm{C}}^{i} \cdot z}{\|N_{\mathrm{C}}^{i}\| \|z\|}\right) \tag{6}$$

where N_{C}^{i} is the representative normal of cluster C_i, z is the unit vector of the z-axis (i.e., $z = [0, 0, 1]^T$), $\cdot$ denotes the dot product, and $\|\|$ denotes the magnitude of the vector. The angle α_i is calculated by the arc-cosine of the dot product of N_{C}^{i} and z, divided by the product of their magnitudes.

Subsequently, by setting a threshold θ, we can determine whether this angle meets the requirement. If the angle α_i

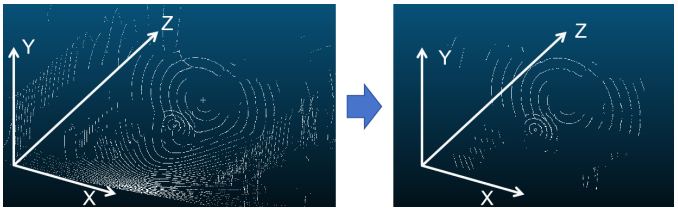

Fig. 4: Most of the irrelevant point clouds in the left figure are filtered out, leaving only the point clouds mainly face to the xoy plane in the right figure.

between a cluster's normal and the z-axis is greater than θ, i.e., $\alpha_i > \theta$, then the point cloud belonging to that cluster will be filtered out, Fig. 4 shows the comparison of point cloud before and after filtering.

Final step is plane point cloud extraction, given the complexity of the actual calibration environment, it is possible that after point cloud filtering, multiple planes still exist. It is necessary to detect the planes belonging to the checkerboard among these multiple planes. This can be achieved by leveraging the prior information obtained from Section IV-A. Specifically, we have computed an approximate distance l between the checkerboard and the camera. For the general LiDAR-camera systems installed in close proximity, this l can be directly utilized as an approximate distance threshold. If there is a significant disparity in the installation distance between the LiDAR and the camera, we can also manually measure and adjust the value of l. We aim to identify the filtered plane cluster C_i whose distance d_i to the LiDAR is closest to l. The point clouds in cluster C_i can be represented as a $3 \times n$ matrix $\mathbf{A}_i = [x_1, y_1, z_1; x_2, y_2, z_2; ...; x_n, y_n, z_n]^T$. Summing across each row of the matrix, we can obtain a 3×1 matrix $\mathbf{S}_i = [\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i, \sum_{i=1}^{n} z_i]^T$. The distance d_i can be obtained by the following equation:

$$d_{\mathrm{i}} = \frac{1}{n} \cdot \sqrt{\mathbf{S}_i^T \cdot \mathbf{S}_i} \tag{7}$$

Where $\mathbf{S}_i^T$ is the transpose of $\mathbf{S}_i$. In complex environments, it is possible to encounter multiple d_i values close to or even equal to l. To address this, we introduce the point cloud density ρ as another assessment, where the value of ρ equals the number of point clouds in the cluster. In our calibration environment requirements, we aim to use relatively large checkerboard, ensuring that the point cloud density ρ on the checkerboard is maximal. Through this process, we can ultimately determine the point cloud in the LiDAR data that belongs to the checkerboard by finding the cluster with the closest d_i value to l as well as the largest point cloud density ρ in the candidate cluster. Finally, we extract the point cloud in the selected cluster, denoted as P_{L}^i. These are the corresponding points of checkerboard in LiDAR point clouds. Fig. 5 illustrates the precise extraction of corresponding points in different calibration scenarios.

C. Co-planar Constraint and Extrinsic Calculation

For the point cloud P_{L}^i extracted in the previous step, if we know the extrinsic parameters R_{CL} and t_{CL}, we can project it from the LiDAR coordinate system to the camera coordinate system, yielding the point cloud P_{C}^i:

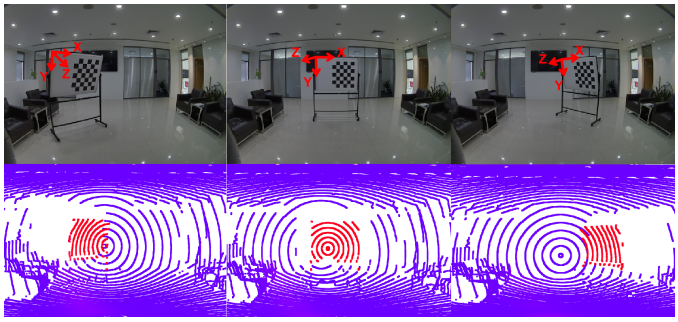

Fig. 5: Plane point cloud extraction in different scenario.

