

Extrinsic Calibration of Multiple LiDARs for a Mobile Robot based on Floor Plane And Object Segmentation

Shun Nijjima¹, Atsushi Suzuki¹, Ryoichi Tsuzaki¹, Masaya Kinoshita¹

Abstract—The utilization of mobile robots equipped with multiple light detection and ranging (LiDAR) sensors, capable of perceiving their surroundings, is on the rise due to the miniaturization and cost reduction of LiDAR technology. This paper introduces a target-less extrinsic calibration method for multiple LiDARs with non-overlapping fields of view (FoV). The proposed method leverages accumulated point clouds of the floor plane and objects obtained during robot motion. It enables accurate calibration, even in challenging configurations where LiDARs are directed towards the floor plane, which can introduce biased feature values. Additionally, the method incorporates a noise removal module that takes into account the scanning pattern to address bleeding points, which are significant sources of error in point cloud alignment when using high-density LiDARs. Evaluations conducted through simulation demonstrate that the proposed method achieves higher accuracy in extrinsic calibration with two and four LiDARs compared to conventional methods, regardless of the type of objects. Furthermore, experiments conducted using a real mobile robot validate the effectiveness of our proposed noise removal module in precisely eliminating noise compared to conventional methods. The estimated extrinsic parameters successfully contribute to the creation of consistent 3D maps.

I. INTRODUCTION

A light detection and ranging (LiDAR) is recognized as a critical sensor in autonomous driving technology due to its high accuracy observation, and it has been advancing in miniaturization and cost reduction. Consequently, the incorporation of LiDARs into mobile robots is increasing. The spreading of high-density LiDARs [1] is also enhancing the perception capabilities of mobile robots. However, it is difficult for a single LiDAR to cover the entire surroundings of a mobile robot, including the travel surface. It is common to mount multiple LiDARs in different directions, such as front and back or left and right, as shown in Fig. 1(a). In the future, it is expected that widespread adoption of these mobile robots equipped with multiple LiDARs. With the proliferation of such mobile robots, there arises a need for robot inspection sites around the world to calibration at the time of shipment, time of deterioration over time, after overhauls and so on.

Accurate integration of data from multiple LiDARs requires robust extrinsic calibration. A general and useful method involves observing a common object with distinctive features such as spheres or planes simultaneously with multiple LiDARs [2] [3]. However, since each LiDAR is positioned to recognize different directions, their field of view (FoV) rarely overlap, if at all. Therefore, placing a common

¹ Sony Group Corporation, Minato-ku, Tokyo, Japan, 108-0075 (email: shun.nijjima@sony.com).

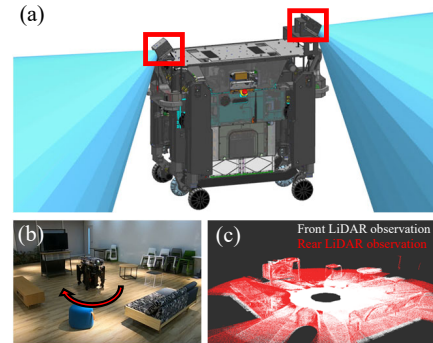


Fig. 1. The proposed method achieves extrinsic calibration of multiple LiDARs with non-overlapping FoV such as mobile robot in (a). Our approach is accumulating data while moving similar to (b) and creating a common FoV to evaluate the consistency of the 3D map. The proposed method achieves accurate extrinsic calibration and generates a consistent 3D map even when most of the observations are on the floor, shown in (c).

object in the shared FoV is difficult for actual mobile robots, making these target-based methods challenging to apply.

To avoid additional tasks such as limitations of the Field of View (FoV) and the preparation of specific components, methods have been proposed that create a common FoV among sensors by accumulating point clouds while in motion [4] [5] similar to Fig. 1(b). These methods minimize observation errors between LiDARs by using the entire set of environmental features such as planes and line features accumulated during movement. However, for robots that travel on the ground, LiDARs are often mounted to detect the floor surface, resulting in observations that are predominantly of the floor shown in Fig. 1(c). Consequently, the features obtained may be insufficient, or the floor features may become dominant, leading to unsuccessful extrinsic calibration.

This paper proposes a method estimating the extrinsic parameters of multiple LiDARs with non-overlapping FoV by utilizing point clouds of floor plane and objects accumulated while in motion. Our method pre-segment the floor plane and object point clouds, and perform a two-step estimation process that utilizes each point cloud. It achieves accurate extrinsic calibration even when the majority of LiDAR observations are of the floor.

Furthermore, as reported in this research [6], high-density LiDARs are known to exhibit prominent bleeding points, which are generated as they extend from the edge of the objects. This noise is caused by the mixing of pulses hitting both foreground and background objects. These noises are a major source of error in extrinsic calibration. Our method includes a module that can remove noise more accurately

compared to conventional methods, based on a very simple concept.

Our contributions are as follows:

- We propose the extrinsic calibration method for multiple LiDARs with non-overlapping FoV. Our method estimates the movement trajectory and extrinsic parameters simultaneously using segmented floor and object point clouds. It enables accurate calibration even when floor information is dominant and challenging for conventional methods.
- We propose a simple noise removal module useful for calibrating with high-density LiDARs.
- Through evaluation in simulation of different environments where various types of objects are placed on the floor plane, and on different configurations of LiDARs with non-overlapping FoV, we demonstrate that our proposed method achieves higher accuracy than the conventional methods.
- The experiments using a real mobile robot has shown that our proposed noise removal module can eliminate noise more precisely than conventional methods, and the estimated extrinsic parameters have successfully created consistent 3D maps by LiDARs with non-overlapping FoV.

