# ERASOR2: Instance-Aware Robust 3D Mapping of the Static World in Dynamic Scenes

Hyungtae Lim* Lucas Nunes† Benedikt Mersch† Xieyunali Chen†
Jens Behley† Hyun Myung*,‡ Cyrill Stachniss†,◇

*Urban Robotics Lab., School of Electrical Engineering and KI-AI, KAIST, Republic of Korea
†Photogrammetry & Robotics Lab, University of Bonn, Germany

***Abstract*— A map of the environment is an essential component for robotic navigation. In the majority of cases, a map of the static part of the world is the basis for localization, planning, and navigation. However, dynamic objects that are presented in the scenes during mapping leave undesirable traces in the map, which can impede mobile robots from achieving successful robotic navigation. To remove the artifacts caused by dynamic objects in the map, we propose a novel instance-aware map building method. Our approach rejects dynamic points at an instance-level while preserving most static points by exploiting instance segmentation estimates. Furthermore, we propose effective ways to consider the erroneous estimates of instance segmentation, enabling our proposed method to be robust even under imprecise instance segmentation. As demonstrated in our experimental evaluation, our approach shows substantial performance increases in terms of both, the preservation of static points and rejection of dynamic points. Our code is available at `https://github.com/url-kaist/ERASOR2`.**

## I. Introduction

A static map is an essential component for many mobile robot platforms to achieve robot navigation [28], [46], [47], [49]. Static landmarks provide geometrical features that are repeatably observed, thus a robot can utilize a map of the static world to localize itself reliably. These maps can be represented in various forms, such as occupancy grid maps [21], feature maps [43], topological maps [8], or structural-variance-robust place representations [24]. In this paper, we focus on 3D point cloud maps [23], [28], which are an output of the accumulation of laser scans from a 3D LiDAR sensor. Thus, we aim for the mapping of the static 3D points.

Laser scanner data captured in urban environments often include dynamic objects. If integrated into a map, this information can degrade the performance of localization and navigation of a mobile robot [28]. Dynamic instances such as buses, cars, and pedestrians, take up space only temporarily. This causes misrecognitions about space occupancy in the map [11], [12], [26]. Therefore, static map building, which rejects dynamic points caused by moving objects, is the key and necessary to reduce these negative effects.

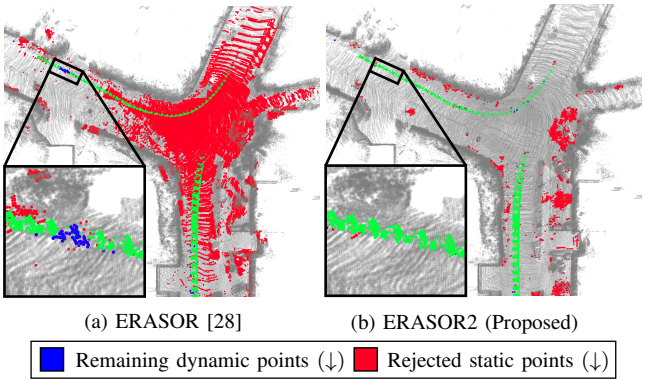



Fig. 1. Performance comparison of (a) ERASOR [28] and (b) our proposed method, called *ERASOR2*, on Seq. `02` of SemanticKITTI dataset [3]. By exploiting instance segmentation estimates, our proposed method can reject dynamic points at an instance-level with less loss of static points compared with ERASOR [28]. For this reason, dynamic points from a motorcyclist are successfully rejected. In this paper, the green, red, blue, and gray colors indicate successfully rejected dynamic points (true positives), wrongly rejected static points (false positives), remaining dynamic points (false negatives), and remaining static points (true negatives) in the map, respectively (best viewed in color).

To tackle the problem of mapping in dynamic environments, numerous researchers have proposed static map building methods exploiting geometrical discrepancies between each scan and a map cloud [2], [17], [23], [36], [44], [56]. In practice, three limitations still exist in most systems. First, once the poses from scan registration or pose graph optimization become imprecise, the geometric correlation between the current scan and map cloud also becomes inaccurate, resulting in the loss of many static points (red points in Fig. 1). Second, existing methods often do not consider any instance-level information, so some points from moving objects could remain in the map cloud (blue points in Fig. 1(a)). Third, some methods use the segmentation information, but the way to deal with noisy or erroneous instance segmentation is still less examined.

In sum, the aim is to bring the number of false positive and false negative dynamic points toward zero even though

‡Corresponding author: Hyun Myung
◇Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

This work was supported by Korea Evaluation Institute of Industrial Technology (KEIT) funded by the Korea Government (MOTIE) under Grant No.20018216, Development of mobile intelligence SW for autonomous navigation of legged robots in dynamic and atypical environments for real application, BK21 FOUR, and the European Union's Horizon Europe research and innovation programme under grant agreement No 101070405 (DigiForest).

realistic and thus imprecise poses and instance segmentation estimates are provided. Several researchers have proposed learning-based moving object segmentation (MOS) methods [11], [12], [32], which segment dynamic and static points for each scan. Although these methods show precise segmentation, the performance can decrease when used in environments that differ significantly from the ones used for training or when using different sensor setups. For these reasons, we believe map-centric management is still required for achieving high-quality maps.

The main contribution of this paper is a novel instance-aware static map building approach, called *ERASOR2*. It shares some of the philosophy of ERASOR [28], a technique that rejects dynamic points in a region-wise manner using so-called *pseudo occupancy*. Our new approach overcomes the shortcomings of existing static map building methods by leveraging instance-level information reducing both, false positives and false negatives, as illustrated in Fig. 1(b). In particular, our new method is not limited to specific instance segmentation algorithms, such that the pipeline easily works with other instance segmentation methods and in various environments. Furthermore, we propose some effective ways to deal with the noise of instance segmentation estimates, i.e. under- and over-segmentation [35], and unlabeled points, enabling our proposed method to be robust against imprecise instance segmentation.

In sum, we make the following three key claims. Our approach (i) precisely rejects dynamic points at an instance-level while preserving static points compared with the state-of-the-art methods, (ii) is robust against the imprecise instance segmentation estimates, and (iii) shows superior performance even in highly crowded environments, demonstrating the robustness of our proposed method and necessity of map-level management by comparing our proposed method to a recent state-of-the-art egocentric learning-based approach. These claims are backed up by the following sections and our experimental evaluation.

## II. Related Work

In general, there are three ways of building a map of the static environment depending on the purpose of dynamic point removal. The first one is to reduce wrong data association within the simultaneous localization and mapping (SLAM) process, so dynamic points are simultaneously rejected when poses are estimated [20], [38], [39], [41], [45], [50]. The second one is to segment moving objects in the surroundings for each scan, which focuses more on an egocentric perception [11]–[14], [25], [32], [56]. The last one is to build a static map as a post-processing of SLAM before the localization or navigation step [16], [22], [23], [28], [40], [42]. In this study, we place more emphasis on the last case. This last group of approaches can be further classified into three subgroups: a) ray tracing-based, b) visibility-based, and c) traversability-based methods.

Occupancy grids [14], [19], OctoMap [5], [21], and TSDF-based methods [27], [33], [51], [52] are typical methods that use ray tracing, which updates the state of some space, such as grid cells or voxels, by checking whether a ray runs through that some space or not. By utilizing the ray tracing concept, Schauer and Nüchter [44] proposed removing dynamic points by traversing a voxel occupancy grid and Pagad *et al*. [36] suggested a combination of object detection and OctoMap. However, these methods are often affected by the quality of estimated poses. If imprecise poses are provided, many static points are removed because the decision on whether the points are static or dynamic becomes challenging (see Section IV.*B*). In addition, ray tracing-based methods are computationally expensive because large parts of the scene are taken into account for each traced ray.

To increase computational efficiency, visibility-based methods [23], [37], [40] have been proposed, which usually detect dynamic points by using the geometrical discrepancies between a query range image and a map range image. That is, if the range value of the pixel on a query image is larger than that on the map image, these visibility-based methods estimate that occlusion occurs at that location corresponding to the pixel of a map range image. Thus, the points projected onto those pixels are considered as dynamic points.

However, owing to the uneven distribution of the 3D point cloud captured by a mechanically spinning LiDAR sensor, the incidence angle, which is an angle between a ray and the normal direction of the ground, becomes more ambiguous [29]. Thus, neighboring ground points far from the sensor frame are projected to the same pixels, resulting in many false positives. To tackle the limitation of the range image, Pomerleau *et al*. [40] exploited the normal vector as another cue to resolve the incidence angle ambiguity. Kim and Kim [23] proposed a window-based pixel comparison and multi-resolution methods. However, from the pixel-wise removal, dynamic points remained scattered in the map, which led to the introduction of traversability-based methods.

Recently, traversability-based methods have been proposed to reject dynamic points based on the fact that dynamic objects in urban environments usually move on the ground [1], [2], [17], [28]. Lim *et al.* [28] proposed an efficient concept to estimate temporarily occupied regions by comparing the min-max height of a query and map, called *pseudo occupancy*. Fu *et al*. [17] proposed robust drivable region proposals to extract ground points. Arora *et al*. [2] suggested offline ground segmentation on the image plane, and then dynamic and static points are discerned. Despite substantial progress being made, two potential limitations still exist. First, the aforementioned works firmly consider the estimated ground points as static points. Yet, the researchers did not propose methods to deal with cases where dynamic points are misclassified as ground points. Second, the proposed methods do not consider instance information, potentially leading to partial false positives, as presented in Fig. 1(a).