$$P_{\mathrm{C}}^i = R_{CL} \cdot P_{\mathrm{L}}^i + t_{CL} \tag{8}$$

Utilizing the extrinsic parameters of the camera to the checkerboard coordinate system, R_{WC} and t_{WC}, we can further project P_{C}^i from the camera coordinate system to the checkerboard coordinate system, resulting in the point cloud P_{W}^i:

$$P_{\mathrm{W}}^i = R_{WC} \cdot (R_{CL} \cdot P_{\mathrm{C}}^i + t_{CL}) + t_{WC} \tag{9}$$

Since P_{W}^i belongs to the xoy plane of the checkerboard, it should lie on the plane of the checkerboard during scanning, indicating that the z-axis coordinates of these points should be zero. Thus, we can establish the constraint equation:

$$z(P_{\mathrm{W}}^i) = z(R_{WC} \cdot (R_{CL} \cdot P_{\mathrm{L}}^i + t_{CL}) + t_{WC}) = 0 \tag{10}$$

We aim to minimize the z-axis coordinates of the checkerboard point P_{W}^i to solve for the unknown extrinsic parameters R_{CL} and t_{CL}. This can be achieved by minimizing the sum of squared errors:

$$\text{minimize} \sum_{i=1}^{N} (z(P_{\mathrm{W}}^i))^2 \tag{11}$$

Here, N is the number of extracted LiDAR points, and $z(P_{\mathrm{W}}^i)$ represents the i-th point's z value in the checkerboard coordinate system. Substituting into Equation 8, it can write as:

$$\arg\min_{R_{CL}, t_{CL}} \sum_{i=1}^{N} \left| z(R_{WC} \cdot (R_{CL} \cdot P_{\mathrm{L}}^i + t_{CL}) + t_{WC}) \right|^2 \tag{12}$$

By solving this optimization problem, we can obtain the desired extrinsic parameters R_{CL} and t_{CL}, thereby achieving the calibration between the LiDAR and the camera. This optimization process relies solely on the constraints provided by the LiDAR point cloud. Moreover, since each point on the plane serves as a constraint, we have established a sufficient number of constraints in one input frame to optimize accurate values for the extrinsic parameters. After several iterations of optimization, we can ultimately determine the precise extrinsic parameters R_{CL} and t_{CL} between the LiDAR and the camera.

The whole calibration algorithm is summarized in Algorithm 1. It is noteworthy that, from processing the LiDAR point cloud to obtaining the extrinsic parameters between the

TABLE I: PROPERTIES OF THE COMPARISON METHODS

Method	Method Type	Calibration Target	Minimum Calibration Frame
Geiger [17]	Fully Automatic	At least four checkerboards	1
Park [4]	Manual	At least three whiteboards	12
Pusztai [32]	Semi-automatic	One box with three perpendicular sides	6
Matlab Toolbox [26]	Semi-automatic	One checkerboard	6
Proposed	**Fully Automatic**	**One checkerboard**	**2**

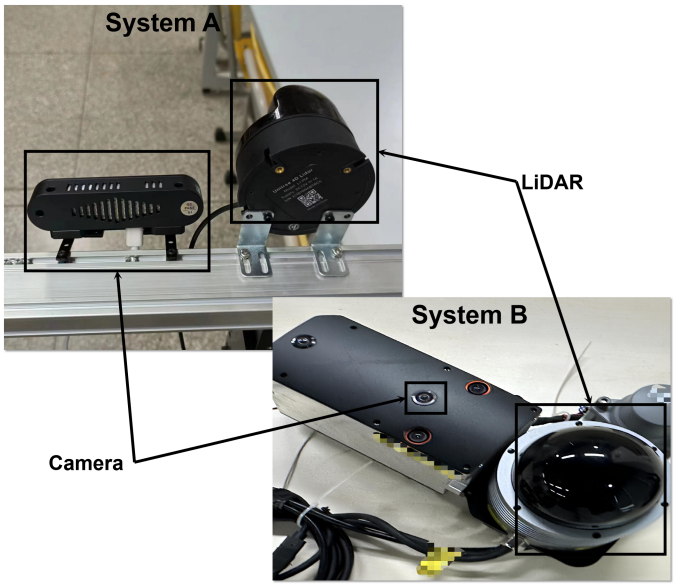

Fig. 6: System A consists of a Gemini Pro camera and a Unitree L1 LiDAR; System B consists of an X1 camera and a Robosense LiDAR.

camera and the LiDAR through optimization, we exclusively handle the LiDAR point cloud and do not perform any corresponding point registration operations. This eliminates the need for registering correspondence between points and reduces potential errors caused by correspondence mismatches.