II. RELATED WORK

1) *Multiple LiDAR calibration*: There are two main approaches to multiple LiDAR calibration: target-based and targetless methods. Target-based methods estimate extrinsic parameters by preparing special components as targets to assist in feature extraction and observing them simultaneously with multiple LiDARs. These research use retro-reflective targets [2], specially processed plates [7], or spherical objects [3] to aid in the extraction of 3D features such as edges, planes, and cones. While target-based methods have relatively low computational complexity, they require the creation of specific targets, which necessitates additional time for preparing. Since the mobile robot may frequently perform extrinsic calibration in various locations, it is desirable to avoid the need to prepare special components. In targetless calibration methods, there are two representative approaches: one is motion-based that estimates extrinsic parameters from motion, and the other is feature-based extracts environmental features to estimate extrinsic parameters. Motion-based approaches [8], [9] assume that each sensor undergoes the same rigid body motion at each time interval, transforming the extrinsic calibration into a hand-eye problem. Methods have also been proposed that facilitate LiDAR motion estimation by introducing other external sensors such as GNSS or INS [10] [11]. The calibration accuracy of these approaches is susceptible to the precision of the sensor's motion estimation results, which may compromise reliability.

Feature-based methods estimate extrinsic parameters without special components by extracting features within the environment. In particular, studies [12], [13] detect planar features in natural environments to obtain correct associations. Research [12] imposes coplanarity and perpendicularity

constraints on line segments extracted from vertical planes encountered in structured scenes to constrain the 6 degrees of freedom (6DoF) parameter space. Study [13] uses three linearly independent planes, which are easier to obtain than vertical planes. These methods are bound by the same limitation as target-based methods: a common FoV is essential.

To solve these problems, methods have been proposed that collect environmental features while in motion to create a common FoV among LiDARs with no FoV overlap [4]. These methods perform global optimization of a graph composed of extrinsic parameters of each LiDAR and movement trajectories to ensure the consistency of the collected point cloud data. Studies [5] [14] have been modified to extract plane and line segment features by adaptive size voxels to improve computational efficiency, and achieves an average translational error of 0.01m and a rotational error of less than 0.1 degrees.

However, these methods needs environments that have sufficient and various direction of features to constrain the 6DoF, such as wall and ceiling. When applying these methods to ground-traveling robots as shown in Fig. 1(a), the point cloud data obtained consists only of the ground and a few objects, leading to insufficient features or dominance of floor point clouds, which can result in unsuccessful extrinsic calibration. To address these issues, our method pre-segment the floor plane and object point clouds, and adopt a two-step estimation process that utilizes each point cloud, and achieves comparable accuracy to above existing methods.

2) *LiDAR noise reduction*: These are two main approaches to noise reduction in LiDAR: point cloud-based and image-based. In point cloud-based methods, among the effective point cloud noise removal methods are the statistical outlier removal filter (SOR) [15] and the radius outlier removal filter (ROR) [16]. These filters remove noise by identifying areas with low surrounding point density. This is based on the characteristic that noise occurs randomly, and its density is sparser than that of points that have hit an object. The aforementioned methods are most effective when the density of the point cloud is consistent regardless of location. However, the density of point clouds observed by LiDAR becomes sparser with increasing distance from the sensor, which means that directly applying these methods to LiDAR data may not yield adequate performance.

To address these issues, the dynamic statistics outlier filter (DSOR) [17] and the dynamic radius outlier removal filter (DROR) [18] have been proposed. These methods dynamically vary the threshold for noise determination according to the distance from the sensor, considering that point density is higher closer to the sensor and lower farther away, successfully removing even nearby dense noise. High-density LiDARs use scanning patterns that are not constant with each scan, so the density of point clouds is not simply dependent on the distance from the sensor but is also significantly affected by the sensor pattern. For example, LiDARs from Livox use a non-repetitive scanning method [1].

Another approach involves converting the observed point

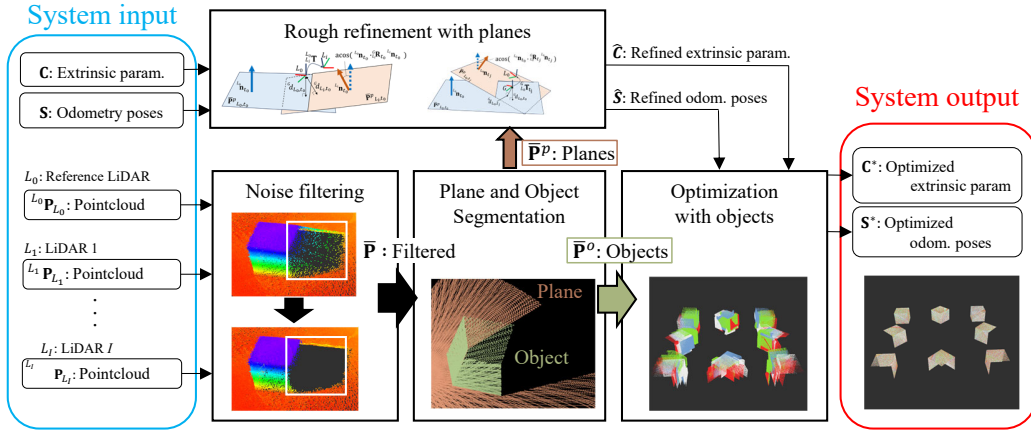


Fig. 2. System overview of proposed multiple LiDAR calibration method

clouds from LiDAR into depth image format and applying computer vision or deep learning for noise classification from LiDAR. High-density LiDARs by non-repetitive scanning [1] obtain dense observation point clouds through the overlay of sparse point clouds, making it difficult to create dense depth images with all values filled, which can hinder the application of the aforementioned image-based methods. These images are fed into the training model to detect features to identify noise from the images. Whethernet [19] and 4DenoiseNet [20] have been shown to be effective in improving data quality from LiDAR with this approach. In certain scenarios, deep learning methods may outperform traditional approaches. However, their performance heavily depends on large-scale labeled datasets, which can be ineffective when labeled data is limited.

