2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

EnhanceERASOR: Two-Stage Static 3D Point Cloud Mapping in Dynamic Scenes

Shuyang Yu[1], Yi Wu[1], Xiaoqing Guan[1], Song Jin[1], Haoxiang Liu[1], You Wang[1,*] and Guang Li[1]

***Abstract*— A clean map of the surrounding environment is essential for autonomous driving systems to ensure reliable localization and safe path planning. However, the existence of dynamic objects introduces ghost traces into the map, significantly degrading its quality. To address this issue, we propose EnhanceERASOR, a two-stage framework for static 3D point cloud mapping, consisting of a lightweight Online-ERASOR stage for real-time static mapping and an Offline-Refinement stage for global optimization. The Online-ERASOR stage utilizes the egocentric ratio of pseudo occupancy between consecutive scans to identify dynamic points, followed by verification and post-processing strategies to suppress false positives and false negatives. The Offline-Refinement stage introduces a submap-to-map consistency check to suppress semi-dynamic and slow-moving objects, and adopts a voxel-guided strategy for dense static mapping. Extensive experiments on diverse datasets with different scenarios and sensors demonstrate the superior performance, robustness, and generalization ability of our proposed method in static map construction.**

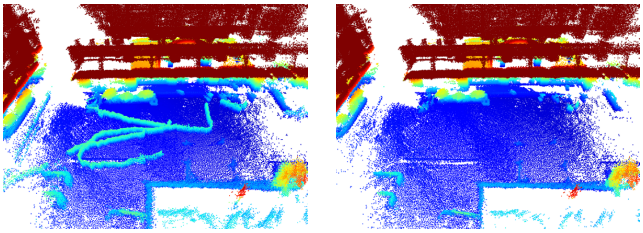
I. INTRODUCTION

With the rapid development of science and technology, unmanned vehicles and autonomous robots have become the current research hotspots and have been gradually applied to various fields, including transportation, logistics, industrial automation, etc. Constructing a map of the surrounding environment is crucial for autonomous mobile platforms to navigate and drive safely. A reliable environmental perception system is necessary for high-quality map construction. Due to the environmental robustness and ability to provide reliable depth estimation, 3D light detection and ranging (LiDAR) sensors have emerged as an essential tool to perceive the surrounding environment for autonomous driving.

Simultaneous Localization and Mapping (SLAM) is a core component of autonomous mobile platforms [2], [3], [4]. Most SLAM algorithms are based on the assumption that the surrounding environment is static and time-invariant. However, dynamic objects inevitably exist in most real-world scenarios, which may introduce ghost tracks into the map (see Fig. 1a) and adversely affecting downstream modules. In the localization task, dynamic objects may reduce accuracy and robustness by introducing ambiguous features or mislead the matching process [5], [6], [7]. In the path planning task, dynamic objects may be mistakenly regarded as static obstacles, leading to unnecessary obstacle avoidance and long path allocation. Thus, the elimination of dynamic points is critically significant.

[1]Authors are with the State Key Laboratory of Industrial Control Technology, Institute of Cyber Systems and Control, Zhejiang University, Hangzhou, 310027, China.

*Corresponding author: You Wang, king_wy@zju.edu.cn



(a) Original Map (b) Static Map

Fig. 1: The original accumulated map and the static map built by our proposed method.

In this paper, we propose EnhanceERASOR, a novel two-stage framework for static 3D point cloud mapping, combining an online scan-to-scan Online-ERASOR stage with a submap-to-map Offline-Refinement stage. Building upon prior work [8], [9], [10], our method utilizes region-wise pseudo occupancy descriptors to represent point clouds and identify potentially dynamic points. The complete pipeline is illustrated in Fig. 2, and the main contributions of our work are summarized as follows:

- We propose a two-stage framework for static 3D point cloud mapping, comprising a lightweight scan-to-scan Online-ERASOR stage for real-time mapping and an Offline-Refinement stage for global map optimization and dense mapping.
- In the Online-ERASOR stage, we extend the height difference and height encoding descriptors proposed in [8], [9] to enhance point cloud representation, and introduce false positive detection and post-processing strategies to mitigate performance degradation under limited temporal observations.
- In the Offline-Refinement stage, we introduce a submap-to-map consistency check module to suppress semi-dynamic and slow-moving objects, and employ a voxel-guided strategy for dense map construction.
- Experiments on diverse datasets [7], [11], [12] demonstrate the effectiveness, robustness, and generalization ability of our method in static 3D point cloud mapping.

II. RELATED WORK

A. Offline Static Mapping

OctoMap [13], Peopleremover [14] and DUFOMap [6] are typically methods that utilize ray tracing to eliminate dynamic points. These methods construct a global voxel occupancy grid and traverse it from the sensor to the measured points to find differences in volumetric occupancy. The

