DORF: A Dynamic Object Removal Framework for Robust Static LiDAR Mapping in Urban Environments

Zhiming Chen*, Kun Zhang*, Hua Chen, Michael Yu Wang, Wei Zhang†, Hongyu Yu†

***Abstract*— 3D point cloud maps are widely used in robotic tasks like localization and planning. However, dynamic objects, such as cars and pedestrians, can introduce ghost artifacts during the map generation process, leading to reduced map quality and hindering normal robot navigation. Online dynamic object removal methods are restricted to utilize only local scope information and have limited performance. To address this challenge, we propose DORF (Dynamic Object Removal Framework), a novel coarse-to-fine offline framework that exploits global 4D spatial-temporal LiDAR information to achieve clean static point cloud map generation, which reaches the state-of-the-art performance among existing offline methods. DORF first conservatively preserves the definite static points leveraging the Receding Horizon Sampling (RHS) mechanism proposed by us. Then DORF gradually recovers more ambiguous static points, guided by the inherent characteristic of dynamic objects in urban environments which necessitates their interaction with the ground. We validate the effectiveness and robustness of DORF across various types of highly dynamic datasets.**

I. Introduction

Autonomous navigation in dynamic urban environments is a very challenging task for robotic systems like self-driving cars and mobile robots. Recent advances in such scenarios owe much to the application of High Definition Map (HD Map). The HD Map provides detailed information to represent the environment, which is the foundation for downstream tasks like localization, place recognition, planning, etc. A 3D point cloud map is a basic form of HD Map. It can easily be transformed into more elaborated forms of HD Map by multi-sensor fusion and manual annotation. With 3D LiDAR sensors, the robotic systems can get the 3D point cloud map about the urban environment through simultaneous localization and mapping (SLAM) algorithms like [1] [2] [3]. However, sometimes moving objects will be built into the map during this process. As shown in Figure 1, a large number of walking pedestrians on the playground are built into the map and become what we call ghost artifacts. Path planning algorithms may fail to generate a safe, efficient, and optimal path in such a map since ghost artifacts may create obstacles that are not present in the environment. Additionally, determining the robot pose based on sensor data and a known map can be problematic as the dynamic objects may lead to errors in point cloud registration or LiDAR descriptor comparison. These issues can cause the robot to become disoriented or lose its localization.

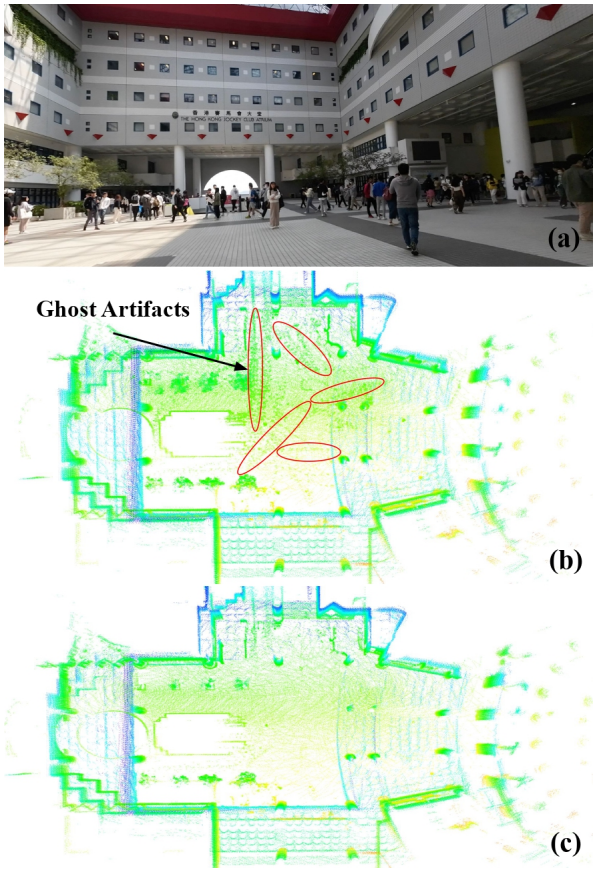

Fig. 1: (a) is the crowded scenario in HKUST campus where our robot and a plethora of pedestrians are walking on a playground; (b) is the prior map built from the crowded scenario by LIO-SAM [1]; (c) is the clean static 3D point cloud map generated by our method.

There are two classes of dynamic object removal algorithms: online, which happens while the data is collected on board a robot, and offline, which happens afterward over a batch of data. Unlike online methods that can only utilize local scope information [4], offline removal can benefit from global scope and sequential future and past observation

This work was supported in part by the grants from Innovation and Technology Commission (ITS/036/21FP) of HKSAR and the Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone (HZQB-KCZYB-2020083).

* denotes equal contributions, † denotes corresponding author.

Zhiming Chen, Kun Zhang, Hongyu Yu are with Robotics Institute, The Hong Kong University of Science and Technology. Hongyu Yu is also with HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen. Emails: zhiming.chen@connect.ust.hk, kun.zhang@connect.ust.hk, hongyuyu@ust.hk

Michael Yu Wang is with the School of Engineering, Great Bay University, Songshan Lake, Dongguan, Guangdong, China. Email: mywang@gbu.edu.cn.