Meanwhile, deep learning-based MOS approaches have led to significant improvement [11], [25], [32], [48]. One might think that a static map can be simply built by accumulating the estimated static points by MOS methods. However, these methods still rely on the supervised labels and it is empirically demonstrated that a static map by the estimates

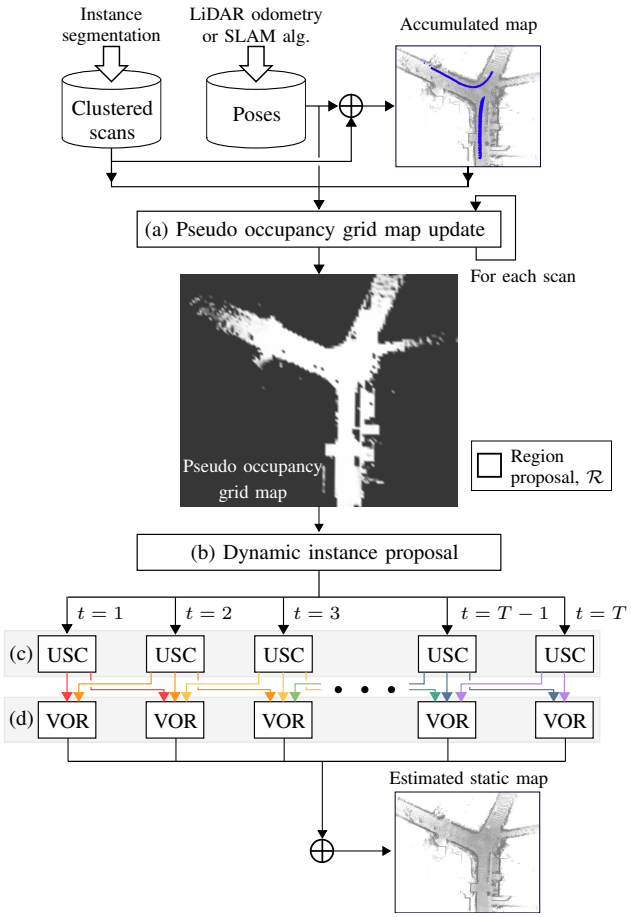



Fig. 2. Overview of our proposed method, which mainly consists of four steps. (a) First, sharing the philosophy of ERASOR [28] that checks the regions where some dynamic objects temporarily occupy by using *pseudo occupancy*, the pseudo occupancy grid map is updated. (b) Using some grid cells that are highly likely to be temporarily occupied, or region proposal, $\mathcal{R}$ (white regions), instances located in $\mathcal{R}$ are considered as moving instances for each scan. (c) Then, *under-segmentation check (USC)* is applied to reject a portion of points from a moving object but segmented with a large static instance. (d) Finally, using estimated dynamic points for each scan, the estimated static points adjacent to these dynamic points are rejected in a spatio-temporal manner via *volumetric outlier removal (VOR)*.

of MOS methods has many undesirable false negatives (see Section IV.*D*). Therefore, map-centric dynamic points management is still necessary to leverage temporal information.

In this paper, we propose a novel instance-aware approach to mapping the static world. Unlike previous static map building approaches that only exploit geometrical discrepancies, our proposed method blends the best of the philosophy of traversability-based methods and instance-aware dynamic point removal, resulting in cleaner static maps with less false positive and false negative points.

## III. Our Approach to Instance-Aware Mapping of the Static World

The schematic diagram of our instance-aware static map building method is presented in Fig. 2. It mainly consists of four parts: pseudo occupancy grid map update, dynamic instance proposal, under-segmentation check, and volumetric outlier removal. Our proposed method aims for dynamic points removal at an instance-level.

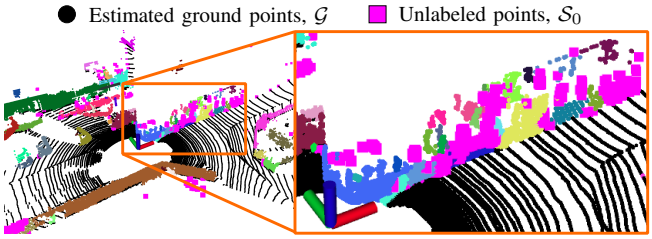



Fig. 3. Example of instance segmentation estimate for Seq. `05` around frame 2,630 on the SemanticKITTI dataset [3]. Owing to the erroneous instance segmentation estimates, the points from a bus are under-segmented and even unlabeled; thus, the unlabeled points, $\mathcal{S}_0$, potentially contain some dynamic points (magenta rectangles). Different colors indicate that the cloud points are clustered as different instances (best viewed in color).

### A. Notations and Problem Definition

We begin by explaining the notations of inputs. Let us define a 3D point cloud as $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\}$ where $N$ denotes the size of point cloud and each point, $\mathbf{p}$, consists of an $x$, $y$, and $z$ value, i.e., $\mathbf{p} = [x, y, z]^\intercal$, in Cartesian coordinates. In this study, we divide $\mathcal{P}$ into two categories: ground points and non-ground points. Those will be further classified into multiple instance segments by the following procedure.

First, $\mathcal{P}$ is taken as an input of a ground segmentation approach [29], so it is divided into ground points, $\mathcal{G}$, and non-ground points, $\mathcal{P}'$, which satisfy $\mathcal{G} \cap \mathcal{P}' = \varnothing$ and $\mathcal{G} \cup \mathcal{P}' = \mathcal{P}$. Next, a clustering method is utilized to separate $\mathcal{P}'$ into unlabeled points, $\mathcal{S}_0$, and $K$ instances, $\mathcal{I}^+ = \{\mathcal{S}_1, \cdots, \mathcal{S}_K\}$, such that $\mathcal{I} = \{\mathcal{S}_0\} \cup \mathcal{I}^+$, each of which follows $\mathcal{S}_a \cap \mathcal{S}_b = \varnothing$ if $a \neq b$. Unlabeled points $\mathcal{S}_0$ are a set of unclustered points, which is an output of most instance segmentation methods due to the failure of clustering caused by the sparse distribution of the cloud points.

Note that $\mathcal{G}$ and $\mathcal{I}$ are estimates, so segments could be partially under- or over-segmented. In particular, the actual dynamic points are sometimes classified into $\mathcal{S}_0$, as shown in Fig. 3. Thus, it is necessary to consider the potential failure of instance segmentation. Therefore, we propose robust dynamic points rejection methods to account for the noises and errors of instance segmentation (see Section III.*D* and Section III.*E*).

In general, static map building methods assume that corrected poses are given [23]. Let $\mathbf{T}_t^w$ be the transformation matrix of the $t$-th body frame with respect to world frame $w$, the map cloud $\mathcal{M}$ can be defined as:

$$\mathcal{M} = \nu\left( \bigcup_{t \in \langle T \rangle} \nu\left( \{ \mathbf{T}_t^w \mathbf{p} \mid \mathbf{p} \in \mathcal{P}_t \} \right) \right), \tag{1}$$

where $\nu(\cdot)$ denotes a voxelization; $T$ is the total time step; $\langle T \rangle$ is equal to $\{1, 2, \ldots, T\}$; $\mathcal{P}_t$ denotes the point cloud whose origin is the $t$-th body frame; $\mathbf{T}_t^w \mathbf{p}$ means that a point expressed in the $t$-th body frame is transformed into world frame $w$. Consequently, $\mathcal{M}$ is an accumulated map, so it still contains all measured dynamic points.

Following Lim *et al.* [28], the problem definition can be expressed as follows:

$$\hat{\mathcal{M}} = \mathcal{M} - \bigcup_{t \in \langle T \rangle} \hat{\mathcal{M}}_t^{\text{dyn}}, \tag{2}$$

where $\hat{\mathcal{M}}$ denotes the estimated static map cloud and $\hat{\mathcal{M}}_t^{\text{dyn}}$ denotes the dynamic points to be rejected from $\mathcal{M}$ at time step $t$.

There are two challenges in (2): a) once the direct subtraction of $\hat{\mathcal{M}}_t^{\text{dyn}}$ from $\mathcal{M}$ wrongly rejects static points, there is no way to revert these static points and b) $\hat{\mathcal{M}}_t^{\text{dyn}}$ is decided without any instance-level information, so points of moving objects could remain in $\hat{\mathcal{M}}$ while only a portion of the objects would be rejected.

To tackle these two problems, we redefine the problem for the work presented here as:

$$\hat{\mathcal{M}} = \nu\Bigg( \bigcup_{t \in \langle T \rangle} \nu\big(\{\mathbf{T}_t^w \mathbf{p} \mid \mathbf{p} \in \mathcal{P}_t - \hat{\mathcal{U}}_t - \bigcup_{\mathcal{S} \in \hat{\mathcal{I}}_t} \mathcal{S}\}\big)\Bigg), \tag{3}$$

where $\hat{\mathcal{I}}_t$ denotes the estimated dynamic instances set, which is a subset of the predictions from the instance segmentation at $t$, $\mathcal{I}_t$, i.e. $\hat{\mathcal{I}}_t \subset \mathcal{I}_t$. $\mathcal{S}$ denotes the moving instance included in $\hat{\mathcal{I}}_t$, and $\hat{\mathcal{U}}_t$ denotes the additionally estimated dynamic points to overcome the erroneous predictions of instance segmentation (see Section III.*E*).

By doing so, (3) has advantages over (2) for the following two reasons. First, the false positives of our proposed method affect the quality of a map less directly. This is because static instances are usually observed multiple times, so false positives at $t$ can be naturally compensated if the corresponding static points are preserved in other viewpoints. In contrast, once false positives occur, the direct subtraction in (2) triggers some empty holes in the map by rejecting static points, which Lim *et al*. [28] has an issue with. Second, by rejecting dynamic points in an instance-aware manner, the aforementioned partial removal issue can be resolved, as shown in Fig. 1(b).

Thus, the following sections explain how to estimate $\hat{\mathcal{I}}_t$ and $\hat{\mathcal{U}}_t$ in (3), overcoming the erroneous estimates of instance segmentation.

### *B. Pseudo Occupancy Grid Map Update*

It is not easy to distinguish moving objects by only using a single scan captured at $t$. Thus, we utilize the temporal information of all scans and fuse the information from the map-centric perspective, as shown in Fig. 2(a). A map cloud is the accumulation of all scans transformed into the reference frame, which includes traces of moving objects over time. Whereas each scan is a snapshot of the surroundings, and traversable regions may only be temporarily occupied by moving objects. Accordingly, we check the geometrical discrepancies due to the moving objects by comparing the $t$-th point cloud transformed into the map frame and map cloud. In doing so, we can estimate the regions that contain the traces of dynamic objects in the map. Finally, we can consider the non-ground points in those regions as moving objects.

Based on these observations, we propose to build a *pseudo occupancy grid map*. Unlike generic occupancy grid maps, which model whether space is occupied or not, our proposed grid map models whether the regions are likely to be temporally occupied by dynamic objects in a probabilistic manner [6], [7], [31]. By estimating the probabilities of which cells could contain moving objects, non-ground points in the grid cells with high probabilities are highly likely to be from dynamic instances.

To this end, we update the probability of each grid cell based on a binary Bayes filter to fuse all the predictions from each scan. In this way, the information is aggregated from the map-centric perspective and can be used to improve decision making about static and dynamic 3D points. Formally, the pseudo occupancy grid map is initialized with a prior probability, $p_0$. Then, the pseudo occupancy grid map is updated, whose joint distribution of the binary state, $\boldsymbol{m}$, is expressed as follows:

$$p\left(\boldsymbol{m} \mid \mathbf{z}_{1:t}\right) = \prod_i p\left(\boldsymbol{m}_i \mid \mathbf{z}_{1:t}\right), \tag{4}$$

where $\boldsymbol{m}_i$ denotes the binary state of whether the $i$-th[1] grid cell could be temporarily occupied or not and $\mathbf{z}_{1:t}$ denotes the observed measurements during whole time steps, each of which is the geometrical discrepancy between the $t$-th scan and the map cloud, which will be explained in the following paragraphs.