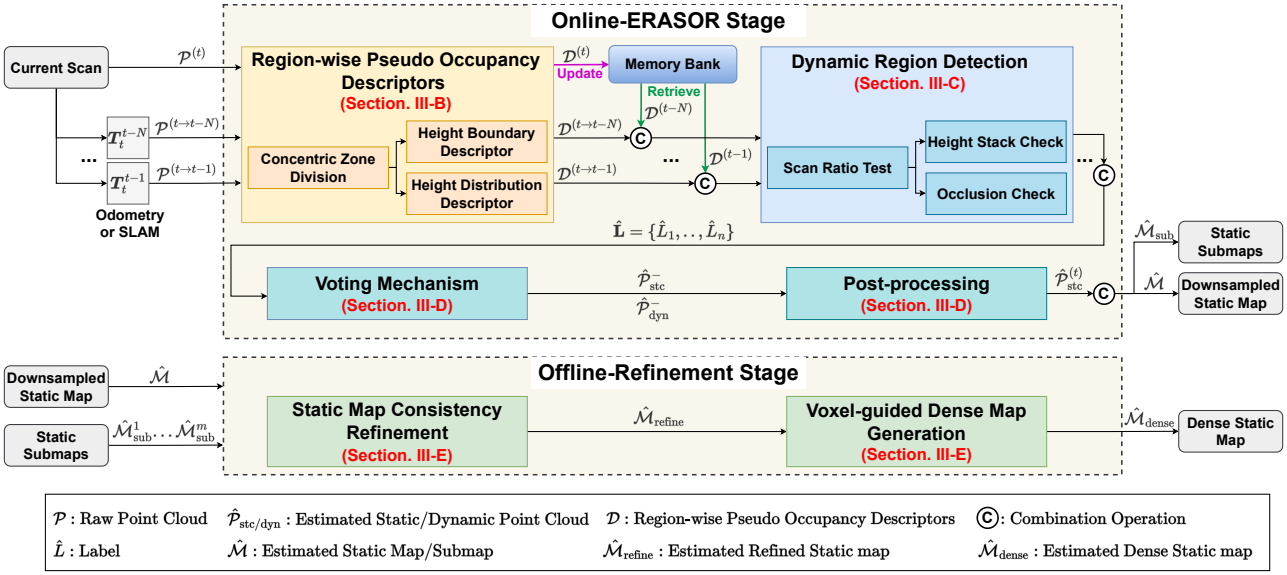


Fig. 2: Framework of EnhanceERASOR. The pipeline consists of two stages: (i) Online-ERASOR removes dynamic points and constructs a downsampled static map in real time by comparing the *Region-wise Pseudo Occupancy Descriptors* between the current and previous scans. (ii) Offline-Refinement performs consistency checks with static submaps to further improve the quality of the static map, and generates a dense static map by voxel-guided comparison.

voxels intersected with the sensor's line of sight are classified as dynamic and the points within them are subsequently rejected. However, ray tracing-based methods are usually computationally costly.

Visibility-based methods are proposed to improve computational efficiency. A point is categorized as dynamic if it occludes the line of sight of a previously observed point. Kim *et al.* [15] introduce Removert, which adopts a range image-based visibility check and extends a multiresolution version for better static map construction. However, both visibility-based and ray tracing-based methods struggle with incidence angle ambiguity and occlusion issues [7].

Egocentric ratio of pseudo occupancy-based methods have been widely applied to static map construction due to their promising performance and low computational load. Lim *et al.* [8] propose ERASOR, which detects dynamic points by comparing the region-wise relative height between a query scan and the accumulated map. Zhang *et al.* [9] extend this approach by introducing a height encoding descriptor to identify dynamic regions more effectively.

B. Online Static Mapping

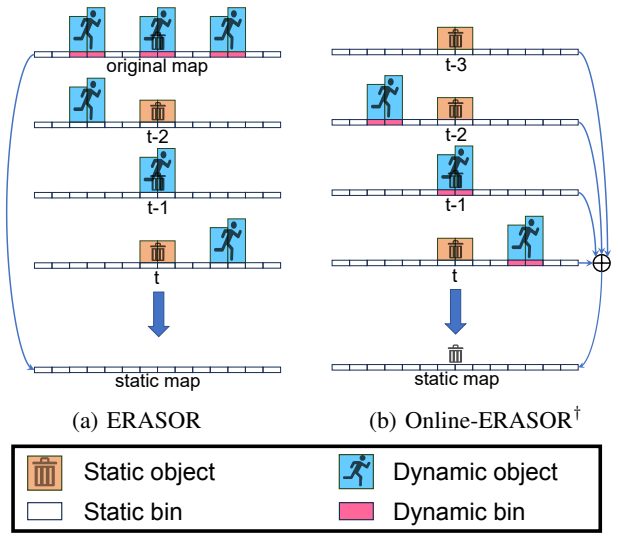
The conventional methods for online static mapping are mainly based on the principles of the above offline methods. Yoon *et al.* [16] use a motion-compensated freespace querying algorithm and classify between dynamic and static labels at the point level. Fan *et al.* [17] propose a framework consisting of scan-to-map front-end and map-to-map back-end modules. It integrates the visibility-based approach to remove dynamic points and the map-based approach to revert false positives. Wu *et al.* [18] propose M-Detector, which leverages depth images to examine the occlusion relation between current and previous points and then uses the occlusion clue to eliminate dynamic points.

Learning-based methods typically involve deep neural networks and supervised training with labeled datasets [7]. Chen *et al.* [19] propose LMNet that concatenates range images with residual images as input to the LiDAR semantic segmentation network [20], [21], [22], and trains it with binary labels to distinguish dynamic points from static points. Sun *et al.* [23] propose a dual-branch neural network structure, consisting of a range image branch to encode the appearance features, a residual image branch to encode the motion features, and a multi-scale motion-guided attention module to fuse them. Mersch *et al.* [24] convert the sequential scans into voxelized sparse 4D point clouds and apply sparse 4D conventions [25] to predict dynamic confidence scores. However, learning-based methods face several challenges, e.g., the requirement for high-quality labeled data, imbalanced data during training, and limited generalizability over different datasets, etc.

III. METHODOLOGY

Our primary objective is to construct a static voxel-downsampled point cloud map from a sequence of 3D LiDAR scans, which is memory-efficient and sufficient to support most downstream tasks. In addition, we provide an offline module to construct a dense static map from the voxel-downsampled map, enabling applications that require high-density maps.

In previous studies [8], [9], [10], the scan-to-map framework has been employed to remove dynamic points from the map, which may falsely reject static points neighboring the estimated dynamic points due to the limitation of bin resolution, as shown in Fig. 3. To address this issue, we



(a) ERASOR (b) Online-ERASOR†

Fig. 3: Comparison of the static mapping process between ERASOR [8] and our designed Online-ERASOR stage. ERASOR falsely removes static objects near the dynamic objects due to its limited bin resolution. In contrast, our scan-to-scan method preserves more static structure by the complement between different scans.

propose an enhanced framework that integrates an online scan-to-scan mapping stage and an offline refinement stage. The Online-ERASOR stage is the core component of our framework, effectively removing the majority of dynamic points while operating in real time. The schematic diagram of our proposed method is illustrated in Fig. 2. Further details of each component are discussed in the following sections.