V. EXPERIMENT

We conducted a series of experiments, encompassing both simulated and real-world data, to evaluate the performance of our proposed method. In Section V-A, we assess the method's effectiveness by calculating calibration accuracy, calibration running time and the translation and rotation errors of extrinsic, leveraging the availability of ground truth data in the simulated environment. In Section V-B experiments, the proposed method has been validated on real-world data collected by two LiDAR-Camera systems A and B shown in Fig.6. We evaluate the algorithm's performance by visualizing the extraction process and re-projecting the LiDAR point cloud onto camera images. We will conduct comparative experiments, comparing the results generated by the proposed method in this paper with the results produced by the four State of the Art(SOTA) methods shown in Table I. All experiments were conducted on a regular laptop with Intel Core i5-8250U CPU and GPU of NVIDIA GeForce MX150.

A. Simulation Experiment

In the simulation experiment, a calibration scenario was created in Gazebo Robot Simulation Environment (Gazebo) to validate both the calibration accuracy and the generalizability of our method across diverse LiDAR-camera systems. We performed calibration on three distinct setups, each employing different combinations of cameras and LiDARs, with variations in the displacement between the camera and LiDAR for each system. Four comparative methods listed in Table I were employed for comparison alongside the proposed method in this paper. For the methods used for comparison, we use the same simulation environment and adhered to their default calibration procedures. Data were collected using three sets of equipment, and both the proposed method and comparative methods were employed to calibrate the camera and LiDAR systems using same collecting data. We employ two primary metrics: rotation error $Error^{\circ}_{\text{Roll,Pitch,Yaw}}$ and translation error $Error^{\text{m}}_{\text{X,Y,Z}}$:

$$Error^{\circ}_{\text{Roll,Pitch,Yaw}} = \arccos\left(\frac{\text{trace}(\mathbf{R}_{\text{c}}\mathbf{R}_{\text{true}}^{T}) - 1}{2}\right) \tag{13}$$

$$Error^{\text{m}}_{\text{X,Y,Z}} = \|\mathbf{t}_{\text{c}} - \mathbf{t}_{\text{true}}\|_2 \tag{14}$$

where $\mathbf{R}_{\text{c}}$ represents the rotation matrix of the calibration result, $\mathbf{R}_{\text{true}}$ represents the ground truth matrix; $\mathbf{t}_{\text{c}}$ represents the translation vector of the calibration result, $\mathbf{t}_{\text{true}}$ represents the ground truth vector. $\text{trace}(\cdot)$ denotes the trace of the matrix. $\|\cdot\|_2$ is the L2 norm of the vector. To more clearly demonstrate the accuracy of the method, we calculated the specific errors in the XYZ, roll, pitch, and yaw dimensions.

Table II presents the mean and standard deviation of these errors for these methods across the three different LiDAR-camera systems. It can be observed that the proposed method exhibits rotation errors of 0.017, 0.043, and 0.066 in terms of roll, pitch, and yaw respectively, and translation errors of 0.0173, 0.0147, and 0.0103 meters in the X, Y, and Z directions respectively. The small errors imply that the results obtained by the proposed method are closest to the ground truth, indicating superior calibration accuracy compared to the comparative methods. Additionally, the proposed method demonstrates the smallest standard deviation across the data, indicating its robustness in accurately calibrating LiDAR-camera systems with different configurations.

After the calibration accuracy experiment, we also computed the mean re-projection error (MRE) for different calibration methods. In the simulated environment, a uniform circular plane was vertically placed in front of the equipment, and all other objects were removed. Data were collected using three equipment, and then, a manual selection of the circular center

TABLE II: THE MEAN AND STANDARD DEVIATION OF CALIBRATION ERRORS FOR THE PROPOSED METHOD AND THE COMPARISON METHODS ACROSS THREE LIDAR-CAMERA SYSTEMS