Hua Chen and Wei Zhang are with the School of System Design and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen, China. Emails: chenh6@sustech.edu.cn, zhangw3@sustech.edu.cn

data, which means more spatial-temporal information can be exploited for dynamic object removal operation. Leveraging more information, offline methods usually achieve better performance than online methods. Thus, we mainly discuss offline removal methods in this paper. Generating an accurate static point cloud map by removing dynamic objects is a challenging task due to the interdependent operations. This actually presents us a *de facto* chicken-or-egg problem: Given a perfectly static map, identifying dynamic points is straightforward; if we accurately distinguish dynamic points, then removing them from incoming sensor data readily builds a clean static map. Unfortunately, obtaining precise information on both static and dynamic objects in real-world urban environments is often unfeasible. To overcome this challenge, we propose a novel coarse-to-fine framework that introduces several mechanisms to address this challenge, offering a novel solution to this long-standing problem. To summarize, the contributions of our paper are as follows:

- We propose a Receding Horizon Sampling (RHS) mechanism that effectively expands the field of view (FOV) for visibility-based removal.
- We present a method that utilizes a 2.5D Polar Elevation Map to improve ground segmentation, thereby more effectively preserving the static environment information from the prior point cloud map.
- We introduce a Bird's-Eye View (BEV) occupancy checking method to perform efficient ray-casting computation.
- We validate the effectiveness, robustness, and generalization ability of our approach on various datasets, including public autonomous driving datasets, extremely crowded simulation scenes, and unstructured real-world environments.

Furthermore, we are committed to open-sourcing our code and related material to facilitate the research community's development. For more information, including experiment videos, please visit our project page at https://sites.google.com/view/dorf-mapping.

II. Related Work

In this section, we review the related work on dynamic object removal, which can be roughly classified into three categories.

A. Segmentation-Based Methods

Traditional segmentation-based methods such as those proposed in [5] and [6] rely on plane fitting. These methods use a fitted ground plane model to differentiate dynamic object candidates from the static ground plane. However, when there are large numbers of dynamic objects on the ground, these methods may encounter a degeneration problem. Recently, MapCleaner [7] based on terrain segmentation has been proposed. However, the researchers did not propose methods to deal with cases where dynamic points are misclassified as ground points. Besides, with the rapid rise of deep learning, typical dynamic objects like cars and pedestrians can be detected by deep neural network models through semantic segmentation [8], instance segmentation [9] [10], or panoramic segmentation [11]. However, the semantic labels provided by deep learning cannot directly tell if the object is moving or not. Following the definition of dynamic object in the SemanticKITTI dataset [12], which is adopted as a benchmark in the field of moving object removal, only the object that once presented motion actions during the mapping process will be treated as dynamic and further removed. Recently, methods such as [13] have leveraged sequential range images as input to capture motion characteristics of dynamic points based on inconsistencies between these sequential frames of data. Nevertheless, these deep learning methods rely heavily on high-quality manually labeled training datasets and require a heavy computational load for training and deploying models.

B. Map-Based Methods

A large percentage of map-based methods utilize ray-casting, as demonstrated in [14] and [15]. These methods count the hits and misses of laser scans in the 3D voxel space and calculate the occupancy probability of the space or traverse the voxel occupancy grid, as described in [16]. However, ray-casting or ray-tracing can be computationally expensive, which may be unaffordable in 3D space. Other methods related to map space division are also categorized in this group. Unlike occupancy grid or voxel grid-based division in Cartesian coordinate frames, some methods divide the map space into a polar coordinate frame. These methods use specific descriptors to discriminate dynamic points on moving objects. However, the descriptor they use such as ERASOR [17] is too light and is not robust in highly dynamic environments. Recently an improved version ERASOR2 [18] of ERASOR [17] is proposed with additional consideration on instance segmentation. However, its core idea inherited from ERASOR [17] about pseudo occupancy relies on a high difference of free space percentage in LiDAR sweep space for most collected scans used in mapping. It may suffer from the same degeneration with ERASOR [17] in continuously crowded dynamic environments.

C. Visibility-Based Methods

Visibility-based methods [4] [19] involve calculating the visibility difference between a query scan point and a map point within a narrow field of view (FOV). The premise is that if the scan point is occluded by the map point, then the closer map point should be dynamic. This visibility mechanism significantly reduces computation load compared to some ray-casting methods. However, the visibility checking assumption may fail in cases with a large incident angle or when occluded by a huge obstacle. These issues are referred to as *visibility issues*. Kim et al. propose Removert [20], which first aggressively removes dynamic points and then iteratively reverts the misclassified static points. However, this reverting mechanism is deeply interwoven with the visibility-checking method. Thus, a single method cannot solve the problems that come with the technique itself.