A. Problem Definition

Let $\mathcal{P} = \left\{ \boldsymbol{p}_k = [x_k, y_k, z_k, 1]^T \mid k = 1, \cdots, N \right\}$ be a 3D point cloud with N points. We denote the current scan in the local sensor frame $\mathcal{S}$ at time step t by $\mathcal{P}_{\mathcal{S}}^{(t)}$ and the sequence of M previous scans by $\mathcal{P}_{\mathcal{S}}^{(l)}$ with $t - M \leq l \leq t - 1$. Let $\boldsymbol{T}_t^w \in \mathrm{SE}(3)$ be the transformation matrix between the t^{th} sensor frame and the world frame, then we can align the u^{th} scan to the viewpoint of v^{th} scan by:

$$\mathcal{P}_{\mathcal{S}}^{(u \to v)} = \left\{ \boldsymbol{T}_u^v \boldsymbol{p}_k \mid \boldsymbol{p}_k \in \mathcal{P}_{\mathcal{S}}^{(u)} \right\} \tag{1}$$

where $\boldsymbol{T}_u^v = (\boldsymbol{T}_v^w)^{-1} \boldsymbol{T}_u^w$.

Let $\hat{\mathcal{M}}$ be the estimated static map, then the static point cloud mapping problem is defined as follows:

$$\hat{\mathcal{M}} = \nu \left(\bigcup_{t \in [T]} \left\{ \boldsymbol{T}_t^w \boldsymbol{p}_k \mid \boldsymbol{p}_k \in \mathcal{P}_{\mathcal{S}}^{(t)} - \hat{\mathcal{P}}_{\mathcal{S},dyn}^{(t)} \right\} \right) \tag{2}$$

where $\nu(\cdot)$ denotes voxelization and $[T]$ is the total set of time steps.

B. Region-wise Pseudo Occupancy Descriptors

Since raw point clouds are inherently unordered and unstructured, it's challenging to process them directly. Moreover, the large volume of raw point clouds leads to high computational costs. Therefore, it is necessary to adopt suitable descriptors to extract distinctive features from the raw point clouds. Based on [8] [9], we utilize two kinds of *Region-wise Pseudo Occupancy Descriptors* (R-POD): *Height Boundary Descriptor* (HBD) and *Height Distribution Descriptor* (HDD).

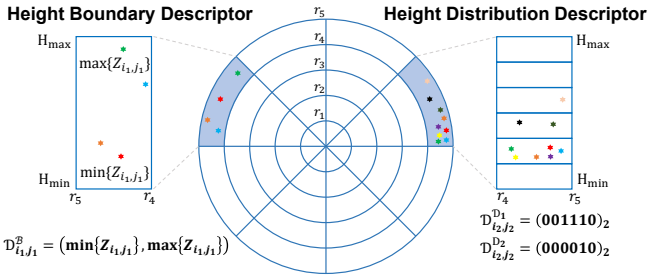


Fig. 4: Visual description of Region-wise Pseudo Occupancy Descriptors. The Height Boundary Descriptor records the minimum and maximum heights to encode vertical boundary information. The Height Distribution Descriptor counts the number of points in each layer to encode vertical distribution information, where $\mathcal{D}^{\mathcal{D}_1}$ represents occupancy feature with a low cardinality threshold and $\mathcal{D}^{\mathcal{D}_2}$ represents occlusion feature with a high cardinality threshold.

When the current scan comes, we first align it to the previous scan's coordinate frame using Eq.(1). Then the point cloud is separated into bins based on azimuthal direction and radial position, i.e., sectors and rings:

$$\begin{aligned} \mathcal{B}_{i,j} = \Bigg\{ \boldsymbol{p}_k \mid \boldsymbol{p}_k \in \mathcal{P}_{\mathcal{S}}, \frac{(i-1)L_{\max}}{N_r} \leq \rho_k < \frac{iL_{\max}}{N_r}, \\ \frac{(j-1)2\pi}{N_\theta} \leq \theta_k < \frac{j2\pi}{N_\theta}, H_{\min} < z_k < H_{\max} \Bigg\} \end{aligned} \tag{3}$$

where $\rho_k = \sqrt{x_k^2 + y_k^2}$ denotes the radial distance, $\theta_k = \arctan 2(y_k, x_k) + \pi$ denotes the azimuthal angle, N_r and N_θ are the numbers of rings and sectors, and $L_{\max}, H_{\min}, H_{\max}$ is physical boundary parameter.

Let $Z_{i,j} = \left\{ z_k \mid \boldsymbol{p}_k = [x_k, y_k, z_k, 1]^T, \boldsymbol{p}_k \in \mathcal{B}_{i,j} \right\}$ be the height set of the points in $\mathcal{B}_{i,j}$. The HBD of each bin is defined as follows:

$$\mathcal{D}_{i,j}^{\mathcal{B}} = \left(\min \left\{ Z_{i,j} \right\}, \max \left\{ Z_{i,j} \right\} \right) \tag{4}$$

To represent the distribution of height values, we further divide each bin into layers. Let $\mathcal{L}_{i,j}(\alpha)$ be the point set in the α^{th} layer of $\mathcal{B}_{i,j}$, which is defined as follows:

$$\begin{aligned} \mathcal{L}_{i,j}(\alpha) = \Bigg\{ \boldsymbol{p}_k \mid \boldsymbol{p}_k \in \mathcal{B}_{i,j}, \frac{(\alpha-1)\left(H_{\max} - H_{\min}\right)}{N_l} \\ \leq z_k < \frac{\alpha \left(H_{\max} - H_{\min}\right)}{N_l} \Bigg\} \end{aligned} \tag{5}$$

where N_l is the number of layers. Subsequently, we utilize binary representation to encode the state of each layer and apply bitwise operation to obtain the HDD of each bin:

$$\mathcal{D}_{i,j}^{\mathcal{D}} = \sum_{\alpha=1}^{N_l} O_{i,j}^{\tau}(\alpha) \cdot 2^{\alpha-1} \tag{6}$$

$$O_{i,j}^{\tau}(\alpha) = \begin{cases} 1, & \text{if } |\mathcal{L}_{i,j}(\alpha)| > \tau \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $|\cdot|$ denotes the cardinality of the set and τ is the constant threshold. We set two thresholds: one with a low value to represent the occupancy information and another with a high value to represent occlusion information.

