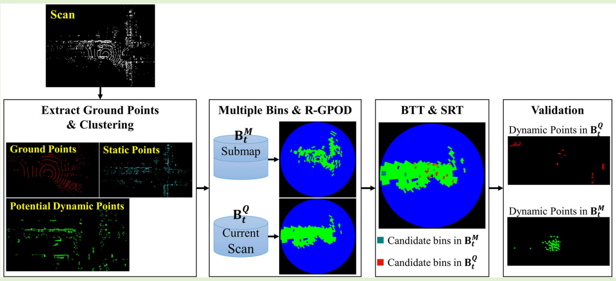
DOR-LINS: Dynamic Objects Removal LiDAR-Inertial SLAM Based on Ground Pseudo Occupancy

Zhoubo Wang, Zhenhai Zhang, Xiao Kang, Miusi Wu, Siyu Chen, and Qilin Li


Scan
Extract Ground Points & Clustering
Ground Points
Static Points
Potential Dynamic Points
Multiple Bins & R-GPOD
$\mathbf{B}_t^M$ Submap
$\mathbf{B}_t^Q$ Current Scan
BTT & SRT
🟦 Candidate bins in $\mathbf{B}_t^M$
🟥 Candidate bins in $\mathbf{B}_t^Q$
Validation
Dynamic Points in $\mathbf{B}_t^Q$
Dynamic Points in $\mathbf{B}_t^M$

***Abstract*—Simultaneous localization and mapping (SLAM) is one of the fundamental capabilities for autonomous vehicles to achieve accurate localization in dynamic urban environments. However, in real-world scenarios, the presence of moving objects such as pedestrians, bicycles, or vehicles affects the localization accuracy of SLAM and leaves behind *ghost trails* in the created map. Hence, it is essential for SLAM to remove dynamic objects in real-time to improve its accuracy. In this article, we present dynamic objects removal LiDAR-inertial SLAM (DOR-LINS), a real-time moving objects removal framework for in light detection and ranging (LiDAR)-Inertial SLAM. Built on the foundation of lidar inertial odometry via smoothing and mapping (LIO-SAM), DOR-LINS integrates the capability of removing dynamic objects, enabling it to remove most of the dynamic objects and achieve precise odometry. Our contributions are reflected in three aspects. First, our method extracts ground points from the current scan and clusters nonground points to segment them into static and potential dynamic clusters. Then, we divide the submap and current scan into multiple bins and introduce a novel concept called *ground pseudo occupancy* to describe the occupancy of each bin. Second, based on the *ground pseudo occupancy*, dynamic clusters, and static clusters, we propose an approach named *beam tracing test* (BTT) and combine it with scan ratio test (SRT) to select candidate dynamic bins. Finally, we employ a dynamic point verification algorithm to filter out actual dynamic points from these candidate dynamic bins. As experimentally evaluated on the UrbanLoco dataset, our proposed method removes many dynamic points and yields promising performance against state-of-the-art methods.**

***Index Terms*—Moving objects, removal, simultaneous localization and mapping (SLAM).**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM) is essential for self-driving cars to navigate unfamiliar environments safely. With the advancement and cost-effectiveness of light detection and ranging (LiDAR) in recent years, LiDAR-based SLAM solutions have attracted increasing attention in autonomous vehicles. Moreover, many excellent SLAM systems, which fuse LiDAR feature points and inertial measurement units (IMUs), have emerged to provide more accurate localization and map. LOAM [1] is widely recognized for proposing a method based on edge and plane feature matching and achieving excellent performance in the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) odometry benchmark [2]. Based on LOAM, LIO-SAM [3] adopts a factor graph strategy by allowing multiple measurements to be incorporated as factors into the system to improve efficiency and accuracy. Fast-LIO [4] and Fast-LIO2 [5] employ tightly coupled iterative extended Kalman filters to achieve robust navigation in fast-motion environments. However, these excellent SLAM systems have a significant limitation: the assumption that the surrounding environment consists of static objects. Intelligent vehicles are typically exposed to dynamic urban scenes, which poses significant challenges to the localization of these methods and causes *ghost trail effects* on the map they generate.

In urban environments, real-world spaces consist of thousands of real-time moving objects (such as cars) and stationary, static objects (such as buildings or parked cars), making distinguishing between dynamic and static objects difficult. For example, Suma++ [6], which is based on deep learning semantic segmentation, cannot differentiate between actual moving objects (such as moving cars) and static objects (such as parked vehicles), which may lead to the removal

Manuscript received 5 August 2023; revised 26 August 2023; accepted 28 August 2023. Date of publication 6 September 2023; date of current version 16 October 2023. This work was supported in part by the Major Pre-Research Background Project under Grant ZW040202, and in part by the National Defense Technology of China under Grant 202020230028. The associate editor coordinating the review of this article and approving it for publication was Prof. Yulong Huang. *(Corresponding author: Zhenhai Zhang.)*

Zhoubo Wang, Zhenhai Zhang, Miusi Wu, Siyu Chen, and Qilin Li are with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: wangzhoubo283@163.com; zhzhang@bit.edu.cn; 3120210165@bit.edu.cn; chensiyu@bit.edu.cn; 3120220150@bit.edu.cn).

Xiao Kang is with the China North Vehicle Research Institute, Beijing 100072, China (e-mail: kangxiao_uvc@126.com).

Digital Object Identifier 10.1109/JSEN.2023.3310484

of valuable features necessary for accurate matching or 3-D reconstruction. Pfreundschuh et al. [7] developed a general unsupervised learning approach to dynamic object perception that successfully achieves reliable discrimination between actual moving and static objects. However, this approach requires sufficient training data to perform well. In the case of autonomous driving vehicles facing unknown environments, the ability to recognize new dynamic objects remains a challenge.