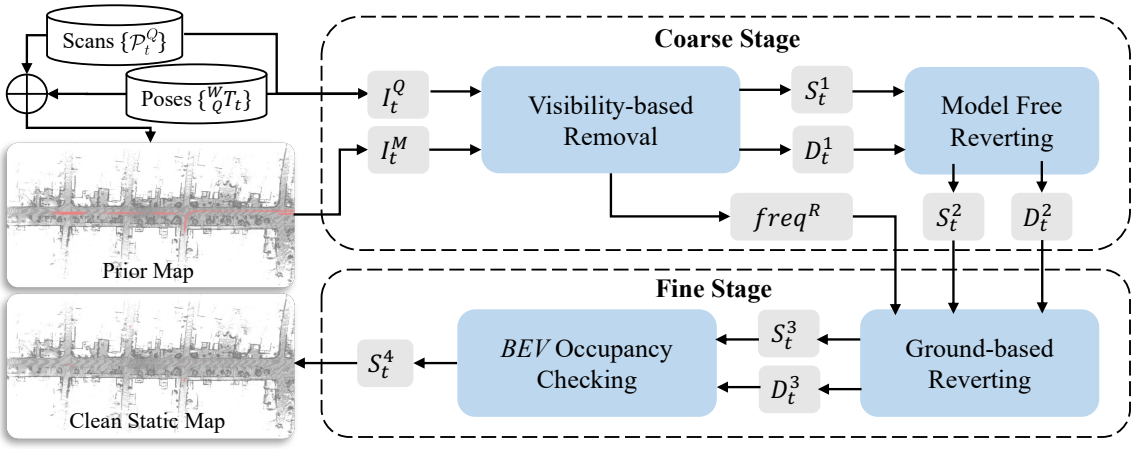

Fig. 2: The pipeline of our method. The capital letters S_t^k and D_t^k represent a point cloud mainly composed of static and dynamic points at the k_{th} phase. In the coarse stage, dynamic points will be coarsely discriminated from static points. The first phase *visibility-based Removal* takes into the range images of the query scan I_t^Q and the submap I_t^M to perform aggressive dynamic points removal leveraging our proposed RHS mechanism. Then the second phase *model free reverting* will revert falsely detected dynamic points based on the result of the first phase through the PCA comparison. In the fine stage, the third phase ground-based reverting is used to revert the ground points to be static based on the statistical status frequency of each point $freq^R$ and the output of the previous stage. Finally, the BEV occupancy checking will conduct an accelerated ray-tracing process from Bird's-Eye View to check the discretized space and revert the remaining possible misclassified points.

III. Methodology

A. Problem Formulation

During the 3D LiDAR mapping process, LiDAR poses can usually be estimated by SLAM algorithms like [1] [2] [3]. Given these poses, it is straightforward to construct the map by merging the scans together. Let denote a point cloud scan received at timestep t as $\mathcal{P}_t^Q \in \mathcal{B}_{scans}$. Let ${}_Q^W\mathbf{R}_t$ be the rotation matrix in SO(3) manifold, ${}_Q^W\mathbf{q}_t$ be the translation direction vector in $\mathbb{R}^3$, and ${}_Q^W\mathbf{T}_t = [{}_Q^W\mathbf{R}_t|{}_Q^W\mathbf{q}_t]$ be the $SE(3)$ pose in $\mathcal{B}_{poses}$ for LiDAR sensor at timestep t in the world frame W. Then the prior 3D point cloud map $\mathcal{M}$ can be expressed as:

$$\mathcal{M} = \bigcup_{t\in\mathcal{T}} {}_Q^W\mathbf{T}_t \boxplus \mathcal{P}_t^Q$$

where, $\mathcal{T} = \{1, 2, \cdots, N\}$ is the set of timestamps, N is the number of scans, and $\boxplus$ stands for the homogeneous transformation operation from the LiDAR frame Q to the world frame W. Our goal in this problem is to obtain a clean static point cloud map, which can be represented as:

$$\hat{\mathcal{M}} = \mathcal{M} - \bigcap_{t\in\mathcal{T}} \hat{\mathcal{M}}_t^D$$

where, $\hat{\mathcal{M}}_t^D$ refers to the estimated dynamic points. As shown in Figure 2, we present a framework that uses a *coarse-to-fine* pipeline to obtain a clean static point cloud map step by step. To the best of our knowledge, DORF is the first approach that integrates the advantages of the current three classes of methods together.

B. Coarse Stage

The coarse stage distinguishes between dynamic and static points on a large spatial-temporal scale and is referred to as a coarse process compared to the finer process that operates on a smaller spatial-temporal scale.

1) Visibility-based Removal: In the first phase, we perform aggressive detection for dynamic points leveraging the visibility-based method in a confined region - *Volume of Interest* (VOI). It is defined as a cylinder of radius r_{max} that extends the height between h_{min} and h_{max}:

$$\mathcal{V}_t = \{\mathbf{p}_k | \mathbf{p}_k \in \mathcal{P}_t^Q, \rho_k < r_{max}, h_{min} < z_k < h_{max}\}$$

where the point $\mathbf{p}_k = (x_k, y_k, z_k)$, $\rho_k = \sqrt{x_k^2 + y_k^2}$. To accelerate the computation, all the points in $\mathcal{V}_t$ are searched by building a *kd-tree* [21] for fast indexing. Inspired by [20], we check the visibility from the comparison between query scan $\mathcal{P}_t^Q \in \mathcal{B}_{scans}$ and the corresponding submap following the chronological order of time step $t \in \mathcal{T}$.

Firstly, we extract $\mathcal{V}_t^Q$, the VOI of the query scan at pose ${}_Q^W\mathbf{T}_t$. Similarly, we extract the VOI of submap $\mathcal{V}_t^M$ from the input map at the same position. In this step, all points in $\mathcal{V}_t^M$ have been transformed to the current LiDAR frame.

Secondly, we project both $\mathcal{V}_t^Q$ and $\mathcal{V}_t^M$ into range images I_t^Q and I_t^M respectively. For each point $\mathbf{p} = (x, y, z)$ with cartesian coordinates, we use a spherical mapping $\Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ to transform it to image coordinates, as follows:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - \arctan(y, x)\pi^{-1}] \cdot w \\ [(f_{\text{up}} - \arcsin(z \cdot r^{-1}))f^{-1}] \cdot h \end{pmatrix}$$

where (u, v) are image coordinates, (h, w) are the height and width of the desired image, $f = f_{\text{up}} - f_{\text{down}}$ is the total vertical field-of-view of the LiDAR, and $r = \sqrt{x^2 + y^2 + z^2}$. For a range image pixel $I_t(u, v)$, it stores the range of the closest point projected into it. Then, the visibility of submap points is calculated via matrix element-wise subtraction as:

Algorithm 1: Receding Horizon Sampling (RHS)

Input: scan buffer $\mathcal{B}_{scans}$, poses buffer $\mathcal{B}_{poses}$, current scan index t, current static map $\mathcal{M}_t^S$
Output: The set of dynamic points $\mathcal{M}_{t+1}^D$, the updated set of static points $\mathcal{M}_{t+1}^S$

