IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

RH-Map: Online Map Construction Framework of Dynamic Object Removal Based on 3D Region-Wise Hash Map Structure

Zihong Yan, Xiaoyi Wu, Zhuozhu Jian, *Graduate Student Member, IEEE*, Bin Lan, and Xueqian Wang

***Abstract*—Mobile robots navigating in outdoor environments frequently encounter the issue of undesired traces left by dynamic objects and manifested as obstacles on map, impeding robots from achieving accurate localization and effective navigation. To tackle the problem, a novel map construction framework based on 3D region-wise hash map structure (*RH-Map*) is proposed, consisting of front-end *scan refresh* and back-end removal modules, which realizes real-time map construction and online dynamic object removal (DOR). First, a two-layer 3D region-wise hash map structure of map management is employed for effective online DOR. Then, in *scan refresh*, region-wise ground plane estimation (R-GPE) is proposed for incrementally estimating and preserving ground information, and Scan-to-Map Removal (S2M-R) is proposed to discriminate and remove dynamic objects. Moreover, the lightweight back-end removal module maintaining keyframes is proposed for further DOR. As experimentally verified on SemanticKITTI, our proposed framework yields promising performance on online DOR of map construction compared with state-of-the-art methods. We also validate the proposed framework in real-world environments.**

***Index Terms*—Mapping, range sensing, dynamic object removal.**

I. Introduction

CLEAN and reliable maps play an important role in navigation and exploration of mobile robot platforms such as Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs) in outdoor environments. And Simultaneously Localization and Mapping (SLAM) techniques are usually applied to provide robust localization and mapping of surroundings [1], [2]. However, most SLAM methods assume that environments are relatively static; therefore, dynamic objects in the surroundings may leave residual traces on map as shown in Fig. 1(a), known as *ghost trail effect* [3]. These usually lead to confusion in environment representation and sub-optimal results of path planning. There are SLAM researches aiming at tackling this assumption by either employing online dynamic object removal (DOR) [4], [5], [6] or utilizing states of dynamic objects in optimization [7] to ensure more reliable localization in dynamic environments. Nevertheless, they focus more on improving the accuracy of localization rather than providing a dense and clean map.

In this work, we focus on real-time DOR for online dense map construction in large-scale environments using 3D LiDAR. Some researches tackling DOR mission in map construction typically require long sequences of historical information, submap or pre-built global map [8], [9], [10], which are unsuitable for

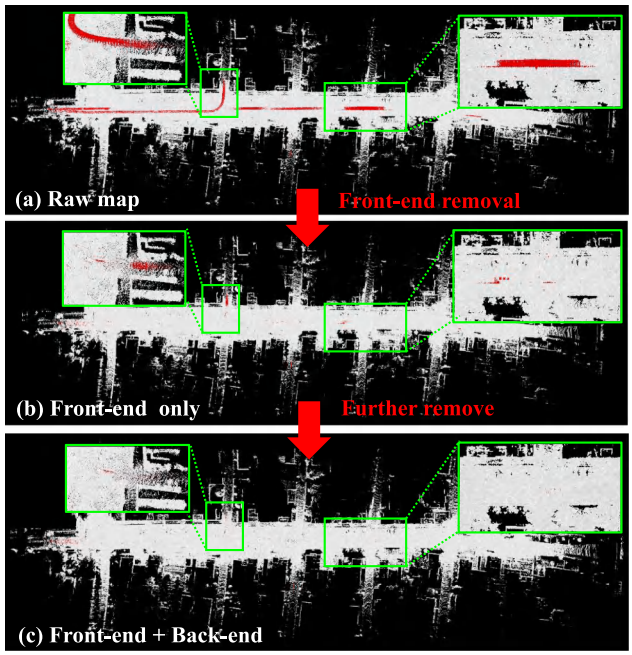

Fig. 1. (a) Raw point cloud map. (b) Processing via the Front-end module only. (c) Further removal of dynamic objects with the Back-end module. Run at 10 Hz on sequence 05 of SemanticKITTI (dynamic objects in red).

Manuscript received 24 July 2023; accepted 26 November 2023. Date of publication 13 December 2023; date of current version 5 January 2024. This letter was recommended for publication by Associate Editor M. Hanheide and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant 62293545. *(Zihong Yan and Xiaoyi Wu are co-first authors.) (Corresponding author: Xueqian Wang.)*

Zihong Yan, Zhuozhu Jian, and Xueqian Wang are with the Center for Artificial Intelligence and Robotics, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China (e-mail: yanzh22@mails.tsinghua.edu.cn; jzz21@mails.tsinghua.edu.cn; wang.xq@sz.tsinghua.edu.cn).

Xiaoyi Wu is with the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: 200320306@stu.hit.edu.cn).

Bin Lan is with the Jianghuai Advance Technology Center, Hefei 230022, China (e-mail: lanbin.thu@gmail.com).

The source code is released to the community. RH-Map: https://github.com/YZH-bot/RH-Map

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3342548, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3342548

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

scenarios with high real-time requirements. Additionally, online DOR methods rely on short-term neighboring frames of LiDAR information, which can filter out most of the dynamic objects in the current scan. However, due to the sparsity and incidence angle ambiguity characteristics of LiDAR in distant areas, some dynamic objects may persist and remain on the map indefinitely while some static points are misclassified, thereby degrading map quality. We summarize the challenges as follows: 1) the difficulty in maintaining the timeliness of map construction while simultaneously filtering out dynamic objects, and 2) the weak performance of online DOR due to characteristics of LiDAR data, negatively impacting the quality of the map.

This work proposes a novel online map construction framework of DOR in dynamic environments referred to as *RH-Map*. First, we adopt a two-layer hash map structure to manage map with 3D voxel blocks and minimum resolution voxels, which leverages the effectiveness of region-wise structure and $O(1)$ complexity of hash map for online DOR and real-time map update. Based on the region-wise structure, a two-stage online mapping framework for constructing static map including front-end and back-end modules is introduced. The front-end module estimates ground information for preservation and removes potential dynamic objects on map utilizing the current scan. And the back-end module maintains a keyframe queue of historical information to further remove residual dynamic objects effectively as shown in Fig. 1. We summarize the contributions as follows:

1) We propose *RH-Map*, an online static map construction framework that fuses ground segmentation-based method and 3D region-wise hash map structure. *RH-Map* facilitates online DOR and ensures real-time map update, achieving real-time performance over 10 Hz on SemanticKITTI.
2) A real-time front-end *scan refresh* and a lightweight *back-end* removal module are proposed. *Scan refresh* achieves robust ground estimation and real-time online DOR, and the *back-end* module realizes further removal of dynamic objects to enhance map quality.
3) We validate the proposed method on SemanticKITTI thoroughly and demonstrate the online performance in real-world scenes. Our method achieves state-of-the-art performance in DOR and can apply to robot navigation in real-world.