This article proposes a novel tightly coupled LiDAR-Inertial SLAM framework with dynamic object removal called dynamic objects removal LiDAR-inertial SLAM (DOR-LINS). Specifically, DOR-LINS first segments the lidar scan into ground and nonground points using the Patchwork++ proposed in [9]. Then, nonground points are clustered into candidate dynamic and static clusters via a two-step clustering approach. Subsequently, both the current lidar scan and the surrounding keyframes, referred to as submap, are divided into multiple bins. Each bin is described using a new concept called region-wise ground pseudo occupancy descriptor (R-GPOD). Based on R-GPOD, we combine the proposed *beam tracking test* with the scan ratio test (SRT) to detect candidate dynamic bins. Finally, we introduce an algorithm for verifying dynamic points within these candidate bins and removing them. After removing dynamic points, DOR-LINS leverages scan-matching to achieve accurate poses in dynamic environments. Using dynamic point removal and odometry tests, we evaluate DOR-LINS on the publicly available Urbanloco dataset [11]. The experimental results demonstrate that DOR-LINS removes dynamic points with less static point loss compared to the state-of-the-art method ERASOR. Furthermore, our approach outperforms the state-of-the-art methods LIO-SAM and LIO-SGEMOT in terms of absolute trajectory accuracy. Moreover, the runtime test indicates that our method exhibits real-time capabilities for dynamic object removal.

This article is organized as follows. Section II discusses the related works. Our method is described in Section III. Section IV discusses the experimental results and analysis. Finally, Section V presents our conclusion.

II. Related Research

With the development of deep learning and convolutional neural networks (CNNs) for scene understanding, semantic segmentation techniques [12], [13], [14] can accurately classify 3-D LiDAR point clouds into driving-related objects such as cars, trees, buildings, and pedestrians. Suma++, a semantic SLAM approach [6], utilizes this semantic information to filter out potential moving objects (e.g., pedestrians, cars) for improved localization. However, distinguishing between actually moving objects (e.g., moving cars) and static objects (e.g., parked cars) remains challenging for Suma++. To address this issue, Chen et al. [15] combine range images and residual images, implementing an end-to-end CNN for moving object detection. Additionally, Pfreundschuh et al. [7] propose an unsupervised learning method for generic dynamic object perception in LiDAR SLAM. They train a 3DMiniNet network using occupancy grids to label and detect dynamic objects. Despite successfully separating moving and static objects, these deep learning-based methods rely on expensive and difficult-to-obtain training data, limiting their effectiveness in detecting dynamic objects in new environments.

Recently, Lin et al. [16] introduced and made available for public use a LiDAR inertial odometry system called LIO-SEGMOT, which incorporates multiobject tracking. It adopts an asynchronous factor graph architecture to estimate the robot's state and the states of the surrounding objects. Additionally, it integrates the teacher–student network proposed by [17] for 3-D object detection. Although the method significantly enhances odometry accuracy, it does not dynamically remove dynamic objects from the generated map.

Pagad et al. [18], Chen [19], Gehrung et al. [20], and Schauer and Nuchter [21] introduce voxel-based approaches which construct a huge regular voxel grid cells map. A dynamic cell is identified for each grid cell if a laser beam initially hits it but is later missed by subsequent rays. The points contained in the dynamic cell are then filtered out. However, due to high computational demand, this method is usually only used for offline post-processing and rarely considers real-time requirements. Ambrus et al. [22] and Pomerleau et al. [23] propose visibility-based methods which find correlations between query and map units in a narrow field of view to improve computational efficiency. However, correlation inaccuracies can occur because of LiDAR motion blur, leading to false removal of dynamic points. To address this issue, Kim and Kim [24] proposed a removing-and-reverting algorithm, which reverts falsely removed points based on multiresolution range images and their confidences. While it can successfully separate moving objects from the background, it cannot achieve online operation. Qian et al. [25] developed a dynamic SLAM framework called RF-LIO, which preliminarily removes moving points using adaptive resolution range images and residual images. Through iterative removal and scan-matching steps, detected dynamic points are removed. However, this method faces challenges in removing points too close to the ground or produced by LiDAR beams parallel to the ground.

ERASOR, proposed in [10], is a novel nonvisibility-based offline approach for constructing a static map. The approach constructs regular grid bins and uses the concept of region-wise pseudo occupancy descriptor (R-POD) to express each unit bin. SRT is then performed between the query bin and map bin to check bins that potentially contain dynamic points, which are subsequently discriminated from static points using region-wise ground plane fitting (R-GPF). With these valuable ideas in mind, we first extract the ground points from the current scan and utilize the R-GPOD to express each bin. Compared with ERASOR, this mechanism reduces the runtime of extracting ground points and the number of queried bins, so our system can run in real-time.

III. Our Method

A. Framework Overview and Notation

LIO-SAM is an impressive system that provides reliable pose estimation in most real-world situations. DOR-LINS builds on the method and achieves more accurate mapping and

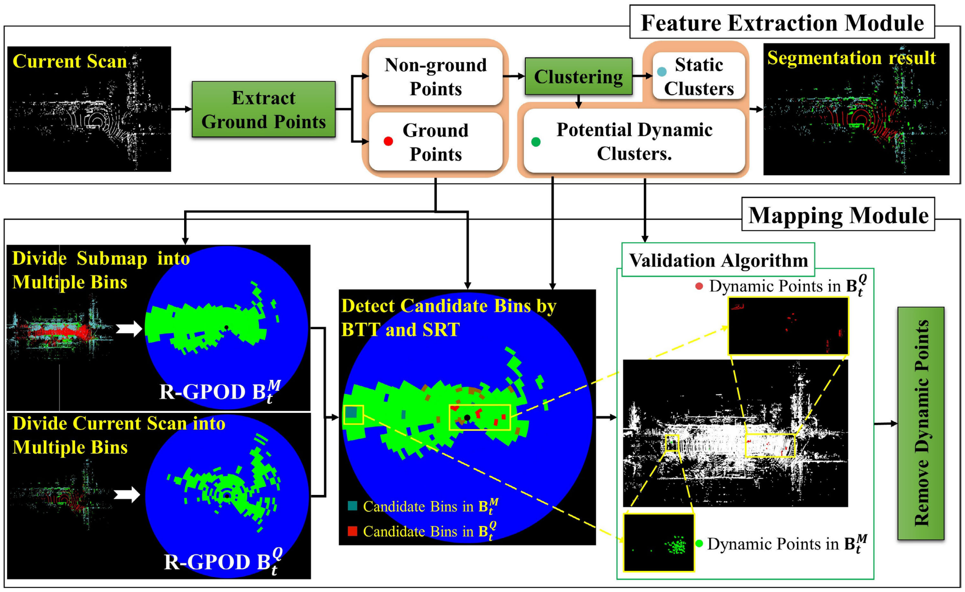

Feature Extraction Module
Current Scan
Extract Ground Points
Non-ground Points
Ground Points
Clustering
Static Clusters
Potential Dynamic Clusters.
Segmentation result
Mapping Module
Divide Submap into Multiple Bins
R-GPOD $\mathbf{B}_t^M$
Divide Current Scan into Multiple Bins
R-GPOD $\mathbf{B}_t^Q$
Detect Candidate Bins by BTT and SRT
Candidate Bins in $\mathbf{B}_t^M$
Candidate Bins in $\mathbf{B}_t^Q$
Validation Algorithm
Dynamic Points in $\mathbf{B}_t^Q$
Dynamic Points in $\mathbf{B}_t^M$
Remove Dynamic Points