$\mathcal{M}_{t+1}^D \leftarrow \emptyset$, $\mathcal{M}_{t+1}^S \leftarrow \mathcal{M}_t^S$
// Construct a horizon from $\mathcal{B}_{scans}$ with size $N+1$:
$\mathcal{H} \leftarrow \{\mathcal{P}_{t-\frac{N}{2}}^Q, \cdots, \mathcal{P}_{t-1}^Q, \mathcal{P}_t^Q, \mathcal{P}_{t+1}^Q, \cdots, \mathcal{P}_{t+\frac{N}{2}}^Q\}$
$\mathcal{H}' \leftarrow$ FisherYatesShuffle($\mathcal{H}$)
$\mathcal{V}_t^M \leftarrow$ ExtractSubmapVOI($\mathcal{M}_{t+1}^S, t$)
$I_t^M =$ ProjectToRangeImage($\mathcal{V}_t^M$)
for $i = 1$ *to* N **do**
  Sample a query scan $\mathcal{P}_i^Q \sim \mathcal{H}'$
  Get the corresponding pose ${}_t^W T_i$ from $\mathcal{B}_{pose}$
  ${}^W\mathcal{P}_i^Q \leftarrow \mathcal{P}_i^Q \boxplus ({}_Q^W T_t{}^{-1} \boxplus {}_Q^W T_i)$
  $I_i^Q =$ ProjectToRangeImage(${}^W\mathcal{P}_i^Q$)
  // Detect dynamic points from residual image
  $I_i^{residual} = I_i^Q - I_i^M$
  $\mathcal{S}_i^D \leftarrow \emptyset$
  foreach *pixel* $p \in I_i^{residual}$ **do**
    row, col = GetPixelIndex(p)
    $r = I_i^{residual}[row, col]$
    if $r > \tau_{dyn}$ **then**
      // Retrieval the point associated with p
      $\mathbf{p}_Q \leftarrow$ RetrievalQueryPoint($\mathcal{P}_i^Q$, p)
      $\mathcal{S}_i^D \leftarrow \mathcal{S}_i^D \cup \mathbf{p}_Q$
  // Update the static map
  $\mathcal{M}_{t+1}^S \leftarrow \mathcal{M}_{t+1}^S - \mathcal{S}_i^D$
  $\mathcal{M}_{t+1}^D \leftarrow \mathcal{M}_{t+1}^D \cup \mathcal{S}_i^D$
return $\mathcal{M}_{t+1}^S$, $\mathcal{M}_{t+1}^D$

$I_t^{\text{residual}} = I_t^Q - I_t^M$. We add a point $\mathbf{p}_{t,(u,v)}^M$ in $\mathcal{V}_t^M$ into the dynamic map $\mathcal{M}_t^D$ if it satisfies this condition:

$$\mathcal{M}_t^D = \{\mathbf{p}_{t,(u,v)}^M | I_t^{\text{residual}}(u,v) > \tau_D\}$$

and static submap $\mathcal{M}_t^S$ is defined as:

$$\mathcal{M}_t^S = \mathcal{M}_t - \mathcal{M}_t^D$$

where $\mathcal{M}_t$ is the set of whole map points at the timestep t, and $\tau_{dyn} = 0.1$ is the minimum resolution threshold to judge a dynamic point.

However, the sparsity of a single scan, compared to a dense submap, leads to a limited visibility field of view (FOV) for the former. To address this issue, we propose incorporating a receding horizon centered at pose ${}_Q^W\mathbf{T}_t$, where neighboring scans within a fixed-size horizon participate in the scan-to-submap visibility checking if their time step is within the horizon. However, a chronological visitation of the point cloud scans in the horizon $\mathcal{H}$ may not be optimal, as dynamic objects may be moving in the same direction as the robot. To mitigate this issue, as shown in Algorithm 1, we propose the *Receding Horizon Sampling Mechanism*, which involves rearrangement for the visiting order of the scans in $\mathcal{H}$ based on Fisher-Yates Shuffle [22]. In each iteration, an unvisited scan $\mathcal{P}_i^Q \in \mathcal{H}$ is selected for the *scan-to-submap visibility check*. Furthermore, for the initial and final time steps of $\mathcal{T}$, a special horizon with a larger fixed size is employed for scans near the beginning and ending, since there are no scans on the outer side of the scan sequence.

2) Model Free Reverting: Visibility-based removal methods can detect dynamic points at a fast speed. However, visibility-based methods usually suffer from large incidence angle issues [14] [19] and occlusion problems [16].

These methods cause a large number of static ground points to be removed. Inspired by [23], we use a model-free method that leverages principle component analysis (PCA) [24] to revert false negative points in the submap VOI region. The observation behind this method is former removal is aggressive while the remaining static points are rather conservative, which strictly preserves the static points.

To process each dynamic point $\mathbf{p}_i^D \in \mathcal{M}_t^D$, we begin by searching for its nearest neighbors in the static map $\mathcal{M}_t^S$. Unlike the approach taken in [23], we utilize PCA to compute the three main components eigenvectors for the set of static nearest points since point cloud data is distributed in 3D space. Next, we select the eigenvector with the maximum eigenvalue. Similarly, we can compute another eigenvector for the union of this set of nearest points and the dynamic point itself. Finally, by comparing the normal distance between these two eigenvectors, we can determine the distribution change resulting from the addition of the dynamic point. If the change is sufficiently small, we revert the dynamic point $\mathbf{p}_i^{\text{D}}$ from $\mathcal{M}_t^D$ to $\mathcal{M}_t^S$.

C. Fine Stage

Although there is a reverting phase in the coarse stage that can help alleviate the *visibility issues*. However, its effect is limited because its nature is tightly coupled with *Visibility-based Removal*. In order to further discriminate the true dynamic points, we propose two decoupled reverting policies from the perspective of a smaller spatial-temporal scale, which are naturally decoupled with the former removal process.