II. Related Works

In this section, we review the related works about DOR of static map construction, which mainly fall into three categories: occupancy map-based methods, visibility-based methods, and segmentation-based methods.

Occupancy map-based methods: Based on the principle of ray propagation, occupancy map-based methods update the probability of ray endpoint as occupied and space traveled by ray as free by using voxel-traversal [11], [12]. However, voxel-traversal suffers from massive calculations [8], [10], [13] and severe misclassification caused by incidence angle ambiguity [9]. Schauer and Nüchter [12] propose to traverse voxel grid to find differences in the occupancy between the scans thus estimating the voxel as dynamic or static, which alleviates the loss of ground information to some extent. Pagad et al. [3] performs a combination of object detection and Octomap with hardware GPU acceleration. Despite the aforementioned algorithmic and engineering optimizations, occupancy map-based methods using voxel-traversal are still challenging for real-time online DOR.

Visibility-based methods: To address the computational complexity of voxel-traversal, visibility-based approaches have been introduced based on range image disparity between LiDAR and map to determine and remove dynamic points on map [8]. However, visibility-based still suffers from incidence angle ambiguity. Kim and Kim [8] propose a visibility-based approach based on multi-resolution range images to construct static map offline. And Fu et al. [14] propose an inverse way of projection, i.e., they back project the map onto each frame and perform a map-to-frame comparison instead to avoid quantization error and achieve state-of-the-art performance with terrain modeling. Compared to offline methods utilizing long sequences of information, online DOR paradigm requires real-time performance and can only rely on short sequences of real-time information, which limits the removal performance [10]. The sparsity issue also impacts the performance of online DOR. Therefore, Fan et al. [10] proposed an online DOR framework integrating submap-based and visibility-based methods and first proposed front-end and back-end framework utilizing historical information for further removal. Nonetheless, they only rely on certain thresholds to restrict the incidence angle, which cannot effectively preserve the ground.

Segmentation-based methods: We further divide this category into scan pre-segmentation methods and ground segmentation methods based on the segmentation stage.

a) Scan pre-segmentation methods primarily rely on frame-to-frame discrepancies or deep learning techniques to detect dynamic objects in the current scan. Yoon et al. [13] utilize inter-frame LiDAR information discrepancies and clustering to detect and remove dynamic regions. In addition, deep learning methods have shown great potential in providing semantic information from pre-trained neural network models and performing filtering for DOR in recent years [15], [16], [17]. To address the issue of mis-segmenting static semantic objects in the environment, for example, parked cars, Sun et al. [18], [19] use LiDAR semantic segmentation networks based on residual range images of sequential data to differentiate between dynamic and static objects in LiDAR scans. However, pre-segmentation methods do not focus on the establishment of static maps [20], resulting in incorrectly segmented points being permanently retained on the static map, and deep learning methods heavily depend on meticulously annotated datasets.

b) Ground segmentation methods refer to the process of separating ground information from the LiDAR scans [21], [22], [23], [24] or maps [9], [14], [25], enabling further dynamic object detection and tracking, and preserving the integrity of ground information. Arora et al. [26] integrate offline ground segmentation into OctoMap to improve the distinction between moving objects and static road backgrounds. Lim et al. [9] apply Region-wise Ground Plane Fitting (R-GPF) by conducting principal component analysis (PCA) iteratively to fit the ground plane on its region-wise map structure with low computational load for static map construction, which leverages the advantages of region-wise data structure to estimate plane precisely. However, they did not take into account the adverse effects of noise points from LiDAR. And wang et al. [27] propose a hierarchical vertical height descriptor in local perspective based on distribution description to better reflect data distribution in each region. However, both of these partitioning methods represent

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

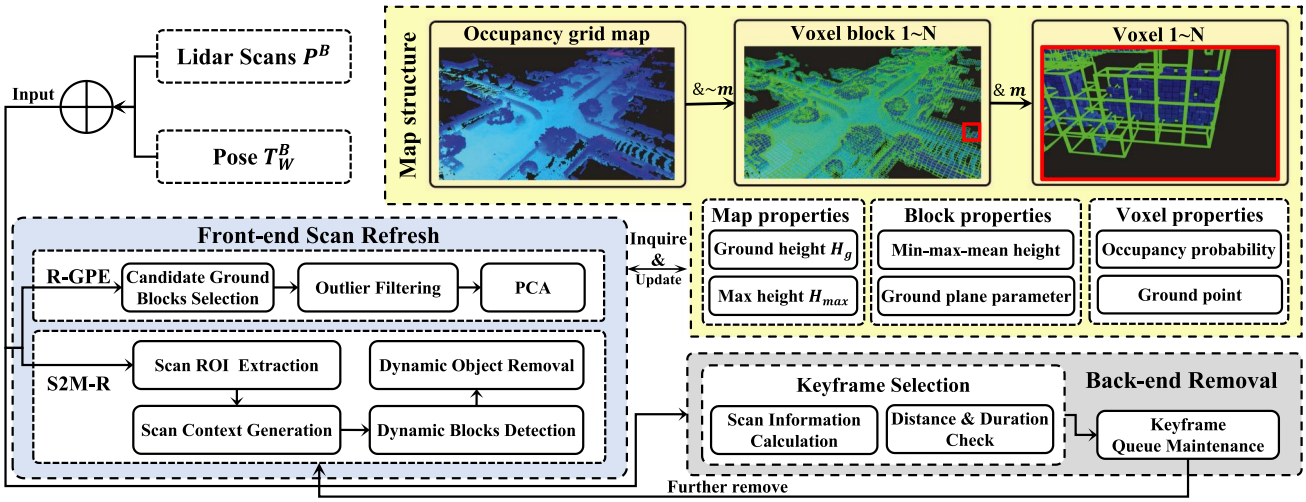

Fig. 2. Overview of RH-Map: the map is constructed based on region-wise map structure as shown in the top right corner. Given the current scan P_B and LiDAR-to-world transformation T_B^W estimated from LiDAR odometry, the region-wise ground plane estimation (R-GPE) estimates the ground plane of each block and updates the properties of map, voxel blocks, and voxels. Meanwhile, Scan-to-Map Removal (S2M-R) detects candidate blocks with dynamic objects and removes dynamic objects based on the properties. With keyframe selection, the back-end removal module maintains a keyframe queue and further removes dynamic object residues on the map.