Method	Metric	Rotation Error(°)			Translation Error(m)		
		Roll	Pitch	Yaw	X	Y	Z
Geiger	Avg	0.4001	0.4067	0.4503	0.0853	0.0903	0.0970
	Std	0.0258	0.0338	0.0141	0.0082	0.0053	0.0041
Park	Avg	0.3633	0.3233	0.3933	0.0892	0.1002	0.0933
	Std	0.0350	0.0661	0.0573	0.0070	0.0121	0.0050
Pusztai	Avg	0.2933	0.3133	0.2833	0.0727	0.0760	0.0733
	Std	0.0140	0.0381	0.0323	0.0037	0.0060	0.0062
Matlab Toolbox	Avg	0.2467	0.2467	0.3233	0.0626	0.0673	0.0623
	Std	0.0281	0.0170	0.0254	0.0033	0.0091	0.0051
Proposed	**Avg**	**0.0171**	**0.0430**	**0.0662**	**0.0173**	**0.0147**	**0.0103**
	Std	**0.0071**	**0.0043**	**0.0051**	**0.0004**	**0.0013**	**0.0002**

TABLE III: COMPARED WITH THE RESULTS OF SOTA CALIBRATION METHODS IN KITTI DATASET.

Method	$Error^{m}_{X}$	$Error^{m}_{Y}$	$Error^{m}_{Z}$	$Error^{m}_{Mean}$	$Error^{\circ}_{Roll}$	$Error^{\circ}_{Pitch}$	$Error^{\circ}_{Yaw}$	$Error^{\circ}_{Mean}$
				$KITTI_1$				
Zhu [33]	0.073	0.085	0.083	0.080	0.388	0.278	0.128	0.264
Ma [34]	0.080	0.063	0.079	0.074	0.236	0.542	0.392	0.390
Calibnet [35]	0.091	0.103	0.077	0.090	0.489	0.321	0.236	0.348
ATOP [36]	0.064	0.063	0.057	0.061	0.258	0.214	0.324	0.265
Proposed	**0.011**	**0.018**	**0.016**	**0.015**	**0.172**	**0.113**	**0.132**	**0.139**
				$KITTI_2$				
Zhu [33]	0.063	0.066	0.071	0.067	0.228	0.211	0.158	0.199
Ma [34]	0.072	0.055	0.069	0.065	0.256	0.343	0.212	0.270
Calibnet [35]	0.053	0.077	0.076	0.068	0.291	0.121	0.251	0.221
ATOP [36]	0.051	0.064	0.053	0.056	0.398	0.241	0.312	0.317
Proposed	**0.014**	**0.011**	**0.009**	**0.011**	**0.122**	**0.083**	**0.092**	**0.099**

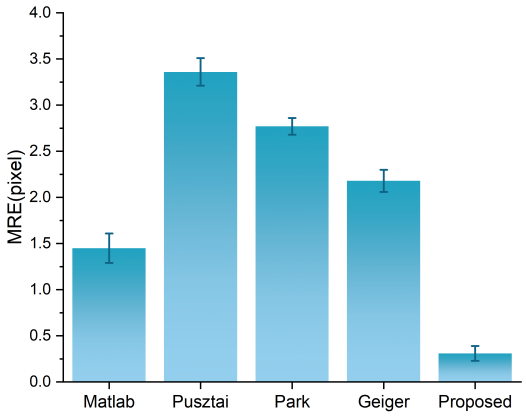

Fig. 7: Mean re-projection error(MRE) for different calibration methods.

point cloud was performed. Subsequently, the center point cloud was re-projected onto the camera coordinate system through the extrinsic parameters. The pixel distance between the re-projected center point $\mu' = (u', v')$ and the center of the circular image $\mu = (u, v)$ in the camera was calculated as the re-projection error:

$$Error_{\text{rep}} = \sqrt{(u' - u)^2 + (v' - v)^2} \tag{15}$$

Fig. 7 demonstrates that the proposed method achieves the minimum MRE , which is less than 0.5 pixels.

To highlight the advantage of the proposed method in requiring less data, in the subsequent experiments, we will set the input calibration frames as a control variable. Calibration frame is defined as an image synchronized with its corresponding LiDAR point cloud data. This approach allows us to assess and compare the calibration accuracy of both the proposed method and the SOTA methods under various quantities of input calibration frames, thereby illustrating the efficiency of our approach in terms of data utilization.