1) Ground-based Reverting: An apparent observation is that dynamic objects are rooted on the ground in urban environments while the ground is naturally static. Therefore, if we can extract the ground points through segmentation, we can leverage these static points on the ground to achieve further reverting. Before the segmentation step, we build a 2.5D polar elevation map as a prior map. We traverse the map by following the chronological order of scans. For a query scan $\mathcal{P}_t$ with pose ${}_Q^W T_t$, we divide the associated volume of interest into n_θ sectors and n_r rings over the regular interval of azimuthal and radial directions, i.e. $\mathcal{V}_t^M = \bigcup_{i\in[n_r]j\in[n_\theta]} B_{(i,j),t}$, where $B_{(i,j),t}$ denotes the (i,j)-th polar bin at timestep t.

From a top-down view, these bins actually form a 2D polar grid map. Inspired by [25], we build a 2.5D elevation map based on the polar grid map for the terrain in polar bins, by introducing distilled elevation information to describe the

point cloud height distribution. Unlike [25], we store the supremum and infimum height values for each polar bin instead of the average height of the point cloud, which is far from the real ground height when a dynamic object is located in it. Let denote $Z_{(i,j),t} = \{z_k \in \mathbf{p}_k | \mathbf{p}_k \in B_{(i,j),t}\}$. That is we store the set for $B_{(i,j),t}$:

$$\mathcal{C}_{(i,j),t} = (\sup\{Z_{(i,j),t}\}, \inf\{Z_{(i,j),t}\})$$

Then we propose an effective heuristic method to give the prior estimation for the height of the ground plane leveraging the elevation information from $\mathcal{C}_{(i,j),t}$. We estimate the upper bound height for the ground plane as:

$$\hat{h} = \alpha \cdot \sup\{Z_{(i,j),t}\} + (1-\alpha) \cdot \inf\{Z_{(i,j),t}\}$$

where $\alpha = 0.9$ for case that $\sup\{Z_{(i,j),t}\} - \inf\{Z_{(i,j),t}\} > 0.2$ and otherwise $\alpha = 0.2$ in all sequences for experiment on SemanticKITTI Dataset.

Based on the polar elevation map, we conduct the bin-wise ground segmentation for points that are lower than $\hat{h}$. Since the volume of a single polar bin is very limited, the terrain surface in a single polar bin can be approximately modeled as a 3D flat plane. In each polar bin, we perform the RANSAC [26] algorithm to extract the plane points. Here we propose a *voting mechanism* to further double-check for the outlier in the ground points. We count the frequency of each point $\mathbf{p}$ that is considered a ground point during the ground segmentation of the entire map as $freq^G(\mathbf{p})$. The point is voted as a real ground point if:

$$freq^G(\mathbf{p}) > freq^R(\mathbf{p})$$

where $freq^R(\mathbf{p})$ is the frequency of the point $\mathbf{p}$ is judged as a dynamic point during the visibility-based removal phase.

2) BEV Occupancy Checking: In the last phase, we propose another reverting policy that can handle the misclassified points above the ground since *Ground-based Reverting* only considers the static ground points. Inspired by recent progress in perception utilizing Bird's-Eye View (BEV) on occupancy map [27] [28] [29], we propose an occupancy checking policy from Bird's-Eye View to conduct more robust checking for dynamic points above the ground.

Following the observation in *Ground-based Reverting*, we project all points onto discretized 2D occupancy grids where dynamic points on dynamic objects are reduced to a few adjacent 2D occupancy grids. The movement of dynamic objects in urban environments is primarily constrained to the horizontal plane and their movement in the vertical dimension is usually minimal. This means that projecting the point cloud onto a 2D plane can still capture the vast majority of the critical information about the movement of the objects, while greatly simplifying the computational complexity of the algorithms used to process the point cloud data. Therefore, the computation load for ray-tracing drops rapidly compared to operation in 3D space like [14]. For each projected laser point, we count the times of hitting $\hat{n}_{occ}$ and passing through $\hat{n}_{free}$ for each grid on the ray-tracing path.

The occupancy probability for each grid is computed as

$$Prob_{occ} = \sum \hat{n}_{occ} / (\sum \hat{n}_{occ} + \sum \hat{n}_{free})$$

after accumulating the count numbers during the checking on the whole map is finished. We have higher confidence to revert the status of laser points in a grid with a high occupancy probability over a certain threshold to be static. The whole phase follows the chronological visiting order of scans in $\mathcal{B}_{scans}$ and poses in $\mathcal{B}_{poses}$ to reach the full checking for the ray-tracing process.

Sequence No.	Method	PR[%]	RR[%]	F_1 score
00	Removert [20]	55.50	**98.44**	0.710
	ERASOR [17]	92.00	97.06	0.945
	Ours	**93.90**	98.39	**0.961**
01	Removert [20]	74.24	**98.40**	0.846
	ERASOR [17]	91.82	94.46	0.931
	Ours	**97.34**	93.97	**0.956**
02	Removert [20]	45.36	98.14	0.620
	ERASOR [17]	82.07	**98.39**	0.895
	Ours	**88.68**	91.23	**0.899**
05	Removert [20]	48.47	94.15	0.640
	ERASOR [17]	86.99	**97.70**	0.920
	Ours	**95.24**	92.68	**0.939**
07	Removert [20]	52.55	89.45	0.662
	ERASOR [17]	**93.91**	**98.69**	**0.962**
	Ours	92.80	76.65	0.840

TABLE I: Comparison with state-of-the-art methods on the SemanticKITTI dataset.