3D information using 2D or 2.5D elevation descriptors, which leads to a loss of information as reflected in Fig. 5(b).

III. Methodology

The schematic diagram of RH-Map is illustrated in Fig. 2. Based on the two-layer hash map structure, RH-Map utilizes 3D voxel blocks to represent the information of local areas for DOR and each block stores the state of voxels (minimum resolution) in the space. And RH-Map consists of two parts: the front-end *scan refresh* and the back-end removal modules based on the proposed data structure.

A. Problem Definition

Different from previous works [8], [9], [12], we focus on online DOR of map constructed by consecutive LiDAR scans in urban environments, and long-term mapping, such as the redevelopment or restoration of buildings, are beyond the scope of this work. And we use state-of-the-art LiDAR-based Odometry or SLAM methods such as [1], [2] to provide SE(3) pose T_B^W between world frame W and LiDAR frame B for real-world application, and receive real-time scan P^B as input.

B. Region-Wise Hash Map Structure

In this section, we employ a two-layer hash map structure for efficient management as shown in Fig. 2, unlike ERASOR's 2D egocentric Region-wise Pseudo Occupancy Descriptor (R-POD) [9], the global map M is divided into voxel blocks, and each block is further divided into minimum resolution voxel. In the region-wise map structure (see Fig. 3), the value of binary mask m and minimum resolution voxel size s_v determines the voxel block size $s_b = (m+1) * s_v$ (e.g. if $m = 00000111$, $s_v = 0.1\,\text{m}$, then $s_b = 0.8\,\text{m}$). For map construction, point cloud from scan is first transformed to the world frame and converted to global index I, then conducted binary operation with the binary mask m on each dimension to get the

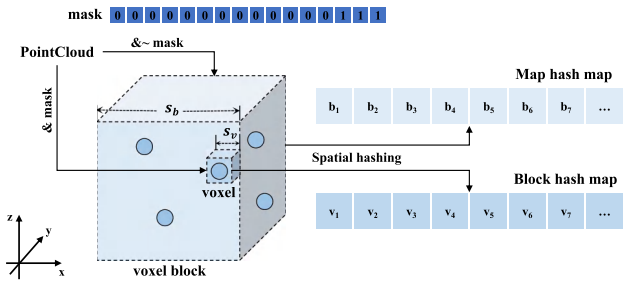

Fig. 3. Structure of 3D region-wise hash map including block and voxel.

corresponding voxel block index I_b and voxel index I_v which compose a unique index in RH-Map:

$$I = \lfloor \frac{1}{s_v} [p_x, p_y, p_z]^T \rfloor,\ I_b = I\ \&\ {\sim}m,\ I_v = I\ \&\ m, \quad (1)$$

where $[p_x, p_y, p_z] \in \mathbb{R}^3$ are the global coordinates of point cloud, and obviously, $I = I_b \| I_v$. After getting the I_b and I_v, we hash the indexes into the map using a 3D hash function like [28] and update the occupancy probability. Our method supports online operation and the hash map structure only updates the state of the occupied places and is free from a presetting range of maps like traditional occupancy grid map or octomap [11], while R-POD [9] can only be applied in an offline manner due to the irregularity of data organization, and the time consumed will increase as the global map becomes larger.

For notations, M_{I_b} is used to access the voxel block with index I_b on the map M. And $O_b = (I_b(x), I_b(y))$ denotes the 2D block index on xy plane. Map M records the max height $H_{\max}$ and average height H_g of ground points on each 2D block index. Each block augments with min-max-mean height and plane parameters. And each voxel records its occupancy probability and ground attribute.

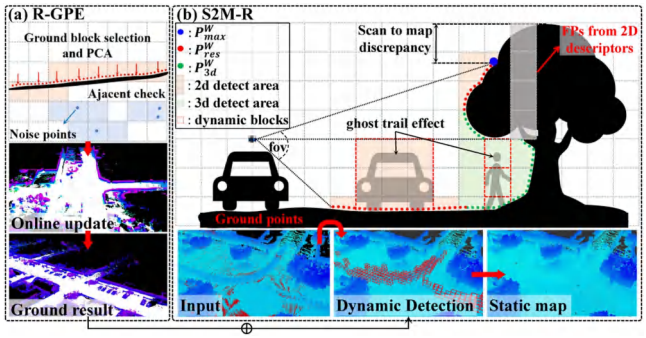

Fig. 4. (a) Illustrates the process of R-GPE, including blocks selection and adjacent check to exclude noise points for ground fitting based on region-wise PCA. It can update ground information online (white points), and the final result is shown at the bottom. (b) Scan P^B is first extracted as $P^B_{\max}$, P^B_{res} and P^B_{3d}, and are transformed to W frame for dynamic blocks detection using 2D and 3D scan context, and finally combine with the ground estimation to remove dynamic objects.

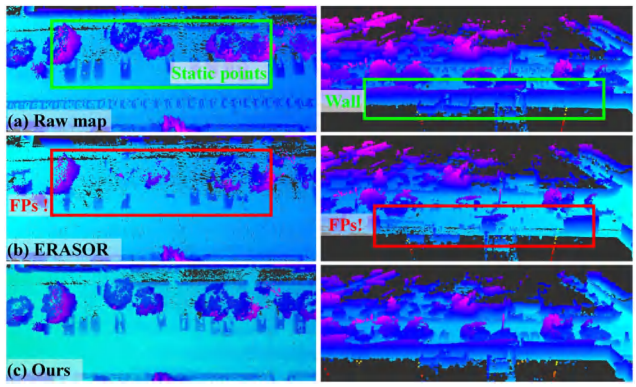

Fig. 5. Effect of ROI extraction on sequence 05 of SemanticKITTI. (a) Raw map. (b) ERASOR [9]. (c) Ours with 2D&3D scan contexts.