The final representation of the R-POD is given by:

$$\mathcal{D} = \left\{ \left(\mathcal{D}_{i,j}^{\mathcal{B}}, \mathcal{D}_{i,j}^{\mathcal{D}_1}, \mathcal{D}_{i,j}^{\mathcal{D}_2} \right)_{i,j} \right\} \tag{8}$$

An intuitive illustration of the R-POD is provided in Fig. 4.

C. Dynamic Region Detection

Due to the online nature of the *Dynamic Region Detection* (DRD) module, the limited temporal context may lead to false positives during the *Scan Ratio Test* (SRT) step. To mitigate this issue, the *Height Stack Check* (HSC) and *Occlusion Check* (OC) are introduced to improve the removal performance by providing additional structural cues.

Firstly, the SRT is utilized to detect potentially dynamic regions. Let $\mathcal{D}_{\mathcal{S}}^{(t\to l)}$ and $\mathcal{D}_{\mathcal{S}}^{(l)}$ be the R-POD of the current scan and the previous scan, where $t - M \leq l \leq t - 1$. Diverging from ERASOR [8], we unify the infimum of height when computing height differences for each pair of bins. This helps prevent false positives caused by occlusion in low layers. The optimized SRT is formulated as follows:

$$\mathcal{R}_{i,j}^{(t\to l)} = \frac{\max\left\{Z_{i,j}^{l}\right\} - \min\left\{Z_{i,j}^{l}, Z_{i,j}^{(t\to l)}\right\}}{\max\left\{Z_{i,j}^{(t\to l)}\right\} - \min\left\{Z_{i,j}^{l}, Z_{i,j}^{(t\to l)}\right\}} \tag{9}$$

The bins with a scan ratio smaller than the threshold τ_r are considered potentially dynamic.

HSC is proposed to revert false positives arising from sensor noise in the SRT step. HSC calculates the number of non-overlapping and overlapping layers in potentially dynamic bins and examines their ratio, as described below:

$$\mathcal{H}_{i,j}^{(t\to l)} = \frac{\mathrm{HW}\left(\mathcal{D}_{i,j}^{\mathcal{D}_1(t\to l)} \wedge \left(\neg \mathcal{D}_{i,j}^{\mathcal{D}_1(l)}\right)\right)}{\mathrm{HW}\left(\mathcal{D}_{i,j}^{\mathcal{D}_1(t\to l)} \wedge \mathcal{D}_{i,j}^{\mathcal{D}_1(l)}\right)} \tag{10}$$

where $\mathrm{HW}(\cdot)$ is the Hamming Weight function that returns the number of 1's in the binary representation, $\wedge$ and $\neg$ denote AND and NOT bitwise operations respectively. The potentially dynamic bins with a ratio smaller than the threshold τ_h are reverted to static regions.

OC is designed to revert false positives arising from occlusion in the SRT step. Due to the varying viewpoints of the LiDAR, the structures observed at the current moment might be occluded at other moments, leading to differences in HDD. OC checks whether the highest point in each potentially dynamic bin of the current scan is visible in the previous scan, as shown in Fig. 5. The bins that may occlude potentially dynamic objects are located in the same sector and are closer to the LiDAR. The indices of the layers that may occlude the highest point can be calculated as follows:

$$L(r) = \left\lfloor \frac{r k_c L_{\max}/N_r - H_{\min}}{H_{\max} - H_{\min}} N_l \right\rfloor \tag{11}$$

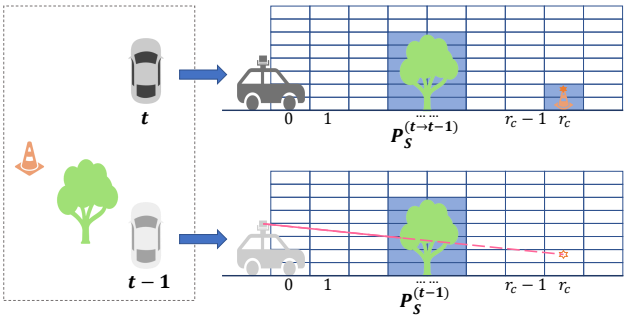


Fig. 5: Illustration of the Occlusion Check. In the driving scenario illustrated, the cone is missed by the LiDAR at time step $t-1$ due to the occlusion by a tree, which results in a discrepancy in the Height Boundary Descriptor and the cone being incorrectly detected as a potentially dynamic object by the SRT. Occlusion Check effectively reverts this false positive by examining the visibility of the potentially dynamic object in the previous scan.

where r is the ring index of the bin to be examined, $k_c = z_c/\rho_c$ is the slope of the highest point, and $\lfloor\cdot\rfloor$ denotes the floor function. The occlusion flag of the potentially dynamic bin is formulated as follows:

$$\mathcal{O}_{i,j}^{(t\to l)} = \bigvee_{r=0}^{r_c-1} \mathrm{LSB}\left(\mathcal{D}_{r,j}^{\mathcal{D}_2(l)} \gg L(r)\right) \tag{12}$$

where $r_c = \lfloor \rho_c N_r / L_{\max} \rfloor$ is the ring index of the highest point, $\gg$ denotes the right-shift bit operation, $\bigvee(\cdot)$ denotes the sequential logic operation, and $\mathrm{LSB}(\cdot)$ denotes the least significant bit of a binary number. The potentially dynamic bins with a positive occlusion flag are marked as unknown and are excluded from the subsequent voting mechanism.

