

# Robust Ground Constrained SLAM for Mobile Robot with Sparse-channel LiDAR

Shaocong Wang<sup>ID</sup>, Fengkui Cao<sup>ID</sup>, Ting Wang<sup>ID</sup>, Shiliang Shao<sup>ID</sup>,  
and Lianqiang Liu<sup>ID</sup>, Senior Member, IEEE

**Abstract**—Although LiDAR SLAM has been widely studied over the past decades, it suffers from degradation situation, particularly for sparse-channel LiDAR. This study proposes a hybrid LiDAR SLAM method combining scan-to-LiDAR-centric sliding submap ICP and scan-to-map feature points matching to adapt to unstructured environments and degraded corridors. First, a ground constraint module based on the principal normal of the planar ground is proposed to avoid vertical drift. To address the lack of geometrical features in corridors and environments with large-scale featureless walls, intensity features combined with geometrical features are adopted for scan-to-map registration. Finally, a joint optimization module integrates the feature correlation, the IMU pre-integration, and the ground constraint into a factor graph to perform low-drift SLAM. Extensive experiments are conducted on the GroundRobotDataset, which include challenging degradation scenarios such as long corridor, obstacle crossing and unstructured outdoor environment with large-size walls in weak texture. Our method achieves better accuracy and robustness than state-of-the-art methods, particularly in inhibiting vertical drift.

**Index Terms**—SLAM, Mobile robot, Ground constraint, Degraded environment, Sparse-channel LiDAR

## I. INTRODUCTION

SIMULTANEOUS localization and mapping (SLAM) is a classic technology for autonomous vehicle to accurately estimate their trajectories and model the surrounding environment, which is essential for the perception, control, and planning modules of autonomous vehicles in GPS-denied environments [1]–[3]. For the perception module, tasks such as dynamic object detection and tracking often require the precise position and velocity of the intelligent vehicle as input [4], [5]. For the control and path planning module, the vehicle needs to know its real-time location and the model of surrounding environment to perform global path optimization and local control [6], [7].

Since LiDAR is not sensitive to illumination and own high ranging accuracy, using LiDAR for location and mapping has

This research is supported by the National Natural Science Foundation of China (Grant No. 62203091 and U20A20201), China postdoctoral Science Foundation (Grant No. GZB20230804), Autonomous Project of State Key Laboratory of Robotics (Grant No. 2024-Z09)(Corresponding author: Fengkui Cao, Ting Wang).

S. Wang, F. Cao, T. Wang, S. Shao and L. Liu are with the State Key Laboratory of Robotics at Shenyang Institute of Automation, and Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China. S. Wang is with the University of Chinese Academy of Sciences, Beijing, China. F. Cao is also with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China. (email: caofengkui@sia.cn, wangting@sia.cn)

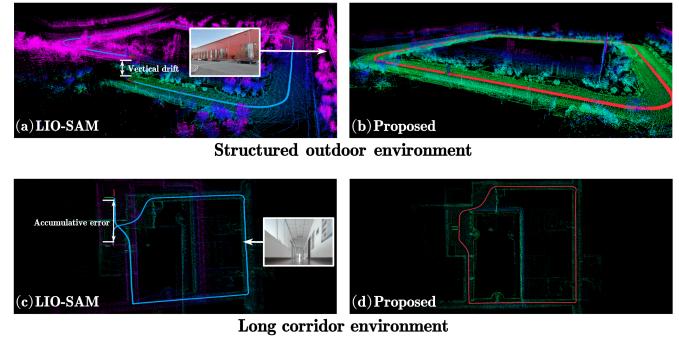


Fig. 1. Examples of degraded environments encountered by different methods. (a)The trajectory of LIO-SAM in a structured outdoor environment, which have serious drift in the vertical direction. (c)The trajectory of LIO-SAM in a long corridor environment, which suffers an obvious degradation. (b), (d)The trajectories of our method in degraded environments, which shows that our method has better performance.

been extensively studied in intelligent vehicle and robotic communities [8]–[10]. Although long-range, high-resolution 3D LiDAR, such as Velodyne VLS-128 and Ouster OS1-128, can capture the fine details of an environment, they are unsuitable for low-cost mobile robots. Thereby, this paper focuses on LiDAR SLAM with sparse-channel LiDAR, such as Velodyne VLP-16.

However, the environmental perception with low-cost LiDAR suffers from several problems. Most of all, it is difficult to capture the vertical texture and structure of the environment due to the low vertical resolution of low-cost LiDAR. Over-structured outdoor environments with sparse and confusing features present challenges for geometrical-feature-based algorithms to converge in the vertical direction. In indoor environments, long corridors usually have similar geometric structures, which results in similar scanning data among continuous scans and confuse the scan-to-scan tracking. In addition, point clouds are clustered in a small area on either side of the robot when the robot is working in a narrow corridor, which can also lead to the vertical drift of the robot's state. Therefore, this work aims at improving the robustness of SLAM with sparse-channel LiDAR in the above challenging situations.

Currently, there exist many excellent LiDAR SLAM methods focusing on capturing environmental structures. However, their performance is somewhat reduced with sparse-channel LiDAR, particularly in over-structured and degradation scenarios. To improve the robustness and output frequency of

the system, many researchers have focused on Lidar-Inertial Odometry (LIO) [11]–[13]. Although the high-precision IMU can compensate the vertical drift and degradation of the LiDAR Odometry (LO) in some degree, the accelerometer can only provide the accurate state estimation in a short time due to the existence of noise. Therefore, the continuous operation of the system still depends on the performance of the LO, which cause LIO methods adopting sparse-channel LiDAR remain challenging in over-structured and degradation environment as shown in Fig. 1. As illustrated in Fig. 1 (a), LIO-SAM [11] has a large drift in the vertical direction when it works in a structured outdoor environment with sparse-channel LiDAR, because it cannot handle the over-structured environments well. Due to the low vertical resolution of spares-channel LiDAR, SLAM directly based on ICP [14] and its variants [15]–[17] also suffer from obvious vertical drift. As illustrated in Fig. 1 (c), it is also challenging for LIO-SAM to steadily operate in featureless long corridors with sparse-channel LiDAR. In confusing long corridors, most LiDAR SLAM methods cannot correctly evaluate whether their state has changed, resulting in the low accuracy of state estimation and distortion of mapping.

To solve these problems, this paper proposes a LiDAR SLAM system with ground constraint that targets the inhibition of drift and degradation of the SLAM system caused by sparse-channel LiDAR. Fig. 1 (b) and (d) shows the performance of our method in degraded environments. The key contributions of this paper are as follows:

- Propose a hybrid LiDAR SLAM method combining scan to LiDAR-centric sliding submap ICP and scan-to-map feature points matching, which combines the advantage of ICP and feature points matching.
- Propose a ground constraint module based on the principal normal of the planar ground, which can effectively inhibit the vertical drift caused by sparse-channel LiDAR. In addition, intensity features combined with geometrical features are adopted for scan-to-map registration to offset the lack of geometrical features in featureless environments.
- To evaluate the performance and limitations of the proposed method, it is extensively tested on GroundRobot-Dataset<sup>1</sup> showing the superiority of the proposed method in inhibiting vertical drift compared with baseline methods. Our code will be publicly acquired at: <https://github.com/ROBOT-WSC/RGC-SLAM>.

The remainder of this paper is organized as follows: Sec. II introduces related works. Sec. III describes the details of the proposed method. Sec. IV describes the experimental setup and the analysis of the results. The article is summarized in Sec. V.

## II. RELATED WORKS

### A. LiDAR-based SLAM

LiDAR Odometry has been extensively studied over the past decades. The most conventional point cloud registration

method is the Iterative Closest Point (ICP) [14]. Though LiDAR SLAM methods [15]–[17] based on ICP and its variants achieve high performance of mapping, registration based on raw point cloud usually needs high time cost. To balance accuracy and efficiency, using feature points to replace the raw point cloud to achieve pose tracking is a relevant topic. LOAM [18], [19] extracts plane and edge points from the raw point cloud to estimate the robot state quickly and accurately, which provided an excellent reference for features-based LiDAR SLAM. LeGO-LOAM [20] introduced the extraction of ground points and a pose graph for global optimization into LOAM. F-LOAM [21] proposed a light-weight framework of LOAM to improve its efficiency. Choi [22] extracted more extensive features of the point cloud based on quadratic surface, which enabled the algorithm to handle sparse point cloud scenarios. MULLS [23] and the method proposed by Guo [24] focused on utilizing PCA to extract more abundant and stable feature points, which have also been widely recognized.

To compensate for the disadvantages of LiDAR, such as its low ranging frequency and motion distortion, many researchers have focused on fusing it with other sensors. LiDAR-Inertial Odometry is the most commonly employed fusion method. LIO-SAM [11] utilized the pre-integration of IMU [25] to constrain the pose increment of the system based on LeGO-LOAM, which improved the frequency and robustness of the system. LIO-Mapping [26] referred to the idea of sliding windows in VINS-Mono [27] to tightly couple IMU and LiDAR to achieve better performance. In contrast to the above methods based on nonlinear optimization, FAST-LIO [12] employed an improved iterative extended Kalman filter to achieve multi-sensor fusion. FAST-LIO2 [13] introduced iKDtree into FAST-LIO, which improved the efficiency of mapping. Based on DLO [15], DLIO [28] achieved a good balance between operation efficiency and accuracy through a novel continuous time motion correction.