Fig. 1. Pipeline of dynamic object removal.

localization by incorporating dynamic object removal. LIO-SAM comprises three modules: IMU preintegration, feature extraction, and mapping. The IMU preintegration module infers the motion between two scans to compensate for point cloud motion distortion. By combining this with LiDAR odometry, it generates IMU odometry. For more detailed information about LIO-SAM, please refer to [3]. Here, we will only introduce the framework for removing dynamic objects in DOR-LINS.

Our dynamic object removal method is implemented in the feature extraction and mapping modules, as shown in Fig. 1. In the feature extraction module, we first use Patchwork++ [9] to segment the point cloud into ground and nonground points. Then, we propose a two-step clustering approach to further divide the nonground points into static and potentially dynamic clusters.

Several key steps are involved in the mapping module.

1) We transform the submap into the current frame using the transformation from IMU odometry. Afterwards, the current scan and the submap are encoded into multiple bins, respectively, and the occupancy of each bin is represented using R-GPOD.
2) We traverse every pair of R-GPODs and apply SRT to extract candidate dynamic and static bins. Additionally, we propose a fast and robust method called *beam tracing test* (BTT) to obtain candidate bins from the submap further.
3) We introduce an algorithm to verify dynamic points for these candidate dynamic bins. Once dynamic points are identified, we immediately remove them from the current scan and the submap.

To help readers understand the content of this article, we provide a list of symbols used in this work, as shown in Table I.

B. Extract Ground Points and Clustering

In urban environments, moving objects (such as people and vehicles) inevitably come into contact with the ground, making ground segmentation a critical task for estimating moving areas. Recently, Lim et al. [9] introduced Patchwork++, a region-wise ground segmentation method. It utilizes a concentric zone model (CZM) polar grid [8] to divide the point cloud into multiple bins, as shown in Fig. 2. Then, for each bin, it simultaneously applies ground plane fitting and ground likelihood estimation to detect the ground. This strategy enables Patchwork++ to achieve fast and accurate ground segmentation. Therefore, we adopt this method for initial ground detection. However, the ground points extracted by the method still include some false positives, whereas nonground points are falsely classified as ground points. It can

TABLE I
SYMBOLS USED IN THIS ARTICLE

Symbols	Meaning
$\mathbf{C}_s^Q$ and $\mathbf{C}_c^Q$	The static and candidate dynamic clusters.
$p_k = \{x_k, y_k, z_k\}$	3D point in Cartesian coordinates.
$\mathbf{P}_t = \{p_1, \cdots, p_n\}$	The point cloud captured at time t.
${}^G\mathbf{P}$ and ${}^{NG}\mathbf{P}$	Ground and non-ground points.
$\mathbf{B}_t^Q$ and $\mathbf{B}_t^M$	Bins of the current scan and the submap at time t.
${}^C\mathbf{B}_t^Q$	Candidate dynamic bins of the current scan at time t.
${}^S\mathbf{B}_t^Q$	Candidate static bins of the current scan at time t.
${}^C\mathbf{B}_t^M$	Candidate dynamic bins of the submap at time t.
Z_m and N_z	The m-th zone and the number of zones.
$N_{r,m}$ and $N_{\theta,m}$	The number of rings and sectors in Z_m.

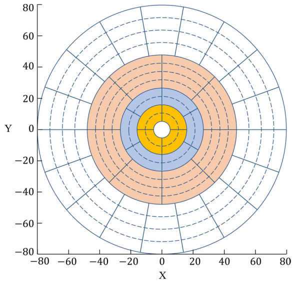


Fig. 2. CZM—based polar grid description.

affect the subsequent method we discuss. Hence, we propose a naive but effective approach to filter out these false positives.

After Patchwork++ divides the point cloud into bins and extracts the ground points ${}^{G}\mathbf{P}$, we traverse each bin and calculate the average z-axis value (H_m) of all the ground points within the bin. If a ground point in the bin has a z-axis value greater than $\tau_h \cdot H_m$ (τ_h is a preset threshold), it is considered a false positive and removed from ${}^{G}\mathbf{P}$. Finally, we classify the point cloud into ground points and nonground points.

Based on prior knowledge of urban environments, where tall objects like trees and buildings are typically static, we perform a two-stage clustering on the nonground points to classify them into static and potential dynamic clusters. In the first stage, we select seed points with z-axis values above a threshold (H_d) and apply Euclidean distance-based clustering to label these objects as static clusters $(\mathbf{C}_s^Q)$. In the second stage, we cluster the remaining nonground points cloud and label them as potential dynamic clusters $(\mathbf{C}_c^Q)$.

C. Region-Wise Ground Pseudo Occupancy Descriptor

By using CZM-based polar grid representation, we encode the current scan and the corresponding submap into multiple bins ($\mathbf{B}_t^Q$ and $\mathbf{B}_t^M$). For each bin, we encapsulate it using a R-GPOD. Its design is inspired by R-POD [10]. R-POD describes the bin using the z-directional difference of point clouds within the bin. However, due to the potential errors in the predicted pose from the IMU odometry, the z-directional difference between different keyframes within the submap may not accurately represent the bin. To address this, we reduce the error by incorporating the z-directional information of the ground points into the z-directional difference.