C. Scan Refresh Front-End

The front-end *scan refresh* contains *Region-wise Ground Plane Estimation* (R-GPE) and *Scan-to-Map Removal* (S2M-R). Our *scan refresh* performs online DOR in S2M-R and applies R-GPE to preserve the integrity of the ground based on the region-wise map structure mentioned in Section III-B.

1) Region-Wise Ground Plane Estimation: Real-time scan data P^B is transformed into world frame $P^W = T^W_B P^B$ and inserted into our map to update max height in $H_{\max}$ of map and max-min-mean height properties of voxel blocks for dynamic blocks detection in Section III-C2. *Candidate ground blocks selection* is conducted to determine the range within which ground plane fitting needs to be performed for acceleration. Moreover, to avoid the influence of noise points on ground estimation, we introduce a simple outlier filtering strategy, categorizing a block as noise if there are no voxels within its adjacent blocks on the xy plane (8 neighbors) as shown on the left of Fig. 4. This is founded on the knowledge that ground usually exhibits continuity, while noise points tend to be isolated. And the candidate ground blocks set ${}^{cgb}\mathbb{I}_b$ is obtained as follows:

$$\begin{aligned} {}^{cgb}\mathbb{I}_b = \{ I^i_b \mid I^i_b = I^i \,\&\sim m, I^i \in \mathbb{I}^{P^W}, \\ I^i(z) < H_g(O^i_b) + \delta, M^N_{I^i_b} \neq \varnothing \} \end{aligned} \tag{2}$$

where $\mathbb{I}^{P^W}$ represents the set of global indexes converted from P^W by Formula (1), $M^N_{I^i_b}$ denotes eight adjacent blocks of $M_{I^i_b}$, δ is a preset height threshold according to different scenes, and $O^i_b = (I^i(x), I^i(y))$. Note that if $H_g(O^i_b)$ is empty, we give an estimated initial value through the height of LiDAR.

Then, we perform principal component analysis (PCA) on each block $M_{I^i_b}$ ($I^i_b \in {}^{cgb}\mathbb{I}_b$). The global indexes of voxels in $M_{I^i_b}$ with smaller z-values than the mean height are extracted as initial ground points set $\mathbb{I}^{I^i_b}$. Then we employ PCA to fit the ground plane and update ground plane parameters of each block $M_{I^i_b}$ with normal vector n_i and plane coefficient $d_i = n_i \overline{I^i}$ where $\overline{I^i}$ denotes the average index of $\mathbb{I}^{I^i_b}$. And the global indexes set ${}^{gro}\mathbb{I}^{I^i_b}$ of ground points in block $M_{I^i_b}$ is extracted as follows:

$${}^{gro}\mathbb{I}^{I^i_b} = \{ I^j \mid I^j \in \mathbb{I}^{I^i_b}, |n_i I^j - d_i| < r_{gro} \} \tag{3}$$

where r_{gro} denotes the distance margin of the plane. Finally, we update the ground property of voxels with indexes in ${}^{gro}\mathbb{I}^{I^i_b}$. Note that R-GPE will be performed only if new candidate ground voxel blocks are detected which enhances the efficiency and noise immunity of ground estimation. The effect of R-GPE is demonstrated on the bottom left of Fig. 4.

2) Scan-to-Map Removal: The essence of ground segmentation-based dynamic object removal methods is to compare the differences in height-related metrics between the scan and the map while preserving the integrity of the ground. Generally, information from a single scan is relatively sparse and information-deprived compared to the map. Existing methods [9], [27] that use 2D or 2.5D descriptors are simple and computationally efficient but may sacrifice some of the 3D information. Firstly, they may fail to detect dynamic points (FNs) such as *ghost trail effect* of pedestrians under the tree as shown in Fig. 4. Moreover, with a fixed detection volume of interest, these methods usually lead to instances of false positives (FPs, falsely classify static points as dynamic points), as seen in the gray blocks containing part of the tree in Fig. 4. More FPs results are illustrated in Fig. 5(b). These situations frequently occur in points that have a larger range distance than the points above them (green points in Fig. 4). To solve these problems, we conduct Regions of Interest (ROI) extraction of scan P^B and introduce 2D and 3D *scan context* for dynamic blocks detection to avoid the FPs and FNs mentioned above.

Scan ROI Extraction: First of all, the limited field of view (FOV) of the LiDAR results in a discrepancy between the observation of sensor and the map, especially in close proximity to the sensor, e.g., the upper part of the tree in Fig. 4. The comparison between scan and map must be constrained within the LiDAR's FOV. Therefore, we project P^B into range image $\mathcal{I}^B_{m\times n}$ and extract the max ring of LiDAR points from P^B, denoted as $P^B_{\max}$, to represent the maximum FOV of LiDAR. Secondly, for the green points, denoted as P^B_{3d}, it is necessary to compare each block (green blocks in Fig. 4) under its upper bound with map separately to determine the presence of objects, instead of using 2D height-related metrics on the corresponding area directly. The extraction of these points and their upper bounds can be

obtained through the following formula:

$$P^B_{3\,d} = \{p_k.ub = Z_{i+t} \mid p_k \in P^B, \\ \mathcal{I}^B(i,j) - \mathcal{I}^B(i+t,j) > r\} \tag{4}$$

where $\mathcal{I}^B_{m\times n}$ denotes the range image of P^B, $\mathcal{I}^B(i,j)$ denotes corresponding range distance of p_k, $\mathcal{I}^B(i+t,j)$, $t \in [0,1\ldots]$ denotes corresponding range distance of point above p_k by searching upwards until the difference in range distance exceeds threshold r, and Z_{i+t} represents the height of the points corresponding to $\mathcal{I}^B(i+t,j)$ and is stored in ub as the height limit for searching potential dynamic blocks.

And the rest points of P^B are represented as P^B_{res} (red points in Fig. 4). Scan P^B is then transformed to world frame P^W as well as the upper bound property for the subsequent part of the algorithm.