Then, by using Bayes' rule and log-odds notation, $l(x) = \text{logit}(p(x)) = \log \frac{p(x)}{1-p(x)}$, (4) is paraphrased as follows:

$$l\left(\boldsymbol{m}_i \mid \mathbf{z}_{1:t}\right) = l\left(\boldsymbol{m}_i \mid \mathbf{z}_{1:t-1}\right) + l\left(\boldsymbol{m}_i \mid \mathbf{z}_t\right) - l\left(\boldsymbol{m}_i\right), \tag{5}$$

where $l\left(\boldsymbol{m}_i \mid \mathbf{z}_{1:t-1}\right)$, $l\left(\boldsymbol{m}_i \mid \mathbf{z}_t\right)$, and $l\left(\boldsymbol{m}_i\right)$ represent a recursive term, update term, and prior term, respectively; $l\left(\boldsymbol{m}_i\right)$ is equal to log-odds of prior probability, i.e. $\text{logit}(p_0)$.

Next, we model the geometrical discrepancy $\mathbf{z}_t$. There are typically three possible occasions for each grid cell as described below:

- Case A: an object occupies the grid cell of map cloud while the corresponding region is free in that of the $t$-th cloud (red bins in Fig. 4).
- Case B: an object occupies both grid cell of the map and the $t$-th cloud (emerald bins in Fig. 4).
- Case C: neither an object occupies the cell of the map nor the $t$-th cloud, i.e. both are unoccupied (white bins in Fig. 4).

Therefore, our objective is to find some regions where case A occurs and then increase the probabilities of these grid cells.

To this end, we use *pseudo occupancy*, which is introduced by Lim *et al*. [28], to efficiently check the discrepancies between the transformed $t$-th scan and map cloud for each grid cell. The pseudo occupancy is described by the min-max $z$ difference $\Delta h$. Then, if there is a large difference of pseudo occupancy between the scan and map cloud, we can determine that the corresponding cell is occupied at another

[1]Strictly speaking, the index $i$ should be expressed as $(u, v)$, where $u$ and $v$ are the indices of the 2D grid map, yet $(u, v)$ is simplified to $i$ for brevity.

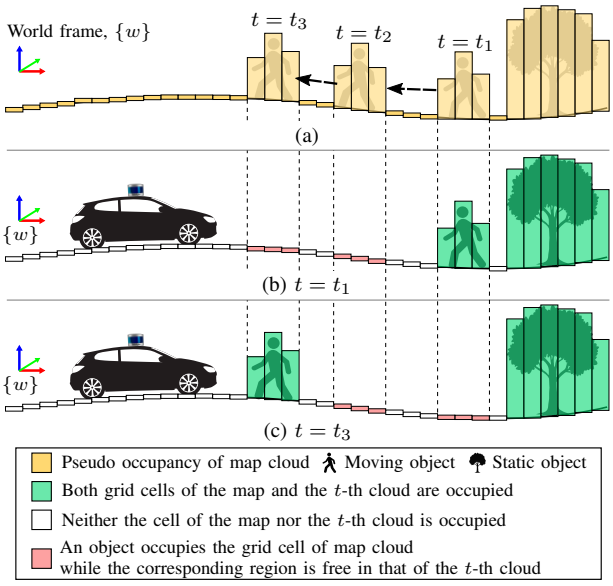



Fig. 4. Visual description of the difference in temporal information between a map cloud and point cloud at time step $t$. For convenience, the 2D grid is represented in 1D. Because a dynamic object moves over time (dashed arrows in (a)), by comparing the geometrical discrepancies between the $t$-th scan and map cloud, we can detect the traces from dynamic objects. The red bins are grid cells of our interest to be cleaned (best viewed in color). (a) Map cloud encoded with pseudo occupancy, where the length of the bins indicates the min-max $z$ value difference in each grid cell. (b) When $t = t_1$. Traces of the moving object at $t_2$ and $t_3$ can be detected. (c) When $t = t_3$. Traces of the moving object at $t_1$ and $t_2$ can be detected. Finally, the probabilities of the grid cells that correspond to red bins increase, which means that the cells are likely to contain dynamic points.

time step $t'$, where $t' \neq t$, which is highlighted as red bins in Fig. 4.

Formally, we first extract the physically significant space where dynamic objects are likely to be located, called *volume of interest (VOI)*, $\mathcal{V}_t$, for each $t$-th point cloud as:

$$\mathcal{V}_t = \{\mathbf{p}_l \in \mathcal{P}_t \mid h_{\min} < z_l < h_{\max}\}, \tag{6}$$

where $h_{\min}$ and $h_{\max}$ denote the minimum and maximum heights of VOI, respectively. Then, the pseudo occupancy of each cell is described using the VOI. By extracting the VOI in advance, we can preserve the purpose of the pseudo occupancy that we intended, i.e. description of being occupied at an instance-level, even in some cases where a roof exists.

Next, the local map, $\mathcal{M}_{\text{local}}$, to be compared with the transformed $\mathcal{V}_t$ is defined as:

$$\mathcal{M}_{\text{local}} = \nu\Big(\bigcup_{t\in\langle T\rangle} \nu\big(\{\mathbf{T}_t^w \mathbf{p} \mid \mathbf{p} \in \mathcal{V}_t\}\big)\Big), \tag{7}$$

where $w$ is the coordinate frame of the local map. By using the local coordinate, our method is available in cases where some pitch angles or $z$-values of poses are large with respect to the world frame.

Let $\mathcal{V}_t^Q = \{\mathbf{T}_t^w \mathbf{p} \mid \mathbf{p} \in \mathcal{V}_t\}$ be the transformed cloud into the $w$ frame and $\mathcal{V}_t^M \subset \mathcal{M}_{\text{local}}$ be the corresponding map VOI, which is the overlapped region with $\mathcal{V}_t^Q$ as illustrated in Fig. 5(a). Finally, the pseudo occupancy for the $i$-th cell

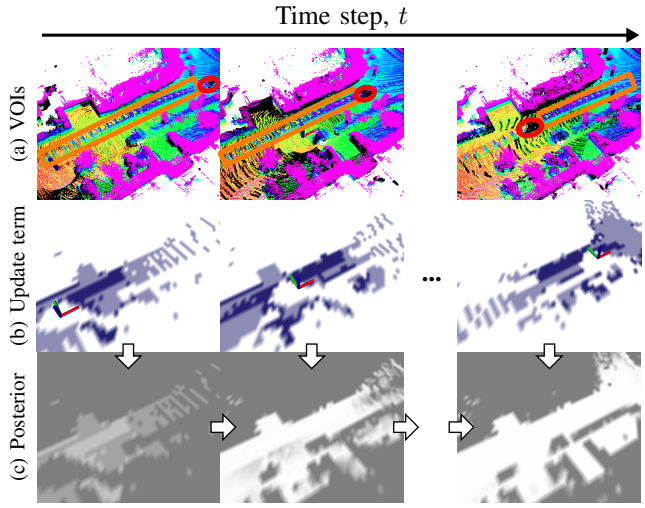



Fig. 5. Procedure of updating pseudo occupancy grid map. (a) By comparing the volume of interest (VOI) of a map (rainbow-colored) and that of the $t$-th point cloud (black), the unoccupied regions can be checked by the geometrical discrepancies between these VOIs (orange rectangles). This is because a moving object temporarily occupies some partial regions at the moment (red circles). (b) Based on the geometrical discrepancy, the update terms for grid cells are adaptively set. Indigo and light blue colors indicate the regions updated by high probability and those by smaller probability, respectively (see Eq. (9)). (c) The updated pseudo occupancy grid map. As time goes by, the probabilities of grid cells corresponding to the regions that are likely to be temporarily occupied by moving objects increase. The whiter the regions are, the higher the probabilities are (best viewed in color).

expressed in the reference frame $\Delta h_{i,t}$ is defined as:

$$\Delta h_{i,t} = \mathcal{H}(\mathcal{V}_{i,t}) = \max\{Z_{i,t}\} - \bar{z}_{i,t}^{\text{lower}}, \tag{8}$$

where $Z_{i,t} = \{z_l \mid \mathbf{p}_l = [x_l, y_l, z_l]^\intercal \in \mathcal{V}_{i,t}\}$ and $\mathcal{V}_{i,t}$ is the subset of VOI corresponding to the region of $i$-th grid cell; let the subsets of the map and transformed scan be $\mathcal{V}_{i,t}^M \subset \mathcal{V}_t^M$ and $\mathcal{V}_{i,t}^Q \subset \mathcal{V}_t^Q$, respectively, then $\bar{z}_{i,t}^{\text{lower}} = \min(\bar{z}_{i,t}^Q, \bar{z}_{i,t}^M)$, where $\bar{z}_{i,t}^Q$ and $\bar{z}_{i,t}^M$ denote the mean $z$ values of ground points within $\mathcal{V}_{i,t}^Q$ and $\mathcal{V}_{i,t}^M$. Thus, by using (8), the pseudo occupancies corresponding to $\mathcal{V}_{i,t}^Q$ and $\mathcal{V}_{i,t}^M$ are set as $\Delta h_{i,t}^Q = \mathcal{H}(\mathcal{V}_{i,t}^Q)$ and $\Delta h_{i,t}^M = \mathcal{H}(\mathcal{V}_{i,t}^M)$, respectively.

$\Delta h_{i,t}^Q$ and $\Delta h_{i,t}^M$ are used to check the case of our interest, i.e. case A. As illustrated in Fig. 4, large difference between $\Delta h_{i,t}^Q$ and $\Delta h_{i,t}^M$ indicates that the $i$-th grid cell is occupied at another time step $t'$ but is not occupied at $t$ resulting in $\Delta h_{i,t}^Q \simeq 0$. To robustly check the difference, we use the ratio test [30] as $\Delta h_{i,t}^Q / \Delta h_{i,t}^M < \tau_{\Delta h}$ where $\tau_{\Delta h}$ denotes the ratio threshold. In addition, we check two conditions to judge if sufficient points are observed and if the cell is not in case C. To this end, first, we check if the number of points in $\mathcal{V}_{i,t}^Q$ and that in $\mathcal{V}_{i,t}^M$ are larger than $N_{\min}$, where $N_{\min}$ denotes the minimum number of points for each grid cell. Second, we check if $\Delta h_{i,t}^M > \Delta h_{\min}$, where $\Delta h_{\min}$ denotes the minimum pseudo occupancy. If all three conditions are satisfied, we consider that case A occurs, which is highlighted as orange rectangles in Fig. 5(a).