### B. Inhibiting degradation LiDAR SLAM

LO and LIO methods are highly dependent on the performance of point cloud registration. However the failure of registration may cause drift and degradation of the whole system, which is particularly prominent in sparse-channel LiDAR. A common way to inhibit the drift of point cloud registration is to use place recognition or loop closing to reduce accumulated errors [29], [30]. Although reliable loop closing can ensure the global consistency of the SLAM system, the drift usually distorts the map corrected by global optimization when the cumulative error is too large. Therefore, many researchers have focused on improving the robustness of point cloud registration in challenging scenarios.

Many researchers have made efforts in recent years to inhibit the vertical drift of LiDAR SLAM. It is a common strategy to add gravity constraint for SLAM systems [31]–[33] to inhibit vertical drift. However, these methods are highly dependent on the accuracy of IMU, which is expensive. Several studies have effectively inhibited vertical drift by defining a global ground [34], [35] adding assumptions of flat ground, which cannot steadily operate in complex environments. GCLO [36] introduced dynamic ground constraint

<sup>1</sup>[https://drive.google.com/drive/folders/1bt9vWPVgTF8I8JXSUO-Dpi3n2vOmG6t9?usp=drive\\_link](https://drive.google.com/drive/folders/1bt9vWPVgTF8I8JXSUO-Dpi3n2vOmG6t9?usp=drive_link)

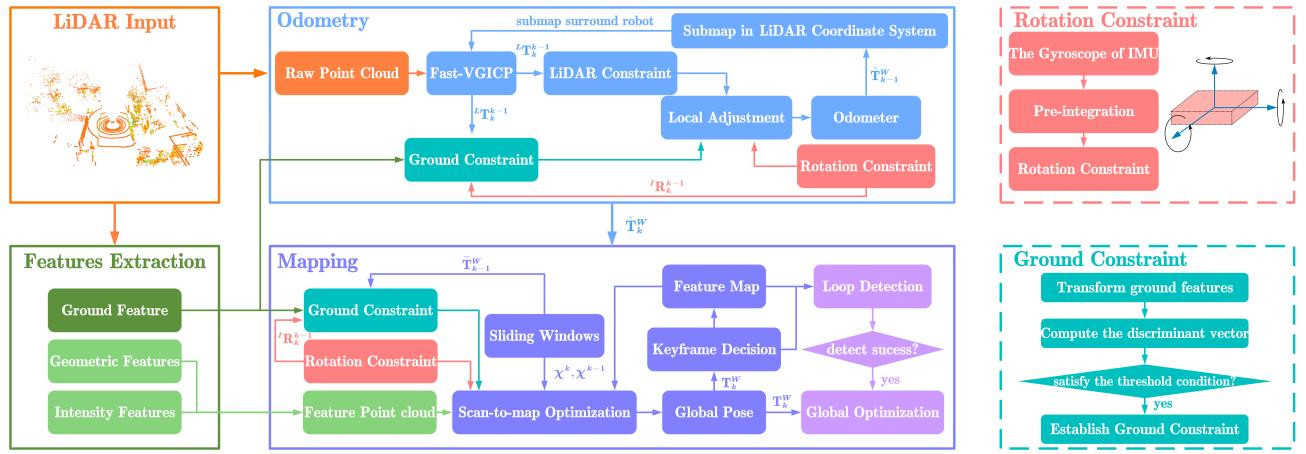


Fig. 2. System overview. The system can be divided into feature extraction, odometry, and mapping. In Feature Extraction module, ground features are extracted from the raw point cloud to construct ground constraints, while geometric and intensity features are extracted for map construction in the mapping stage. In Odometry module, the incremental pose of the robot is first estimated preliminarily using a scan-to-submap Fast-VGICP. Subsequently, the pose increment undergoes local adjustment based on ground and rotation constraints. In Mapping module, a globally consistent map based on feature points and keyframes is maintained. A scan-to-map optimizer based on sliding windows is used to align the robot's pose with the feature map, which considers ground and rotation constraints during the optimization process. Additionally, global optimization based on pose graph and keyframes are used to ensure the global consistency of the trajectory

by fitting ground CP [37] parameters in real time. Although this method could effectively operate in changing ground planes, the ground modeling method makes it less adaptive for general scenarios. FEVO-LOAM [38] inhibited drift by constructing a parallel relationship between the extracted plane features and the ground, but this method could only be applied to scenes with rich structural features, it is not suitable for sparse-channel LiDAR. Chen [39] added SE(2) constraint to a SLAM system based on a vehicle model, which could fade the changes of Z-axis displacement, roll and pitch angles in the state estimation process. However, it is unsuitable for special mobile robots that operate in the field.

The similar geometric structures of the environment primarily cause the degradation of the LO in the direction of motion. Kurz [40] set reflective road signs in the workspace to prevent LO's degeneration. This method is very effective, but it requires the artificial addition of features and is unsuitable for unknown environments. Similarly, Dai [41] focused on extracting some special structures in tunnels as features, which is only applicable to specific scenarios. Recently, using reflection intensity has become an effective scheme to resist degraded environments such as long corridors and caves [42], [43]. However, the reflection intensity of LiDAR beams is easily affected by the incidence angle and range. Therefore, detecting stable features based on intensity is also a concern.

Inspired by the above works, our method focuses on how to inhibit the vertical drift and degradation of sparse-channel LiDAR SLAM. Considering the limitation of the feature-based method, we introduce scan-to-submap ICP for initial state estimation. To further refine the pose tracking and mapping, geometrical features combined with intensity features are used in the scan-to-map mapping process. To remit severe vertical drift introduced by the low vertical resolution of LiDAR, a constraint module based on the principal normal of the planar ground is proposed. Note that, ground constraints

are dynamically constructed for adjacent states of the robot utilizing ground parameters and gyroscope increments, rather than establishing a global ground or presuming that all ground surfaces are flat like [34]–[36], which can adapt to outdoor environments with uneven roads.

### III. METHODOLOGY

#### A. System Overview

An overview of the proposed method is presented in Fig. 2. The IMU employs gyroscope for pre-integration [25], which is used for establish the rotation constraint. The proposed method can be divided into three parts: feature extraction, odometry, and mapping. Ground, geometric and intensity features are extracted for state estimation in feature extraction. In odometry, the state increment of robot is first estimated by ICP of scan to LiDAR-centric sliding submap. Then, a local adjustment based on ground constrain is applied to optimize the registration result. The output of the odometry is considered as a prior by mapping. During mapping, a feature map based on point cloud features and key frames is built, and the global state is optimized by aligning the features into it. At the same time, the ground and IMU rotation constraints are added into the optimizer. For convenience, we summarize some important notations in Table I.

#### B. Calibration and Synchronization of Sensors

Since our method is a loosely coupled LiDAR-Inertial system, the gyroscope bias is obtained via offline calibration and is not optimized online. Additionally, we reference the method proposed by Fangcheng Zhu [44] to calibrate the extrinsic transform of the LiDAR and IMU.

The data frequency of the LiDAR is about 10 Hz, whereas the data frequency of the IMU is approximately 200 Hz, so synchronization between different sensors is necessary. Similar

TABLE I  
DEFINITION OF SOME NOTATIONS

Notations	Meaning
$L$	LiDAR coordinate system
$W$	World coordinate system
$I$	IMU coordinate system
${}^kG$	The ground feature corresponding to frame $k$
${}^k\mathbf{p}_i^l$	The $i$ -th point of the $l$ -th line in the scan of the frame $k$
$\mathbf{T}_k^{k-1}$	The transformation matrix from frame $k$ to frame $k-1$
${}^L\mathbf{T}_k^{k-1}$	The transformation matrix obtained by raw point cloud
$\hat{\mathbf{T}}_k^W$	The output of odometry in frame $k$
$\mathbf{T}_k^W$	The transformation matrix from frame $k$ to world

to VINS-Mono [27], instead of hardware synchronization, we synchronize the timestamps of the IMU and LiDAR at the software level. The high-frequency gyroscope data are interpolated according to the adjacent LiDAR scan timestamp, which can be directly applied to the subsequent pre-integration of the angular velocity.

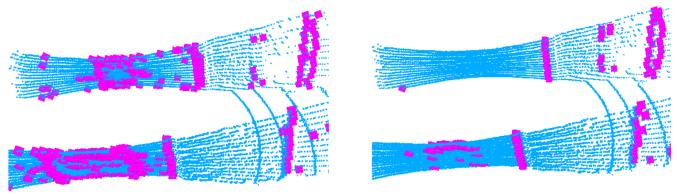
### C. Feature Extraction

*1) Ground feature extraction:* Before extracting the ground features, we need to extract the ground points in the input point cloud roughly. We referring to the ground segmentation method proposed in [45]. Compared with the RANSAC algorithm, this method has a higher speed, which is particularly suitable for the preliminary extraction of ground features. After the ground segmentation is finished, the ground point set  ${}^G P = \{{}^G p_1, {}^G p_2 \dots {}^G p_N\}$  in the  $k$ -th frame can be obtained. The discriminant conditions for ground points in the original point cloud of frame  $k$  are defined as equation (1) :

$$\begin{cases} \left| \left\| {}^k p_i^l \right\|_2 - d^l \right| \leq r_{th} \left( 1 + \frac{l}{l_{max}} \right) \\ {}^k z_i^l \leq h \end{cases} \quad (1)$$

where,  ${}^k z_i^l$  is the Z-axis coordinate of  ${}^k p_i^l$ ;  $h$  is the installation height of LiDAR;  $d^l$  is the ground point range calculated by the  $l$ -th root line of LiDAR based on  $h$ ;  $l_{max}$  is the maximum value of the LiDAR root line utilized to segment the ground, and  $r_{th}$  is a fixed parameter.