As observed from Fig.8, the proposed method significantly outperforms the comparative methods in calibration accuracy when utilizing a small number of calibration frames. This can be attributed to the proposed method's efficient utilization of point cloud information through the extraction of planar point clouds and the establishment of co-planar constraints. Even with minimal data, the method is capable of constructing strong point cloud constraints. In contrast, correspondence registration based methods generally require a larger volume of input to establish substantial constraints that ensure calibration accuracy. This phenomenon is evident as the accuracy of all methods gradually improves with an increasing number of calibration frames. Methods such as the Geiger and Park exhibit some variability in standard deviation as the number of calibration frames increases. This variability is due to the introduction of more correspondence points by additional calibration frames, which, while increasing the number of correspondences, also leads to a higher incidence of corre-

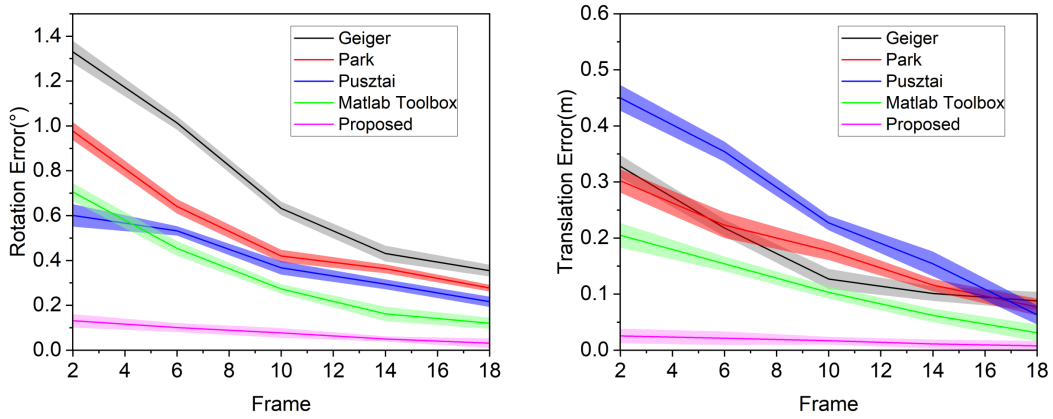

Fig. 8: Error band chart of calibration accuracy for different input frames.

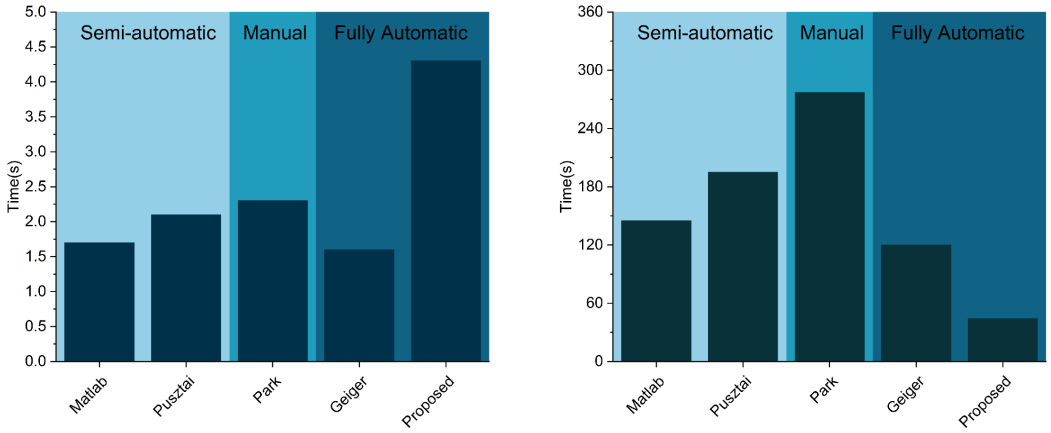

Fig. 9: Left is the time consumption for processing each frame; Right is the total calibration time.

spondence mismatches, thereby causing fluctuations.

Utilizing a increasing number of calibration frames is beneficial for enhancing calibration accuracy; however, the inclusion of additional input data necessitates extended calibration time. Fig.9 illustrates the per-frame processing time as well as the total calibration time across different methodologies. Fully automated approaches hold a significant advantage in terms of processing time, with all methods, without exception, experiencing increased processing times as the volume of calibration data inputs grows. In terms of processing speed, the method proposed in this paper requires a longer duration to process calibration frame. However, the proposed method achieves the accuracy attainable by SOTA methods with a multitude of input datasets using only a limited number of input frames. Therefore, the overall calibration time still remains ahead. In summary, the method introduced in this paper demands less calibration data and possesses an advantage in calibration speed, making it both efficient and effective.

B. Real-world Data Experiment

This real-world data experiment includes re-projecting the LiDAR point clouds into image captured by the camera and evaluating proposed method using KITTI [37] dataset.