However, even though any conditions are not satisfied, if most points in $\mathcal{V}_{i,t}^Q$ are ground points, which means that the cell is currently free, then the $i$-th cell can be considered as an area potentially occupied by a moving object. This is

because the ground regions can be potentially traversable by moving objects. Therefore, the update term is adaptively set based on these observations as follows:

$$l\left(\boldsymbol{m}_i \mid \mathbf{z}_t\right) = \begin{cases} \kappa_{\text{inc}} \cdot \text{logit}(p_{\text{inc}}), & \text{if case A occurs} \\ \text{logit}(p_{\text{inc}}), & \text{if } \frac{|\mathcal{G}^Q_{i,t}|}{|\mathcal{V}^Q_{i,t}|} > \tau_{\text{ground}} \\ \text{logit}(p_0), & \text{otherwise.} \end{cases} \tag{9}$$

where $\kappa_{\text{inc}} > 1$ is the incremental gain, $p_{\text{inc}} > p_0$ is the update probability, $|\cdot|$ denotes the cardinality of a set, $\mathcal{G}^Q_{i,t}$ denotes the ground points within $\mathcal{V}^Q_{i,t}$, and $\tau_{\text{ground}}$ denotes the ratio threshold that is close to 1. $p_0$ is prior probability, which makes $l\left(\boldsymbol{m}_i \mid \mathbf{z}_t\right) = l(\boldsymbol{m}_i)$ in (5), so update does not happen, i.e. $l\left(\boldsymbol{m}_i \mid \mathbf{z}_{1:t}\right) = l\left(\boldsymbol{m}_i \mid \mathbf{z}_{1:t-1}\right)$, which is represented as white regions in Fig. 5(b).

### C. Dynamic Instance Proposal

After updating the pseudo occupancy grid map, we consider the instances corresponding to the non-ground points in the grid cells with high probabilities as initial dynamic instances $\hat{\mathcal{I}}^{\text{init}}_t$. Formally, we define the indices set of grid cells which correspond to the high probabilities as $\mathcal{R}$:

$$\mathcal{R} = \left\{i \mid p(\boldsymbol{m}_i \mid \mathbf{z}_{1:t}) > p_{\text{rp}}\right\}, \tag{10}$$

where $p_{\text{rp}}$ is a region proposal threshold. $\mathcal{R}$ is expressed as white regions in Fig. 6(b).

Let $f: \mathbf{p} \mapsto i$ be the function that returns an index of the grid cell where a point, $\mathbf{p}$, is located, $\hat{\mathcal{I}}^{\text{init}}_t$ is defined as follows:

$$\hat{\mathcal{I}}^{\text{init}}_t = \left\{\mathcal{S}_{k,t} \in \mathcal{I}^+_t \mid \exists f(\mathbf{p}) \in \mathcal{R},\ \mathbf{p} \in \mathcal{S}_{k,t}\right\}, \tag{11}$$

where $\mathcal{I}^+_t$ is a provided instance set at $t$, except by the unlabeled points $\mathcal{S}_{0,t}$. Meanwhile, unlabeled points that are located in the region proposals are handled separately as

$$\hat{\mathcal{U}}^{\text{init}}_t = \left\{\mathbf{p} \in \mathcal{S}_{0,t} \mid f(\mathbf{p}) \in \mathcal{R}\right\}, \tag{12}$$

where $\mathcal{S}_{0,t}$ denotes all the unlabeled points at $t$. By doing so, the noisy points, which are highlighted in Fig. 3, are filtered out.

However, $\hat{\mathcal{I}}^{\text{init}}_t$ also includes some undesirable static instances, which should not be rejected, because a portion of static instances is located in $\mathcal{R}$, as shown in Fig. 6(c). Therefore, to check whether most parts of the objects are located on grid cells with high probabilities, we propose a novel metric called *dynamic instance score* $\texttt{dyn}(\cdot)$, which is defined as:

$$\texttt{dyn}(\mathcal{S}_{k,t}) = \frac{1}{|\mathcal{S}_{k,t}|} \sum_{\mathbf{p} \in \mathcal{S}_{k,t}} \text{logit}(g(\mathbf{p})), \tag{13}$$

where $g(\cdot)$ denotes a function that adaptively returns a probability of the grid cell corresponding to $\mathbf{p}$ depending on whether the grid cell is updated or not as:

$$g\left(\mathbf{p}\right) = \begin{cases} p(\boldsymbol{m}_i \mid \mathbf{z}_{1:t}), & \text{if } p(\boldsymbol{m}_i \mid \mathbf{z}_{1:t}) > p_0 \\ p_{\text{neg}}, & \text{otherwise} \end{cases} \tag{14}$$

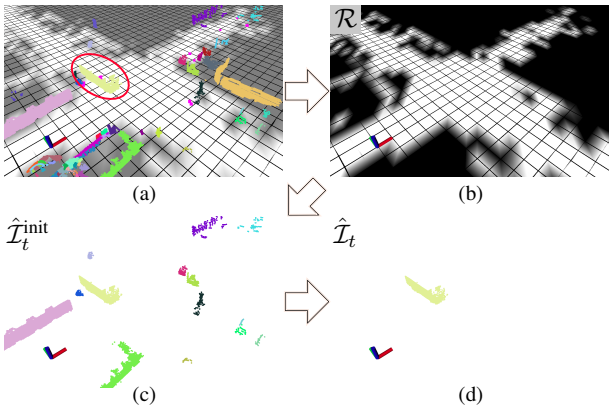



Fig. 6. Procedure of dynamic instance proposal for Seq. 07 around frame 632 on the SemanticKITTI dataset. (a) Estimated instance segmentation and the updated pseudo occupancy grid map. A truck (red circle) is a moving object. (b) Region proposal, $\mathcal{R}$ (white cells). (c) Initial dynamic instances, $\hat{\mathcal{I}}^{\text{init}}_t$, whose portions are in $\mathcal{R}$. (d) Final dynamic instances, $\hat{\mathcal{I}}_t$, which are filtered by our *dynamic instance score*. (best viewed in color).

where $p_{\text{neg}} < 0.5$ is a constant parameter.

The objective of $g(\cdot)$ is to reduce $\texttt{dyn}(\mathcal{S}_{k,t})$ if an instance is static. The cells fully occupied by static instances do not satisfy the aforementioned conditions in Section III.*B*. so these cells are not updated, preserving $p(\boldsymbol{m}_i \mid \mathbf{z}_{1:t})$ as $p_0$. Thus, even if small portions of static objects are in $\mathcal{R}$ so that these instances are wrongly considered as initial dynamic instances, their dynamic instance scores are much lower than those of the actual dynamic instances due to $\text{logit}(p_{\text{neg}}) < 0$, which makes $\texttt{dyn}(\mathcal{S}_{k,t})$ be negative.

Therefore, the final dynamic instances, $\hat{\mathcal{I}}_t$, are selected as follows:

$$\hat{\mathcal{I}}_t = \left\{\mathcal{S}_{k,t} \in \hat{\mathcal{I}}^{\text{init}}_t \mid \text{logit}^{-1}(\texttt{dyn}(\mathcal{S}_{k,t})) > p(\bar{\mathbf{p}}_{k,t})\right\}, \tag{15}$$

where $p(\bar{\mathbf{p}}_{k,t})$ is an adaptive probability threshold depending on the distance between the origin of the body frame and the centroid of the $k$-th segment, $\bar{\mathbf{p}}_{k,t}$. That is, $p(\bar{\mathbf{p}}_{k,t})$ returns $p_{\text{soft}}$ to make the threshold less conservative if an instance is located far from the body frame, and $p_{\text{hard}}$, otherwise, where $p_{\text{soft}} < p_{\text{hard}}$. By doing so, wrongly rejected static instances are successfully reverted, as shown in Fig. 6(d). This adaptive thresholding slightly increases undesirable false positives, but substantially increases the rejection rate of moving objects (see Section IV.*C*).

### D. Under-Segmentation Check

The approach presented so far works well, but we occasionally observe that under-segmentation, which means that at least one static and one dynamic object are segmented together, occasionally triggers false negatives. This phenomenon is illustrated in Figs. 7(a) and 7(b). For instance, once an actual static object with a large size and an actual dynamic object are clustered together, the large static instance makes $\texttt{dyn}(\mathcal{S}_{k,t})$ smaller because the most static points located in the non-updated cells return minus values of log-odds in (13). Consequently, $\texttt{dyn}(\mathcal{S}_{k,t})$ becomes

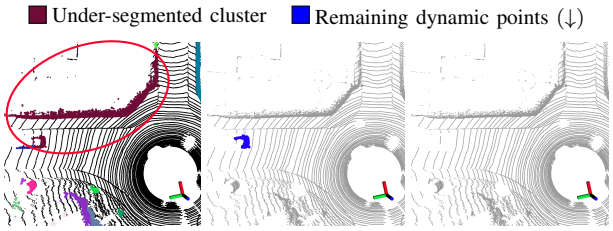



Fig. 7. (a) Example of under-segmentation for Seq. 05 around frame 2,414 on the SemanticKITTI dataset, where different colors indicate different instances. A wall and the back part of a car are segmented as one instance (brown color in red circle). (b)-(c) Before and after the application of our *under-segmentation check*, where blue points denote the false negatives (best viewed in color).

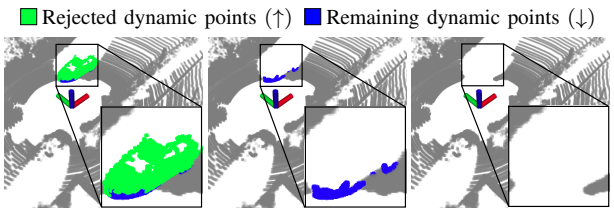



Fig. 8. (a)-(b) Successfully rejected points and partially remaining dynamic points caused by erroneous instance segmentation estimates for Seq. 02 around frame 878 on the SemanticKITTI dataset. The green and blue colors indicate the true positives and false negatives (best viewed in color). (c) After the application of our *volumetric outlier removal* (best viewed in color).

smaller than the adaptive threshold of (15), so the under-segmented clusters with dynamic points are considered to be static, resulting in false negatives.