The ground feature extracted at the  $k$ -th frame is an 11-dimensional vector  ${}^kG$ , which includes three orthogonal three-dimensional vectors  ${}^k v, {}^k m, {}^k n \in \mathbb{R}^3$ ; the distance from LiDAR to the ground  ${}^k h$ , and ground confidence  ${}^k s$  for subsequent optimization. To extract these parameters, PCA technique is performed on the point set  ${}^c P$ . Unlike the usual PCA, we assign a weight to each point negatively related to  $l$ , which enables the ground feature to be concentrated in the robot's vicinity. Three orthogonal eigenvectors  ${}^k v, {}^k m, {}^k n$  can be obtained by PCA, whose corresponding eigenvalues are  ${}^k \lambda_v > {}^k \lambda_m > {}^k \lambda_n$ . We take the eigenvector  ${}^k n$  with the smallest eigenvalue as the normal vector of the ground at the  $k$ -th frame, while the two mutually orthogonal vectors  ${}^k v$  and  ${}^k m$  can represent a plane parallel to the ground. In addition, we define a direction vector  ${}^k u$  formed by the ground center



**(a) Before compensating**

**(b) After compensating**

Fig. 3. The extracted intensity feature points. Intensity feature points with pink color. (a) The feature points before intensity compensating. (b) The feature points after intensity compensating.

${}^g p_{center}$  and LiDAR's origin, and then the direction of  ${}^k n$  can be normalized according to the direction of the vector  ${}^k u$ .

For the ground point  ${}^G p_j$ , we can determine  ${}^k s_j$  as the confidence value, which is determined by the perpendicularity between  ${}^k n$  and the vector made by  ${}^G p_j$  and  ${}^G p_{center}$ .  ${}^k s_j$  is defined as equation (2) :

$$^k s_j = 1 - r_{th2} \left| {}^k n \odot \frac{({}^G p_j - {}^G p_{center})}{|{}^G p_j - {}^G p_{center}|} \right| \quad (2)$$

where,  $r_{th2}$  is an amplification factor, whose function is to increase the differentiation of each ground points.

Since the projection of the ground point on  ${}^k n$  can be expressed as the height of the LiDAR from the ground.  ${}^k h$  and  ${}^k s$  are defined equation (3) :

$$\left\{ \begin{array}{l} {}^k h = \frac{\sum\limits_{j=1}^N \left( {}^k s_j \left( {}^k n \odot {}^G p_j \right) \right)}{\sum\limits_{j=1}^N {}^k s_j} \\ {}^k s = \frac{\sum\limits_{j=1}^N {}^k s_j}{N} \end{array} \right. \quad (3)$$

where,  ${}^k s$  is the average value of  ${}^k s_j$ , which is used as the weight of the ground constraint in subsequent optimizations.

2) *Point cloud feature extraction*: For the extraction of geometric features, our extraction method is similar with LOAM [18]. In the  $k$ -th frame, the edge point set  $\alpha^k$  and the surface point set  $\beta^k$  are extracted. The difference from LOAM is that the depth factor  ${}^k f_i^l$  is added to each point  ${}^k p_i^l$  when we calculate the curvature.  ${}^k f_i^l$  is negatively related to the range of  ${}^k p_i^l$  and can be applied to subsequent state estimate.

In degraded scenarios, such as long corridor, there are only a few edge feature points in the geometric features, which can cause incorrect estimation of the robot's pose. To improve the environmental adaptation of this system, we additionally extract reflection intensity features for mapping.

Owing to the low resolution of the reflection intensity, we only extract edge features from the region with a steep reflection intensity change rather than directly relying on the value of the intensity. Similar to extracting edge points in LOAM, we compress the curvature calculation from the three-dimensional Euclidian space to the one-dimensional intensity space to obtain the intensity curvature  $k^I l_i$  of  $k^p l_i$ . If a point has larger intensity curvature, it will be considered as an intensity

edge feature, and the qualified point will be added to  $\eta^k$ , the set of intensity feature points.

Since the reflection intensity is affected by the range of the measuring point and the cosine value of the laser incidence angle, as shown in Fig. 3 (a), it is easy to produce a large distortion when the range is small and the incidence angle is close to  $90^\circ$ . The distorted reflection intensity can cause incorrect feature points following the robot continuously. Therefore, we compute the cosine of  ${}^k p_i^l$ 's incidence angle.  ${}^k \cos \theta_i^l$  is defined as equation (4) :

$$\begin{aligned} {}^k \cos \theta_i^l &= \left| \frac{{}^k n_i^l \odot {}^k p_i^l}{\| {}^k n_i^l \|_2 \| {}^k p_i^l \|_2} \right|, \\ {}^k n_i^l &= ({}^k p_i^l - {}^k p_{i-5}^l) \otimes \left( {}^k p_i^l - \frac{{}^k p_{i-5}^l + {}^k p_{i+5}^l}{2} \right) \end{aligned} \quad (4)$$

where,  ${}^k n_i^l$  is the normal vector of  ${}^k p_i^l$ ;  ${}^k p_{i-5}^l$  and  ${}^k p_{i+5}^l$  are two points close to  ${}^k p_i^l$  on the same scan line, which can improve the efficiency of calculating  ${}^k n_i^l$ .

For computational efficiency, we only compensate for points whose range is less than  $r_{th5}$ , and their  ${}^k I_i^l$  values are multiplied by the sum of  ${}^k \cos \theta_i^l$  and  $r_{th6}$  on the original basis, where  $r_{th6}$  is adopted to prevent  ${}^k \cos \theta_i^l$  from being too small. The feature points extracted after the compensation are presented in Fig. 3 (b). The optimization weights of these intensity features can also be adjusted in this way.

#### D. Odometry

As illustrated in Fig. 2, the process of odometry can be divided into tracking and local adjustment. The optimization variable is  $\chi_k^{k-1} = [R_k^{k-1} \ t_k^{k-1}]^T$ , where  $R_k^{k-1}$  and  $t_k^{k-1}$  represent the state increment between adjacent states  $\chi_k$  and  $\chi_{k-1}$  respectively, whose cumulative process is defined as equation (5) :

$$\check{T}_k^w = \begin{bmatrix} \check{R}_{k-1}^w R_k^{k-1} & \check{t}_{k-1}^w + \check{R}_{k-1}^w t_k^{k-1} \\ 0 & 1 \end{bmatrix} \quad (5)$$

where,  $\check{T}_k^w$  is the output of odometry in frame k;  $\check{R}_{k-1}^w$  and  $\check{t}_{k-1}^w$  are the rotation and translation components of  $\check{T}_{k-1}^w$ .

1) *Scan-to-submap tracking*: To obtain more accurate and robust increment state estimation in LO, we implement scan to LiDAR-centric sliding submap tracking based on the Fast VGICP [46]. At the k-th frame, we select some historical keyframes in a sliding window according to the distance increment and angle increment. The point clouds in these keyframes are uniformly converted to the LiDAR coordinate system of the k-1-th frame, as follows:

$$P_{sub}^{k-1} = \bigcup_i^N \left( (\check{R}_{k-1}^w)^{-1} (\check{R}_i^w P^i + \check{t}_i^w) - (\check{R}_{k-1}^w)^{-1} \check{t}_{k-1}^w \right) \quad (6)$$

where  $P_{sub}^{k-1}$  represents the submap in the LiDAR coordinate system of the k-1-th frame; N represents the size of the sliding window;  $P^i$  represents the input point cloud corresponding to the i-th key frame, and  $\check{R}_i^w$ ,  $\check{t}_i^w$  represents the output of odometry corresponding to the i-th keyframe.

We then utilize Fast-VGICP to complete the estimate of the state increment  ${}^L T_k^{k-1}$ , which can be expressed as a maximum likelihood estimate defined as equation (7) :

$${}^L \check{T}_k^{k-1} = \varphi({}^L \check{T}_k^{k-1}, P_s^k, P_t^k) \quad (7)$$

where  $P_s^k$  is the source cloud, which equal to  $P^k$  in our method;  $P_t^k$  is the target cloud, which equal to  $P_{sub}^{k-1}$  in our method;  ${}^L \check{T}_k^{k-1}$  is a prior information, which can be divided into the rotation provided by the pre-integration result  ${}^L R_k^{k-1}$  of the gyroscope and the translation provided by the constant velocity model based on  $t_{k-1}^{k-2}$ .