In this paper, we propose a novel point cloud-based noise filter that dynamically changes the noise threshold considering the scanning pattern to efficiently remove noise from LiDARs that obtain non-constant observations with each scan.

III. METHODS

A. Problem setting

The goal of extrinsic calibration for multiple LiDARs is to determine the coordinate transformations $\mathbf{C} = \{\mathbf{T}_{L_1}^{L_0}, \dots, \mathbf{T}_{L_I}^{L_0}\}$ between a reference LiDAR L_0 and other LiDARs $\{L_1 \dots L_I\}$. Here, $\mathbf{T}_{L_i}^{L_0} = (\mathbf{R}_{L_i}^{L_0}, \mathbf{t}_{L_i}^{L_0}) \in SE(3)$ represents the coordinate transformation from frame A to frame B, consisting of a rotation component $\mathbf{R}_{L_i}^{L_0} \in SO(3)$ and a translation component $\mathbf{t}_{L_i}^{L_0} \in \mathbb{R}^3$.

To create a common FoV among LiDARs with non-overlapping FoV, the mobile robot accumulates data at multiple timestamps by alternating between stopping and moving. For formalization, assume the robot captures movement trajectory $\mathbf{S} = \{\mathbf{T}_{L_0}^{G, t_0}, \dots, \mathbf{T}_{L_0}^{G, t_j}\}$ denotes the global frame of reference for the movement trajectory.

To assess the consistency of the 3D maps constructed from point clouds accumulated by the reference LiDAR $\{L_0\}$ and other LiDARs $\{L_1 \dots L_I\}$ while in motion, these point

clouds are transformed into a single global coordinate system. The global coordinates of LiDAR L_i at each timestamp t_j can be expressed using the movement trajectory $\mathbf{T}_{L_i}^{G, t_j}$ and the coordinate transformation $\mathbf{T}_{L_i}^{L_0}$ as follows:

$$\mathbf{T}_{L_i}^{G, t_j} = \mathbf{T}_{L_0}^{G, t_j} \mathbf{T}_{L_i}^{L_0} \quad (1)$$

The point cloud observed by LiDAR L_i at time t_j in the LiDAR L_i coordinate system is denoted as \mathbf{P}_{L_i, t_j} , and the corresponding point cloud in the global coordinate system \mathbf{P}_{L_i, t_j}^G is obtained as follows:

$$\begin{aligned} \mathbf{P}_{L_i, t_j}^G &= \mathbf{T}_{L_i}^{G, t_j} \mathbf{P}_{L_i, t_j} \\ &= \mathbf{R}_{L_i}^{G, t_j} \mathbf{P}_{L_i, t_j} + \mathbf{t}_{L_i}^{G, t_j}, \end{aligned} \quad (2)$$

where $\mathbf{p}_{L_i, t_j} = \{x, y, z\}$ represents the coordinates of each point.

The problem of extrinsic calibration for multiple LiDARs with non FoV can be described as finding the optimal movement trajectory \mathbf{S}^* and coordinate transformations \mathbf{C}^* that minimize the error in the 3D maps constructed from each LiDAR's point cloud represented in the global coordinate system. Let $e(\mathbf{P}_{L_0, t}, \mathbf{P}_{L_i, t}^G)$ be a function representing the error between the 3D map constructed by the reference LiDAR and the 3D map constructed by other LiDARs, the problem can be formulated as follows:

$$(\mathbf{C}^*, \mathbf{S}^*) = \arg \min_{\mathbf{C}, \mathbf{S}} \sum_{i=1}^I e(\mathbf{P}_{L_0, t}, \mathbf{P}_{L_i}^G) \quad (3)$$

Various functions can be used to represent the error, such as point-to-point [21], point-to-plane [22], or quadratic entropy [23]. Additionally, to improve computational efficiency, only 3D features may be used in advance [4].

B. System overview

LiDARs mounted to observe the travel surface predominantly capture point clouds of the floor. While the floor plane provides a strong constraint, it can lead to an optimization problem where only the alignment of the floor is prioritized,

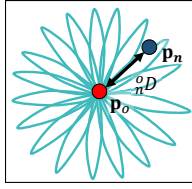


Fig. 3. Scanning pattern of a non-repetitive LiDAR

Algorithm 1 Overview of the noise filter algorithm

Input: Point cloud: $\mathbf{P} = \{\mathbf{p}_0 \cdots \mathbf{p}_n\}$, $\mathbf{p}_n = (x_n, y_n, z_n)$

Output: Filtered point cloud: $\bar{\mathbf{P}}$

```

1:  $\mathbf{P} \leftarrow \text{KdTree}$ 
2: for  $\mathbf{p}_n \in \mathbf{P}$  do
3:   mean distances  $\leftarrow \text{nearestKSearch}(k)$ 
4: end for
5: calculate mean ( $\mu$ ) and std. ( $\sigma$ )  $\leftarrow$  mean distances
6: calculate global threshold ( $H_g$ )  $\leftarrow \mu + (\sigma \times C_s)$ 
7: determine observation center ( $\mathbf{p}_o$ )
8: for  $\mathbf{p}_n \in \mathbf{P}$  do
9:    $\text{Dis.}({}^o_n D) \leftarrow \sqrt{(x_n - x_o)^2 + (y_n - y_o)^2 + (z_n - z_o)^2}$ 
10:  calculate dynamic threshold ( $H_d$ )  $\leftarrow H_g \times C_r \times {}^o_n D$ 
11:  if mean distance ( $\mu_n$ )  $< H_d$  then
12:     $\bar{\mathbf{P}} \leftarrow \mathbf{p}_n$ 
13:  end if
14: end for
15: return  $\bar{\mathbf{P}}$ 

```

potentially failing to find the correct movement trajectory and extrinsic parameters. To address this issue, our process is divided into a rough refinement process using the floor point cloud and an optimization process using the objects point cloud with the floor removed.