Dynamic Blocks Detection and Removal: For $P^W_{\max}$ and P^W_{res}, we introduce 2D region-wise *scan context* ${}^{2\,d}S$ to detect the differences between scan and map for dynamic objects detection, which is defined as follows:

$${}^{2\,d}S_{O^i_b} = Z^{\max}_{O^i_b}, O^i_b \in \mathbb{O}_b^{\{P^W_{\max}, P^W_{res}\}} \tag{5}$$

where $\mathbb{O}_b^{\{P^W_{\max}, P^W_{res}\}}$ represents the set of 2D block indexes of $P^W_{\max}$ and P^W_{res}, $Z^{\max}_{O^i_b}$ denotes the maximum height of $\{P^B_{\max}, P^B_{res}\}$ on 2D block index O^i_b. Note that $\mathbb{O}_b^{P^W_{res}}$ and $\mathbb{O}_b^{P^W_{\max}}$ might have an intersection due to the block size. We calculate the $ratio_1$ between ${}^{2d}S$ and the corresponding area of map M as follows:

$$ratio_1^{O^i_b} = \begin{cases} \frac{{}^{2\,d}S_{O^i_b}}{H_{\max}(O^i_b)}, & O^i_b \in \mathbb{O}_b^{P^W_{res}} \setminus \mathbb{O}_b^{P^W_{\max}} \\ 1, & O^i_b \in \mathbb{O}_b^{P^W_{\max}} \end{cases} \tag{6}$$

where $H_{\max}(O^i_b)$ represents the max height of map M at 2D block index O^i_b. If $ratio_1^{O^i_b}$ is less than removal thresholds δ_1, we consider the blocks with 2D block index O^i_b to be dynamic (red dashed squares in Fig. 4).

For the green blocks in Fig. 4, 2D descriptor cannot distinguish whether there are dynamic objects such as the points left by pedestrians under the tree. Therefore, it is necessary to identify and compare the blocks between these points to detect dynamic objects. First, we convert $P^W_{3\,d}$ to block indexes $\mathbb{I}_b^{P^W_{3\,d}}$, and search upward to reach the minimum upper bound ub of each block in $\mathbb{I}_b^{P^W_{3\,d}}$ for the set of potential dynamic blocks $\mathbb{I}^*_b$. And then we introduce 3D *scan context* ${}^{3\,d}S$ to perform dynamic object detection within these blocks. We define ${}^{3\,d}S$ as follows:

$${}^{3\,d}S_{I^i_b} = \{Z_{\max}, Z_{\min}\}, I^i_b \in \mathbb{I}_b^{P^W_{3\,d}} \cup \mathbb{I}^*_b \tag{7}$$

where $Z_{\max}$, Z_{min} are parameters that denote the maximum, minimum height within block I^i_b, respectively. We calculate the $ratio_2$ between ${}^{3\,d}S$ and the corresponding block of map M as follows:

$$ratio_2^{I^i_b} = \frac{{}^{3\,d}S_{I^i_b}.Z_{\max} - {}^{3\,d}S_{I^i_b}.Z_{\min}}{M_{I^i_b}.Z_{\max} - M_{I^i_b}.Z_{\min}}, I^i_b \in \mathbb{I}_b^{P^W_{3\,d}} \cup \mathbb{I}^*_b \tag{8}$$

Obviously, $ratio_1$ and $ratio_2$ are close to 1 if the scan perfectly matches the global map. If $ratio_1^{O^i_b} < \delta_1$ and $ratio_2^{I^i_b} < \delta_2$, we consider the blocks located at O^i_b and the block $M_{I^i_b}$ to be dynamic as shown in Fig. 4. After dynamic object detection, dynamic blocks are removed excluding ground points extracted in Section III-C1. The effect of combining the 2D and 3D scan contexts and the procedure of *S2M-R* are demonstrated in Figs. 4 and 5(c).

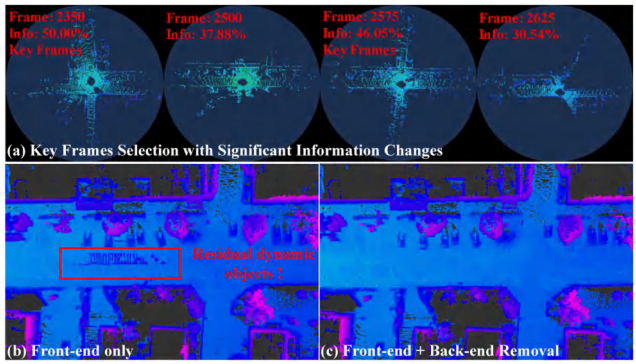

Fig. 6. Back-end removal module. (a) Information content of different LiDAR frames. (b) Front-end module only with progressively accumulateda residues on the map (red box). (c) Effect after incorporation of the Back-end removal module.

D. Back-End Removal

To obtain higher-quality static maps, historical information such as multiple submaps [10] is considered for further removal online, compensating for the insufficiency of information from short-term scans. Regardless of the methods employed, the information becomes sparser as the distance from ego center increases, ultimately resulting in inevitable presence of dynamic object residue on the map by using short-term scans as shown in Fig. 6(b). Therefore, it is essential to introduce back-end to reuse historical information which is first proposed in [10]. However, the use of sub-maps introduces a significant computational cost. In this section, we design a lightweight back-end removal module that can run at high frequency with the front-end leveraging historical information to handle residual dynamic objects.

Our back-end removal module maintains a keyframe queue, from which keyframe information is processed using the S2M-R algorithm for further DOR when the robot moves a certain distance away. The selection of keyframe primarily depends on distance, time interval and the amount of information captured by the scan. We define information content of a LiDAR frame as follows:

$$V(\mathcal{I}^B_{m\times n}) = \frac{\sum_{j=1}^{n} \max\limits_{1\le i\le m} \mathcal{I}^B_{m\times n}(i,j)}{r_{\max} \times n} \times 100\% \tag{9}$$

where $r_{\max}$ denotes the max range distance of LiDAR. The area occupied by one scan frame within the blue circle as shown in Fig. 6(a), represents the amount of information it contains. The information content at an intersection is often greater than that on a single-lane road. When the information content of a frame changes significantly compared to the previous frames, it is selected as a keyframe to avoid information loss. When the distance or time interval between current frame and previous keyframe exceeds given thresholds, it is also selected as a keyframe. The effect after the application of back-end removal module is shown in Fig. 6(c).