The specific form of  $\varphi({}^L \check{T}_k^{k-1}, P_s^k, P_t^k)$  is as equation (8) :

$$\begin{aligned} \varphi({}^L \check{T}_k^{k-1}, P_s^k, P_t^k) &= \arg \min_{{}^L T_{k-1}^k} \sum_i^{N_S} \left( N_i \tilde{d}_i^T \tilde{C}_i^{-1} \tilde{d}_i \right) \\ \tilde{d}_i &= \frac{\sum_j^{N_T} p_j^T}{N_i} - {}^L T_k^{k-1} p_i^s, \\ \tilde{C}_i &= \frac{\sum_j^{N_T} C_j^T}{N_i} + {}^L T_k^{k-1} C_i^s ({}^L T_k^{k-1})^T \end{aligned} \quad (8)$$

where,  $p_i^s$  is the i-th point in the source cloud;  $p_i^T$  is the j-th point in the target cloud;  $C_i^s$  and  $C_i^T$  are their corresponding covariance;  $N_i$  is the number of points adjacent to  $p_i^s$ , and  ${}^L \check{T}_k^{k-1}$  is the initial value in the optimization process.

After completing the estimation of  ${}^L T_k^{k-1}$ , the residuals of LiDAR constraint can be defined as equation (9) :

$$\begin{aligned} r_L({}^L T_k^{k-1}, \chi_k^{k-1}) &= [\delta R_k^{k-1} \ \delta t_k^{k-1}]^T \\ &= \left[ 2 \left( ({}^L q_k^{k-1})^{-1} \otimes q_k^{k-1} \right) \right. \\ &\quad \left. t_k^{k-1} - {}^L t_k^{k-1} \right] \end{aligned} \quad (9)$$

where,  $q$  is the quaternion expression of the rotation matrix R, and  $\chi_k^{k-1} = [R_k^{k-1} \ t_k^{k-1}]^T$  is the variable to be optimized by odometry in the k-th frame.

2) *Local adjustment based on ground constraints*: Before creating the ground constraint, it is necessary to judge whether the current ground of the robot satisfies the conditions of ground constraint, because the incorrect ground constraint will affect the result of the state estimation. We utilize the  ${}^L T_k^{k-1}$  obtained in the last part as the prior information to make a rigorous judgment on whether the ground plane has changed and whether the ground constraint can be added. The ground normal vector  ${}^k n$  and LiDAR's height  ${}^k h$  at k-th frame are firstly transformed to the LiDAR coordinate system at k-1-th frame, which is defined as equation (10) :

$$[{}^k \dot{n} \ {}^k \dot{h}]^T = \left[ {}^L R_k^{k-1} {}^k n \right. \\ \left. {}^k h + ({}^L R_k^{k-1} {}^k n) \odot {}^L t_k^{k-1} \right] \quad (10)$$

where,  ${}^k \dot{n}$  and  ${}^k \dot{h}$  are the corresponding parameters of  ${}^k n$  and  ${}^k h$  in the LiDAR coordinate system at k-1-th frame.

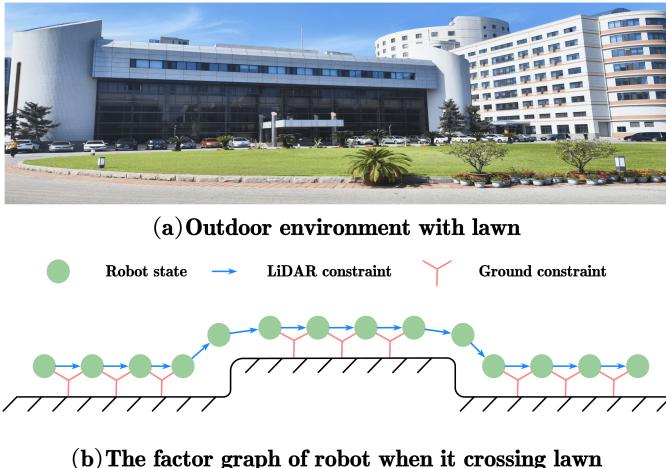


Fig. 4. Example of utilization ground constraint. (a) An outdoor environment with lawn, if the robot wants to cross the lawn, it must climb a step indicated in red circle. (b) The factor graph of robot crossing the lawn, red lines are the ground constraint added to it.

Then we employ  ${}^k\dot{n}$  and  ${}^k\dot{h}$  to construct a five-dimensional discriminant vector  $e_G$  to determine whether the robot is on the same ground as before.  $e_G$  is defined as equation (11) :

$$e_G = [e_1 \ e_2 \ e_3 \ e_4 \ e_5]^T \\ = \begin{bmatrix} \|{}^{k-1}h^{k-1}n - {}^k\dot{h}{}^k\dot{n}\|_2 \\ |{}^{k-1}v \odot {}^k\dot{n}| \\ |{}^{k-1}m \odot {}^k\dot{n}| \\ |roll({}^tR_k^{k-1})| \\ |pitch({}^tR_k^{k-1})| \end{bmatrix} \quad (11)$$

where,  $e_1$  can judge whether the ground normal vector and the LiDAR height change as a whole;  $e_2$  and  $e_3$  can further judge the direction change of the normal vector in two degrees of freedom in order to avoid the singular situation;  $e_4$  and  $e_5$  are utilized to judge the instantaneous changes of the robot's roll angle and pitch angle by utilizing the results of the pre-integration of the gyroscope. When any component of  $e_G$  is larger than the set threshold, we consider that the ground plane has changed, and the ground constraint will not work. To adapt complex terrain, the ground constraint is constructed only when there is no ground change for  $N_G$  consecutive frames. Fig. 4 illustrates the working process of the ground constraint module with the robot crossing a lawn, the ground constraint module is only active on stable ground.

To reflect the global consistency of the same ground, we save robot's pose when the ground is changing. This pose is defined as  $\check{R}_G^W$ , which is utilized as the reference pose of the current ground to constrain the Z-axis displacement at adjacent moments. After each ground change, we find the pose  $\check{R}_H^W$  in the historical ground reference pose that is most similar to the global roll and pitch angles of the current ground reference pose. If the similarity of these two params is within the threshold range, we consider that there is a parallel relationship between the current ground and the historical ground, making  $\check{R}_G^W$  equal to  $\check{R}_H^W$ .

After the above operations, when the  $k$ -th frame satisfies the requirement of applying ground constraint, the residuals of ground constraint can be defined as equation (12) :

$$\begin{aligned} r_G^k ({}^Gz_k^{k-1}, \chi_k^{k-1}) &= [\delta t_k^{k-1} \ \delta \theta_k^{k-1} \ \delta \vartheta_k^{k-1}]_{3 \times 1}^T \\ &= \begin{bmatrix} |{}^{k-1}h - ({}^k h + (\check{R}_{k-1}^G {}^L R_k^{k-1} n) \odot (\check{R}_{k-1}^G t_k^{k-1}))| \\ |{}^{k-1}v \odot (R_k^{k-1} n)| \\ |{}^{k-1}m \odot (R_k^{k-1} n)| \end{bmatrix} \\ &= {}^G z_k^{k-1} = [\check{R}_{k-1}^G {}^L R_k^{k-1} {}^{k-1}G {}^k G]^T \end{aligned} \quad (12)$$

where,  $\check{R}_{k-1}^G$  is equal to  $(\check{R}_G^W)^{-1} \check{R}_{k-1}^W$ ;  $\delta t_k^{k-1}$  constrains the displacement change of the robot in the direction of the current ground normal vector in the ground reference coordinate system;  $\check{R}_{k-1}^G t_k^{k-1}$  represents the displacement change of the robot in the ground reference coordinate system;  $\check{R}_{k-1}^G {}^L R_k^{k-1} n$  represents the ground normal vector that the robot transforms to the ground coordinate system at the  $k$ -th frame;  $\delta \theta_k^{k-1}$  and  $\delta \vartheta_k^{k-1}$  are depend on  ${}^{k-1}v$  and  ${}^{k-1}m$  to constrain the direction of the ground normal vector to complete the constraints on the roll angle and pitch angle of  $R_k^{k-1}$ .

After determining the ground constraint, we complete the maximum a posteriori estimate of  $\chi_k^{k-1}$  by minimizing the objective function defined as equation (13):

$$\begin{aligned} \hat{\chi}_k^{k-1} &= \arg \min \left\{ \|\rho^L r_L^k ({}^L T_k^{k-1}, \chi_k^{k-1})\|^2 \right. \\ &\quad \left. + \|\rho^I r_I^k ({}^I R_k^{k-1}, \chi_k^{k-1})\|^2 + \|\rho^G r_G^k ({}^G z_k^{k-1}, \chi_k^{k-1})\|^2 \right\} \end{aligned} \quad (13)$$

where  $r_I^k ({}^I R_k^{k-1}, \chi_k^{k-1})$  is the IMU rotation constraint, and its function is to utilize  ${}^I R_k^{k-1}$  to constrain  $R_k^{k-1}$ ;  $\rho^L$ ,  $\rho^I$ , and  $\rho^G$  are the weights of each constraint, respectively;  $\rho^L$  depends on the fitness score of the VGICP matching result;  $\rho^I$  is proportional to the Euler angle of  ${}^I R_k^{k-1}$ ;  $\rho^G$  is equal to  $k_s$ , when the ground constraint does not satisfy the application conditions,  $\rho^G$  equals zero. This objective function can be solved as a nonlinear least squares problem, which can be solved by Levenberg-Marquardt algorithm.

### E. Mapping

The main task of the mapping is to estimate the robot's state more accurately utilizing scan-to-map matching. This part is implemented with two parallel threads: scan-to-map optimization and global optimization.