The system overview of the proposed method is illustrated in Fig. 2. Each module is represented by a rectangle. The inputs are the observed point clouds ${}^{L_i}\mathbf{P}_{L_i}$ from the LiDARs, initial extrinsic parameters \mathbf{C} , and the movement trajectory \mathbf{S} . First, the noise removal module outputs denoised point cloud ${}^{L_i}\bar{\mathbf{P}}_{L_i}$ from each LiDAR ${}^{L_i}\mathbf{P}_{L_i}$. Next, the point clouds are divided into planes ${}^{L_i}\bar{\mathbf{P}}_{L_i}^p$ and objects ${}^{L_i}\bar{\mathbf{P}}_{L_i}^o$ by the segmentation module. There are no constraints on the segmentation method, this paper uses simple RANSAC and Euclidean clustering. Subsequently, the rough refinement module uses the plane point clouds ${}^{L_i}\bar{\mathbf{P}}_{L_i}^o$ to coarsely correct the movement trajectory \mathbf{S} and initial extrinsic parameters \mathbf{C} , and outputs rough refine extrinsic parameters $\hat{\mathbf{C}}$ and $\hat{\mathbf{S}}$. Here, the movement trajectory and extrinsic parameters are corrected using the fact that LiDARs observing the travel surface are observing the same plane. Details are explained in Section III.D. Finally, the optimization module outputs the optimized set of extrinsic parameters \mathbf{C}^* and movement trajectory \mathbf{S}^* by solving equation (3) using the object point clouds ${}^{L_i}\bar{\mathbf{P}}_{L_i}^o$. Details are discussed in Sections III.E.

C. Noise filter for non-repetitive scan

The purpose of the noise filter module is to output point clouds ${}^{L_i}\bar{\mathbf{P}}_{L_i,t_j}$ from LiDAR observations ${}^{L_i}\mathbf{P}_{L_i,t_j}$ that have been denoised. The proposed noise removal module follows the traditional filter concept that noise occurs randomly, and thus its density is sparser than that of point clouds from actual objects. DSOR considers the dynamic change in the density of observed point clouds with distance from the LiDAR, varying the noise threshold dynamically with distance. However, we focus on the fact that for LiDARs to observe the travel surface, where the distance between the sensor and observation points does not change significantly, the density of the LiDAR's point clouds is more dominantly affected by the LiDAR's scanning pattern than by the distance to the observation points. For example, Livox's non-repetitive scanning method, as shown in Fig. 3, results in varying densities depending on the distance from center of observation.

We propose a noise removal module that dynamically changes the noise threshold based on the distance from the sensor's observation center approximating that the density of these scanning methods simply changes with the distance ${}^o_n D$ between the observation center \mathbf{p}_o and the observation points \mathbf{p}_n . The algorithm overview is shown in Algorithm 1.

- Preliminary preparation: First, a kd-tree is constructed for the observation point cloud (L.1). Then, for each point $\mathbf{p}_n = \{x_n, y_n, z_n\}$, the average distance to k nearest neighbors is calculated (L.2-4). The average distance μ and standard deviation σ are also calculated for the entire point cloud (L.5).
- Global threshold calculation: Using the calculated average distance μ and standard deviation σ for the entire point cloud, the global threshold H_g is calculated (L.6).

$$H_g = \mu + \sigma \times C_s \quad (4)$$

Here, C_s is the constant multiplication factor for standard deviation.

- Noise estimation: Next, the observation center $\mathbf{p}_o = \{x_o, y_o, z_o\}$ is determined (L.7). For each point \mathbf{p}_i , the distance ${}^o_i D$ to the observation center \mathbf{p}_o is calculated (L.9).

$${}^o_n D = \sqrt{(x_n - x_o)^2 + (y_n - y_o)^2 + (z_n - z_o)^2} \quad (5)$$

Finally, the dynamic threshold for noise filtering is calculated (L.10).

$$H_d = H_g \times C_r \times {}^o_n D \quad (6)$$

Here, C_r is the constant multiplication factor for range. If the average distance μ_n for each point \mathbf{p}_n is greater than the dynamic threshold H_d , it is deemed noise and removed (L.12).

D. Rough refinement with planes

1) *Rough refinement of movement trajectory:* Using the plane point clouds $\bar{\mathbf{P}}^p$ detected by the segmentation module, we correct the movement trajectory \mathbf{S} of the reference

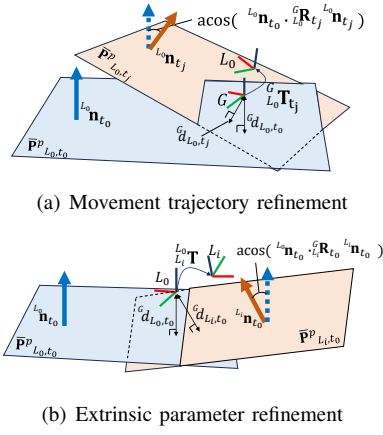


Fig. 4. Movement trajectory and extrinsic parameter refinement with planes

LiDAR L_0 . Specifically, we adjust the movement trajectory so that the normal ${}^{L_0}\mathbf{n}_{t_0}$ and the distance ${}^G d_{L_0,t_0}$ from the plane ${}^G \bar{\mathbf{P}}_{L_0,t_0}^p$ detected by the reference LiDAR L_0 at time t_0 match the normal ${}^{L_0}\mathbf{n}_{L_0,t_j}$ and the distance ${}^G d_{L_0,t_j}$ from the plane ${}^G \bar{\mathbf{P}}_{L_0,t_j}^p$ detected at each timestamp t_j . Figure 4(a) shows an image of the rough refinement of movement trajectory. The change in pose $\Delta_{L_0}^G \mathbf{T} = \{\Delta_{L_0}^G \mathbf{R}, \Delta_{L_0}^G \mathbf{t}\}$ is derived as follows:

$$\Delta_{L_0}^G \mathbf{R} = M(\text{acos}({}^{L_0}\mathbf{n}_{t_0} \cdot {}^G \mathbf{R}_{L_0,t_j} {}^{L_0}\mathbf{n}_{t_j})), \quad (7)$$

$$\Delta_{L_0}^G \mathbf{t} = \mathbf{n}_{t_0} ({}^G d_{L_0,t_j} - {}^G d_{L_0,t_0}) \quad (8)$$

where $M()$ is a function that converts an angle into a rotation matrix.

2) *Rough refinement of extrinsic parameter:* After correcting the movement trajectory, the extrinsic parameters $\mathbf{C} = \{{}^{L_1}\mathbf{T}, \dots, {}^{L_{n-1}}\mathbf{T}\}$ for each LiDAR are corrected similar conditions. The correction of extrinsic parameters involves transforming the coordinates so that the normal \mathbf{n}_{t_0} and distance d_{t_0} to the plane at time t_0 for the reference LiDAR match the normal ${}^{L_i}\mathbf{n}_{t_0}$ and ${}^{L_i}d_{t_0}$ and distance d_{t_0} observed by each LiDAR $\{L_i\}$. Figure 4(b) shows an image of the rough refinement of extrinsic parameter. The change in pose $\Delta_{L_i}^{L_0} \mathbf{T} = \{\Delta_{L_i}^{L_0} \mathbf{R}, \Delta_{L_i}^{L_0} \mathbf{t}\}$ is derived as follows:

$$\Delta_{L_i}^{L_0} \mathbf{R} = M(\text{acos}({}^{L_0}\mathbf{n}_{L_0,t_0} \cdot {}^G \mathbf{R}_{L_i,t_0} {}^{L_i}\mathbf{n}_{L_i,t_0})), \quad (9)$$

$$\Delta_{L_i}^{L_0} \mathbf{t} = {}^{L_0}\mathbf{n}_{L_0,t_0} ({}^G d_{L_0,t_0} - {}^G d_{L_i,t_0}) \quad (10)$$

E. Optimization with objects

We refine the movement trajectory $\hat{\mathbf{S}}$ and the extrinsic parameters $\hat{\mathbf{C}}$ of each LiDAR using the object point clouds $\bar{\mathbf{P}}^o$ detected by segmentation. To solve the optimization problem in equation (3), we define an error function. Considering that the point clouds from LiDARs observing the travel surface may not capture objects with sufficient features, we have chosen to use a simple distance between point clouds without features as the error function.

In each optimization process, a kd-tree is constructed using the object point cloud ${}^G \bar{\mathbf{P}}_{L_0}^o$ observed by the reference LiDAR, and the nearest neighbor ${}^G \hat{\mathbf{P}}_{L_0}^o$ is found for the ${}^G \bar{\mathbf{P}}_{L_i}^o$ transformed into global coordinates using the current movement trajectory $\hat{\mathbf{S}}$ and the extrinsic parameters $\hat{\mathbf{C}}$ of each LiDAR. The error is calculated as the sum of the distances between these point clouds.

$$\begin{aligned} e({}^G \hat{\mathbf{P}}_{L_0}^o, {}^G \bar{\mathbf{P}}_{L_i}^o) &= \sum_{n=1}^{N_o} \| {}^G \hat{\mathbf{p}}_{n,L_0} - {}^G \bar{\mathbf{p}}_{n,L_i} \| \\ &= \sum_{n=1}^{N_o} \| {}^G \hat{\mathbf{p}}_{n,L_0} - ({}_{L_i}^G \hat{\mathbf{R}} \bar{\mathbf{p}}_{n,L_i} + {}_{L_i}^G \mathbf{t}) \|, \end{aligned} \quad (11)$$

where N_o is the number of points in the object point cloud.

Equation (11) is dependent on the poses $\hat{\mathbf{S}}$ and extrinsic parameters $\hat{\mathbf{C}}$ to be optimized. By substituting these error functions into equation (3) and recursively updating until convergence, similar to ICP, the optimized movement trajectory and extrinsic parameters are output. This paper solves the nonlinear optimization problem using the Ceres Solver¹ implemented in C++ with the Levenberg-Marquardt (LM) method.

IV. EXPERIMENTS

In this chapter, we evaluated the proposed methods using both simulation and real-world robot experiments. We estimated the extrinsic parameters of two and four non-overlapping FoV LiDARs in a simulation environment. We demonstrate that our proposed method, which includes rough refinement using the floor plane, can estimate extrinsic parameters more accurately compared to traditional methods. Furthermore, we demonstrate that the proposed calibration method is applicable in the real world using data from an actual mobile robot. We also show that the proposed noise filter module can effectively remove bleeding points noise, which occurs in real observations, compared to traditional methods.

A. Simulation evaluation

1) *Settings:* We created a mobile platform equipped with multiple LiDARs that have non-overlapping FoV in the simulation. The configuration of the multiple LiDARs and simulation environment are shown in Fig. 5. We used the Livox Mid-70 for the LiDARs². We tested our proposed method using rotational motion in place and evaluated the error against the ground truth of the extrinsic parameters. The platform performed repeated stopping and rotating actions at poses where both the object and the floor surface could be observed, accumulating data at a total of 16 poses.