D. Voting Mechanism and Post-processing

After a series of scan-to-scan comparisons and checks, M pointwise predictions for the current scan are obtained. We fuse these results and utilize post-processing steps to further improve the precision of dynamic points estimation.

Voting Mechanism (VM) is applied to calculate the dynamic confidence score for each point by fusing all scan-to-scan predictions. The confidence score is defined as the ratio of the number of dynamic votes to the total number of valid votes. Points with a score exceeding the probability threshold τ_{d1} are identified as dynamic, and the corresponding bins are classified as high-confidence dynamic regions.

Clustering Filter (CF) is employed to remove isolated dynamic points, which are more likely to be noise. To improve processing speed, the *Depth-First Search* (DFS) algorithm is utilized to extract the connected high-confidence dynamic bins, and then *Euclidean Clustering* (EC) is applied to filter out noise from each bin cluster.

Dynamic Region Growth (DRG) is designed to further eliminate the low-confidence points near the high-confidence dynamic points. Although DRG may mistakenly remove static points, the missing static structures can be recovered

Algorithm 1 Post-processing

Input: Raw point cloud $\mathcal{P}^{(t)}$, prior dynamic point cloud $\hat{\mathcal{P}}^{-}_{\text{dyn}}$, pointwise dynamic score $\boldsymbol{S}$, threshold τ_{d2}.
Output: Posterior dynamic point cloud $\hat{\mathcal{P}}^{(t)}_{\text{dyn}}$.

$\hat{\mathcal{P}}^{(t)}_{\text{dyn}} = \emptyset$
$\boldsymbol{B}_{\text{dyn}} = \text{ConcentricZoneDivision}(\mathcal{P}^{-}_{\text{dyn}})$ ▷ The set of non-empty dynamic bins.
$\boldsymbol{B}^{(t)} = \text{ConcentricZoneDivision}(\mathcal{P}^{(t)})$ ▷ The set of non-empty bins.
$\mathcal{C}_{\text{dyn}} = \text{DepthFirstSearch}(\boldsymbol{B}_{\text{dyn}})$ ▷ Extract connected dynamic bins.
for each cluster $\boldsymbol{C}_i$ in $\mathcal{C}_{\text{dyn}}$ **do**
  $\hat{\mathcal{P}}^{(t)}_{\text{dyn}} \leftarrow \hat{\mathcal{P}}^{(t)}_{\text{dyn}} \cup \text{EuclideanCluster}(\boldsymbol{C}_i)$
  for each bin $\mathcal{B}_j$ in $\text{RegionGrow}(\boldsymbol{C}_i)$ **do**
    for each point $\boldsymbol{p}_k$ in $\mathcal{B}_j$ **do**
      if $\boldsymbol{S}[k] > \tau_{d2}$ **then**
        $\hat{\mathcal{P}}^{(t)}_{\text{dyn}} \leftarrow \hat{\mathcal{P}}^{(t)}_{\text{dyn}} \cup \{\boldsymbol{p}_k\}$
      end if
    end for
  end for
end for

by other scans owing to our scan-to-scan framework. DRG first extracts the connected high-confidence dynamic bins using the DFS algorithm. Then the bin cluster is expanded to include surrounding bins, and a smaller probability threshold τ_{d2} is used for classification in the expansion space. A pseudo-code of the post-processing is shown in Algorithm. 1.

Our approach is based on the assumption that most dynamic objects are inevitably in contact with the ground [8]. Following Patchwork [26], we adopt *Region-wise Ground Plane Fitting* (R-GPF) to discriminate ground points in dynamic regions. The identified ground points are restored to static points, while others above the ground are rejected. Finally, the cleaned scan is accumulated into the static map.

E. Offline Static Map Refinement

Static Map Consistency Refinement (SMCR) leverages multiple submaps to perform consistency analysis from a global perspective, aiming to suppress semi-dynamic points and refine the static map. We still employ region-wise pseudo occupancy features for inconsistency detection. Compared with the online stage, the offline refinement adopts stricter criteria to minimize the loss of static structures, as accumulated submaps provide more stable observations than single frames, while most dynamic points have already been removed during the online stage. Regions are classified as inconsistent if the scan ratio between the global map and the local submaps exceeds a predefined threshold τ_r and a sufficient number of low-height ground points are observed within the submap. Then the non-ground points within these inconsistent regions are removed from the global map.

Voxel-guided Dense Map Generation (VDMG) is designed to produce a dense static map from the refined downsampled static map. We first construct a voxel occupancy grid from the refined map and apply morphological closing to fill voxel gaps. Then, by performing frame-by-frame voxel-wise comparison, points falling within occupied voxels are identified as static and accumulated to form a dense static map. To improve segmentation accuracy near the ground, which typically contains a mixture of dynamic and static points, the maximum point height within each ground voxel is recorded as a threshold for identifying static points, thereby achieving sub-voxel-level precision.

TABLE I: Ablation Study of the Online-ERASOR Stage on the SemanticKITTI Sequence 05.

Method	PR	RR	F_1
Ours	98.67	97.45	**0.981**
w/o HSC	97.34	97.56	0.974
w/o OC	82.63	**98.59**	0.899
w/o CF	96.01	98.45	0.972
w/o DRG	**99.93**	69.54	0.820

TABLE II: Ablation Study of the Offline-Refinement Stage on the Semi-indoor Dataset.

Method	SA	DA	AA
Ours	96.78	**93.17**	**94.96**
w/o SMCR	97.32	56.98	74.47
w/o sub-voxel strategy	**97.86**	85.28	91.35

IV. EXPERIMENTAL RESULTS

A. Experimental Setups

The experimental datasets include subsequences from SemanticKITTI (VLP-64) [11], [12], semi-indoor dataset (VLP-16) [7], and our own outsquare dataset (MID360), covering a variety of sensor types and scenarios. We evaluate the performance of our online stage using the benchmark proposed by Lim *et al.* [8], which utilizes voxel-wise metrics *Preservation Rate* (PR), *Rejection Rate* (RR), and F_1 score to quantitatively assess the quality of the estimated static map. Additionally, the benchmark proposed by Zhang *et al.* [7] is used to evaluate the quality of the dense map generated in the offline stage, which employs point-wise metrics *Static Accuracy* (SA), *Dynamic Accuracy* (DA), and *Associate Accuracy* (AA).