Fig.10 represents the re-projection outcome, demonstrating the superior performance of proposed method, particularly evident at the edges of the checkerboard. In our method, the laser points belonging to the checkerboard are accurately projected within its boundaries, whereas alternative approaches tend to project some of these points outside the checkerboard, to varying degrees. This observation underscores the higher calibration accuracy and generalizability achieved by proposed method.

As detailed in Section IV, our method relies solely on plane identification when camera parameters are known. Consequently, our approach can be evaluated using the KITTI dataset by manually selecting the appropriate plane, given the availability of camera parameters. In contrast, other calibration methods previously compared necessitate specific calibration targets not found in the KITTI dataset. Accordingly, we selected several SOTA calibration methods [33], [34], [35], [36] suitable for the KITTI dataset for comparison. We selected two sequences from the KITTI dataset that contain the requisite plane for calibration, enabling the calibration of the camera-LiDAR system and the computation of errors relative to ground truth, as presented in Table III.

From Table III, it is evident that our method consistently

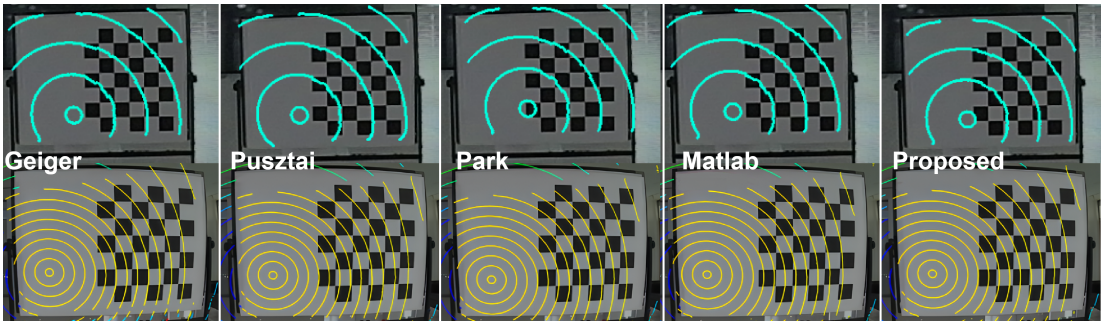

Fig. 10: The first row shows the re-projection results of different calibration methods applied to LiDAR-camera System A, while the second row shows the re-projection results of the same methods applied to System B.