For the convenience of describing the calculation of $\mathbf{B}_t^Q$ and $\mathbf{B}_t^M$, we use $\mathbf{B}_t$ to represent them. Then $\mathbf{B}_t$ can be expressed as follows:

$$\mathbf{B}_t = \bigcup_{m \in N_z, i \in N_{r,m}, j \in N_{\theta,m}} \mathbf{B}_{(m,i,j),t} \tag{1}$$

where $\mathbf{B}_{(m,i,j),t}$ represents the (m, i, j)th bin. Let $r_k = (x_k^2 + y_k^2)^{1/2}$, $\theta_k = \arctan(y_k, x_k)$. Let $L_{\min,m}$ and $L_{\max,m}$ indicate the maximum and minimum boundary of mth zones. Then each $\mathbf{B}_{(m,i,j),t}$ consists of the points that satisfy the following condition:

$$\mathbf{B}_{(m,i,j),t} = \Big\{ p_k \in \mathbf{P}_t | L_{\min,m} < r_k < L_{\max,m}, \frac{(i-1)\cdot \Delta L_m}{N_{r,m}} < r_k - L_{\min,m} < \frac{i \cdot \Delta L_m}{N_{r,m}}, \frac{(j-1)\cdot 2\pi}{N_{\theta,m}} - \pi < \theta_k < \frac{j \cdot 2\pi}{N_{\theta,m}} - \pi \Big\} \tag{2}$$

where $\Delta L_m = L_{\max,m} - L_{\min,m}$.

Afterwards, each bin in $\mathbf{B}_t^M$ and $\mathbf{B}_t^Q$ assigns two real values to describe ground pseudo occupancy, $\Delta h_{(m,i,j),t}^M$ and $\Delta h_{(m,i,j),t}^Q$. Let $\mathbf{Z}_{(m,i,j),t}^M = \{z_k \in p_k | p_k \in \mathbf{B}_{(m,i,j),t}^M\}$. If the number of ground points in $\mathbf{B}_{(m,i,j),t}^M$ is greater than the threshold N_G, the ground pseudo occupancy of $\mathbf{B}_{(m,i,j),t}^M$ is encoded as follows:

$$\begin{cases} \Delta h_{(m,i,j),t}^M = \max\left\{\mathbf{H}_{(m,i,j),t}^M\right\} - \min\left\{\mathbf{H}_{(m,i,j),t}^M\right\} \\ \mathbf{H}_{(m,i,j),t}^M = \left\{z_k^n - \mathbf{G}^M | z_k^n \in \mathbf{Z}_{(m,i,j),t}^M\right\} \end{cases} \tag{3}$$

where z_k^n represents the kth point's z-directional information of the nth keyframe in the submap. Notably, If the number of ground points for the nth keyframe in $\mathbf{B}_{(m,i,j),t}^M$ is more than 0, $\mathbf{G}^M$ is the average value in z-directional information of the ground points for the nth keyframe in the $\mathbf{B}_{(m,i,j),t}^M$; otherwise, $\mathbf{G}^M$ is the average value in z-directional information of all ground points for the nth keyframe.

Similarly, let $\mathbf{Z}_{(m,i,j),t}^Q = \{z_k \in p_k | p_k \in \mathbf{B}_{(m,i,j),t}^Q\}$. If the number of ground points in $\mathbf{B}_{(m,i,j),t}^Q$ is greater than the threshold N_p, the ground pseudo occupancy of $\mathbf{B}_{(m,i,j),t}^Q$ is encoded as follows:

$$\begin{cases} \Delta h_{(m,i,j),t}^Q = \max\left\{\mathbf{H}_{(m,i,j),t}^Q\right\} - \min\left\{\mathbf{H}_{(m,i,j),t}^Q\right\} \\ \mathbf{H}_{(m,i,j),t}^Q = \left\{z_k - \mathbf{G}^Q | z_k \in \mathbf{Z}_{(m,i,j),t}^Q\right\} \end{cases} \tag{4}$$

where z_k denotes the z-directional information of the kth point. Notably, if there is no ground point in $\mathbf{B}_{(m,i,j),t}^Q$, $\mathbf{G}^Q$ is the average value in z-directional information of all ground points in the current scan; otherwise, $\mathbf{G}^Q$ is the average value in z-directional information of all ground points in $\mathbf{B}_{(m,i,j),t}^Q$.

D. Detect Candidate Dynamic and Static Bins

1) Using SRT to Detect Dynamic and Static Bins: The SRT has been proven effective at offline detection of potential dynamic bins [10]. In this study, we extend this method further. Given a pair of R-GPODs, the SRT not only detects candidate dynamic bins from $\mathbf{B}_t^M$ and $\mathbf{B}_t^Q$ but also identifies candidate static regions from $\mathbf{B}_t^Q$. The strategy allows us to leverage the algorithm proposed in Section III-E for extracting static points, ensuring their preservation during subsequent analysis. Let the term scan ratio (S_r) refers to the ratio of ground pseudo occupancy between each pair of R-GPODs, and it is expressed as follows:

$$S_r = \min\left\{\frac{\Delta h_{(m,i,j),t}^Q}{\Delta h_{(m,i,j),t}^M}, \frac{\Delta h_{(m,i,j),t}^M}{\Delta h_{(m,i,j),t}^Q}\right\}. \tag{5}$$

Typically, bins with candidate dynamic points have a lower S_r, while bins with only static points are those whose S_r is close to 1. Based on our experiments, we set the threshold for dynamic bins at 0.3 (τ_c) and for static bins at 0.7 (τ_s), which proved to be strict enough. Fig. 1 illustrates the results retrieved from SRT, with red bins indicating candidate dynamic bins with S_r below 0.3, while dark red bins denote static bins with S_r exceeding 0.7. Notably, each pair of R-GPODs must contain sufficient points to prevent SRT from skipping over them.