For all experimental sequences, we maintain an R-POD Memory Bank using the previous ten scans sampled at intervals of four frames for comparison, and set the voxel size to 0.2 m for voxel-wise operations. For the R-POD, we set $N_r = 120$, $N_\theta = 160$, and $N_l = 32$. For the SRT, HSC, and SMCR, we set $\tau_r = 0.3$ and $\tau_h = 0.5$. For the voting mechanism, we set $\tau_{d1} = 0.5$ and $\tau_{d2} = 0.1$ by default.

B. Ablation Study

As the core part of our method, the Online-ERASOR stage directly determines the quality of the constructed static map. To this end, an ablation study is designed to evaluate the effectiveness of each component within the Online-ERASOR stage. As illustrated in Table I, the complete

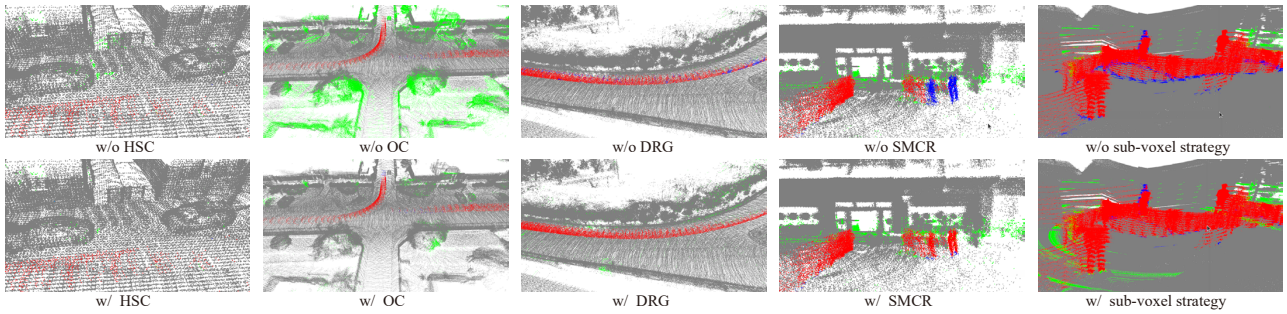


Fig. 6: Visual evaluation of the effect of module removal in ablation studies. The first three columns correspond to the Online-ERASOR stage, while the last two columns correspond to the Offline-Refinement stage. The red, green, gray, and blue points indicate true positives, false positives, true negatives, and false negatives, respectively.

TABLE III: Comparison with State-of-the-Art Methods Based on Voxel-Wise Metrics. † Means Online Methods.

	KITTI small town (00)			KITTI Residential (05)			Semi-indoor			Square-MID360		
Methods	PR ↑	RR ↑	F_1 ↑	PR ↑	RR ↑	F_1 ↑	PR ↑	RR ↑	F_1 ↑	PR ↑	RR ↑	F_1 ↑
ERASOR[8]	93.98	97.08	0.955	88.73	98.26	0.921	90.2	91.95	0.911	88.94	90.71	0.898
ERASOR++[9]	96.82	96.10	0.965	96.53	97.67	0.971	-	-	-	-	-	-
BeautyMap[5]	95.31	97.94	0.966	92.33	97.41	0.948	90.31	78.44	0.840	84.34	84.18	0.843
DUFOMap[6]	98.70	98.58	**0.986**	98.13	88.98	0.933	99.92	70.81	0.829	99.74	61.27	0.759
Octomap w GF†[7]	88.29	98.88	0.933	88.06	96.14	0.919	73.5	90.59	0.812	85.84	71.79	0.782
DynamicFilter†[17]	90.07	91.09	0.906	90.17	84.65	0.873	-	-	-	-	-	-
Dynablox†[28]	98.68	86.85	0.924	98.75	78.07	0.872	85.63	71.02	0.776	95.06	80.01	0.869
Online-ERASOR†(ours)	97.79	97.45	0.976	98.67	97.45	**0.981**	92.83	94.68	0.937	92.49	97.51	0.949
EnhanceERASOR (ours)	97.34	98.12	0.977	98.07	98.09	**0.981**	92.58	99.04	**0.957**	92.34	99.58	**0.958**

Online-ERASOR stage achieves state-of-the-art performance over other ablated variants, showing the highest F_1 score. Fig. 6 presents a visual comparison to further demonstrate the effects of removing different modules. HSC improves the PR marginally by reverting false positives caused by sensor noise, which is particularly beneficial for low-height objects where noise points may lead to an abnormal scan ratio. OC discards unreliable votes from occluded frames, thereby effectively reducing false positives, which is critical in dense scenarios with frequent occlusions and limited visibility. DRG eliminates additional points near high-confidence dynamic points, significantly improving RR with only a slight decrease in PR and achieving a better trade-off between PR and RR. This aggressive strategy is particularly effective for large dynamic objects and slow-moving objects.

The second experiment is conducted to evaluate the impact of the SMCR module and the sub-voxel strategy in the Offline-Refinement stage. As shown in Fig. 6, SMCR effectively eliminates temporarily static semi-dynamic objects, and the sub-voxel strategy enhances the precision of static point identification near the ground. These modules significantly improve DA while causing only a minor decrease in SA, as reported in Table II.