IV. Experiment

A. Experimental Setups

In order to evaluate the resulting quality for the generation of static 3D point cloud map after dynamic objects removal, we utilize the *preservation rate* (PR) and *rejection rate* (RR) from ERASOR [17] as the evaluation metrics. Compared to the precision-recall model, they exhibit lower sensitivity to the voxelization size. The metrics are calculated voxel-wise and we set the voxel size as 0.2 during the evaluation, which is the same with [17]. The definitions for the metrics are as follows:

- PR: $\frac{\#\text{ of preserved static points by the static map}}{\#\text{ of total static points on the raw map}}$
- RR: $1 - \frac{\#\text{ of preserved dynamic points by the static map}}{\#\text{ of total dynamic points on the raw map}}$

Besides PR and RR, we also calculated the F1 score for evaluation, which is the harmonic mean value of the PR and RR.

B. Experiments on Public Autonomous Driving Datasets

1) Comparison With SOTA Methods: One of the most well-known publicly available datasets for autonomous driving is the KITTI Dataset [30]. It's a de facto benchmark for many tasks like SLAM, perception, and planning in research. J.Behley et al added semantic labels to the KITTI Dataset to form the SemanticKITTI Dataset [12]. The SemanticKITTI Dataset provides point-wise ground-truth labels and each frame has a pose estimated by SuMa [31]. We select the top-5 time frames (Sequence $00, 01, 02, 05, 07$) in the SemanticKITTI Dataset that contain the maximum number of

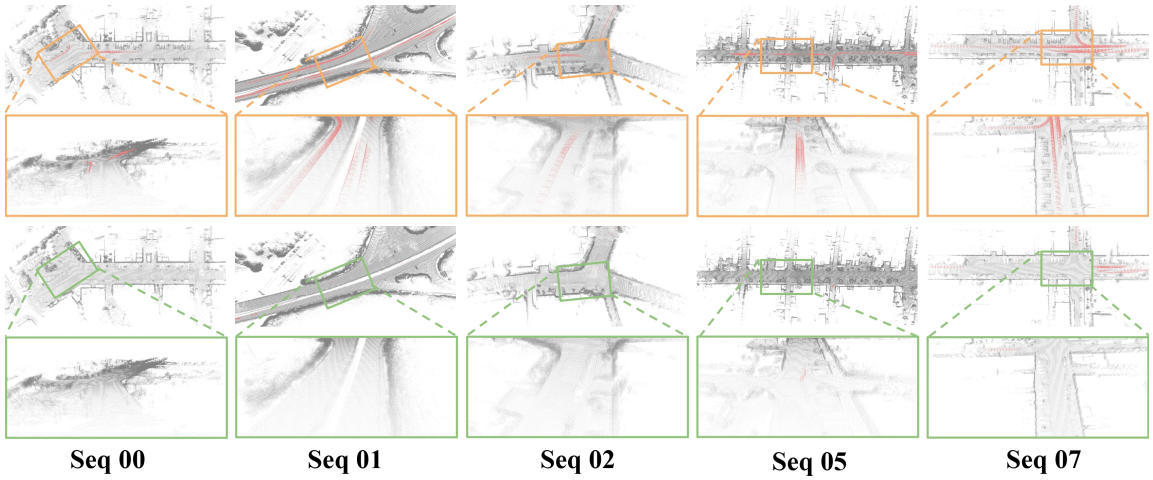

Fig. 3: Evaluation Result of Our Method On The SemanticKITTI Dataset (Sequence 00, 01, 02, 05 and 07). The first row is the Bird's-Eye View of the original point cloud map. The second row is the first-person view of the point cloud at the orange-boxed place. The third row is the Bird's-eye view of the clean point cloud map generated by our method. The fourth row is the first-person view of the clean point cloud at the green-boxed place. The red points are dynamic points while the grey points are static points.

Sequence No.	Method	PR[%]	RR[%]	F_1 score
00	RHS-only	82.64	**99.56**	0.903
	Without-MFR	93.86	98.29	0.960
	Without-GS	86.23	99.21	0.923
	GS-without-voting	**94.21**	93.97	0.941
	Without-BEV-OC	91.19	98.75	0.948
	Ours	93.90	98.39	**0.961**
01	RHS-only	88.09	**98.17**	0.929
	Without-MFR	97.32	93.99	0.956
	Without-GS	91.68	94.67	0.931
	GS-without-voting	**97.91**	86.10	0.916
	Without-BEV-OC	94.37	97.48	**0.959**
	Ours	97.34	93.97	0.956
02	RHS-only	76.37	**95.62**	0.849
	Without-MFR	88.57	91.05	0.898
	Without-GS	82.05	93.15	0.873
	GS-without-voting	**92.71**	59.72	0.726
	Without-BEV-OC	84.38	93.02	0.885
	Ours	88.68	91.23	**0.899**
05	RHS-only	83.44	**95.52**	0.891
	Without-MFR	95.20	92.81	**0.940**
	Without-GS	87.88	93.35	0.905
	GS-without-voting	**95.64**	79.21	0.867
	Without-BEV-OC	91.37	94.82	0.931
	Ours	95.24	92.68	0.939
07	RHS-only	82.90	**78.37**	0.806
	Without-MFR	92.76	76.73	**0.840**
	Without-GS	**93.16**	73.72	0.823
	GS-without-voting	93.14	73.67	0.823
	Without-BEV-OC	90.50	78.07	0.838
	Ours	92.80	76.65	**0.840**

TABLE II: Ablation Study for our method on SemanticKITTI Dataset

appearances of dynamic objects to evaluate our algorithms. We select the state-of-the-art methods Removert[1][20] and ERASOR[2][17] as the comparison baselines, both of which have open source code for easier comparison. Another recent state-of-the-art method ERASOR2[3][18] is not included here temporarily because its source code is not fully released at present. In order to keep the fairness of comparison, all the algorithms are tested on the same PC with an Intel CPU i7-12700F 4.90GHz processor. As shown in Table I, our method outperforms the existing methods [17] [20] on most of the sequences with the highest F1 score. A more clear and more intuitive qualitative result is shown in Figure 3.