2) Beam Tracing Test: SRT requires marking every pair of R-GPODs ($\mathbf{B}_{(m,i,j),t}^M$ and $\mathbf{B}_{(m,i,j),t}^Q$) as occupied. However, if $\mathbf{B}_{(m,i,j),t}^Q$ is unoccupied and $\mathbf{B}_{(m,i,j),t}^M$ is occupied, SRT fails at detecting whether $\mathbf{B}_{(m,i,j),t}^M$ is a dynamic bin. Hence, relying solely on SRT is insufficient to extract all candidate

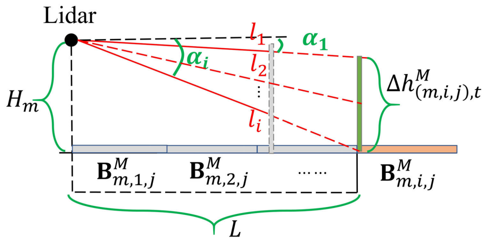


Fig. 3. Red lines are potential beams that miss the $\rhd m, i, j \lhd$th bin.

dynamic bins from $\mathbf{B}_t^M$, which leads to dynamic points within the submap affecting the extraction of dynamic bins in the following scans. To overcome this challenge, we propose the BTT, an approach to detect candidate dynamic bins within $\mathbf{B}_t^M$ for the specific pair of R-GPODs. The principle of this approach is similar to ray-casting: the occupied $\mathbf{B}^M_{(m,i,j),t}$ is considered a candidate dynamic bin if it is missed by the specific laser beams from the current scan; otherwise, it is not. These specific laser beams are illustrated in Fig. 3.

The angles α_1 and α_i are calculated as follows:

$$\begin{cases} \alpha_1 = \arctan\left(\left(H_m - \Delta h^M_{(m,i,j),t}\right), L\right) \\ \alpha_i = \arctan(H_m, L) \end{cases} \tag{6}$$

where H_m denotes the mean of the z-components of the ground points in $\mathbf{B}^M_{(m,n,j),t}$, L is the minimum radial boundary of $\mathbf{B}^M_{(m,n,j),t}$, and $\Delta h^M_{(m,i,j),t}$ can be obtained from (3). We determine the vertical emission angles of each laser beam from the device manual to calculate effortlessly the beams $(l_1, l_2, \ldots, l_i)$ that fall within the range between α_1 and α_i. We then count the number of nonground points that contain these beams before the (m, i, j)th bin. If the number of these points is zero, it indicates that the beams miss the (m, i, j)th bin, implying a candidate dynamic bin. Otherwise, the bin is static. Finally, we add these candidate dynamic bins to the set ${}^C\mathbf{B}_t^M$. Fig. 1 depicts the results retrieved from BTT, with teal bins indicating candidate dynamic bins.

E. Validation of Static and Dynamic Points

After extracting candidate dynamic (${}^C\mathbf{B}_t^Q$ and ${}^C\mathbf{B}_t^M$) and static (${}^S\mathbf{B}_t^Q$) bins, we need to validate these bins' actual dynamic and static points. As discussed in Section III-B, the current scan is clustered and labeled into static ($\mathbf{C}_s^Q$) and candidate dynamic clusters ($\mathbf{C}_c^Q$). To identify which clusters in $\mathbf{C}_c^Q$ are dynamic or static, we evaluate each cluster $\mathbf{C}_i$. For a cluster $\mathbf{C}_i$, if $\mathbf{C}_i$ is dynamic, most of its points should be detected as candidate dynamic points; conversely, if $\mathbf{C}_i$ is static, most of its points should be detected as candidate static points. We use N_i to represent the total number of points in $\mathbf{C}_i$. Then, we obtain the numbers of potential dynamic and static points (N_i^D and N_i^S) labeled as i from ${}^C\mathbf{B}_t^Q$ and ${}^S\mathbf{B}_t^Q$, respectively. Next, we calculate the candidate dynamic points ratio (R_c) as $R_c = N_i^D/N_i$, and the candidate static points ratio (R_s) as $R_s = N_i^S/N_i$. If R_c exceeds the threshold of 0.6 (τ_e), all points within $\mathbf{C}_i$ are considered dynamic points. If R_s exceeds the threshold of 0.6 (τ_r), all points within $\mathbf{C}_i$ are considered static points. Afterwards, we remove dynamic points from the current scan and label static points.

Additionally, regarding the candidate dynamic bins (${}^C\mathbf{B}_t^M$) from the submap, some static points may exist in the candidate dynamic bins extracted by SRT and BTT. Fortunately, we have already labeled some static points, extracting only the points labeled as potential dynamic clusters from ${}^C\mathbf{B}_t^M$. Afterwards, we remove dynamic points from the submap. We summarize this process in Algorithm 1 to make it more clear.

Algorithm 1 Dynamic Points Validation and Removal

```
Input: Candidate dynamic bins: ${}^C\mathbf{B}_t^Q$ and ${}^C\mathbf{B}_t^M$.
       Static bins: ${}^S\mathbf{B}_t^Q$.
       Static clusters: $\mathbf{C}_s^Q$.
       Potential dynamic clusters: $\mathbf{C}_c^Q$
Output: Static points in the current scan and static
        points in the submap.
for $\mathbf{C}_i$ in $\mathbf{C}_c^Q$ do
    Get the number of all points in $\mathbf{C}_i$: $N_i$.
    Get the number of candidate dynamic points in $\mathbf{C}_i$
      from ${}^C\mathbf{B}_t^Q$: $N_i^D$.
    Get the number of candidate static points in $\mathbf{C}_i$
      from ${}^S\mathbf{B}_t^Q$: $N_i^S$.
    $R_c = N_i^D/N_i$.
    $R_s = N_i^S/N_i$.
    if $R_c > \tau_e$ then
        Label $C_i$ as dynamic cluster.
    if $R_s > \tau_r$ then
        Add $C_i$ to the static clusters $\mathbf{C}_s^Q$.
Remove dynamic clusters from current scan.
for $\mathbf{B}_i$ in ${}^C\mathbf{B}_t^M$ do
    for $p_i$ in $\mathbf{B}_i$ do
        Get the label $i$ of $p_i$.
        if $i$ in $\mathbf{C}_s^Q$ then
            $p_i$ is a static points.
        else
            $p_i$ is a dynamic points.
Remove dynamic points from submap.
```

F. LIO Subsystem

We employ LiDAR-Inertial odometry from LIO-SAM, which maintains a factor graph for global pose optimization. IMU preintegration, LiDAR odometry, and loop closure factors are added to the graph and jointly optimized to obtain the LiDAR odometry. Due to space limitations, we refer readers to [3] for details on the IMU preintegration and loop closure factors. Here, we only describe the LiDAR odometry factors relevant to this article.