To resolve this problem, we partially reject the points in the grid cells with definitely high probabilities from the abnormally large instances. Formally, if the following three conditions are satisfied: a) the total area where a potential static instance is occupied is larger than $A_{\text{USC}}$, b) the ratio of the non-updated grids is larger than $\tau_{\text{USC}}$, where $\tau_{\text{USC}}$ is the ratio threshold, and c) other grid cells with definitely high probabilities exist, we segment the partial points located in the grids with high probabilities as a new dynamic instance. By doing so, false negatives caused by under-segmentation can be successfully rejected, as shown in Fig. 7(c). The estimated dynamic points are added to $\hat{\mathcal{I}}_t$.

### E. Volumetric Outlier Removal

Finally, VOR is proposed to reject partially remaining dynamic points caused by the erroneous estimates of the instance and ground segmentation, by leveraging spatio-temporal information. As presented in Figs. 8(a) and 8(b), even though most dynamic instances are rejected by the dynamic instance proposal, some noisy points still remain in the map. Even, some lower parts of dynamic instances are misclassified as ground points in some cases. Therefore, it is necessary to additionally reject dynamic points at the final stage.

The basic idea of VOR is to reject points of the estimated static points neighboring to the estimated dynamic points because the adjacent volumes of dynamic instances are also likely to be occupiable; thus, our VOR works like volumetric erosion. To this end, first, let the estimated dynamic points be $\mathcal{Q}_t$ (green points in Fig. 8(a)) and the estimated static points be $\mathcal{T}_t$, which is the complement of $\mathcal{Q}_t$. Then, the neighboring search, $\mathcal{N}(\cdot)$, to obtain neighboring noisy points is defined as:

$$\mathcal{N}(\mathcal{Q}_t, \mathcal{T}_t, r) = \{\mathbf{p} \in \mathcal{T}_t \mid \min \|\mathbf{p} - \mathbf{q}\| < r\}, \tag{16}$$

where $\mathbf{q} \in \mathcal{Q}_t$ and $r$ denotes a search radius.

$\mathcal{Q}_t$ is set by utilizing dynamic instances on adjacent frames as well as those estimated at time $t$. For each time step, there are two elements: a) the selected dynamic segments $\mathcal{P}_t^{\mathcal{Q}}$, which is expressed as:

$$\mathcal{P}_t^{\mathcal{Q}} = \bigcup_{\mathcal{S}_{k,t} \in \hat{\mathcal{I}}_t} \mathcal{S}_{k,t}[\text{logit}^{-1}(\texttt{dyn}(\mathcal{S}_{k,t})) > p_{\text{v}}], \tag{17}$$

where $[\cdot]$ denotes the Iverson bracket, which is an indicator function returning 1 if the statement inside the bracket is true, and 0, otherwise; $p_{\text{soft}} < p_{\text{v}} < p_{\text{hard}}$ is a probability threshold to select more reliable dynamic segments, and b) the unlabeled points included in (12) and close to $\mathcal{P}_t^{\mathcal{Q}}$, which is defined as $\mathcal{U}_t^{\mathcal{Q}} = \mathcal{N}(\mathcal{P}_t^{\mathcal{Q}}, \hat{\mathcal{U}}_t^{\text{init}}, r_{\text{v}})$, where $r_{\text{v}}$ is defined as $\kappa_{\text{v}} \cdot \xi$; $\kappa_{\text{v}} > 1$ denotes the volumetric gain and $\xi$ denotes the voxel size. Finally, let $t' \in W$ be the window indices set, e.g. if $|W| = 3$, $W = \{t-1, t, t+1\}$, then $\mathcal{Q}_t$ is set as:

$$\mathcal{Q}_t = \bigcup_{t' \in W} \{(\mathbf{T}_t^{w\prime})^{-1} \mathbf{T}_{t'}^{w\prime} \cdot \mathbf{p} \mid \mathbf{p} \in \mathcal{P}_{t'}^{\mathcal{Q}} \bigcup \mathcal{U}_{t'}^{\mathcal{Q}}\}. \tag{18}$$

Next, the estimated static points, $\mathcal{T}_t$, is defined as:

$$\mathcal{T}_t = \bigcup_{\mathcal{S}_{k,t} \in (\mathcal{I}_t^{+} - \hat{\mathcal{I}}_t)} \mathcal{S}_{k,t} + \mathcal{S}_{0,t} - \hat{\mathcal{U}}_t^{\text{init}}, \tag{19}$$

which is the complement of the estimated dynamic points, i.e. cloud points from $\hat{\mathcal{I}}_t$ and $\hat{\mathcal{U}}_t^{\text{init}}$. Consequently, the rejected points, $\hat{\mathcal{P}}_t^{\text{v}}$, are defined as $\hat{\mathcal{P}}_t^{\text{v}} = \mathcal{N}(\mathcal{Q}_t, \mathcal{T}_t, r_{\text{v}})$.

Finally, $\hat{\mathcal{U}}_t$ in (3) is set to $\hat{\mathcal{U}}_t = \hat{\mathcal{P}}_t^{\text{v}} \bigcup \hat{\mathcal{U}}_t^{\text{init}}$. As a result, some remaining dynamic points in the map are successfully rejected, as shown in Fig. 8(c).

## IV. Experimental Evaluation

The main focus of this work is a robust map building approach for the static 3D points that rejects the dynamic points at an instance-level while preserving static points, overcoming the potentially erroneous estimates of instance segmentation.

We present our experiments to show the capabilities of our method. The results of our experiments also support our key claims, which are: our approach (i) precisely rejects dynamic points by using instance segmentation information, (ii) has effective ways that suppress the effect of erroneous estimates of instance segmentation, and (iii) shows superior robustness in a more crowded environment compared to state-of-the-art learning-based methods. Our experimental evaluation backs up these claims.

TABLE I. Parameters of our proposed method.

| Parameter | Value | Description |
|---|---|---|
| $p_0$ | 0.5 | Prior probability |
| $N_{\min}$ | 5 | Minimum number of points for each grid cell |
| $\Delta h_{\min}$ | 0.4 m | Minimum pseudo occupancy |
| $\tau_{\Delta h}$ | 0.2 | Ratio threshold of pseudo occupancy |
| $\tau_{\text{ground}}$ | 0.95 | Percentage of ground points for a query scan |
| $\kappa_{\text{inc}}$ | 2.0 | Incremental gain for update |
| $p_{\text{inc}}$ | 0.5374 | Update probability s.t. $\text{logit}(p_{\text{inc}}) = 0.15$ |
| $p_{\text{rp}}$ | 0.8 | Region proposal threshold |
| $p_{\text{hard}}$ | 0.999999 | Hard threshold of dynamic instance score |
| $p_{\text{soft}}$ | 0.75 | Soft threshold of dynamic instance score |
| $A_{\text{USC}}$ | $56\,\text{m}^2$ | Minimum area threshold of USC |
| $\tau_{\text{USC}}$ | 0.25 | Ratio threshold of non-updated regions |
| $p_{\text{v}}$ | 0.953 | Probability threshold for query cloud of VOR |
| $\kappa_{\text{v}}$ | 1.732 | Voxel gain of VOR |
| $\xi$ | 0.2 m | Voxel size |

### A. Experimental Setup

To perform our analysis, we exploit static map building benchmark, proposed by Lim *et al.* [28]. The benchmark uses the SemanticKITTI dataset [3], [18], which provides point-wise annotations of moving instances. This benchmark consists of five sequences with a large number of dynamic points as follows: Seq. `00` (4,390 - 4,530), Seq. `01` (150 - 250), Seq. `02` (860 - 950), Seq. `05` (2,350 - 2,670), and Seq. `07` (630 - 820), where the numbers in parentheses indicate the start and end frames. These subsequences are the five sequences with the largest number of dynamic objects, so the capability of static map building methods can be effectively evaluated. Note that this benchmark uses the estimated poses by SuMa [4], not ground truth poses, to check the robustness using realistically uncertain poses [23].

The datasets that have point-wise labels [9] have only a few dynamic points, so the phenomenon that different dynamic objects occupy the same space does not frequently occur. Therefore, we manually labeled Seq. `19` of the KITTI object tracking dataset, which contains more moving objects per scan. By doing so, we evaluate the performance of baseline methods and our method in a crowded environment.

For all experiments, we use *Preservation Rate (PR)*, *Rejection Rate (RR)*, and *$F_1$ score* [28] defined as:

- PR: $\frac{\text{\# of preserved static voxels}}{\text{\# of total static voxels on the naively accumulated map}}$,
- RR: $1 - \frac{\text{\# of remaining dynamic voxels}}{\text{\# of total dynamic voxels on the naively accumulated map}}$,
- $\text{F}_1$: $2\text{PR} \cdot \text{RR}/(\text{PR} + \text{RR})$.

See Lim *et al.* [28] for more details on these metrics. In addition, we summarize the parameters of our approach in Table I. Experiments on parameter tuning are provided in the appendix.