2) Abalation Study: Additionally, we investigate five variants of our framework on SemanticKITTI Dataset to reveal the principles of DORF:

- *RHS-only* method only uses the first phase on receding horizon sample in the coarse stage.
- *Without-MFR* method removes the second phase about model-free reverting in the coarse stage.
- *Without-GS* method removes the first phase of ground-based reverting in the fine stage.
- *GS-without-voting* method removes the voting mechanism in ground-based reverting and directly reverts all ground points to be static without double checking.
- *Without-BEV-OC* method removes the second phase about BEV occupancy checking in the fine stage.

Table II demonstrates that the *RHS-only* approach obtains the highest score on RR but the lowest score on PR. This illustrates the effectiveness of our newly proposed receding horizon sampling mechanism for aggressive removal, which results in a higher removal rate of dynamic points at the cost of falsely removing some true static points. The *Without-MFR* approach indicates that the second phase of the coarse stage contributes only a small amount to the removal of dynamic points, as it is tightly integrated with the previous phase. Comparing our approach to *Without-GS* reveals that the *Ground-based Reverting* phase plays the most significant role in reverting misclassified true static points. The *Without-BEV-OC* approach shows that BEV occupancy checking has a similar and comparable contribution to ground-based

[1]https://github.com/irapkaist/removert
[2]https://github.com/LimHyungTae/ERASOR
[3]https://github.com/url-kaist/ERASOR2

reverting. The *GS-without-voting* approach demonstrates that the voting mechanism can help reduce the number of falsely reverted true dynamic points. Overall, our method achieves the best balance between removal and reverting.

# of pedestrians	Method	PR[%]	RR[%]	F_1 score
50	Removert [20]	70.97	60.86	0.655
	ERASOR [17]	84.82	68.05	0.755
	Ours	**95.26**	**98.63**	**0.969**
100	Removert [20]	73.11	55.55	0.631
	ERASOR [17]	74.39	56.94	0.645
	Ours	**95.81**	**97.93**	**0.969**
150	Removert [20]	72.30	53.85	0.617
	ERASOR [17]	73.03	44.19	0.551
	Ours	**96.33**	**98.44**	**0.974**

TABLE III: Comparison with state-of-the-art methods in the Gazebo pedestrian simulation environment.

C. Highly Crowded Simulation Evaluation

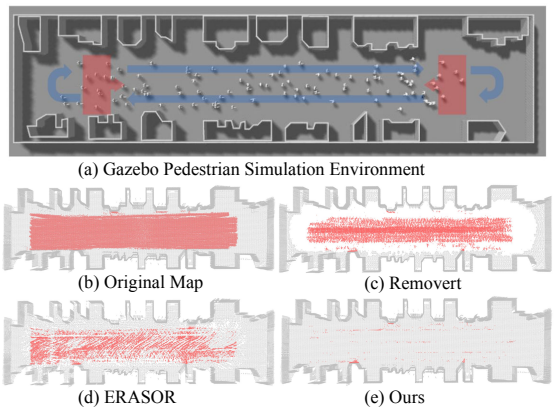

Fig. 4: (a) is the highly dynamic Gazebo [32] simulation scenario with 70 m × 10 m. The red arrow is the moving flow of pedestrians. The blue arrow is the robot trajectory when collecting data; (b) is the original prior map with huge amounts of dynamic points in the gazebo simulation environment with 50 pedestrians. The pink points are dynamic points from pedestrians, the gray points are static environment points, and the blank area means no point left there. (c) (d) (e) are the removal results in the same scenario for Removert [20], ERASOR [17], and our method respectively.

Although previous works are tested on the SemanticKITTI Dataset, it's actually not so challenging since most of the frames contain only one moving object. In urban environments, a more common scenario is a continuous flow of moving objects like crowded streams of cars or pedestrians. In order to further validate the robustness of our approach under such continuously dynamic environments, we set a highly crowded simulation environment as shown in the first row of Figure 4. Our simulation environment is built in the Gazebo Simulator [32] where the Menge [33] simulates the moving motion of crowded pedestrians and a mobile robot with a 3D LiDAR is moving among these simulated pedestrians for two rounds. The quantitative evaluation result is in Table III and Figure 5. We set three scenarios with different numbers of pedestrians (50, 100, 150) respectively. The dynamic points in all these scenarios are over 50%.

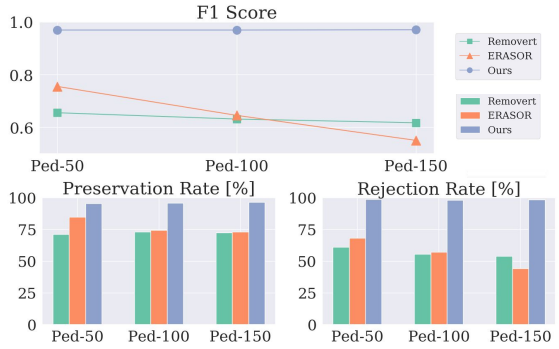

Fig. 5: Visualization for Evaluation Metrics In Gazebo Pedestrian Simulation. Ped-50, Ped-100, and Ped-150 are scenarios for 50, 100, and 150 pedestrians respectively.