Verify that the proposed method is able to correctly adjust the extrinsic parameters even when randomly applied with a maximum noise of 0.2 m in translation and 5.0 deg in

¹<http://ceres-solver.org/>

²<https://www.livoxtech.com/mid-70>

rotation. Similarly, a maximum noise of 0.2 m and 3.0 deg is randomly added to the motion trajectory. To demonstrate that estimation is possible with various objects, we used three types of objects: box, cylinder, and capsule. In particular, cylinders and capsules are challenging objects for traditional feature-based methods due to their lack of planar features. The simulation environment was constructed using Mujoco³.

For accuracy comparison, we compared with feature-based method [5] and the entire cloud-based method, which uses the entire point cloud as described in [4] and methods that only use segmentation without rough refinement by planes (proposed w.o. RR). To maintain similar conditions, we used the same cost function as the proposed method for the entire cloud-based method, which is the sum of distances to points. We evaluated the accuracy of the translation error and the rotation error. Each method was optimized for 200 iterations during a evaluation process.

2) *Extrinsic error evaluation*: Figure 6 shows the average values of translation error and rotation error for each method and object, which were tested 10 times. The evaluation results indicate that the proposed method achieves the highest accuracy compared to other methods, and could reduce the mean initial translation error of 0.12 m by up to 92%, resulting in an translation error to below 0.01 m.

Figure 7 shows the point clouds before and after optimization in each method for Set 1. It is shown that the proposed method can calibrate more accurately than conventional methods, regardless of object type, and that through rough refinement improves the calibration performance.

3) *Comparison of iteration number to convergence* : Figure 8 shows a graph with iterations on the horizontal axis and translation error and rotation error on the vertical axis in set 1. These results demonstrate that rough refinement by the floor plane in the proposed method reduces the number of iterations required for convergence and achieves the high accuracy.

B. Real world evaluation

1) *Settings*: We used our developed mobile robot Tachyon3 [24], equipped with two Livox Mid-70s as shown in Fig. 1. By design, the front LiDAR was tilted at 35.4 deg

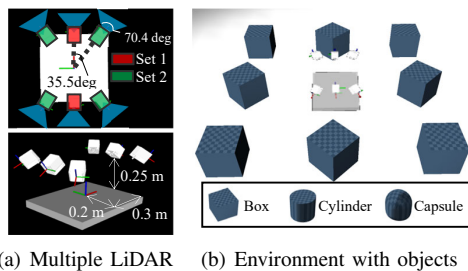


Fig. 5. Configuration of multiple LiDAR and environment in simulation. (a) The LiDARs are divided into two sets: Set 1 (red) with two LiDARs and Set 2 (green) with four LiDARs, each arranged to avoid overlap. (b) These LiDARs are moved in an environment where objects with a height of 0.25 m are placed around.

³<https://mujoco.org/>

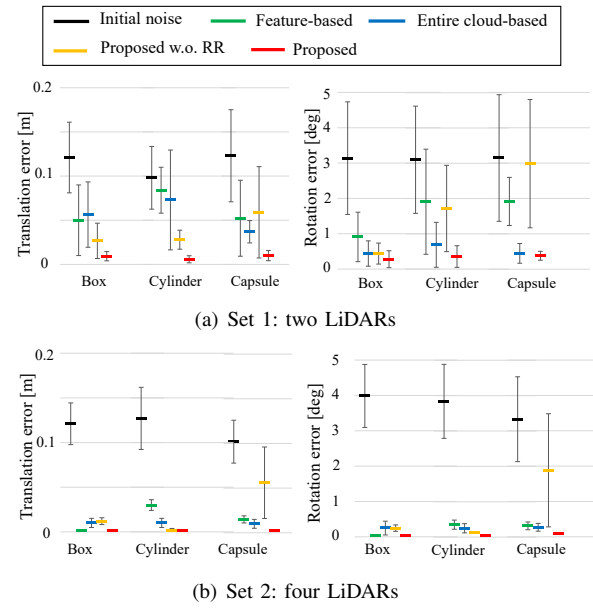


Fig. 6. Comparison of translation and rotation error in each method. The proposed method demonstrates the best accuracy regardless of the type of object. In particular, it can be confirmed that the proposed method reduces the translation error to below 0.01 m and rotation error to below 0.3 deg in calibration using two LiDARs, while the accuracy of other methods decreases when using a cylinder or a capsule. In particular, feature-based method tends to be less accurate when using cylinder or capsule compared to using boxes. This discrepancy is likely due to the bias or lack of plane and edge features. In set 2, where four LiDAR systems are employed, the proposed method also exhibited the highest accuracy. Although all the methods achieved relatively higher accuracy due to the abundance of data, methods without rough refinement sometimes converge to an incorrect position in the capsule. This further confirms the usefulness of the proposed rough refinement with planes.

and mounted at a height of 0.34 m from a base frame, while the rear LiDAR was tilted at 34.4 deg and also mounted at a height of 0.28 m from a base frame. The height of base frame was 0.54 m from a ground. Note that the design values were not reliably accurate and include errors that are difficult to measure, such as mounting errors.

As one of the evaluation, we defined the sum of distances between normalized point clouds as the metric error, since it was difficult to measure the true values of extrinsic parameters in real-world experiments.

$$\frac{\sum_{i=1}^I e^{(G \hat{\mathbf{P}}_{L_0}^o, G \bar{\mathbf{P}}_{L_i}^o)}}{\sum_{i=1}^I (N_{L_i})} \quad (12)$$

N_{L_i} represent the number of point clouds observed by the LiDARs to be calibrated.