When a new scan arrives, we first extract its edge and plane features. Using our method introduced in Sections III-B–III-E, we remove dynamic points from both the current scan and

TABLE II
DATASET INFORMATION

Dataset	Length (m)	Scans	Dynamic level
CAMarketStreet	5688.4	14471	High
CARussianHill	3570.4	15860	High
CAChinaTown	2887.7	12930	High
CAColiTower	1349.8	2439	High
CALombardStreet	1196.9	2484	Low
HK20190426-1	556.4	1508	High
HK20190426-2	741.3	2607	High

TABLE III
PARAMETERS EMPLOYED IN DOR-LINS

Parameter	N_G	N_p	τ_c	τ_s	τ_e	τ_r	τ_h	H_r	N_z
Value	100	5	0.3	0.7	0.6	0.6	1.2	2.3	4

submap, including those in the edge and plane features of the current frame. The new edge and plane features are denoted as $\mathbf{F}_e$ and $\mathbf{F}_p$, respectively. We then transform $\mathbf{F}_e$ and $\mathbf{F}_p$ from the current LiDAR coordinate system to the world coordinate system using predicted poses from the IMU. We also extract n surrounding keyframes, transform them to the world coordinate system, and represent them as $\mathbf{M}$. Finally, we perform scan-matching to match the LiDAR scan $\mathbf{F}_e$ and $\mathbf{F}_p$ to $\mathbf{M}$, obtaining the LiDAR odometry factors. Our experiments show that removing dynamic points improves LiDAR odometry factors, resulting in higher localization accuracy.

IV. EXPERIMENTS AND RESULTS

A. Experiment Setup

To evaluate our method, DOR-LINS, we conduct comprehensive tests using the publicly available UrbanLoco dataset [11]. UrbanLoco is collected in densely populated areas of San Francisco and Hong Kong and features a wide range of dynamic elements, such as pedestrians, vehicles, and bicyclists, which can provide a realistic representation of the urban environments typically encountered by autonomous vehicles. We class the dataset into high and low dynamic datasets based on the number of moving objects, as detailed in Table II.

We conduct two tests to evaluate the performance of our method. First, comparative experiment against ERASOR are conducted to assess the effectiveness of our dynamic point removal. Second, comparative experiments against LIO-SAM and LIO-SEGMOT are carried out to evaluate trajectory accuracy and runtime. Notably, DOR-LINS, LIO-SAM, and LIO-SEGMOT only use LiDAR and IMU without GPS. For all experiments, we set $[N_{r,1}, N_{r,2}, N_{r,3}, N_{r,4}] = [2, 6, 12, 10]$, $[N_{\theta,1}, N_{\theta,2}, N_{\theta,3}, N_{\theta,4}] = [12, 24, 36, 60]$, $[L_{\min,1}, L_{\max,1}) = [2\text{ m}, 6\text{ m})$, $[L_{\min,2}, L_{\max,2}) = [6\text{ m}, 22\text{ m})$, $[L_{\min,3}, L_{\max,3}) = [22\text{ m}, 52\text{ m})$, and $[L_{\min,4}$, and $L_{\max,4}) = [52\text{ m}, 80\text{ m})$ for the CZM. Other parameters are shown in Table III. All methods are performed on a desktop computer equipped with an Intel i5-11600k CPU, 16 GB RAM, and Quadro RTX 5000 10 GB graphics card.

B. Dynamic Points Removal Test

In this test, we quantitatively compare DOR-LINS with the state-of-the-art method, ERASOR. To assess their performance, we use the rejection rate (RR) and preservation rate (PR), proposed by [10], as quantitative metrics specifically designed for a static map. Additionally, F_1 score is also used to evaluate their performance. These metrics are defined as follows.

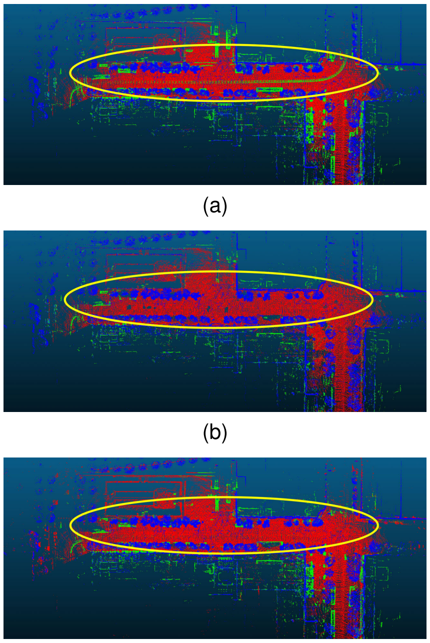

(a)
(b)
(c)

Fig. 4. Mapping results of LIO-SAM, ERASOR, and DOR-LINS in sequence 01 of CAMarketStreet. (a) Original map generated by LIO-SAM. (b) ERASOR. (c) DOR-LINS.

TABLE IV
COMPARISON WITH ERASOR ON THE SEQUENCES 01-03 OF CAMARKETSTREET DATASET

Seq.	Method	PR [%]	RR [%]	F_1 score
01	ERASOR	35.9	**91.8**	0.516
	DOR-LINS	**75.1**	88.8	**0.814**
02	ERASOR	47.6	**94.2**	0.632
	DOR-LINS	**73.5**	86.2	**0.793**
03	ERASOR	41.6	**93.8**	0.537
	DOR-LINS	**71.3**	85.6	**0.778**

1) *PR:* (# of preserved static points)/(# of total static points on the original map).
2) *RR:* $1 -$ (# of preserved dynamic points)/(# of total dynamic points on the original map).
3) *F_1 Score:* $2 \cdot (\text{PR} \cdot \text{RR})/(\text{PR} + \text{RR})$.

It should be noted that ERASOR requires a prior map and LiDAR poses. To ensure a fair comparison between ERASOR and DOR-LINS, we used the map and poses generated by LIO-SAM as inputs for ERASOR. Furthermore, since DOR-LINS employs the same feature extraction method as LIO-SAM, we used the map generated by LIO-SAM as the evaluation standard. We selected three short sequences from the CAMarketStreet dataset for testing. Fig. 4(a) shows the original map generated by LIO-SAM for sequence 01, with 1 236 624 static points and 11 768 dynamic points. Fig. 4(b) and (c) depict the maps generated after removing dynamic points using ERASOR and DOR-LINS, respectively. Fig. 4 shows that although

TABLE V

Absolute Trajectory RMSE [m] for LIO-SAM, LIO-SEGMOT, and DOR-LINS

Dataset	LIO-SAM	LIO-SEGMOT	DOR-LINS
CAMarketStreet	52.88	–	**20.61**
CARussianHill	51.18	–	**21.37**
CAChinaTown	27.46	–	**15.12**
CAColiTower	5.63	7.43	**4.16**
CALombardStreet	7.73	8.24	**7.26**
HK20190426-1	2.32	2.51	**2.01**
HK20190426-2	2.69	**2.17**	2.38

ERASOR rejects most moving points, it falsely removes many static points. This observation is supported by Table IV, which presents the quantitative metrics for ERASOR and DOR-LINS. Despite having a higher RR, ERASOR erroneously removes many static points, resulting in a lower PR. This indicates that, compared to our method, ERASOR is more susceptible to motion blur and prone to losing more static points. Consequently, our method has a better F_1 scores across all the sequences.