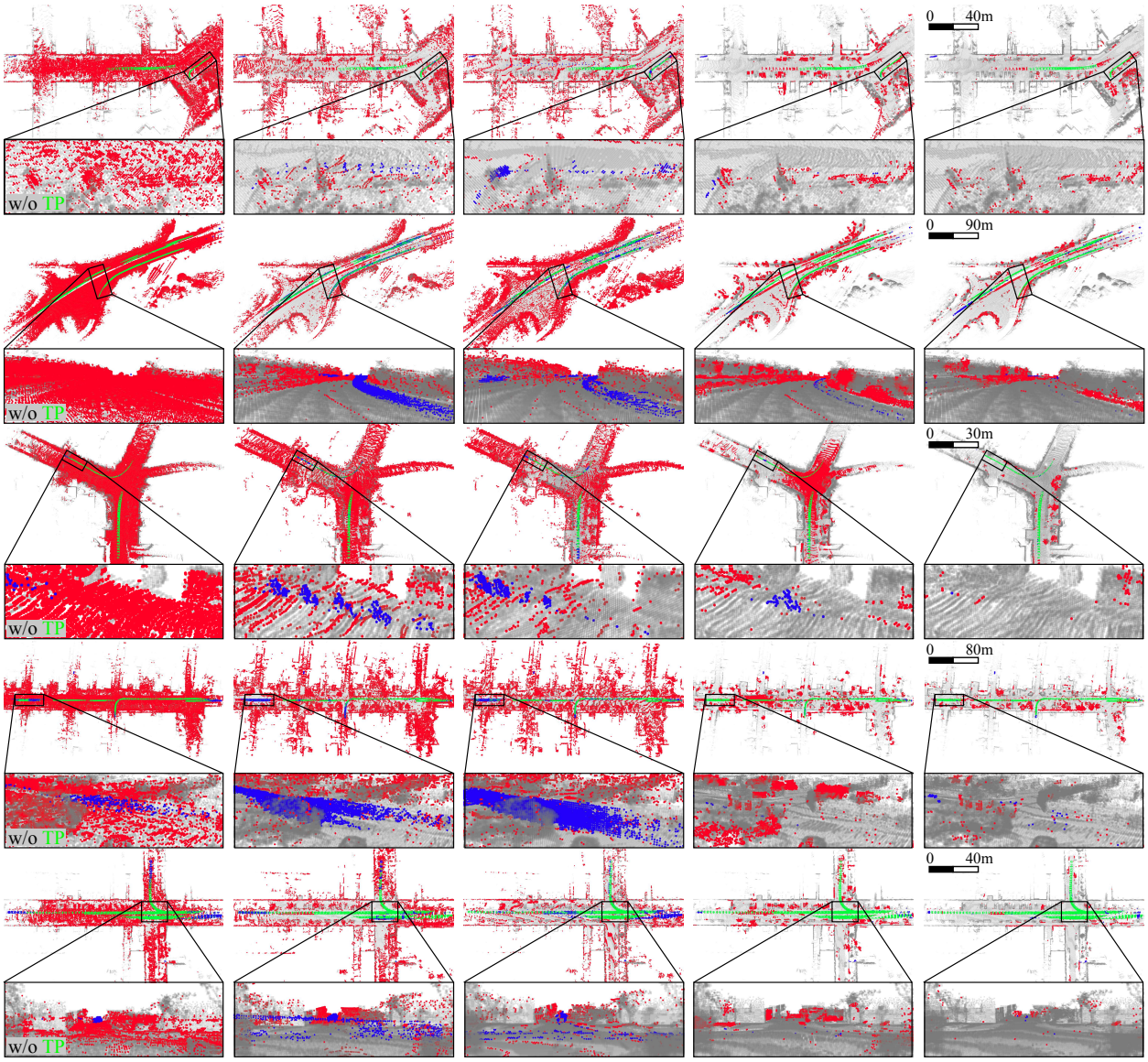
### B. Static Map Building Performance

The first experiment evaluates the performance of our proposed method and baseline methods, supporting the claim that our approach precisely rejects dynamic points while preserving static points as many as possible by leveraging instance segmentation information. For our comparison, we used the following baseline methods: OctoMap [21] that employs a clamping technique [54] with voxel size 0.05 and 0.2; Peopleremover [44], Removert [23], where `RM3` denotes the results after three Removal stages with resolutions of 0.4°, 0.45°, and 0.5° when projecting points into the range images. `RM3-RV1` refers to the result of `RM3` followed by a Revert stage with the resolution of 0.55°; DynamicFilter [15], Auto-MOS [12], ERASOR [28], and Park *et al.* [38]. We use the results or default parameters provided by the authors.

TABLE II. Comparison with state-of-the-art methods on the static map benchmark using the SemanticKITTI dataset, which is proposed in [28] (PR: Preservation Rate, RR: Rejection Rate).

| Seq. | Method | PR [%] | RR [%] | $\text{F}_1$ score |
|---|---|---|---|---|
| `00` | OctoMap - 0.05 [21] | 76.731 | 99.124 | 0.865 |
| | OctoMap - 0.2 [21] | 34.568 | **99.979** | 0.514 |
| | Peopleremover [44] | 37.523 | 89.116 | 0.528 |
| | Removert - `RM3` [23] | 85.502 | 99.354 | 0.919 |
| | Removert - `RM3+RV1` [23] | 86.829 | 90.617 | 0.887 |
| | DynamicFilter [15] | 90.070 | 91.090 | 0.906 |
| | Auto-MOS [12] | 86.164 | 94.329 | 0.901 |
| | ERASOR [28] | 93.980 | 97.081 | 0.955 |
| | Park *et al.* [38] | 94.050 | 97.190 | 0.956 |
| | ERASOR2 (Proposed) | **98.788** | 98.582 | **0.987** |
| `01` | OctoMap - `0.05` [21] | 53.163 | 99.663 | 0.693 |
| | OctoMap - `0.2` [21] | 20.777 | **99.863** | 0.344 |
| | Peopleremover [44] | 36.349 | 93.116 | 0.523 |
| | Removert - `RM3` [23] | 94.221 | 93.608 | 0.939 |
| | Removert - `RM3+RV1` [23] | 95.815 | 57.077 | 0.715 |
| | DynamicFilter [15] | 87.950 | 87.690 | 0.878 |
| | Auto-MOS [12] | 89.597 | 90.130 | 0.899 |
| | ERASOR [28] | 91.487 | 95.383 | 0.934 |
| | Park *et al.* [38] | 91.815 | 94.096 | 0.929 |
| | ERASOR2 (Proposed) | **96.879** | 94.629 | **0.957** |
| `02` | OctoMap - `0.05` [21] | 54.112 | 98.769 | 0.699 |
| | OctoMap - `0.2` [21] | 23.746 | **99.792** | 0.384 |
| | Peopleremover [44] | 29.037 | 94.527 | 0.444 |
| | Removert - `RM3` [23] | 76.319 | 96.799 | 0.853 |
| | Removert - `RM3+RV1` [23] | 83.293 | 88.371 | 0.858 |
| | DynamicFilter [15] | 88.020 | 86.100 | 0.871 |
| | Auto-MOS [12] | 88.539 | 93.352 | 0.909 |
| | ERASOR [28] | 87.731 | 97.008 | 0.921 |
| | Park *et al.* [38] | 91.208 | 95.510 | 0.933 |
| | ERASOR2 (Proposed) | **98.523** | 99.709 | **0.991** |
| `05` | OctoMap - `0.05` [21] | 76.341 | 96.785 | 0.854 |
| | OctoMap - `0.2` [21] | 33.904 | **99.882** | 0.506 |
| | Peopleremover [44] | 38.495 | 90.631 | 0.540 |
| | Removert - `RM3` [23] | 86.900 | 87.880 | 0.874 |
| | Removert - `RM3+RV1` [23] | 88.170 | 79.981 | 0.839 |
| | DynamicFilter [15] | 90.170 | 84.650 | 0.873 |
| | Auto-MOS [12] | 87.931 | 72.333 | 0.794 |
| | ERASOR [28] | 88.730 | 98.262 | 0.933 |
| | Park *et al.* [38] | 93.820 | 95.740 | 0.947 |
| | ERASOR2 (Proposed) | **97.582** | 98.992 | **0.983** |
| `07` | OctoMap - `0.05` [21] | 77.838 | 96.938 | 0.863 |
| | OctoMap - `0.2` [21] | 38.183 | **99.565** | 0.552 |
| | Peopleremover [44] | 34.772 | 91.983 | 0.505 |
| | Removert - `RM3` [23] | 80.689 | 98.822 | 0.888 |
| | Removert - `RM3+RV1` [23] | 82.038 | 95.504 | 0.883 |
| | DynamicFilter [15] | 87.940 | 86.800 | 0.874 |
| | Auto-MOS [12] | 86.641 | 91.785 | 0.891 |
| | ERASOR [28] | 90.624 | 99.271 | 0.948 |
| | Park *et al.* [38] | 91.146 | 97.284 | 0.941 |
| | ERASOR2 (Proposed) | **98.977** | 98.459 | **0.987** |

The comparison of such baseline to our method is shown in Table II and Fig. 9. The state-of-the-art methods provided an accurate static map, filtering out over 90 % of dynamic points. Our proposed method exhibits noticeable improvements in both PR and RR, showing the highest $\text{F}_1$ scores. In particular, our proposed method even rejected the lower part of the dynamic instances more clearly (the 2nd row, Fig. 9) while losing fewer static points compared to ERASOR.




(a) OctoMap [21] (b) Removert [23] (c) Auto-MOS [12] (d) ERASOR [28] (e) ERASOR2 (Proposed)

Fig. 9. Comparison of static map generation results produced by state-of-the-art methods and our proposed method on Semantic KITTI dataset (T-B): Seqs. 00, 01, 02, 05, and 07. The zoomed views are the actual remaining points, so true positives are omitted. Green, red, and blue points indicate true positives, false positives and false negatives, respectively, and the fewer red and blue points there are, the better (best viewed in color).

OctoMap and Removert lost many static points while leaving dynamic points on the top part of the bus (the 4th row, Fig. 9). This is because, from the query coordinate, behind the upper part of the large object is free space, so there is no point to check hit or miss along the ray or range discrepancy between the query and map cloud. In addition, Auto-MOS, which is a tracking-based MOS method, also failed because tracking algorithms are highly sensitive to the parameters setting [53], [55], so tracking usually fails to track when too large or too small moving objects were detected. For these reasons, it was demonstrated that our proposed method showed better performance compared with ray tracing methods, visibility-based methods, and MOS-based methods.

The cause of lower RR in Seq. 01 and Seq. 07 compared with ERASOR is because the probability values of the pseudo occupancy grid behind the median barriers and end parts of a map were less observed, so these values were not sufficiently updated. However, additional sensor measurements from the opposite lanes would easily overcome this limitation.

We conclude that our instance-aware dynamic object removal is effective for accurate static map building compared with existing methods.

### C. Robustness Against Noise in Instance Segmentation

The second experiment evaluates the ability of our method to deal with noisy instance segmentation. As shown in Table III, the performance of our approach does not degrade significantly even if noisy instance segmentation is used. According to Nunes *et al.* [34], it was shown that the instance segmentation of the approach [34] for known objects is

TABLE III. Performance comparison with different instance segmentation methods.

| Seq. | Instance seg. method | PR [%] | RR [%] | $F_1$ score |
|---|---|---|---|---|
| 00 | HDBSCAN [10] | 98.649 | **98.582** | 0.986 |
| | Nunes *et al.* [34] | **98.788** | **98.582** | **0.987** |
| 01 | HDBSCAN [10] | 93.554 | **94.951** | 0.943 |
| | Nunes *et al.* [34] | **96.879** | 94.629 | **0.957** |
| 02 | HDBSCAN [10] | 98.339 | **99.709** | 0.990 |
| | Nunes *et al.* [34] | **98.523** | **99.709** | **0.991** |
| 05 | HDBSCAN [10] | 97.473 | **99.113** | **0.983** |
| | Nunes *et al.* [34] | **97.582** | 98.992 | **0.983** |
| 07 | HDBSCAN [10] | 98.767 | **98.800** | **0.988** |
| | Nunes *et al.* [34] | **98.973** | 98.459 | 0.987 |

substantially better than HDBSCAN [10]. Nevertheless, note that no matter what instance segmentation method was used, our proposed method showed similar performance.