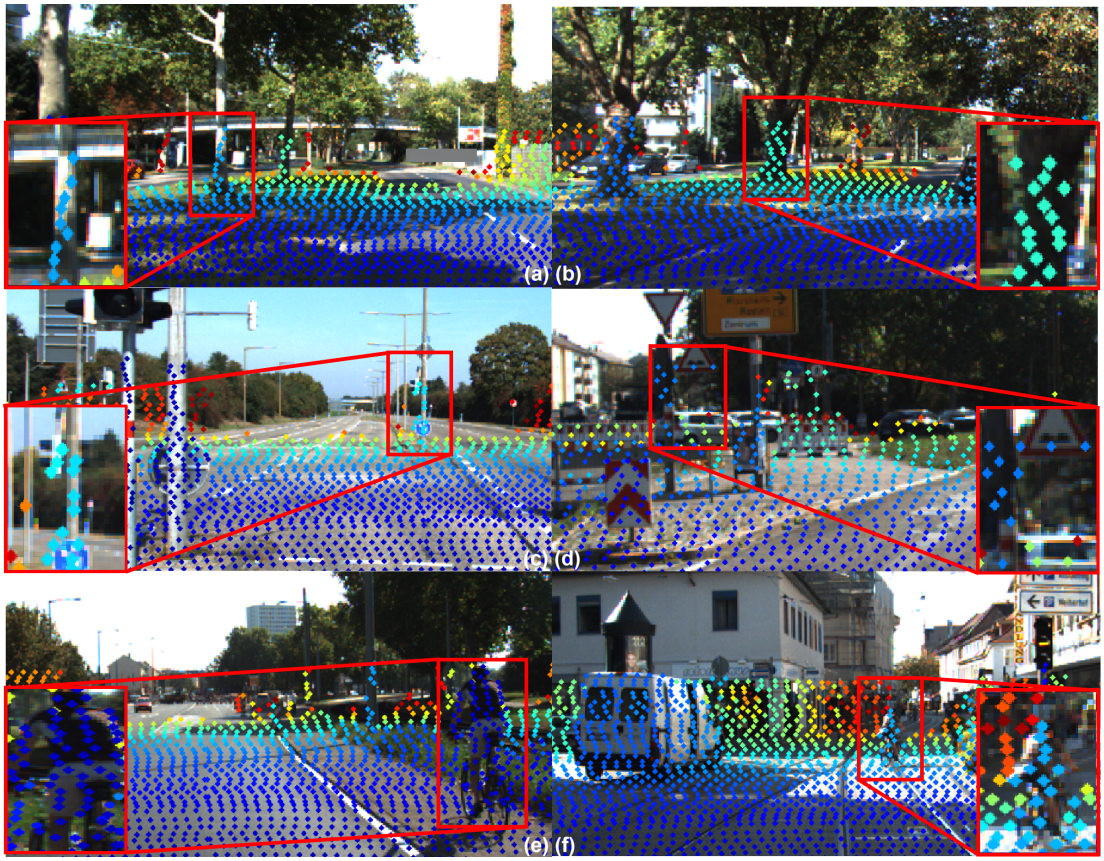

Fig. 11: (a)(b), (c)(d), (e)(f) illustrate the re-projection of trees, roadside poles, and cyclist respectively. The point cloud of trees and poles is concentrated on the main body without exceeding the boundaries, and the point cloud projection on the cyclist is uniform in color (depth), indicating precise extrinsic calibration.

achieves outstanding performance on the KITTI dataset, compared to other methods, the calibration accuracy on the two sequence has been improved by up to 64.36% and 68.77%, respectively. This means that in outdoor environments, the measurement errors caused by calibration inaccuracies of the LiDAR-camera system at a distance of 20 m has decreased from a maximum of 14 cm to 5 cm. Moreover, we utilize our calibration parameters to re-project outdoor LiDAR point clouds, as depicted in Fig. 11. We selected two trees, two roadside poles, and two cyclists as our observation targets. The color of the point clouds is determined by depth, revealing consistent and evenly distributed colors on the observation targets, aligning well with their contours. This indicates that our method yields highly accurate re-projection results.

VI. CONCLUSION

This paper introduces a novel LiDAR-camera calibration method that is straightforward to implement and does not rely on human intervention. The key innovation of this work lies in circumventing corresponding point registration using in current most calibration methods. This is achieved through an algorithm capable of extracting plane point clouds in complex environments and establishing co-planar constraints. Specifically, the algorithm calculates point cloud normals,

fits planes, filters candidate planes using normal filtering, and applies distance and density thresholds to finally extract LiDAR point clouds. Co-planar constraints are then built using the extracted LiDAR point clouds, and the extrinsic parameters between LiDAR and camera are introduced. These parameters are ultimately obtained through optimization. This process circumvents the errors introduced by mismatched corresponding points in registration-based methods and maximizes the utilization of point cloud information, thereby reducing the required calibration input data. Extensive experiments on both simulation and real-world data have demonstrated that this method exhibits high accuracy in extrinsic calibration compared to current methods. The average rotation and translation errors of the calibration results are both less than 0.05° and 0.015m, respectively.

Although the proposed method offers these advantages, there are still areas for improvement and enhancement. Firstly, the algorithm for extracting planar point clouds in the method may struggle to accurately extract the checkerboard plane when there are many planes in the environment that resemble the checkerboard. Additionally, when the checkerboard is placed too far from the LiDAR-camera setup, the sparse point cloud of its plane may lead to insufficiently strict co-planar constraints, thereby affecting the accuracy of calibration parameters. Future work involves further enhancing the effectiveness and robustness of the planar point cloud extraction method in complex scenarios and extending the algorithm to more general scenes. Moreover, efforts will be made to organize the entire work, develop calibration software, and release it as open-source, to serve a wider audience of researchers and industrial practitioners.

REFERENCES

[1] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-level semantic feature detection: A new perspective for pedestrian detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5182–5191.

[2] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Occlusion-aware r-cnn: Detecting pedestrians in a crowd," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 637–653.

[3] A. Humeau-Heurtier, "Texture feature extraction methods: A survey," *IEEE access*, vol. 7, pp. 8975–9000, 2019.

[4] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3d lidar instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[5] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, "Revisiting the pnp problem: A fast, general and optimal solution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2344–2351.

[6] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," *arXiv preprint arXiv:2103.02690*, 2021.

[7] A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets," *Image and vision computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[8] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7163–7172.