The robot was moved remotely to collect data, and extrinsic calibration was performed for each method. Similar to the simulation, the initial position of the extrinsic parameters was randomly perturbed with a maximum noise of 0.2 m in translation and 5.0 deg in rotation from design value. For the estimation of the motion trajectory, we used an in-house SLAM device mounted on the robot. We conducted optimization using the proposed method by utilizing both the SLAM trajectory and initial extrinsic parameters as input. Figure 9(a) shows the observation scenes. In the scene 1,

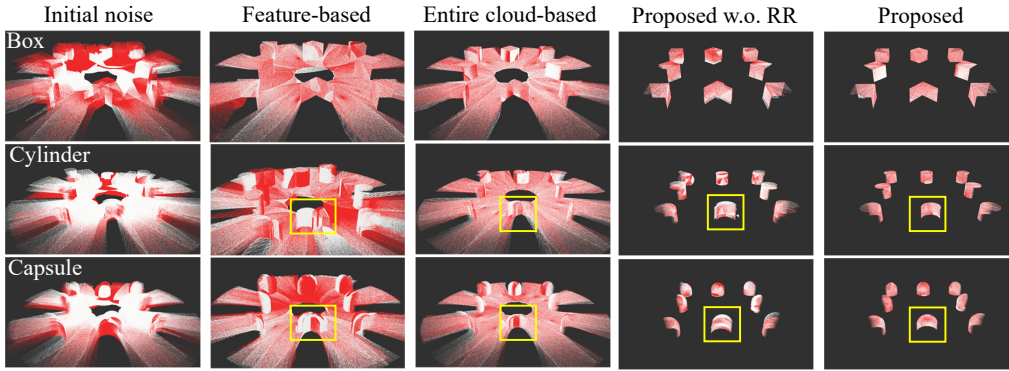


Fig. 7. Comparison of multiple LiDAR point clouds before and after optimization in set 1. The white clouds represent reference LiDAR observations, and the red clouds represent calibration LiDAR observations. The conventional method that uses the entire cloud prioritizes matching the floor, resulting in a translation error when matching the object. The feature-based method has a larger error when using a cylinder or capsule than when using box. The proposed method without rough refinement converges to the wrong position. However, the proposed method accurately aligns two LiDAR point clouds regardless of the object type. The yellow squares indicate the areas where these differences are most pronounced.

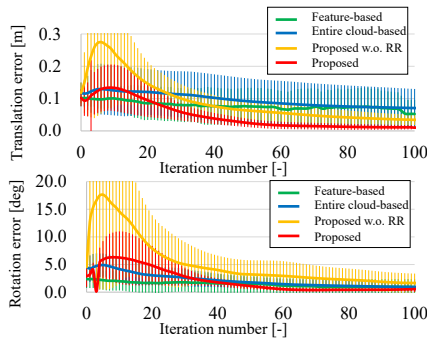


Fig. 8. Comparison graph shows the iteration number and extrinsic parameter errors for each method. The results of the graph indicate that rough refinement using planes leads to faster convergence. It also demonstrates convergence in fewer iterations and higher accuracy compared to conventional methods that use the entire point cloud.

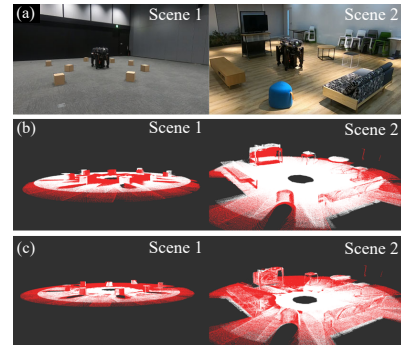


Fig. 9. (a) Real world experiment environment. Comparison of (b) initial 3D map and (c) optimized 3D map after calibration. The white point clouds represent observations from the front LiDAR, which serves as the reference, while the red point clouds represent observations from the rear LiDAR.

boxes that were 0.25 m height and these were placed around the robot. The various type of objects were placed around the robot in scene 2.

To evaluate the proposed noise removal module, we applied it to the object point clouds and calculated the precision, recall and F-score. To obtain ground truth for the box point cloud, we performed plane removal and manually cropped the point cloud based on the size of the box. For comparison, we compared these values with traditional methods which are SOR [15] and DSOR [17]. We used the same parameters $(k, C_s) = (20, 0.01)$ for all methods and used the same parameter $C_r = 3.0$ for DSOR and proposed method.

2) *Noise filter module evaluation:* Table I summarizes the precision, recall and F-score for each method, and Figure 10 shows the comparison results of the noise point clouds after applying each method in Scene 1. The table shows that the proposed method had the highest F-score. The SOR achieved high recall, but misrecognizes distant sparse floor surfaces, resulting in low precision. The DSOR reduced misrecognition of distant floor surfaces but had low recall because it did not fully remove dense noise at the center of

the observations. It was confirmed that the proposed method correctly removes the high-density noise near the center of observation with few misidentification.

3) *3D map evaluation:* The Table II shows the average metric error before and after optimization. It indicates that the proposed method reduces the average metric error. These results show that the two point clouds overlap with high precision and are integrated into one coordinate system.

Figures 9(b) and 9(c) show the point clouds before and after optimization for each scene. The figures demonstrate that the proposed method can correctly integrate observations from two LiDARs with non-overlapping FoV in real world.

TABLE I
COMPARISON OF PRECISION AND RECALL FOR EACH METHODS.

Methods	Precision	Recall	F-score
SOR	0.036	0.646	0.068
DSOR	0.703	0.312	0.433
Proposed	0.638	0.718	0.676

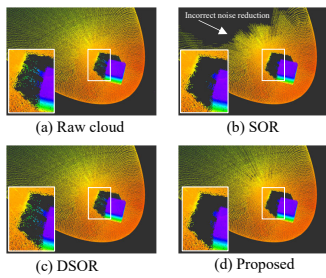


Fig. 10. Comparison of noise in point clouds for each method. The raw cloud shows noise from the top of the box to the floor (indicated by the white square). SOR correctly removes noise but misidentifies sparse areas away from the sensor as noise (indicated by the white arrow). DSOR has fewer misidentification but does not detect enough noise to be removed. The proposed method removes the noise correctly with few misidentification.