From Seq. 01, we also notice how our instance-aware approach helps not only on removing dynamic objects but also on preserving static points. Because Seq. 01 is recorded in a highway scene, lots of bushy vegetation are represented as noisy points. When considering the more precise instance predictions from Nunes *et al*. [34], the PR is increased because more complete instance information is provided. Therefore, the noisy points from static bushes can be correctly preserved.

To further support our claim and examine the effectiveness of each module more closely, we conduct an ablation study, as shown in Table IV. There was a trade-off between PR and RR, yet our proposed modules showed a greater improvement in the RR than a decrease in the PR. In particular, our USC and VOR significantly increased RR for the following reasons: USC enables rejection of a portion of dynamic points clustered with large static instances and VOR rejects some false negative points neighboring to the estimated dynamic points, as presented in Figs. 7 and 8.

Therefore, these results support our claim that robust modules, USC and VOR, against erroneous instance segmentation estimates allow the performance of our approach to be less variant to the quality of instance segmentation.

### *D. Static Map Building Performance in Highly Crowded Environments*

This subsection backs up our third claim that the proposed method is more robust than existing methods, also in crowded environments. As shown in Table V and Fig. 10, our proposed method showed promising performance compared with other state-of-the-art methods. In particular, Removert did not remove the traces of pedestrians successfully and lost static points all over the region (Fig. 10(a)). ERASOR showed better dynamic object removal performance, yet ERASOR created some holes in the map, i.e., some ground points and a portion of walls are also removed (Fig. 10(b)). In contrast, our proposed method successfully rejected dynamic points by fusing all the predictions about moving objects to the pseudo occupancy grid map, while losing fewer static points than Removert and ERASOR (Fig. 10(d)).

TABLE IV. Ablation study: performance according to the presence or absence of each component of our proposed method on Seq. 05 of the SemanticKITTI dataset (USC: Under-segmentation check, VOR: Volumetric outlier removal).

| Adaptive thresh. (Section III.*C*) | USC (Section III.*D*) | VOR (Section III.*E*) | PR [%] | RR [%] | $F_1$ |
|---|---|---|---|---|---|
| | | | **99.397** | 94.480 | 0.969 |
| ✓ | | | 98.335 | 95.840 | 0.971 |
| ✓ | ✓ | | 98.326 | 96.686 | 0.975 |
| ✓ | ✓ | ✓ | 97.582 | **98.992** | **0.983** |

TABLE V. Comparison with state-of-the-art methods on the KITTI tracking 19 dataset.

| Method | PR [%] | RR [%] | $F_1$ score |
|---|---|---|---|
| 4DMOS | **99.602** | 78.209 | 0.876 |
| Removert [23] | 80.622 | 88.394 | 0.843 |
| ERASOR [28] | 75.411 | 89.782 | 0.820 |
| ERASOR2 (Proposed) | 94.316 | **93.162** | **0.937** |

TABLE VI. Runtime per iteration on Seq.01 of SemanticKITTI dataset on Intel(R) Core(TM) i7-7700K CPU.

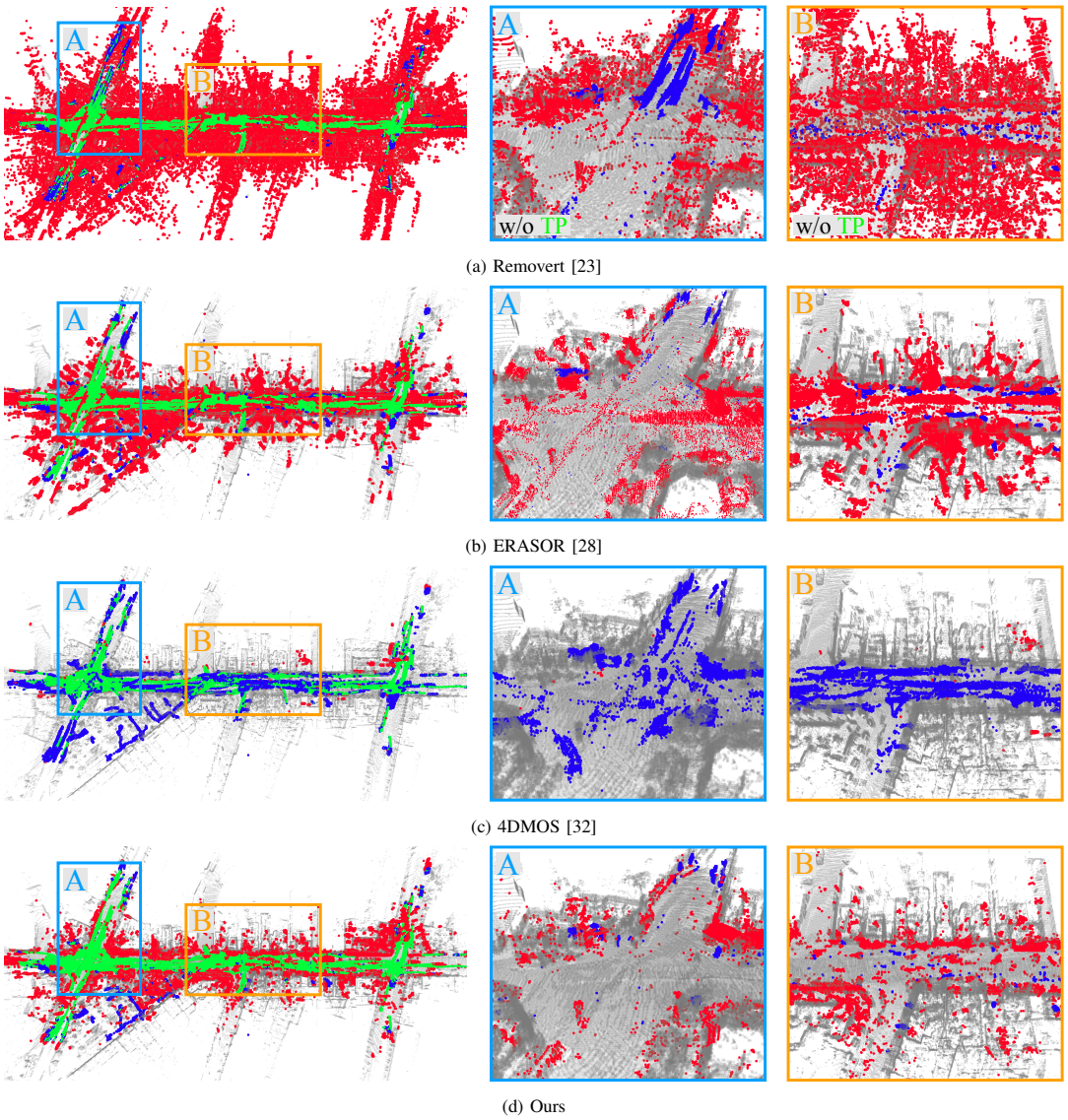
| Method | Runtime/# of iterations [s] |
|---|---|
| OctoMap [21] | 1.077 |
| Peopleremover [44] | 1,000 |
| Removert [23] | 0.8307 |
| ERASOR [28] | **0.0732** |
| ERASOR2 (Proposed) | 0.2034 |

We also compare our proposed method with the state-of-the-art learning-based MOS method, *4DMOS* [32]. That is, we generate the map by only using the estimated static points by 4DMOS. One interesting thing is that the MOS methods care more about precision, thus being too conservative and leading to more false negatives. For this reason, 4DMOS showed the highest PR, but its RR is substantially lower than existing static map building methods, as shown in Table V. In particular, naively accumulating the results of MOS cannot cover some false negatives at $t$. Thus, these false negatives directly degrade the quality of the estimated static map, as shown in Fig. 10(c).

### *E. Runtime of Our Approach*

Finally, we investigate the runtime of our approach. The runtime per scan of the static map building methods was analyzed in Seq. 01, which is the largest scale, so the speed difference becomes more noticeable depending on how map cloud is handled and geometrical discrepancies are estimated.

As presented in Table VI, our approach runs fast at around 0.2 s per scan. More specifically, pseudo occupancy grid map update, dynamic instance proposal, and the other refinement stages take 6.6 ms, 52.6 ms, and 144.2 ms, respectively. It takes some time to perform VOR because we use a K-d tree for searching neighboring points as in (16). Nevertheless, static map building is often used as a post-processing before the localization or navigation, which is an offline process, so the runtime is not very important. Nevertheless, the sufficiently fast speed implies that our proposed method efficiently builds static maps.




(a) Removert [23]

(b) ERASOR [28]

(c) 4DMOS [32]

(d) Ours

Fig. 10. Comparison of static map generation results produced by state-of-the-art methods and our proposed method on KITTI tracking `19` dataset. The zoomed views are the actual remaining points, so true positives are omitted. Green, red, and blue points indicate true positives, false positives and false negatives, respectively, and the fewer red and blue points there are, the better (best viewed in color).

## V. Conclusion

In this paper, we proposed a novel instance-aware static map building method, called *ERASOR2*. By exploiting instance segmentation estimates, our proposed method can precisely reject more dynamic points at an instance-level while preserving most static points. In particular, we proposed pseudo occupancy grid map update to estimate the regions that contain the traces of moving objects in the map. Furthermore, we present novel ways to deal with noisy instance segmentation estimates, which allow us to provide high performance, overcoming somewhat imprecise instance segmentation. Consequently, our proposed method shows promising performance compared with existing conventional and deep learning-based methods. Finally, all claims made in the paper have been experimentally supported.

## References

[1] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss. Mapping the static parts of dynamic scenes from 3D LiDAR point clouds exploiting ground segmentation. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, pages 1–6, 2021.

[2] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss. Static map generation from 3D LiDAR point clouds exploiting ground segmentation. *Journal on Robotics and Autonomous Systems (RAS)*, 159:104287–104294, 2023.

[3] 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 *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 9297–9307, 2019.

[4] J. Behley and C. Stachniss. Efficient surfel-based SLAM using 3D laser range data in urban environments. In *Proc. of Robotics: Science and Systems (RSS)*, pages 59–68, 2018.

[5] J. Behley, V. Steinhage, and A.B. Cremers. Efficient radius neighbor search in three-dimensional point clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3625–3630, 2015.

[6] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *Intl. Journal of Robotics Research (IJRR)*, 24(1):31–48, 2005.