[9] L. Cheng, S. Chen, X. Liu, H. Xu, Y. Wu, M. Li, and Y. Chen, "Registration of laser scanning point clouds: A review," *Sensors*, vol. 18, no. 5, p. 1641, 2018.

[10] S. Wu, A. Hadachi, D. Vivet, and Y. Prabhakar, "This is the way: Sensors auto-calibration approach based on deep learning for self-driving cars," *IEEE Sensors Journal*, vol. 21, no. 24, pp. 27 779–27 788, 2021.

[11] J. Han, Z. Zhang, Z. Wang, J. Li, X. Guo, and X. Kang, "Extrinsic calibration of a binocular camera and lidar based on neural networks," *IEEE Sensors Journal*, vol. 23, no. 23, pp. 29 271–29 282, 2023.

[12] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[13] P. Li, R. Wang, Y. Wang, and W. Tao, "Evaluation of the icp algorithm in 3d point cloud registration," *IEEE access*, vol. 8, pp. 68 030–68 048, 2020.

[14] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[15] H. Zhao, Y. Chen, and R. Shibasaki, "An efficient extrinsic calibration of a multiple laser scanners and cameras' sensor system on a mobile platform," in *2007 IEEE Intelligent Vehicles Symposium*. IEEE, 2007, pp. 422–427.

[16] X. Gong, Y. Lin, and J. Liu, "3d lidar-camera extrinsic calibration using an arbitrary trihedron," *Sensors*, vol. 13, no. 2, pp. 1902–1918, 2013.

[17] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3936–3943.

[18] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3d laser scanner and an omnidirectional camera," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 336–341, 2010.

[19] T. Tóth, Z. Pusztai, and L. Hajder, "Automatic lidar-camera calibration of extrinsic parameters using a spherical target," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8580–8586.

[20] 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.

[21] M. Nie, W. Shi, W. Fan, and H. Xiang, "Automatic extrinsic calibration of dual lidars with adaptive surface normal estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2022.

[22] D. Zhang, L. Ma, Z. Gong, W. Tan, J. Zelek, and J. Li, "An overlap-free calibration method for lidar-camera platforms based on environmental perception," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–7, 2023.

[23] C. Ye, H. Pan, and H. Gao, "Keypoint-based lidar-camera online calibration with robust geometric network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2021.

[24] J. Zhu, X. Li, Q. Xu, and Z. Sun, "Robust online calibration of lidar and camera based on cross-modal graph neural network," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[25] Z. Hu, Y. Li, N. Li, and B. Zhao, "Extrinsic calibration of 2-d laser rangefinder and camera from single shot based on minimal solution," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 915–929, 2016.

[26] 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," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5562–5569.

[27] I. Miyagawa, H. Arai, and H. Koike, "Simple camera calibration from a single image using five points on two orthogonal 1-d objects," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1528–1538, 2010.

[28] H.-T. Chen, "Geometry-based camera calibration using five-point correspondences from a single image," *IEEE Transactions on Circuits and systems for Video Technology*, vol. 27, no. 12, pp. 2555–2566, 2016.

[29] R. Abbasi, A. K. Bashir, H. J. Alyamani, F. Amin, J. Doh, and J. Chen, "Lidar point cloud compression, processing and learning for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 962–979, 2023.

[30] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[31] A. Jalal, M. Z. Sarwar, and K. Kim, "Rgb-d images for objects recognition using 3d point clouds and ransac plane fitting," in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, 2021, pp. 518–523.

[32] Z. Pusztai and L. Hajder, "Accurate calibration of lidar-camera systems using ordinary boxes," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 394–402.

[33] Y. Zhu, C. Li, and Y. Zhang, "Online camera-lidar calibration with sensor semantic information," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4970–4976.

[34] T. Ma, Z. Liu, G. Yan, and Y. Li, "Crlf: Automatic calibration and refinement based on line feature for lidar and camera in road scenes," *arXiv preprint arXiv:2103.04558*, 2021.
[35] J. Pei, T. Jiang, H. Tang, N. Liu, Y. Jin, D.-P. Fan, and P.-A. Heng, "Calibnet: Dual-branch cross-modal calibration for rgb-d salient instance segmentation," *arXiv preprint arXiv:2307.08098*, 2023.
[36] Y. Sun, J. Li, Y. Wang, X. Xu, X. Yang, and Z. Sun, "Atop: An attention-to-optimization approach for automatic lidar-camera calibration via cross-modal object matching," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 696–708, 2022.
[37] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.