As shown in Fig. 2, the scan-to-map optimization of the mapping is performed in a sliding window to estimate the global pose of the current frame and the previous frames simultaneously at each window. The variable to be optimized for the  $k$ -th frame is  $\chi^k = [R_{k-1}^W \ t_{k-1}^W \ R_k^W \ t_k^W]^T$ , which means that the robot's pose at each frame is optimized twice.

To ensure the system's operational efficiency, scan-to-map optimization did not use the whole cloud like odometry, but adopted the feature points extracted in Section III-C. In order to maintain great global performance, scan-to-map is adopted to match feature points. After completing each sliding window optimization, if the  $k-1$ -th frame can be judged as a keyframe,

its feature points are added to the feature point cloud map, and the selection of keyframes is similar to LeGO-LOAM [20]. In the matching process of the point cloud, the edge and intensity edge points adopt a point-to-line matching mode, whereas the plane points adopt a point-to-plane matching mode. Please refer to LOAM [18] for the specific residual construction process.

To increase the accuracy of the ground constraint model, only the  $k-1$ -th frame in the sliding window is subjected to the ground constraint. In fact, As the pose of  $k-1$ -th frame has already been estimated once in the previous sliding window, this pose can be adopted as prior information to determine the conditions that the ground constraint is applied. Therefore, we extend equation (10) to obtain the equation (14) :

$$\begin{aligned} & \begin{bmatrix} {}^{k-1}\dot{n} & {}^{k-1}\dot{h} \end{bmatrix}^T \\ &= \left[ \begin{array}{c} \left( R_{k-2}^W \right)^{-1} \bar{R}_{k-1}^W {}^{k-1}n \\ {}^{k-1}h + {}^{k-1}\dot{n} \odot \left( \left( R_{k-2}^W \right)^{-1} (\bar{t}_{k-1}^W - t_{k-2}^W) \right) \end{array} \right] \quad (14) \end{aligned}$$

where  $\bar{R}_{k-1}^W$  and  $\bar{t}_{k-1}^W$  are the pose obtained from the previous sliding window. Subsequently, we substitute the calculation results into equation (11), and the discriminant vector  $e_G$  is calculated to determine whether the ground on which the robot is located has changed. Similarly, the residual of ground constraints is similarly extended. Due to space limitations, this is not detailed here.

The final optimization objective function of the scan-to-map optimization is defined as equation (15):

$$\begin{aligned} \chi^k = \arg \min \left\{ \sum_{(i,j)} \left( \left\| \rho_j^{k_i} r_j^{k_i} ({}^M z^k, \chi^k) \right\|^2 \right) \right. \\ \left. + \left\| \rho_I^I r_I^k ({}^I R_k^{k-1}, \chi^k) \right\|^2 + \left\| \rho_G^G r_G^k ({}^G z^k, \chi^k) \right\|^2 \right\} \quad (15) \end{aligned}$$

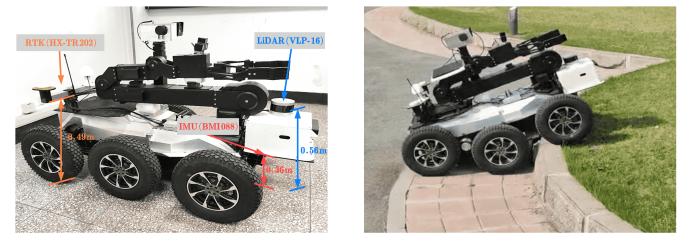
where  $r_j^{k_i} ({}^M z^k, \chi^k)$  represents the constraint generated by the  $j$ -th point of LiDAR in the  $i$ -th frame of the sliding window;  ${}^M z^k$  is the feature map of the corresponding point cloud in the sliding window,  $\rho_j^{k_i}$  is the optimization weight of the feature points;  $r_G^k ({}^G z^k, \chi^k)$  is the ground constraint imposed at the  $k-1$ -th frame, where  ${}^G z^k = [R_G^W \ R_{k-2}^W \ \bar{R}_{k-1}^W \ {}^{k-2}G \ {}^{k-1}G]^T$ ,  $R_G^W$  is the pose of reference ground; the IMU is still used to constrain rotation increments.

With the long-term movement of the robot, it is difficult to estimate its state to avoid accumulated errors. Global optimization based on pose graph is an effective method for suppressing the accumulated errors. With reference to the LeGO-LOAM [20], we construct the keyframe as a pose graph. A keyframe closest to the current frame in the pose graph can be selected, and the ICP algorithm [14] is used to calculate the relative pose between these two frames. After successful matching, we use the relative pose to correct the map and poses of keyframes.

## IV. EXPERIMENTS AND RESULTS

### A. Experiment Setup

In this section, the GroundRobotDataset is used to verify the validity of our method, which is created by our self-developed



(a)Setup of ground robot (b)Crossing obstacle

Fig. 5. Ground robot employed for experiment. (a) The robot utilized in the experiment is equipped with sparse-channel LiDAR, IMU, and RTK, with installation heights of 0.56m, 0.36m, and 0.49m. (b) The process of robot crossing a step of lawn.

TABLE II  
PERFORMANCE IN LONG CORRIDOR ENVIRONMENT.

Sequences	Type	Characteristics	LiDAR
01	Indoor	Office	
02	Indoor	Long corridor	
03	Outdoor	Uneven road, steps, lawn	VLP-16
04	Outdoor	Uneven road, weak texture	
05	Outdoor	Uneven road, field, slope	

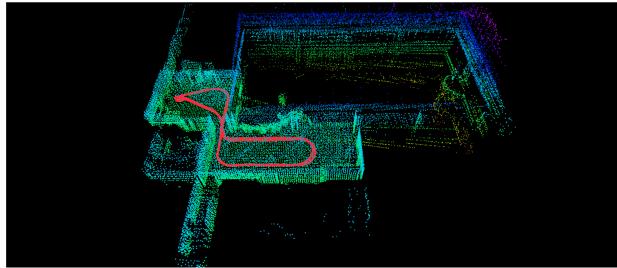
ground robot as shown in Fig. 5 (a), and open-sourced in our previous work [45]. For outdoor sequences, the groundtruth are collected by the high-precision RTK combine point cloud offline registration. For RTK-denied indoor sequences, the groundtruth are collected by the wheel odom combine point cloud offline registration. As shown in Fig. 5 (b), this robot can collect various outdoor data with excellent obstacle-crossing performance. The details of data sequences used in this work are presented in TABLE II.

The Absolute Pose Error(APE) is adopted to evaluate the accuracy and cumulative drift of the proposed method. We calculate the mean, maximum, and root-mean-square errors(RMSE) to analyze the accuracy and robustness of each algorithm quantitatively. To evaluate the ability of the proposed method to inhibit vertical drift, we also calculate the RMSE in the vertical direction.

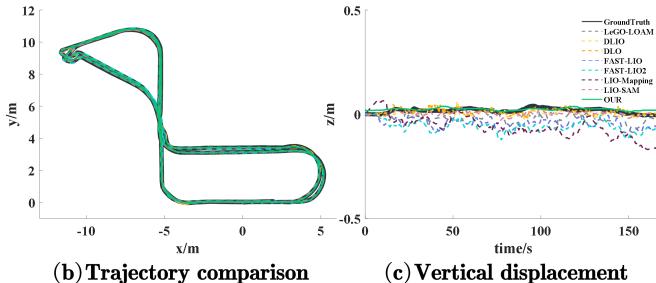
To evaluate the proposed method objectively, loop-closed detection is disabled by default in all experiments below. In addition, we select seven types of SLAM algorithms that have been widely adopted in recent years for comparative experiments, including LiDAR-only SLAM methods such as LeGO-LOAM [20] and DLO [15] and LiDAR-Inertial SLAM methods such as DLIO [28], FAST-LIO [12], FAST-LIO2 [13], LIO-Mapping [26], and LIO-SAM [11]. All the algorithms are deployed on a computer equipped with an Intel i7-10875H CPU for testing, and the test environment is Ubuntu 20.04.

### B. Evaluation in the Indoor Environment

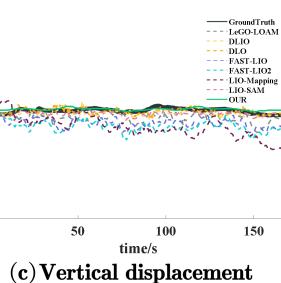
1) *Office environment:* Figure. 6 (a) represents the map and the estimated trajectory established by our method in an office environment, which can be considered as a simple and closed scene. As illustrated in Fig. 6 (b), the trajectories estimated by different algorithms in this environment can fit the ground truth well. Owing to this scene's small-scale and rich structural



(a) Mapping by our method



(b) Trajectory comparison



(c) Vertical displacement

Fig. 6. Office environment. (a) The mapping result of the proposed method is that the red line is the trajectory of the robot. (b) Comparison of trajectories obtained by each algorithm, solid black line is the ground truth. (c) Variation of different trajectories in the vertical direction.

TABLE III  
PERFORMANCE COMPARISON IN SEQUENCE 1.