TABLE II
AVERAGE METRIC ERROR BEFORE AND AFTER OPTIMIZATION.

Scene	Metric Error [m]	
	Initial	Optimized
Scene 1	0.020	0.0010
Scene 2	0.025	0.0008

V. CONCLUSION

This paper proposes a calibration method for multiple LiDARs with non-overlapping fields of view. By leveraging floor information and point clouds obtained during operation, we achieve precise calibration of LiDARs directed towards the ground. Additionally, we incorporate a noise removal module that takes into account the scanning pattern, effectively reducing noise. Through simulation-based evaluations, we demonstrate the ability of our method to correct up to 92% of randomly given noise when observing the floor and various objects using two or four LiDARs. Furthermore, using real hardware, we confirm the superior noise reduction effects of the noise removal module compared to conventional methods, contributing to the consistent creation of a 3D map. Our proposed method is cost-effective and does not require specialized calibration equipment, making it well-suited for future robot inspection facilities. However, it is important to note that our method assumes operation in flat environments and may experience decreased accuracy in irregular terrains such as outdoor settings. In our future research, we plan to focus on expanding the system to include multimodal devices integrated into robots.

REFERENCES

- [1] Z. Liu, F. Zhang, and X. Hong, "Low-cost retina-like robotic lidars based on incommensurable scanning," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 58–68, Feb. 2021.
- [2] B. Xue, J. Jiao, Y. Zhu, L. Zhen, D. Han, M. Liu, and R. Fan, "Automatic calibration of dual-lidars using two poles stickered with retro-reflective tape," in *Proc. 2019 IEEE International Conference on Imaging Systems and Techniques (IST)*, Abu Dhabi, 2019, pp. 1–6.
- [3] J. Zhang, Q. Lyu, G. Peng, Z. Wu, Q. Yan, and D. Wang, "Lb-l2l-calib: Accurate and robust extrinsic calibration for multiple 3d lidars with long baseline and large viewpoint difference," in *Proc. 2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, 2022, pp. 926–932.

- [4] X. Liu and F. Zhang, "Extrinsic calibration of multiple lidars of small fov in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2036–2043, Apr. 2021.
- [5] 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, May 2022.
- [6] 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, July 2021.
- [7] J. Domhof, J. F. Kooij, and D. M. Gavrilu, "An extrinsic calibration tool for radar, camera and lidar," in *Proc. 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, 2019, pp. 8107–8113.
- [8] J. Lin, X. Liu, and F. Zhang, "A decentralized framework for simultaneous calibration, localization and mapping with multiple lidars," in *Proc. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, NV, 2020, pp. 4870–4877.
- [9] L. Heng, "Automatic targetless extrinsic calibration of multiple 3d lidars and radars," in *Proc. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, NV, 2020, pp. 10669–10675.
- [10] D. Chang, R. Zhang, S. Huang, M. Hu, R. Ding, and X. Qin, "Versatile multi-lidar accurate self-calibration system based on pose graph optimization," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, June 2023.
- [11] J. Levinson and S. Thrun, "Unsupervised calibration for multi-beam lasers," in *Proc. Experimental Robotics: The 12th International Symposium on Experimental Robotics*, Berlin, 2014, pp. 179–193.
- [12] E. Fernández-Moral, J. Gonzalez-Jimenez, and V. Arévalo, "Extrinsic calibration of 2d laser rangefinders from perpendicular plane observations," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1401–1417, May 2015.
- [13] J. Jiao, Q. Liao, Y. Zhu, T. Liu, Y. Yu, R. Fan, L. Wang, and M. Liu, "A novel dual-lidar calibration algorithm using planar surfaces," in *Proc. 2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, 2019, pp. 1499–1504.
- [14] J. Jiao, Y. Yu, Q. Liao, H. Ye, R. Fan, and M. Liu, "Automatic calibration of multiple 3d lidars in urban environments," in *Proc. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macao, 2019, pp. 15–20.
- [15] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, pp. 345–348, Aug. 2010.
- [16] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Proc. 2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, 2011, pp. 1–4.
- [17] X. Zhong, R. Lu, L. Li, X. Wang, and Y. Zheng, "Dsor: A traffic-differentiated secure opportunistic routing with game theoretic approach in manets," in *Proc. 2019 IEEE Symposium on Computers and Communications (ISCC)*, Barcelona, 2019, pp. 1–6.
- [18] M. H. Prio, S. Patel, and G. Koley, "Implementation of dynamic radius outlier removal (dror) algorithm on lidar point cloud data with arbitrary white noise addition," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–7.
- [19] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, "Cnn-based lidar point cloud de-noising in adverse weather," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2514–2521, Dec. 2020.
- [20] A. Seppänen, R. Ojala, and K. Tammi, "4denoisenet: Adverse weather denoising from adjacent point clouds," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 456–463, Sep. 2022.
- [21] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," vol. 1611, pp. 586–606, Apr. 1992.
- [22] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, Apr. 1992.
- [23] A. Rényi, "On measures of entropy and information," vol. 4, pp. 547–562, Jan. 1961.
- [24] N. Takasugi, M. Kinoshita, Y. Kamikawa, R. Tsuzaki, A. Sakamoto, T. Kai, and Y. Kawanami, "Real-time perceptive motion control using control barrier functions with analytical smoothing for six-wheeled-telescopic-legged robot tachyon 3," 2023.