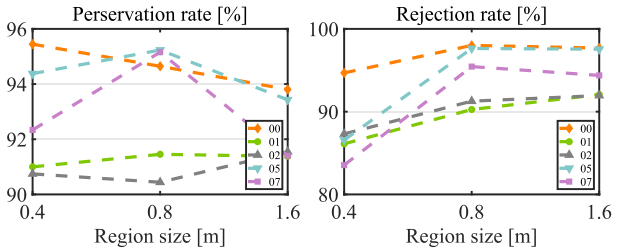

Fig. 7. Performance on SemanticKITTI dataset of different voxel block sizes.

IV. Experiments

A. Experimental Setup

To evaluate the performance of our algorithm, we conduct comprehensive comparisons against existing open-source methods on SemanticKITTI [29]. To assess the quality of the static map that has been retained after removal of dynamic points, we adopt the preservation rate (PR) and rejection rate (RR) as evaluation metrics [9]. The evaluation metrics are defined as follows:

$$PR = N^{TN}/N^{sta} \tag{10}$$

$$RR = 1 - N^{TP}/N^{dyn} \tag{11}$$

$$F1\ score = 2PR \cdot RR/(PR + RR) \tag{12}$$

where N^{sta}, N^{dyn} represent the nums of total static and dynamic points on the ground truth, and N^{TN}, N^{TP} represent the nums of preserved static and dynamic points of generated map. F1 score is also computed to reflect the harmonic mean of precision and recall metrics.

Additionally, we conduct real-world experiments of the proposed RH-Map in campus environments with pedestrians and vehicles. Moreover, we integrate RH-Map into open-source navigation system PUTN [30] to demonstrate the effectiveness for motion planning of mobile robots.

B. Evaluation

1) Impact of Voxel Block Size: Firstly, we investigate the influence of block size with 0.4 (16-bit m: $2^2 - 1$), 0.8 (16-bit m: $2^3 - 1$) and 1.6 m (16-bit m: $2^4 - 1$) on online DOR effect of our RH-Map. As shown in Fig. 7, it can be observed that increasing block size leads to a decline in PR value, and setting region size as 0.4 m led to a decline in RR value. We believe that a smaller block size can lead to inaccurate ground plane fitting, resulting in poor DOR performance. On the other hand, increasing the block size does not significantly improve the DOR performance and actually reduces the retention of static points. We attribute this to the fact that larger blocks may misclassify some non-ground static points as dynamic points when removing points except the ground point. Therefore, we set 0.8 m as the block size in RH-Map for the balance.

2) Comparison With State-of-The-Art Methods: We conduct the comparative experiments with Octomap[1] [11] (voxel size: 0.1 m), Removert-RM3[2] [8] (3 remove stages with pre-built map of voxel size 0.05 m), ERASOR[3] [9] (pre-built map of voxel size 0.1 m) and Ours (voxel size: 0.1 m). Table I presents quantitative comparisons of the results of DOR and the quality of static map can be seen in Fig. 8. All the results can be reached at the open-source repository.

We perform Octomap offline to filter out dynamic objects as much as possible in this experiment due to its poor real-time performance. As described in Table I, Octomap achieves the highest performance in terms of DOR. However, due to the angle ambiguity of voxel-traversal, vast quantities of static points are falsely classified as dynamic, resulting in lower PR values. Removert achieves good results on most sequences, but it generates a significant number of FPs on sequence 02 with slopes, resulting in a relatively low PR value. We believe that it is due to the incidence angle ambiguity of visibility method, which requires more handling of the incident angles. ERASOR, belonging to segmentation methods, achieves a low PR value on certain sequences due to the use of 2D R-POD descriptors with information loss, leading to some FPs. In contrast, our method is based on a 3D spatial data structure, which helps to alleviate this issue. Note that in the 02 sequence on the highway, there are guardrails in the middle, making it challenging to detect vehicles traveling in the opposite lane, which may hinder our algorithm from achieving better performance. Nonetheless, the presence

TABLE I
Comparison With State-of-The-Art Methods on SemanticKITTI

Seq.	method	PR(%)	RR(%)	F1 score
00	OctoMap* [11]	57.5321	**99.9623**	0.730317
	Removert-RM3* [8]	93.2048	99.3169	0.961638
	ERASOR* [9]	90.5091	95.9936	0.978302
	RH-Map (Front-end)	**97.8724**	97.788	**0.978302**
	RH-Map (Ours)	94.6446	98.005	0.962955
01	OctoMap* [11]	52.6362	**99.8929**	0.68944
	Removert-RM3* [8]	**95.3201**	95.0313	**0.951755**
	ERASOR* [9]	89.6322	94.1192	0.918209
	RH-Map (Front-end)	92.9481	77.1859	0.843369
	RH-Map (Ours)	91.454	90.2589	0.908525
02	OctoMap* [11]	41.8728	**99.9293**	0.590163
	Removert-RM3* [8]	79.4783	93.0953	0.857496
	ERASOR* [9]	80.4437	96.2932	0.876578
	RH-Map (Front-end)	**98.855**	77.2788	0.867
	RH-Map (Ours)	90.4409	91.2651	**0.908511**
05	OctoMap* [11]	49.1314	**99.774**	0.658409
	Removert-RM3* [8]	91.7955	86.4054	0.890189
	ERASOR* [9]	96.5607	92.9508	0.947214
	RH-Map (Front-end)	**97.2321**	96.6255	**0.969278**
	RH-Map (Ours)	94.713	98.5007	0.965697
07	OctoMap* [11]	65.9977	**99.2425**	0.792758
	Removert-RM3* [8]	91.7955	86.4054	0.890189
	ERASOR* [9]	91.5798	98.536	0.949306
	RH-Map (Front-end)	**97.3686**	91.2949	0.94234
	RH-Map (Ours)	96.4697	97.8803	**0.971699**

"*" means that the algorithm operates offline.