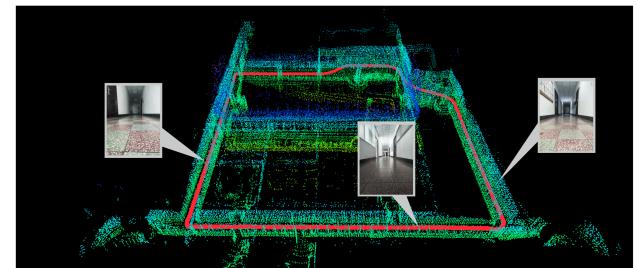
Method	Max	Mean	RMSE	Vertical Error
LeGO-LOAM	0.148	0.040	0.046	0.020
DLO	0.094	0.028	0.033	0.013
DLIO	<b>0.085</b>	<b>0.028</b>	<b>0.032</b>	0.013
FAST-LIO	0.251	0.122	0.131	0.022
FAST-LIO2	0.258	0.140	0.149	0.026
LIO-Mapping	0.174	0.059	0.067	0.048
LIO-SAM	0.608	0.043	0.072	<b>0.009</b>
OURS	0.171	0.038	0.043	0.012

features, none of the algorithms demonstrate too much drift in the vertical direction, as illustrated in Fig. 6 (c).

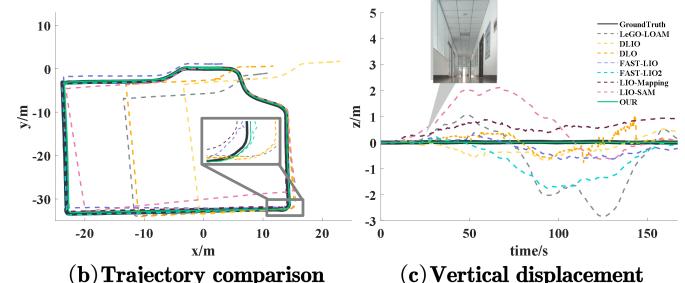
TABLE III presents the performance metrics of different algorithms in this environment. Due to their special scan-to-map matching mechanisms, DLO and DLIO achieve the best performance in this scenario. Our method is ranked next to DLO and DLIO, and its accuracy is even higher than LeGO-LOAM and LIO-SAM with loop-closed detection.

2) *Long corridor environment*: Note sequence 2 is a long corridor with similar geometric structure. As shown in Fig. 7 (b), DLO, DLIO, and LeGO-LOAM have serious degradation in the moving direction of the robot, which is caused by the similar geometric structure in the long corridor. Although LIO-SAM inhibits instantaneous degradation by fusing IMU, IMU cannot correct the degradation errors accumulated for a long time. As shown in Fig. 7 (a) and (b), because of the introduction of intensity features, our method can steadily work in the long corridor, which can fit the ground truth on the x and y axes well.

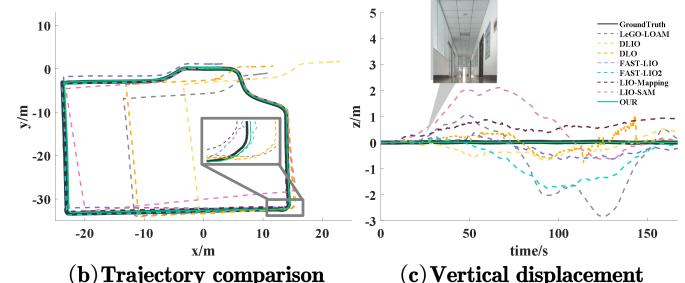
As shown in Fig. 7 (a), the degradation in the moving direction of FAST-LIO, FAST-LIO2 and LIO-Mapping in the x and y axes are alleviated by the tight coupling of LiDAR and



(a) Mapping by our method



(b) Trajectory comparison



(c) Vertical displacement

Fig. 7. Long corridor environment. (a) The mapping result of the proposed method is that the red line is the trajectory of the robot. (b) Comparison of trajectories obtained by each algorithm, solid black line is the ground truth. (c) Variation of different trajectories in the vertical direction, most algorithms drift after entering the corridor.

TABLE IV  
PERFORMANCE IN SEQUENCE 2.

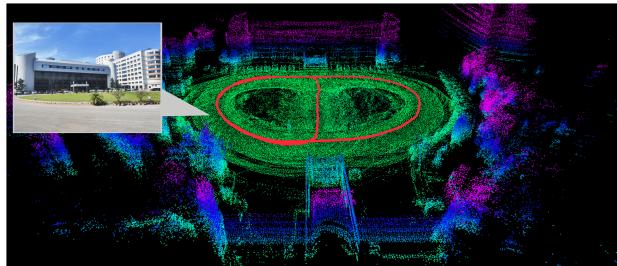
Method	Max	Mean	RMSE	Vertical Error
LeGO-LOAM	11.723	7.244	8.686	1.314
DLO	12.319	7.295	8.867	0.303
DLIO	26.842	13.067	16.206	0.339
FAST-LIO	1.799	0.928	1.107	0.358
FAST-LIO2	1.774	0.778	0.977	0.959
LIO-Mapping	1.089	0.732	0.769	0.615
LIO-SAM	5.245	2.419	2.971	1.163
OURS	<b>0.538</b>	<b>0.273</b>	<b>0.317</b>	<b>0.009</b>

IMU. However, Fig. 7 (c) demonstrates that these methods have different degrees of drift in the vertical direction after entering the corridor, because of the insufficient constraints in the vertical direction. As shown in Fig. 7 (c), our method limits vertical drift in real time through the ground constraint module, which allows our method to maintain a small vertical error in this environment.

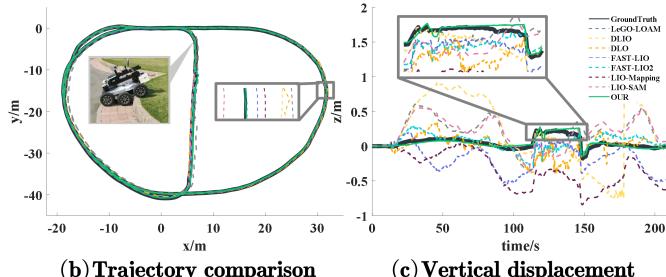
TABLE IV statistics the performance indicators of different methods in the long corridor. In such an environment, our method is significantly superior to other methods in terms of overall accuracy. Note that the error of our method in the vertical direction is greatly reduced compared with other methods.

### C. Evaluation in outdoor environment

1) *Crossing obstacle scene*: Sequence 3 is an open outdoor scene with a big lawn, which includes the process of climbing steps. As shown in Fig. 8 (b), the trajectories of various algorithms in the X and Y-axes can fit the ground truth well. However, as illustrated in Fig. 8 (c), other methods in the



(a) Mapping by our method



(b) Trajectory comparison

(c) Vertical displacement

Fig. 8. Crossing obstacle scene. In this environment, robot will climb the flower bed. (a) The mapping result of the proposed method is that the red line is the trajectory of the robot. (b) Comparison of trajectories obtained by each algorithm, the solid black line is ground truth. (c) Variation of different trajectories in the vertical direction.

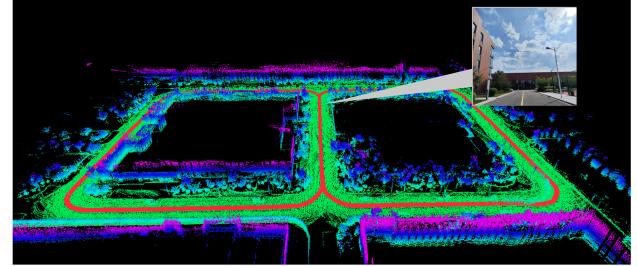
TABLE V  
PERFORMANCE IN SEQUENCE 3.

Method	Max	Mean	RMSE	Vertical Error
<b>LeGO-LOAM</b>	1.010	0.677	0.693	0.278
<b>DLO</b>	0.746	0.217	0.259	0.161
<b>DLIO</b>	0.982	0.486	0.559	0.522
<b>FAST-LIO</b>	0.550	0.271	0.304	0.277
<b>FAST-LIO2</b>	<b>0.500</b>	0.166	0.179	0.114
<b>LIO-Mapping</b>	0.787	0.387	0.438	0.413
<b>LIO-SAM</b>	0.703	0.303	0.326	0.285
<b>OUR</b>	0.551	<b>0.119</b>	<b>0.137</b>	<b>0.031</b>

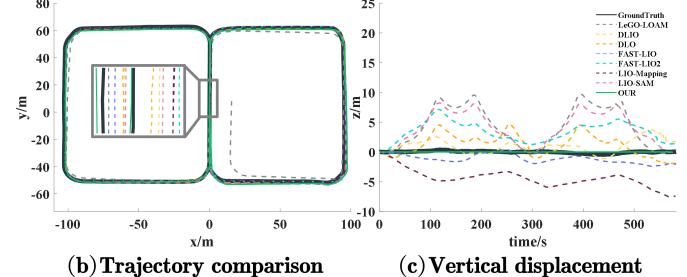
vertical direction accumulate errors in varying degrees. Particularly, the severe vibration caused by the climbing process intensifies the drift of these algorithms in the vertical direction.

In combination with TABLE V, we find that LeGO-LOAM is prone to producing large errors in such a scenario. The trajectory estimated by DLIO significantly deviates in the vertical direction due to the misestimation of IMU state during the strong vibrations. As shown in Fig. 8 (c) and TABLE V, although DLO and FAST-LIO2 have higher accuracy and disturbance immunity, their vertical trajectories still show a noticeable drift over time. Compared to these methods, our method has smallest vertical error, which is only 27% of FAST-LIO2. Moreover, from the vertical trajectory, our method can still operate robustly under complex ground conditions on uneven lawn.