C. Comparison of Static Mapping Performance

The quantitative comparison with other state-of-the-art methods based on voxel-based metrics is summarized in Table III, with the qualitative results illustrated in Fig. 7. Our method achieves state-of-the-art performance among baseline methods, obtaining the highest F_1 scores on most sequences. Compared to offline ERASOR [8] and ERASOR++ [9], our Online-ERASOR stage preserves more static structures owing to the complementary nature between consecutive scans, achieving an improvement in PR while maintaining comparable RR. Furthermore, the Offline-Refinement stage enables the complete system to remove semi-dynamic and slow-moving objects better, as demonstrated by the results on the semi-indoor and Square-MID360 datasets. The static points incorrectly removed and the dynamic points falsely retained by our method are primarily concentrated near the map boundaries, where LiDAR observations are normally insufficient to identify as static regions accurately. By leveraging statistical features along the vertical dimension for dynamic point identification, our method shows improved robustness in sparse scenarios compared to voxel-occupancy methods such as OctoMap[7], [13] and DUFOMap [6].

Based on the high-quality voxel-downsampled static maps generated by our method, the offline voxel-guided dense mapping module achieves excellent performance, attaining the highest AA on most sequences, as shown in Table IV. A visualization of dense mapping results is presented in Fig. 8. Most dynamic points are effectively removed, while residual points are mainly located near the ground and static obstacles due to the limited voxel resolution.

D. Robustness under Sparse LiDAR Inputs

To quantitatively evaluate the robustness of our proposed method under sparse input conditions, we simulate sparse

TABLE IV: Comparison with State-of-the-Art Methods Based on Point-Wise Metrics. † Means Online Methods.

	KITTI small town (00)			KITTI highway (01)			KITTI residential (05)			Semi-indoor		
Methods	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑
ERASOR[8]	66.70	98.54	81.07	98.12	90.94	94.46	69.40	99.06	82.92	94.90	66.26	79.30
BeautyMap[5]	96.76	98.38	97.56	99.17	92.99	95.98	96.34	98.29	97.31	93.69	90.67	92.17
DUFOMap[6]	97.96	98.72	**98.34**	98.09	94.20	96.12	-	-	-	99.64	83.00	90.94
Octomap w GF†[7]	93.06	98.67	80.64	97.27	88.18	95.78	93.54	92.48	93.01	96.79	73.50	83.55
Dynablox†[28]	96.76	90.68	93.62	96.33	68.01	79.73	97.80	88.68	93.02	98.81	36.49	53.30
EnhanceERASOR (ours)	98.79	96.99	97.89	99.10	94.50	**96.78**	99.44	96.32	**97.89**	96.78	93.17	**94.96**

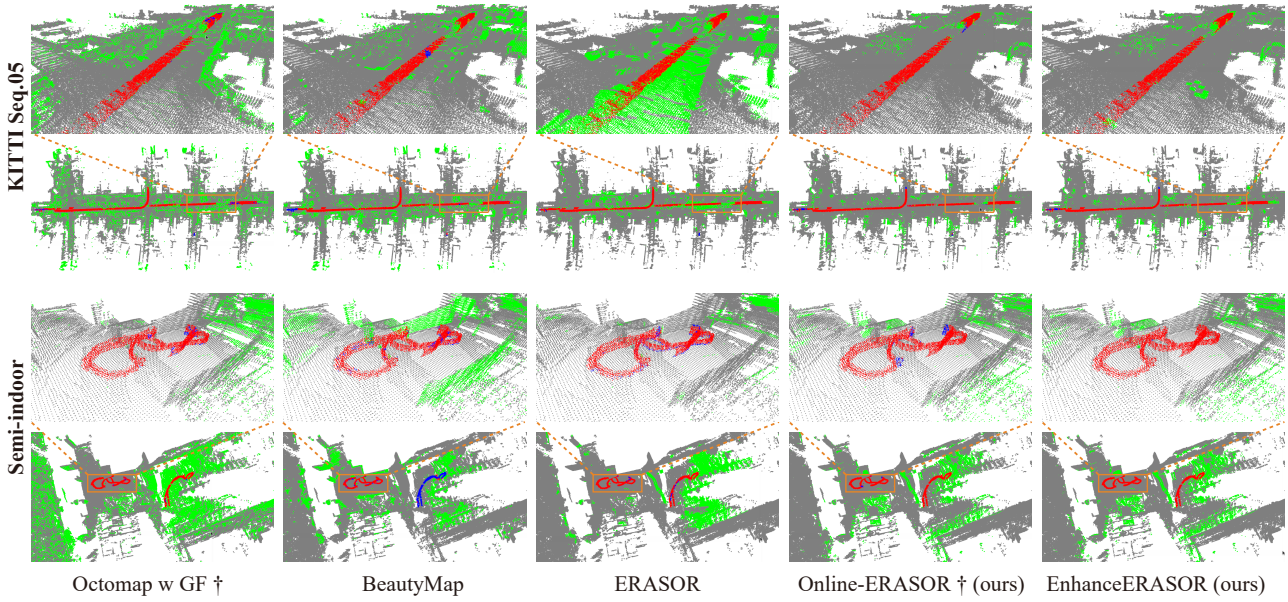


Fig. 7: Qualitative comparison of different static mapping methods. The red, green, gray, and blue points indicate true positives, false positives, true negatives, and false negatives, respectively.

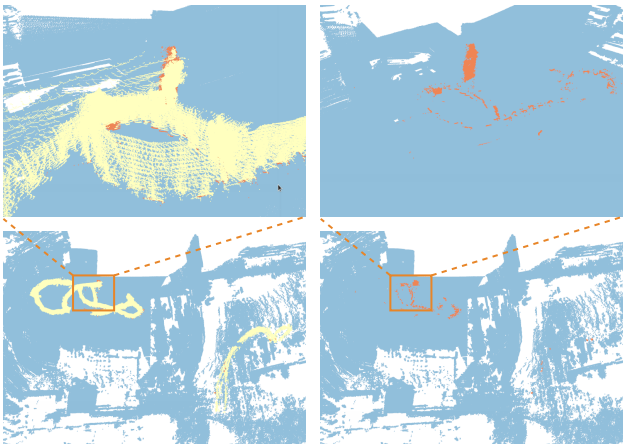


Fig. 8: Visualization of dense mapping results by our method, with blue, yellow, and orange indicating true negatives, true positives, and false negatives, respectively.

point clouds by downsampling the LiDAR beams to 32 and 16, which are commonly deployed on cost-effective platforms. As illustrated in Table V, our method demonstrates strong robustness, maintaining comparable performance despite a significant reduction in input density. In the highway scenario of Seq.01, the PR decreases by approximately 10%, primarily due to the high vehicle speed with sparse LiDAR beams, which results in severely insufficient observations. Consequently, our method tends to make conservative predictions, mistakenly removing more static points. In the residential scene of Seq.05 with a slow vehicle speed, our method maintains consistent scores with negligible degradation.