TABLE II
Real-Time Performance of RH-Map on SemanticKITTI

Seq.	Runtime (ms)	Frequency (hz)
00	63.9573	15.6354
01	93.2319	10.7259
02	66.9781	14.9303
05	64.6114	15.4771
07	51.0236	19.5988

[1][Online]. Available: https://octomap.github.io/

[2][Online]. Available: https://github.com/irapkaist/removert

[3][Online]. Available: https://github.com/LimHyungTae/ERASOR

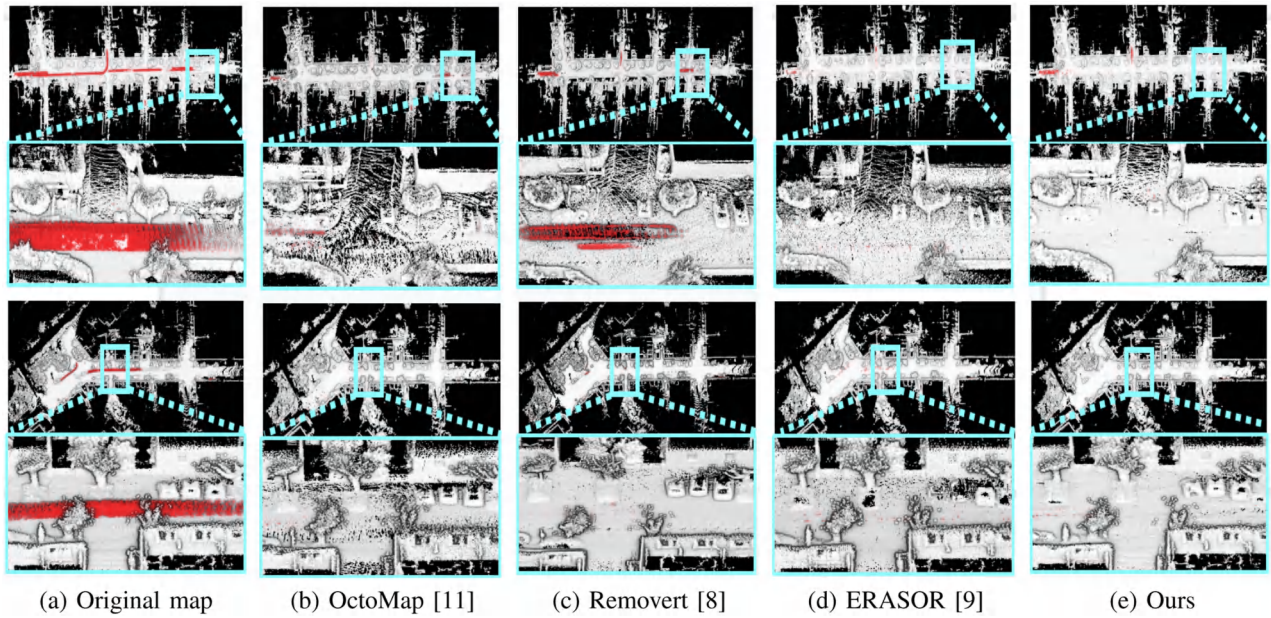

Fig. 8. Comparison of the final DOR effect conducted by our proposed method and existing methods on sequence 00 and 05 of SemanticKITTI. Red points are dynamic points that are not filtered out. And the fewer red points and more white points preserved, the better effect.

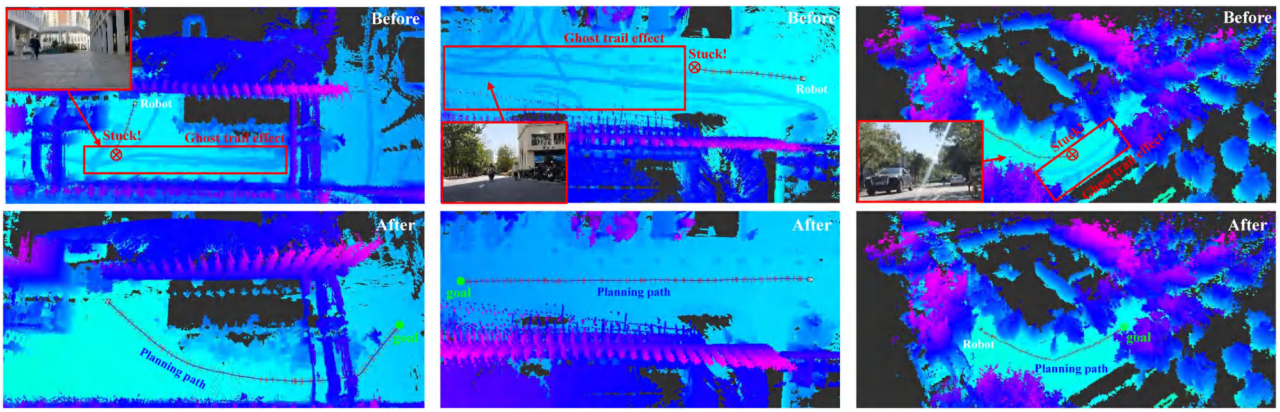

Fig. 9. Real-world experiments in different scenes with moving pedestrians and vehicles. The blue curves represent the generated global paths by PUTN.

of dynamic objects in the opposite lane has minimal impact on robot navigation. As can be observed, our method achieves comparable performance (ranked first on sequences 00, 02, 05, 07 in F1 score) to that of state-of-the-art approaches (ERASOR, Removert). In addition, our method is free from pre-built map and runs online directly.

3) Ablation Study: We also conduct ablation experiments to test the effectiveness of the back-end removal module, including RH-Map (front-end) and combined front-end and back-end approaches RH-Map (Ours). From Table I, it can be observed that the front-end-only approach achieves the highest F1 score on sequences 00 and 05. However, in scenes with a higher number of vehicles or faster-moving vehicles such as 01 and 02, there is a great amount of residual dynamic objects on the map due to the reasons mentioned in Section III-D. After introducing the back-end module, it can be seen that the RR increases, although it may result in a slight decrease in the PR. Nevertheless, RH-Map (Ours) consistently maintains higher F1 scores across all sequences, indicating a higher level of robustness.