[7] M. Bennewitz, W. Burgard, and S. Thrun. Adapting navigation strategies using motions patterns of people. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 2000–2005, 2003.

[8] F. Blochliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart. Topomap: Topological mapping and navigation based on visual SLAM maps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3818–3825, 2018.

[9] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020.

[10] R.J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Proc. of Pacific-Asia. Conf. on Knowledge Discovery and Data Mining*, pages 160–172, 2013.

[11] 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 (RA-L)*, 6:6529–6536, 2021.

[12] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss. Automatic labeling to generate training data for online LiDAR-based moving object segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):6107–6114, 2022.

[13] A. Dewan, T. Caselitz, G.D. Tipaldi, and W. Burgard. Motion-based detection and tracking in 3D LiDAR scans. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 4508–4513, 2016.

[14] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[15] T. Fan, B. Shen, H. Chen, W. Zhang, and J. Pan. DynamicFilter: An online dynamic objects removal framework for highly dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 7988–7994, 2022.

[16] M. Fehr, F. Furrer, I. Dryanovski, J. Sturm, I. Gilitschenski, R. Siegwart, and C. Cadena. TSDF-based change detection for consistent long-term dense reconstruction and dynamic object discovery. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 5237–5244, 2017.

[17] H. Fu, H. Xue, and G. Xie. MapCleaner: Efficiently removing moving objects from point cloud maps in autonomous driving scenarios. *Remote Sensing*, 14(18):4496, 2022.

[18] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[19] D. Hahnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 496–501, 2002.

[20] M. Henein, G. Kennedy, V. Ila, and R. Mahony. Simultaneous localization and mapping with dynamic rigid objects. *arXiv preprint*, arXiv:1805.03800, 2018.

[21] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on Octrees. *Autonomous Robots*, 34:189–206, 2013.

[22] C. Jiang, D.P. Paudel, Y. Fougerolle, D. Fofi, and C. Demonceaux. Static-map and dynamic object reconstruction in outdoor scenes using 3-D motion segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):324–331, 2016.

[23] G. Kim and A. Kim. Remove, then revert: Static point cloud map construction using multiresolution range images. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 10758–10765, 2020.

[24] G. Kim, B. Park, and A. Kim. 1-day learning, 1-year localization: Long-term LiDAR localization using scan context image. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1948–1955, 2019.

[25] J. Kim, J. Woo, and S. Im. RVMOS: Range-view moving object segmentation leveraged by semantic and motion features. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):8044–8051, 2022.

[26] J. Kim and W. Chung. Robust localization of mobile robots considering reliability of LiDAR measurements. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 6491–6496, 2018.

[27] T. Kühner and J. Kümmerle. Large-scale volumetric scene reconstruction using LiDAR. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 6261–6267, 2020.

[28] 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 (RA-L)*, 6(2):2272–2279, 2021.

[29] 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 (RA-L)*, 6(4):6458–6465, 2021.

[30] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[31] M. Luber, J.A. Stork, G.D. Tipaldi, and K.O. Arras. People tracking with human motion predictions from social forces. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 464–469, 2010.

[32] 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 (RA-L)*, 7(3):7503–7510, 2022.

[33] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena. C-blox: A scalable and consistent TSDF-based dense mapping approach. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 995–1002, 2018.

[34] L. Nunes, X. Chen, R. Marcuzzi, A. Osep, L. Leal-Taixé, C. Stachniss, and J. Behley. Unsupervised class-agnostic instance segmentation of 3D LiDAR data for autonomous vehicles. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):8713–8720, 2022.

[35] M. Oh, E. Jung, H. Lim, W. Song, S. Hu, E.M. Lee, J. Park, J. Kim, J. Lee, and H. Myung. TRAVEL: Traversable ground and above-ground object segmentation using graph representation of 3D LiDAR scans. *IEEE Robotics and Automation Letters (RA-L)*, pages 7255–7262, 2022.

[36] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and G. Yalla. Robust method for removing dynamic objects from point clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 10765–10771, 2020.

[37] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[38] J. Park, Y. Cho, and Y.S. Shin. Nonparametric background model-based LiDAR SLAM in highly dynamic urban environments. *IEEE Trans. on Intelligent Transportation Systems (ITS)*, 23(12):24190–24205, 2022.

[39] P. Pfreundschuh, H.F. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart, and A. Cramariuc. Dynamic object aware LiDAR SLAM based on automatic generation of training data. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 11641–11647, 2021.

[40] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart. Long-term 3D map maintenance in dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3712–3719, 2014.

[41] C. Qian, Z. Xiang, Z. Wu, and H. Sun. RF-LIO: Removal-first tightly-coupled LiDAR inertial odometry in high dynamic environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4421–4428, 2021.

[42] P. Ruchti and W. Burgard. Mapping with dynamic-object probabilities calculated from single 3D range scans. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 6331–6336, 2018.

[43] P.E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12716–12725, 2019.

[44] J. Schauer and A. Nüchter. The Peopleremover—Removing dynamic objects from 3D point cloud data by traversing a voxel occupancy grid. *IEEE Robotics and Automation Letters (RA-L)*, 3(3):1679–1686, 2018.

[45] S. Song, H. Lim, A.J. Lee, and H. Myung. DynaVINS: A visual-inertial SLAM for dynamic environments. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):11523–11530, 2022.

[46] C. Stachniss. *Robotic Mapping and Exploration*, volume 55. 2009.

[47] C. Stachniss and W. Burgard. Mobile robot mapping and localization in non-static environments. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 1324–1329, 2005.

[48] 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 *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 11456–11463, 2022.

[49] C. Sung, S. Jeon, H. Lim, and H. Myung. What if there was no revisit? Large-scale graph-based SLAM with traffic sign detection in an HD map using LiDAR inertial odometry. *Intelligent Service Robotics (ISR)*, 15(2):161–170, 2022.

[50] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V.A. Prisacariu, O. Kähler, D.W. Murray, S. Izadi, P. Pérez, et al. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 75–82, 2015.

[51] I. Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss. Poisson surface reconstruction for LiDAR odometry and mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 5624–5630, 2021.

[52] I. Vizzo, T. Guadagnino, J. Behley, and C. Stachniss. VDBFusion: Flexible and efficient TSDF integration of range sensor data. *Sensors*, 22(3):1296–1320, 2022.

[53] X. Weng, J. Wang, D. Held, and K. Kitani. 3D multi-object tracking: A baseline and new evaluation metrics. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 10359–10366, 2020.

[54] M. Yguel, O. Aycard, and C. Laugier. Update policy of dense maps: Efficient algorithms and sparse representation. In *Proc. of the Intl. Conf. on Field and Service Robotics (FSR)*, pages 23–33, 2008.

[55] M. Yokozuka, K. Koide, S. Oishi, and A. Banno. LiTAMIN: LiDAR-based tracking and mapping by stabilized ICP for geometry approximation with normal distributions. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5143–5150, 2020.

[56] D. Yoon, T. Tang, and T. Barfoot. Mapless online detection of dynamic objects in 3D LiDAR. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, pages 113–120, 2019.

## APPENDIX A
## PERFORMANCE CHANGES WITH DIFFERENT PARAMETERS

Among the parameters in Table I, we stress that $\kappa_{\text{inc}}$ and $p_{\text{inc}}$ should be set appropriately. As shown in Fig. A1, there is a trade-off between PR and RR, so two parameters directly affect the performance. For instance, if both parameters are too large, too many region proposals occur, so false positives increase, resulting in a lower preservation rate and a high rejection rate. In contrast, if both parameters are too small, i.e. when $\kappa_{\text{inc}} = 1.0$ and $\text{logit}(p_{\text{inc}}) = 0.05$, the probabilities of the grids are not allowed to be over $p_{\text{rp}}$, leading to more false negatives. As a result, dynamic points are not successfully filtered out, showing low rejection rate as 32.93 %. Except in that case, our method successfully preserves static points ($> 98\%$) and rejects dynamic points ($> 90\%$) even though parameters are changed.

## APPENDIX B
## PERFORMANCE CHANGES WITH DIFFERENT WINDOW SIZES

Next, we present performance changes with different window sizes. If the window size $W$ (see Section III.*E*) becomes large, more points are likely to be wrongly rejected, so the preservation rate slightly decreases and rejection rate increases, and vice versa. In Seqs. `00`, `02`, `05`, and `07`, dynamic points from moving objects are already successfully rejected ($> 98\%$), so the performance change with different window sizes is insignificant. But in more crowded environments, i.e. Seqs. `01` and `19`, the performance change

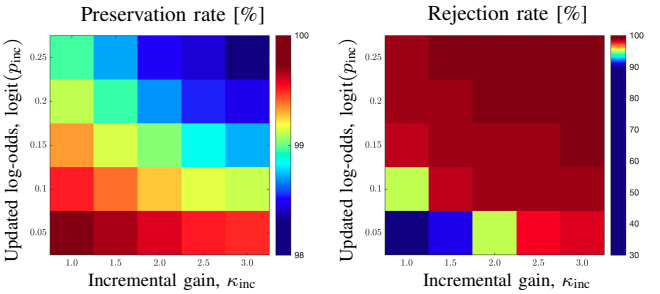



Fig. A1. Preservation and rejection rates for Seq. `07` on the SemanticKITTI dataset depending on $\kappa_{\text{inc}}$ and $\text{logit}(p_{\text{inc}})$. If both parameters are too small, dynamic points are not successfully filtered out, showing low rejection rate, e.g. when $\kappa_{\text{inc}} = 1.0$ and $\text{logit}(p_{\text{inc}}) = 0.05$, rejection rate is 32.93 % (best viewed in color).

of rejection rate becomes significant because moving objects frequently re-occupy the spaces that were occupied by another moving object just before. However, if $W$ is set to be too large, the performance of the preservation rate becomes worse because the number of points to be preserved is wrongly rejected. Based on these observations, we set $W = 3$ as a moderate value.

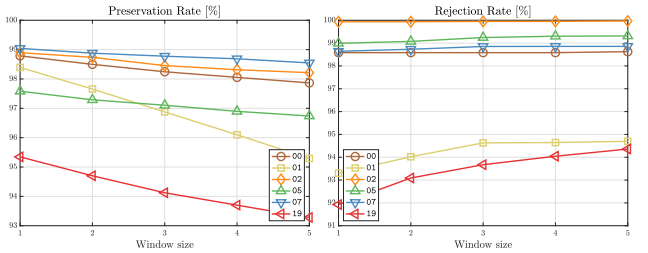



Fig. A2. Preservation and rejection rates on the SemanticKITTI dataset depending on window size $W$.