C. Odometry Test and Runtime Test

In this test, since LIO-SEGMOT and DOR-LINS adopt the LiDAR measurement model as LIO-SAM, we compare these three methods to show the performance of DOR-LINS in the general case. Table V lists the translational absolute trajectory error (ATE), in terms of root-mean-square errors (RMSEs), as the evaluation metrics for odometry. In practice, LIO-SEGMOT is tested by reducing the data playback speed by ten times, as it is nonreal-time, while LIO-SAM and DOR-LINS produce real-time results. The results indicate that LIO-SEGMOT performs well only in HK20190426-2 but performs worst in other datasets. This is because LIO-SEGMOT is not a real-time system. Its mapping thread slows down over time, resulting in slower processing speed. When the mapping thread becomes slow enough, it affects the results of factor graph optimization. However, the factor graph optimization of LIO-SAM and DOR-LINS can perform well. Therefore, LIO-SEGMOT performs poorly in some sequences and fails to run in the first three sequences. On the other hand, our method, DOR-LINS, shows the highest accuracy in the first six datasets. Compared to LIO-SAM, DOR-LINS improves trajectory accuracy by 60.02% in CAMarketStreet and 6.08% in CALombardStreet. This difference can be attributed to more dynamic objects and longer travel distances in CAMarketStreet, while CALombardStreet had fewer dynamic objects and shorter travel distances.

Additionally, Fig. 5(a) displays snapshots of CAMarketStreet along with the ground truth trajectory aligned with satellite images. Fig. 5(b)–(d) represent the maps generated by these methods in CAMarketStreet, with Fig. 5(c) being the result before LIO-SEGMOT fails. It is evident that DOR-LINS generates a better map that closely matches the driving trajectories.

Real-time capability is another crucial metric for SLAM. The feature extraction and mapping threads are the primary threads of both LIO-SEGMOT and DOR-LINS.

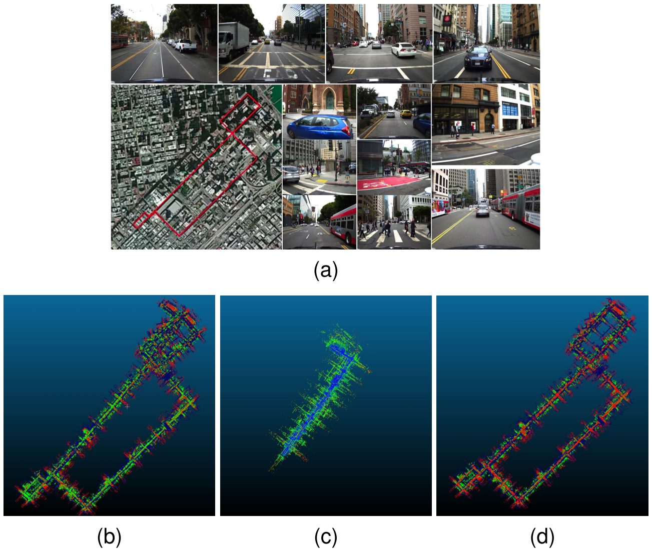


Fig. 5. Mapping results of LIO-SAM, LIO-SEGMOT, and DOR-LINS in CAMarketStreet. (a) Snapshots and ground truth of CAMarketStreet. (b) LIO-SAM. (c) LIO-SEGMOT. (d) DOR-LINS.

TABLE VI

Runtime of FET and MT. FET: Feature Extraction Thread, MT: Mapping Thread

Dataset	LIO-SEGMOT		DOR-LINS	
	FET	MT	FET	MT
CAMarketStreet	**38 ms**	3283 ms	55 ms	**78 ms**
CARussianHill	**37 ms**	3158 ms	53 ms	**76 ms**
CAChinaTown	**38 ms**	3158 ms	54 ms	**72 ms**
CAColiTower	**36 ms**	2154 ms	53 ms	**70 ms**
CALombardStreet	**36 ms**	1587 ms	56 ms	**65 ms**
HK20190426-1	**37 ms**	1959 ms	55 ms	**71 ms**
HK20190426-2	**37 ms**	1858 ms	57 ms	**69 ms**

Table VI presents the runtime of these two threads. The results indicate that LIO-SEGMOT is faster than DOR-LINS in the feature extraction module. It is because DOR-LINS incorporates additional ground extraction and object cluster capabilities in this thread, making it slightly slower than LIO-SEGMOT. However, DOR-LINS still maintains real-time performance in the feature extraction thread, with a runtime below 100 ms (the output frequency of the LiDAR). Regarding the mapping thread, LIO-SEGMOT is considerably slower, taking over 3s in the first three datasets, which leads to failure in the odometry test. Moreover, DOR-LINS demonstrates superiority across all datasets, consistently achieving computation times below 100 ms. Overall, our method ensures real-time processing in both threads.

V. Conclusion

This study introduces DOR-LINS, a LiDAR-Inertial odometry framework capable of real-time removal of dynamic objects. The feature extraction module of DOR-LINS initially identifies ground points and then clusters and labels nonground points as potential dynamic objects. Subsequently, based on this information, DOR-LINS removes moving points and performs state estimation during the mapping module. Notably, Our moving object removal method does not rely on prior training data, nor do specific moving objects limit it. Therefore, DOR-LINS can be applied in various urban environments. Through comprehensive experiments on the publicly available UrbanLoco dataset, our method demonstrates superior performance compared to state-of-the-art methods.

However, ongoing work remains for the refinement of DOR-LINS. In particular, specific challenging environments may pose difficulties in extracting ground points, thereby affecting the effectiveness of our method.