2) *Structured outdoor environment*: Sequence 4 depicts a campus scene, in which, the length of robot's trajectory is approximately 863 m. Such open and long-distance scenario can easily generate large cumulative errors in the vertical direction especially for sparse-channel LiDAR. In addition, some structured buildings in this environment can produce



(a) Mapping by our method



(b) Trajectory comparison

(c) Vertical displacement

Fig. 9. Structured outdoor environment. (a) The mapping result of the proposed method is that the red line is the trajectory of the robot. (b) Comparison of trajectories obtained by each algorithm, solid black line is the ground truth. (c) Variation of different trajectories in the vertical direction.

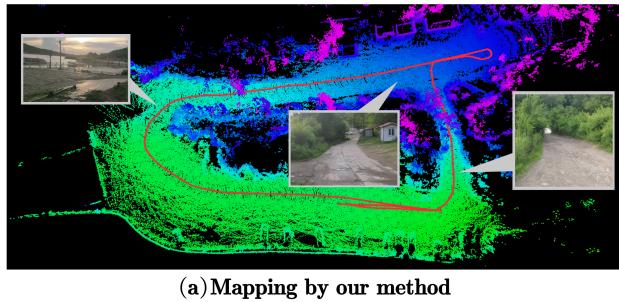
TABLE VI  
PERFORMANCE IN SEQUENCE 4.

Method	Max	Mean	RMSE	Vertical Error
<b>LeGO-LOAM</b>	16.649	7.407	9.037	5.738
<b>DLO</b>	11.244	4.079	4.653	4.049
<b>DLIO</b>	5.260	2.893	3.133	2.354
<b>FAST-LIO</b>	5.333	2.588	2.942	1.734
<b>FAST-LIO2</b>	10.125	4.458	5.081	4.182
<b>LIO-Mapping</b>	7.579	5.086	5.372	4.732
<b>LIO-SAM</b>	10.798	4.462	5.446	5.108
<b>OUR</b>	<b>2.905</b>	<b>1.806</b>	<b>1.967</b>	<b>0.174</b>

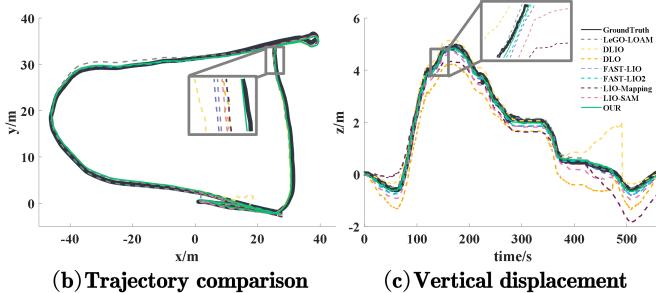
sparse line and plane features almost perpendicular to the ground plane, which can make it difficult for the algorithm that is highly dependent on feature points to converge in the vertical direction.

TABLE VI, Fig . 9 (b) and (c) shows that most algorithms produce large vertical error with time. Although LeGO-LOAM and LIO-SAM enable loop-closed detection, the map exhibits serious distortion after optimization due to vertical drift. As shown in TABLE VI, DLIO performs well in this experiment, which is attributed to the flat ground. As shown in TABLE VI and Fig . 9 (c), compared with others, our method has the best performance depending on the combination of geometrical features, intensity features and ground constraints, which can successfully inhibit the vertical error particularly.

3) *Field environment with slope*: Sequence 5 is collected from the field environment along the coastal tidal flats, which includes uneven terrain, tree obstructions, and frequently changing ground slopes. As shown in Fig . 10 (a), the robot needs to traverse various types of slopes during its operation, which means that the ground below the robot is constantly changing. Additionally, as can be observed from Fig . 10 (c), the slopes in this scene have irregular gradients, and the robot's



(a) Mapping by our method



(b) Trajectory comparison

(c) Vertical displacement

Fig. 10. Field environment with slope. (a) The mapping result of the proposed method is that the red line is the trajectory of the robot. (b) Comparison of trajectories obtained by each algorithm, solid black line is the ground truth. (c) Variation of different trajectories in the vertical direction.

TABLE VII  
PERFORMANCE IN SEQUENCE 5.

Method	Max	Mean	RMSE	Vertical Error
LeGO-LOAM	1.908	0.962	1.038	0.108
DLO	1.998	1.017	1.054	0.664
DLIO	1.908	0.881	0.982	0.473
FAST-LIO	1.852	0.802	0.846	0.153
FAST-LIO2	1.953	0.823	0.870	0.139
LIO-Mapping	<b>1.697</b>	0.991	0.980	0.560
LIO-SAM	2.132	0.903	0.964	0.298
OUR	1.984	<b>0.720</b>	<b>0.765</b>	<b>0.090</b>

vertical trajectory experiences a rapid increase followed by a slow decline.

As indicated in Fig. 10 and TABLE VII, our method demonstrates the highest accuracy and the smallest vertical error in Sequence 5. The experimental results show that the ground constraint method we propose can effectively adapt to transitions between slopes with different gradients, allowing the trajectory to closely align with the groundtruth in the vertical direction.

#### D. Ablation experiment

To verify the effectiveness of the proposed method, ablation experiments are conducted on the ground constraint module and intensity features.

For the ground constraint module, we select sequences 3, 4 and 5 for the experiment. The proposed method is divided into three types: without ground constraint (*Proposed: w/o GC*), with global ground constraint like [34], [35] (*Proposed: GC like [34][35]*), and with the ground constraint module that we propose(*Proposed*). As shown in Fig. 11 (b), for sequences 4 with flat ground, both our ground constraint module and the

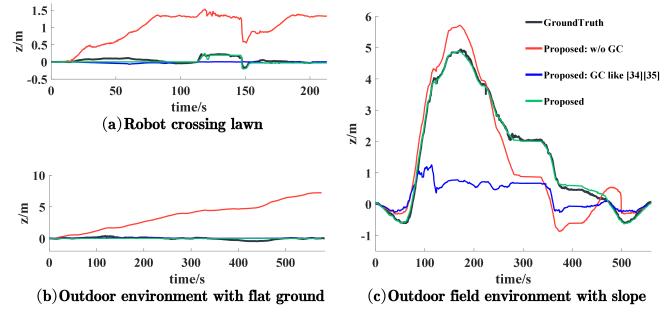


Fig. 11. Ablation experiment of the proposed ground constraint module. (a), (b) and (c) respectively display the vertical trajectories of Sequences 3, 4, and 5 under different ground constraint conditions, where the "GC" means ground constraint.

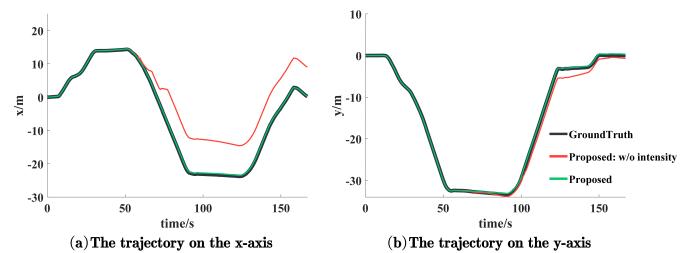


Fig. 12. Ablation experiment of the intensity features. (a) The trajectory comparison on the x-axis. (b) The trajectory comparison on the y-axis.

global ground constraint in [34], [35] are effective in suppressing the system's drift in the vertical direction. However, as shown in Fig. 11 (a) and (c), for sequences 3 and 5 with complex terrains, the method in [34], [35] uses a single global ground plane as a constraint, which is unable to adapt to the changes in ground plane transitions while introduces erroneous estimates of the system's state in the vertical direction.

It should be noted that despite the experiments showing that our ground constraint module has stronger environmental adaptability and robustness compared to the traditional ground constraint methods in [34], [35], our method still has certain limitations. When the ground changes are not enough (all dimensions of  $e_G$  are smaller than the ground change discrimination threshold), failure of ground change detection may cause the trajectory to drift over time. For example, between 390-460s in Fig. 11 (c), due to the inadequate estimation of the gradual ground change, there remains an unavoidable minor drift between the trajectory estimated by our method and the groundtruth. Therefore, our method has limited performance in such scenarios with small and continuous ground changes.

We evaluate the role of intensity features in sequence 2. Fig. 12 demonstrates that the extracted intensity features effectively inhibit the degradation of the robot trajectory in the corridor.

#### E. Runtime

We evaluate the efficiency of different methods in sequence 4. TABLE VIII shows the time cost of different algorithms for each stage. Since the current versions of FAST-LIO and FAST-LIO2 use the same framework, with the only difference being whether they distinguish between feature points, they

TABLE VIII

THE MEAN COMPUTATION TIME OF DIFFERENT ALGORITHMS EXECUTED ON A COMPUTER WITH INTEL i7-10875H CPU.