4) Algorithm Speed: As our algorithm is designed for real-time online processing, we measure the average time and frequency of our proposed algorithm on SemanticKITTI with an Intel(R) Core(TM)i7-12700H CPU only. From Table II, we can observe that our algorithm achieves average processing frequencies above 10 Hz on each sequence, surpassing the typical data publishing rates of LiDAR sensors.

C. Real-World Experiments

To assess the effectiveness and practicality of our method, we conduct real-world experiments in campus environments with pedestrians and vehicles using Scout2 robot base equipped with a 32-beams RS-Helios LiDAR and imu. All the calculation is performed on an Intel(R) Core(TM)i7-12700H CPU which can reach 30 Hz for online DOR with a resolution of 0.1 m. Fast-LIO [2] is used to provide $SE(3)$ poses for RH-Map. Fig. 9 displays three scenes with dynamic objects: the top images represent the raw map without DOR, where the red

boxes indicate the presence of *ghost trail effect*, and the bottom images demonstrate the online DOR results achieved by our approach. In addition, we integrate RH-Map with PUTN [30] as shown in Fig. 9: the blue curves represent the generated global paths, and the green dots represent local goals. It can be observed that without DOR on the original map, the navigation algorithm results in sub-optimal global paths or planning failures as shown in the top images. After integrating RH-Map, the robot can navigate successfully and plan effectively in dynamic environments, which is advantageous for outdoor navigation and exploration tasks of mobile robots. More details can be available at. [4]

V. Conclusion

In this letter, a novel online map construction framework of DOR in dynamic environments, RH-Map, is proposed. Based on the region-wise hash map data structure, RH-Map consists of a real-time front-end module Scan Refresh for ground estimation and online DOR, and a lightweight back-end module for further removal of residual dynamic objects. Comparative validations on SemanticKITTI and real-world experiments are conducted to evaluate the proposed method. The experimental results demonstrate the efficiency of RH-Map in online static map construction of DOR.

References

[1] 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.*, 2020, pp. 5135–5142.

[2] 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.

[3] 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. Automat.*, 2020, pp. 10765–10771.

[4] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa : Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4530–4537.

[5] C. Qian, Z. Xiang, Z. Wu, and H. Sun, "RF-LIO: Removal-first tightly-coupled LiDAR inertial odometry in high dynamic environments," 2022, *arXiv:2206.09463*.

[6] J. Park, Y. Cho, and Y.-S. Shin, "Nonparametric background model-based LiDAR SLAM in highly dynamic urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24 190–24 205, Dec. 2022.

[7] X. Tian, Z. Zhu, J. Zhao, G. Tian, and C. Ye, "Dl-slot: Dynamic LiDAR SLAM and object tracking based on collaborative graph optimization," 2022, *arXiv:2212.02077*.

[8] 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.*, 2020, pp. 10758–10765.

[9] 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. Automat. Lett.*, vol. 6, no. 2, pp. 2272–2279, Apr. 2021.

[10] T. Fan, B. Shen, H. Chen, W. Zhang, and J. Pan, "DynamicFilter: An online dynamic objects removal framework for highly dynamic environments," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 7988–7994.

[11] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. robots*, vol. 34, pp. 189–206, 2013.

[12] J. Schauer and A. Nüchter, "The peopleremover–removing dynamic objects from 3-D point cloud data by traversing a voxel occupancy grid," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1679–1686, Jul. 2018.

[13] D. Yoon, T. Tang, and T. Barfoot, "Mapless online detection of dynamic objects in 3D LiDAR," in *Proc. 16th Conf. Comput. Robot Vis.*, 2019, pp. 113–120.

[14] H. Fu, H. Xue, and G. Xie, "MapCleaner: Efficiently removing moving objects from point cloud maps in autonomous driving scenarios," *Remote Sens.*, vol. 14, no. 18, 2022, Art. no. 4496.

[15] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds," in *Proc. Int. Symp. Vis. Comput.*, 2020, pp. 207–222.

[16] S. Li, X. Chen, Y. Liu, D. Dai, C. Stachniss, and J. Gall, "Multi-scale interaction for real-time LiDAR data segmentation on an embedded platform," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 738–745, Apr. 2022.

[17] X. Chen et al., "Automatic labeling to generate training data for online LiDAR-based moving object segmentation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6107–6114, Jul. 2022.

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

[19] J. Sun et al., "Efficient spatial-temporal information fusion for LiDAR-based 3D moving object segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 11 456–11 463.

[20] B. Han, J. Wei, J. Zhang, Y. Meng, Z. Dong, and H. Liu, "GardenMap: Static point cloud mapping for garden environment," *Comput. Electron. Agriculture*, vol. 204, 2023, Art. no. 107548.

[21] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3D LiDAR scans," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4508–4513.

[22] A. Asvadi, C. Premebida, P. Peixoto, and U. Nunes, "3D LiDAR-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes," *Robot. Auton. Syst.*, vol. 83, pp. 299–311, 2016.

[23] 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. Automat. Lett.*, vol. 6, no. 4, pp. 6458–6465, Oct. 2021.

[24] 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.*, 2022, pp. 13 276–13 283.

[25] Y. Wang, B. Zhang, P. Li, T. Cao, and B. Zheng, "Dynamic object separation and removal in 3D point cloud map building," in *Proc. 6th Int. Conf. Robot. Automat. Sci.*, 2022, pp. 247–252.

[26] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Static map generation from 3D LiDAR point clouds exploiting ground segmentation," *Robot. Auton. Syst.*, vol. 159, 2023, Art. no. 104287.

[27] Y. Wang, W. Yao, B. Zhang, J. Fu, J. Yang, and G. Sun, "DRR-LIO: A dynamic-region-removal-based LiDAR inertial odometry in dynamic environments," *IEEE Sensors J.*, vol. 23, no. 12, pp. 13175–13185, Jun. 2023.

[28] M. Teschner, B. Heidelberger, M. Müller, D. Pomerantes, and M. H. Gross, "Optimized spatial hashing for collision detection of deformable objects," in *Proc. Workshop Vis. Model. Visualization*, 2003, pp. 47–54.

[29] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.

[30] Z. Jian et al., "PUTN: A plane-fitting based uneven terrain navigation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7160–7166.

[4] Video: https://youtu.be/BEjzsPvN4ZA.