References

[1] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. Conf.*, Jul. 2014, pp. 1–9, doi: 10.15607/RSS.2014.X.007.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361, doi: 10.1109/CVPR.2012.6248074.

[3] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142, doi: 10.1109/IROS45743.2020.9341176.

[4] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021, doi: 10.1109/LRA.2021.3064227.

[5] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022, doi: 10.1109/TRO.2022.3141876.

[6] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537, doi: 10.1109/IROS40897.2019.8967704.

[7] P. Pfreundschuh, H. F. C. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart, and A. Cramariuc, "Dynamic object aware LiDAR SLAM based on automatic generation of training data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11641–11647, doi: 10.1109/ICRA48506.2021.9560730.

[8] H. Lim, M. Oh, and H. Myung, "Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3D LiDAR sensor," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6458–6465, Oct. 2021, doi: 10.1109/LRA.2021.3093009.

[9] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3D point cloud," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 13276–13283, doi: 10.1109/IROS47612.2022.9981561.

[10] H. Lim, S. Hwang, and H. Myung, "ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2272–2279, Apr. 2021, doi: 10.1109/LRA.2021.3061363.

[11] W. Wen et al., "UrbanLoco: A full sensor suite dataset for mapping and localization in urban scenes," in *Proc. ICRA*, Paris, France, May 2020, pp. 2310–2316, doi: 10.1109/ICRA40945.2020.9196526.

[12] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, "3D-MiniNet: Learning a 2D representation from point clouds for fast and efficient 3D LiDAR semantic segmentation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5432–5439, Oct. 2020, doi: 10.1109/LRA.2020.3007440.

[13] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu, "PointSeg: Real-time semantic segmentation based on 3D LiDAR point cloud," 2018, *arXiv:1807.06288*.

[14] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1887–1893, doi: 10.1109/ICRA.2018.8462926.

[15] X. Chen et al., "Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6529–6536, Oct. 2021, doi: 10.1109/LRA.2021.3093567.

[16] Y.-K. Lin, W.-C. Lin, and C.-C. Wang, "Asynchronous state estimation of simultaneous ego-motion estimation and multiple object tracking for LiDAR-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 10616–10622, doi: 10.1109/ICRA48891.2023.10161269.

[17] W. Zheng, W. Tang, L. Jiang, and C. Fu, "SE-SSD: Self-ensembling single-stage object detector from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14489–14498, doi: 10.1109/CVPR46437.2021.01426.

[18] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and G. Yalla, "Robust method for removing dynamic objects from point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 10765–10771, doi: 10.1109/ICRA40945.2020.9197168.

[19] C. Chen and B. Yang, "Dynamic occlusion detection and inpainting of in situ captured terrestrial laser scanning point clouds sequence," *ISPRS J. Photogramm. Remote Sens.*, vol. 119, pp. 90–107, Sep. 2016, doi: 10.1016/j.isprsjprs.2016.05.007.

[20] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, "An approach to extract moving objects from MLS data using a volumetric background representation," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 107–114, Jun. 2017, doi: 10.5194/isprs-annals-IV-1-W1-107-2017.

[21] J. Schauer and A. Nüchter, "The peopleremover—Removing dynamic objects from 3-D point cloud data by traversing a voxel occupancy grid," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1679–1686, Jul. 2018, doi: 10.1109/LRA.2018.2801797.

[22] R. Ambrus, N. Bore, J. Folkesson, and P. Jensfelt, "Meta-rooms: Building and maintaining long term spatial models in a dynamic world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1854–1861, doi: 10.1109/IROS.2014.6942806.

[23] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3D map maintenance in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2014, pp. 3712–3719, doi: 10.1109/ICRA.2014.6907397.

[24] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10758–10765, doi: 10.1109/IROS45743.2020.9340856.

[25] C. Qian, Z. Xiang, Z. Wu, and H. Sun, "RF-LIO: Removal-first tightly-coupled LiDAR inertial odometry in high dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4421–4428, doi: 10.1109/IROS51168.2021.9636624.

Zhoubo Wang received the M.S. degree in mechanical engineering from the Inner Mongolia University of Technology, Hohhot, China, in 2016. He is currently pursuing the Ph.D. degree in mechanical engineering with the Beijing Institute of Technology, Beijing, China.

His research interests include simultaneous localization and mapping (SLAM) for autonomous vehicles.

Zhenhai Zhang received the Ph.D. degree in mechatronics engineering from the Beijing Institute of Technology, Beijing, China, in 2008.

He is an Associate Professor with the School of Mechatronical Engineering, Beijing Institute of Technology. He has published more than 60 SCI and EI papers in IEEE Sensors Journal, *Measurement*, *Optics Express*, *Optics Letters*, and other academic journals. His research interests include environmental perception, recognition, and control for intelligent unmanned system.

Dr. Zhang has obtained the Second Prize of the Defense Technological Invention Award.

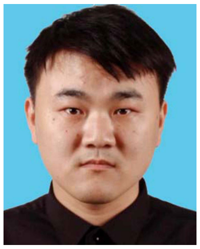
Xiao Kang received the Ph.D. degree in mechatronics engineering from the Beijing Institute of Technology, Beijing, China, in 2013.

She is currently a Researcher with China North Vehicle Research Institute, Beijing. Her research focuses on the object detection recognition and information fusion for unmanned systems.

Dr. Kang serves as a Council Member of Sensor Branch of China Instrument and Control Society.

Miusi Wu received the B.S. degree in mechatronics engineering from the Beijing Institute of Technology, Beijing, China, in 2021, where he is currently pursuing the M.S. degree in mechanical engineering.

His research interests include design of mechatronic systems and inertial navigation.



Siyu Chen received the B.S. degree in process equipment and control engineering from the Hefei University of Technology, Hefei, China, in 2022. He is currently pursuing the M.S. degree in mechanical engineering with the Beijing Institute of Technology, Beijing, China.

His research interests include LiDAR-based moving object detection for autonomous driving.

Qilin Li received the B.S. degree in mechanical engineering from the Beijing University of Technology, Beijing, China, in 2022. He is currently pursuing the M.S. degree in mechanical engineering with the Beijing Institute of Technology, Beijing.

His research interests include object detection and tracking in complex environment.