Method	Avg. Comp.(ms)		
	Odometry	Mapping	Sum
LeGO-LOAM	2.36	63.53	65.89
DLO	28.46	—	28.46
DLIO	25.33	—	25.33
FAST-LIO/FAST-LIO2	18.53	—	18.53
LIO-Mapping	171.39	312.22	483.61
LIO-SAM	—	64.59	64.59
OUR	13.88	33.91	47.79

are statistically analyzed together here. Compared with two typical LIO methods, DLIO and FAST-LIO, our odometry has a close or even higher computing efficiency. Our method also has better real-time performance than whole LiDAR SLAM systems, such as LeGO-LOAM, LIO-SAM and LIO-mapping. Experiments reveal that our SLAM framework can operate in real time.

## V. CONCLUSION

In this paper, a LiDAR SLAM system with ground constraints is proposed. The problem that this work focuses on is the inhibition of the drift and degradation of the SLAM system caused by sparse-channel LiDAR. Firstly, we proposed a ground constraint module based on the principal normal of planar ground that could effectively inhibit the vertical drift caused by sparse-channel LiDAR. Then, we extracted additional intensity features for feature point matching to offset the lack of geometrical features in the featureless environment. In addition, the proposed method employed a hybrid LiDAR SLAM framework that combines scan to LiDAR-centric sliding submap ICP and scan-to-map feature points matching, which can achieve a good balance between accuracy and real-time performance. We conducted extensive experiments based on the GroundRobotDataset, and the results demonstrated the superiority of the proposed method in inhibiting drift and degradation compared with baseline methods.

In fact, the proposed method only employed the LiDAR and gyroscope of the IMU, which did not fuse multiple sensors information. Therefore, our future study will focus on fusing more sensors, such as camera and wheel encoder, into the proposed method to obtain more accurate and robust pose estimation.

## REFERENCES

- [1] X. Ding, Y. Wang, R. Xiong, D. Li, L. Tang, H. Yin, and L. Zhao, "Persistent stereo visual localization on cross-modal invariant map," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4646–4658, 2020.
- [2] P. Wang, Z. Fang, S. Zhao, Y. Chen, M. Zhou, and S. An, "Vanishing point aided lidar-visual-inertial estimator," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 120–13 126.
- [3] P. Yin, A. Abuduweili, S. Zhao, L. Xu, C. Liu, and S. Scherer, "Bioslam: A bioinspired lifelong memory system for general place recognition," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4855–4874, 2023.
- [4] Y. Cui, Z. Fang, Z. Li, S. Li, and Y. Lin, "Hygfnet: Hybrid geometry-flow learning network for 3d single object tracking," *IEEE Transactions on Intelligent Vehicles*, 2024.
- [5] H. Wu, Y. Li, W. Xu, F. Kong, and F. Zhang, "Moving event detection from lidar point streams," *Nature communications*, vol. 15, no. 1, 2024.
- [6] H. Kim, J. Cho, D. Kim, and K. Huh, "Intervention minimized semi-autonomous control using decoupled model predictive control," in *2017 28TH IEEE Intelligent Vehicles Symposium (IV 2017)*, 2017, pp. 618–623.
- [7] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2021.
- [8] X. Xia, N. P. Bhatt, A. Khajepour, and E. Hashemi, "Integrated inertial-lidar-based map matching localization for varying environments," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 10, pp. 4307–4318, 2023.
- [9] Z. Zhou, X. Feng, S. Di, and X. Zhou, "A lidar mapping system for robot navigation in dynamic environments," *IEEE Transactions on Intelligent Vehicles*, pp. 1–20, 2023.
- [10] L. He, W. Li, Y. Guan, and H. Zhang, "Igicp: Intensity and geometry enhanced lidar odometry," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2023.
- [11] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.
- [12] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, APR 2021.
- [13] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, AUG 2022.
- [14] P. BESL and N. MCKAY, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, FEB 1992.
- [15] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, APR 2022.
- [16] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5580–5586.
- [17] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp - simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, FEB 2023.
- [18] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, 2014.
- [19] J. Zhang, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, FEB 2017.
- [20] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.
- [21] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam : Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4390–4396.
- [22] S. Choi, H.-W. Chae, Y. Jeung, S. Kim, K. Cho, and T.-w. Kim, "Fast and versatile feature-based lidar odometry via efficient local quadratic surface approximation," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 640–647, 2023.
- [23] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "Mulls: Versatile lidar slam via multi-metric linear least square," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 633–11 640.
- [24] S. Guo, Z. Rong, S. Wang, and Y. Wu, "A lidar slam with pca-based feature extraction and two-stage matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.
- [25] C. Forster, L. Carbone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.
- [26] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3144–3150.
- [27] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [28] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," in *2023 IEEE*

- International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3983–3989.
- [29] F. Cao, F. Yan, S. Wang, Y. Zhuang, and W. Wang, “Season-invariant and viewpoint-tolerant lidar place recognition in gps-denied environments,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 563–574, 2021.
- [30] M. Liao, Y. Zhang, J. Zhang, L. Liang, S. Coleman, and D. Kerr, “Semantic topological descriptor for loop closure detection within 3d point clouds in outdoor environment,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2856–2863.
- [31] Z. Wang, L. Zhang, Y. Shen, and Y. Zhou, “D-liom: Tightly-coupled direct lidar-inertial odometry and mapping,” *IEEE Transactions on Multimedia*, vol. 25, pp. 3905–3920, 2023.
- [32] V. Kubelka, M. Vaidis, and F. Pomerleau, “Gravity-constrained point cloud registration,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4873–4879.
- [33] Y. Jiang, T. Wang, S. Shao, and L. Wang, “3d slam based on ndt matching and ground constraints for ground robots in complex environments,” *Industrial Robot-the International Journal Of Robotics Research And Application*, vol. 50, no. 1, pp. 174–185, JAN 2 2023.
- [34] W. Wei and L.-T. Hsu, “Agpc-slam: Absolute ground plane constrained 3d lidar slam,” *Navigation-Journal of the Institute of Navigation*, vol. 69, no. 3, FAL 2022.
- [35] K. Koide, J. Miura, and E. Menegatti, “A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419841532, 2019.
- [36] X. Wei, J. Lv, J. Sun, E. Dong, and S. Pu, “Gclo: Ground constrained lidar odometry with low-drifts for gps-denied indoor environments,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2229–2235.
- [37] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, “Lips: Lidar-inertial 3d plane slam,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 123–130.
- [38] Z. Wang, L. Yang, F. Gao, and L. Wang, “Fovo-loam: Feature extraction and vertical optimized lidar odometry and mapping,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 086–12 093, 2022.
- [39] J. Chen, H. Wang, M. Hu, and P. N. Suganthan, “Versatile lidar-inertial odometry with se(2) constraints for ground vehicles,” *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3486–3493, 2023.
- [40] G. Kurz, S. A. Scherer, P. Biber, and D. Fleer, “When geometry is not enough: Using reflector markers in lidar slam,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4880–4887.
- [41] X. Dai, W. Song, Y. Wang, Y. Xu, Y. Lou, and W. Tang, “Lidar-inertial integration for rail vehicle localization and mapping in tunnels,” *IEEE Sensors Journal*, vol. 23, no. 15, pp. 17 426–17 438, 2023.
- [42] H. Wang, C. Wang, and L. Xie, “Intensity-slam: Intensity assisted localization and mapping for large scale environment,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1715–1721, 2021.
- [43] H. Li, B. Tian, H. Shen, and J. Lu, “An intensity-augmented lidar-inertial slam for solid-state lidars in degenerated environments,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.
- [44] F. Zhu, Y. Ren, and F. Zhang, “Robust real-time lidar-inertial initialization,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3948–3955.
- [45] Y. Su, T. Wang, S. Shao, C. Yao, and Z. Wang, “Gr-loam: Lidar-based sensor fusion slam for ground robots on complex terrain,” *Robotics and Autonomous Systems*, vol. 140, JUN 2021.
- [46] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, “Voxelized gicp for fast and accurate 3d point cloud registration,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 054–11 059.



**Shaocong Wang** received the B.S. degree in Department of Automation from Chang'an University of China in 2022. He is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences. His current research interests include robot control, simultaneous localization and mapping.



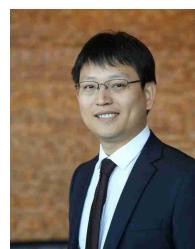
**Fengkui Cao** received the bachelor's degree in vehicle engineering from Shandong Agricultural University, Shandong, China, in 2013, and the Ph.D. degree in control science and engineering from the Dalian University of Technology, Dalian, China, in 2020. He is currently working in the State Key Laboratory of Robotics at Shenyang Institute of Automation, Shenyang, China. His research interests include unmanned ground vehicles' autonomous navigation and long-term localization, SLAM, and robotic vision.



**Ting Wang** received the B.S. degree in Department of Automation from Dalian University of Technology in 2001. In 2007, he graduated from Shenyang Institute of Automation, majoring in pattern recognition and intelligent systems, and obtained the Ph.D. degree. Since then he has been working in the State Key Laboratory of Robotics at Shenyang Institute of Automation. His research interests include robot control, special robot technology and pattern recognition and intelligent system.



intelligent systems.



**Lianqing Liu** (Senior Member, IEEE) received the B.S. degree in industry automation from Zhengzhou University, Zhengzhou, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2009. He is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences. His current research interests include micro/nanorobotics, biosyncretic robotics, and intelligent control.