Similarly, we perform further comparison in above scenarios. The existing methods [17] [20] degrade dramatically in crowded dynamic scenarios. The performance of [20] drops due to the severe *visibility issues*. While [17] is fragile to handle a highly dynamic environment due to the reliance on a high difference of free space percentage in the LiDAR sweeping space among input scans. Our method outperforms them by a large margin on all evaluation metrics. A more intuitive quality comparison result is in Figure 4. For [20], large amounts of static ground points are falsely removed and quite a number of dynamic points are kept in the middle of the original map. [17] preserves more ground points but still leaves lots of dynamic points. Compared to them, our method removes the most dynamic points while preserving the most static points at the same time.

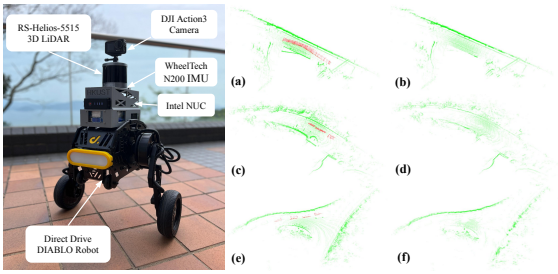

Fig. 6: Quality comparison for different real-world scenarios including (a) downhill (c) uphill, and (e) inclined intersection. The left part is the robotic system used to collect sensor data. The red points in the middle column are the ghost artifacts left by passing-by vehicles. The green points are from the static environment. The last columns (b) (d) (f) are the results after removing moving objects for the three scenes respectively.

D. Real World Experiments

To realistically validate the performance of our approach in the urban environment, we recorded a crowded pedestrian dataset on a playground and three datasets on different terrains. As shown in Figure 1, the crowded playground has quite highly dense pedestrians coming from every direction.

We also test our approach in three uneven urban terrains as shown in Figure 6. Unlike some methods [17] that may be limited to flat ground scenes, our approach also has a significant effect on uneven terrains including uphill, downhill, and inclined intersection scenes.

V. CONCLUSION

This work proposes a novel framework for dynamic object removal that follows a coarse-to-fine pipeline to remove potential dynamic points and recover miss-identified points back to static. We validate our approach quantitatively and qualitatively on the SemanticKITTI Dataset and demonstrate its robustness in a super-crowded gazebo pedestrian simulation environment. Additionally, we test our framework on several unstructured terrain scenes in the real world, showcasing its generalization ability. Our method generates a cleaned static 3D point cloud map to improve downstream navigation and localization tasks for mobile robotic systems. There also exist some limitations to our work. This work relies on the relatively flat ground conditions in urban environments. It may have limited extending ability in a wild environment, which presents different terrain configurations. Another limitation is we detect the moving objects solely relying on motion properties. Combining semantics with motion property may reinforce the accuracy and robustness. In future works, we plan to generalize our framework to fit both online and offline modalities. We are also interested in extending this work beyond urban environments.

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," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, 2020.

[2] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, 2014.

[3] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, pp. 4861–4868, 2022.

[4] D. J. Yoon, T. Y. Tang, and T. D. Barfoot, "Mapless online detection of dynamic objects in 3d lidar," *2019 16th Conference on Computer and Robot Vision (CRV)*, pp. 113–120, 2018.

[5] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Motion-based detection and tracking in 3d lidar scans," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4508–4513, 2016.

[6] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Static map generation from 3d lidar point clouds exploiting ground segmentation," *Robotics Auton. Syst.*, vol. 159, p. 104287, 2022.

[7] H. Fu, H. Xue, and G. Xie, "Mapcleaner: Efficiently removing moving objects from point cloud maps in autonomous driving scenarios," *Remote. Sens.*, vol. 14, p. 4496, 2022.

[8] C. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017.

[9] H. Zhang, Y. Wang, J. Cai, H.-M. Hsu, H. Ji, and J.-N. Hwang, "Lifts: Lidar and monocular image fusion for multi-object tracking and segmentation," 2020.

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

[11] Z. Zhou, Y. Zhang, and H. Foroosh, "Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 189–13 198, 2021.

[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," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9296–9306, 2019.

[13] 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, pp. 6529–6536, 2021.

[14] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap : A probabilistic , flexible , and compact 3 d map representation for robotic systems," 2010.

[15] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and V. Yalla, "Robust method for removing dynamic objects from point clouds," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10 765–10 771, 2020.

[16] 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, pp. 1679–1686, 2018.

[17] 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, pp. 2272–2279, 2021.

[18] 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.

[19] F. Pomerleau, P. Krüsi, F. Colas, P. T. Furgale, and R. Y. Siegwart, "Long-term 3d map maintenance in dynamic environments," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3712–3719, 2014.

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

[21] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, pp. 509–517, 1975.

[22] I. Febriani, R. Ekawati, U. Supriadi, and M. M. I. Abdullah, "Fisher-yates shuffle algorithm for randomization math exam on computer based-test," 2021.

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

[24] H. T. Shen, "Principal component analysis," in *Encyclopedia of Database Systems*, 2009.

[25] B. Douillard, J. P. Underwood, N. Melkumyan, S. P. N. Singh, S. Vasudevan, C. J. Brunner, and A. J. Quadros, "Hybrid elevation maps: 3d surface models for segmentation," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1532–1538, 2010.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," *ArXiv*, vol. abs/2205.13542, 2022.

[28] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," *arXiv preprint arXiv:2302.07817*, 2023.

[29] A.-Q. Cao and R. de Charette, "Monoscene: Monocular 3d semantic scene completion," in *CVPR*, 2022.

[30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[31] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," *Robotics: Science and Systems XIV*, 2018.

[32] N. P. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149–2154 vol.3, 2004.

[33] S. Curtis, A. Best, and D. Manocha, "Menge: A modular framework for simulating crowd movement," 2016.