TABLE V: Performance under Different Simulated LiDAR Beam Configurations on the SemanticKITTI Dataset.

	KITTI highway (01)			KITTI residential (05)		
Beam	PR	RR	F_1	PR	RR	F_1
64	97.01	95.55	0.963	98.67	97.45	0.981
32	92.80	95.51	0.941	98.33	97.49	0.979
16	87.06	91.22	0.891	97.21	95.19	0.962

E. Algorithm Speed

We analyze the computational complexity of each module and investigate the time consumption on an onboard PC equipped with an Intel i7-8559U CPU (2.7 GHz, 4 cores). As shown in Table VI, the worst-case time complexity of our Online-ERASOR stage is $\mathcal{O}(N \log N)$, and in scenarios where the majority of objects are static, the time complexity

TABLE VI: Computational Cost Analysis of the Online-ERASOR Stage using the SemanticKITTI Dataset.

Module	Time Complexity	Parallelisation	Runtime/frame[s]
R-POD	$\mathcal{O}(N)$		
DRD	$\mathcal{O}(N_\theta N_r^2)$	Frame-level	0.041
VM	$\mathcal{O}(N)$		
CF	$\mathcal{O}(N_d \log N_d)$		
DRG	$\mathcal{O}(N_d)$	Bin-level	0.014
R-GPF	$\mathcal{O}(N_d)$		
Accumulation	$\mathcal{O}(N)$	-	0.006

N_d denotes the number of dynamic points.

reduces to $\mathcal{O}(N)$. By employing parallelization techniques, our Online-ERASOR stage operates at 61 ms per frame (16.4 Hz) on average, which is faster than the common sensor frame rate of 10 Hz, indicating that our Online-ERASOR stage can run in real time to generate static maps. The voxel-guided dense mapping, executed in the offline stage without strict runtime requirements, operates at 96 ms per frame (10.4 Hz) and can be accelerated via frame-level parallelization.

V. CONCLUSIONS

In this paper, a novel two-stage framework for static 3D point cloud mapping, called EnhanceERASOR, has been presented. The lightweight Online-ERASOR stage enables real-time dynamic point removal, while the Offline-Refinement stage enhances the quality of the static map, and generates a dense static map via the voxel-guided strategy. Experiments on diverse datasets demonstrate the superior performance and robustness of our method. In future works, we plan to integrate our method into an online SLAM framework to improve the localization accuracy and mapping quality, thereby better supporting downstream modules.

REFERENCES

[1] B. Ravi Kiran, L. Roldao, B. Irastorza, R. Verastegui, S. Suss, S. Yogamani, V. Talpaert, A. Lepoutre, and G. Trehard, "Real-time dynamic object detection for autonomous driving using prior 3d-maps," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[2] 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, 2021.

[3] 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, 2022.

[4] 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.

[5] M. Jia, Q. Zhang, B. Yang, J. Wu, M. Liu, and P. Jensfelt, "Beautymap: Binary-encoded adaptable ground matrix for dynamic points removal in global maps," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6256–6263, 2024.

[6] D. Duberg, Q. Zhang, M. Jia, and P. Jensfelt, "Dufomap: Efficient dynamic awareness mapping," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5038–5045, 2024.

[7] Q. Zhang, D. Duberg, R. Geng, M. Jia, L. Wang, and P. Jensfelt, "A dynamic points removal benchmark in point cloud maps," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 608–614.

[8] 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 Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272–2279, 2021.

[9] J. Zhang and Y. Zhang, "Erasor++: Height coding plus egocentric ratio based dynamic object removal for static point cloud mapping," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 4067–4073.

[10] H. Lim, L. Nunes, B. Mersch, X. Chen, J. Behley, H. Myung, and C. Stachniss, "Erasor2: Instance-aware robust 3d mapping of the static world in dynamic scenes," 2023.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[12] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9296–9306.

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

[14] J. Schauer and A. Nüchter, "The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1679–1686, 2018.

[15] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 758–10 765.

[16] D. Yoon, T. Tang, and T. Barfoot, "Mapless online detection of dynamic objects in 3d lidar," in *2019 16th Conference on Computer and Robot Vision (CRV)*, 2019, pp. 113–120.

[17] T. Fan, B. Shen, H. Chen, W. Zhang, and J. Pan, "Dynamicfilter: an online dynamic objects removal framework for highly dynamic environments," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7988–7994.

[18] H. Wu, Y. Li, W. Xu, F. Kong, and F. Zhang, "Moving event detection from lidar point streams," *nature communications*, vol. 15, no. 1, p. 345, 2024.

[19] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, "Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6529–6536, 2021.

[20] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*. Springer, 2020, pp. 207–222.

[21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet ++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4213–4220.

[22] 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 Robotics and Automation Letters*, vol. 7, no. 2, pp. 738–745, 2022.

[23] J. Sun, Y. Dai, X. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 456–11 463.

[24] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss, "Receding moving object segmentation in 3d lidar data using sparse 4d convolutions," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7503–7510, 2022.

[25] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3070–3079.

[26] 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 Robotics and Automation Letters*, vol. 6, no. 4, pp. 6458–6465, 2021.

[27] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in *Robotics: science and systems*, vol. 2018, 2018, p. 59.

[28] L. Schmid, O. Andersson, A. Sulser, P. Pfreundschuh, and R. Siegwart, "Dynablox: Real-time detection of diverse dynamic objects in complex environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6259–6266, 2023.