ERASOR: Egocentric Ratio of Pseudo Occupancy-based Dynamic Object Removal for Static 3D Point Cloud Map Building

Hyungtae Lim[1], *Student Member, IEEE*, Sungwon Hwang[2], and Hyun Myung[1], *Senior Member, IEEE*

Abstract— Scan data of urban environments often include representations of dynamic objects, such as vehicles, pedestrians, and so forth. However, when it comes to constructing a 3D point cloud map with sequential accumulations of the scan data, the dynamic objects often leave unwanted traces in the map. These traces of dynamic objects act as obstacles and thus impede mobile vehicles from achieving good localization and navigation performances. To tackle the problem, this paper presents a novel static map building method called *ERASOR*, Egocentric RAtio of pSeudo Occupancy-based dynamic object Removal, which is fast and robust to motion ambiguity. Our approach directs its attention to the nature of most dynamic objects in urban environments being inevitably in contact with the ground. Accordingly, we propose the novel concept called *pseudo occupancy* to express the occupancy of unit space and then discriminate spaces of varying occupancy. Finally, Region-wise Ground Plane Fitting (R-GPF) is adopted to distinguish static points from dynamic points within the candidate bins that potentially contain dynamic points. As experimentally verified on SemanticKITTI, our proposed method yields promising performance against state-of-the-art methods overcoming the limitations of existing ray tracing-based and visibility-based methods.

I. INTRODUCTION AND RELATED WORKS

For most mobile platforms such as Unmanned Ground Vehicles (UGVs), Unmanned Aerial Vehicles (UAVs), or autonomous cars, retrieving a point cloud map is essential in achieving long-term autonomy [1]–[10]. Robots may utilize static information of their neighboring geometrical structure provided by the map to achieve successful localization or navigation. The map can be represented in many different forms, such as feature map-based representation [11], occupancy-based representation [10], [12], or structural-variance-robust place representation [3]. This paper limits the scope of the map to a 3D point cloud map, which is generated by accumulations of laser scans obtained by a three-dimensional LiDAR sensor.

Unfortunately, because scan data presents a snapshot of the surroundings, scan data in urban environments inevitably

includes representations of dynamic objects, such as vehicles, pedestrians, and so forth [13]–[16]. Moreover, because a 3D point cloud map is the product of sequential accumulations of the scan data, there might be traces of dynamic objects, or the *ghost trail effect* [8], [13], as shown in Fig. 1 on the left. These traces of dynamic objects act as obstacles in the map and thus impede mobile robots from exhibiting good localization and navigation performance.
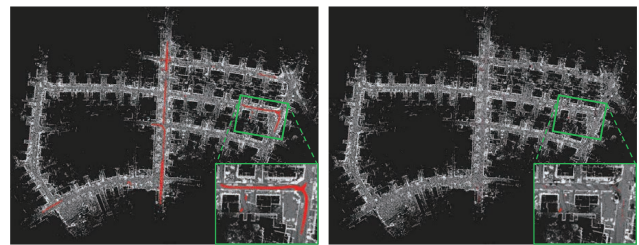

Fig. 1: Before and after the application of our proposed method called *ERASOR* on sequence 05 of SemanticKITTI. Dynamic objects are highlighted in red (best viewed in color).

Conventional methods that tackle this problem are mainly divided into two categories: a) rejection of dynamic objects at the time of map generation [16], [17] and b) post-rejection of dynamic objects with a complete set of generated map [7], [8], [15]. The methodology presented in this paper focuses on the latter case. Accordingly, the latter category can be further classified into three main methodologies: a) segmentation-based, b) ray tracing-based, and c) visibility-based methods.

First of all, segmentation-based methods are usually based on clustering [16], [18], [19]. Litomisky and Bhanu [18] discriminated dynamic clusters among static clusters by using the Viewpoint Feature Histogram (VFH) and Yin *et al.* [19] utilized feature matching to check false correspondences and then used them as seeds to extract clusters from dynamic objects. Yoon *et al.* [16] proposed a region growing-based method.

In addition, the deep learning-based semantic segmentation or detection has led to additional significant improvement in performance [20]. Given correct segmented points with dynamic labels, a map can be straightforwardly constructed by excluding the predicted dynamic points [15], [21]. However, existing segmentation-based approaches only focus on the dynamic points rejection at the moment and not show the final static map. Besides, these are heavily rely on the supervised labels; therefore, they are vulnerable to labeling errors or dynamic objects of unlabeled classes [22]. To mitigate this problem, considering the scan-to-map relationship is necessary.

Occupancy grid [17] and *Octomap* [12] are typical methods that use ray tracing, which count hits and misses of scans in the gridmap space and decide the occupancy of the space. By extension, Schauer and Nüchter [14] proposed the removal of dynamic points by traversing a voxel occupancy grid and Pagad *et al.* [13] suggested a combinination of object detection and Octomap. However, ray tracing-based methods are computationally expensive, which led to the introduction of visibility-based methods to reduce the computational cost [8], [10], [15].

Visibility-based methods are based on the premise that if a query point is observed behind a previously acquired point in the map, then the previously acquired point is dynamic [8], [15], [16]. However, this assumption has *incidence angle ambiguity*; as the range measurement from the ground becomes longer, the incidence angle of a point becomes more ambiguous, so the neighboring ground points may be falsely considered as dynamic objects (Fig. 2(a)). To resolve the problem, Pomerleau *et al.* [8] utilized a Bayesian approach using normal and incidence angles to represent the state for each point. By extension, Kim and Kim [15] proposed a pixel-to-window comparison method to take incidence angle ambiguity into account. However, all previous works struggled with the occlusion issue (Fig. 2(b)) and it was not possible to remove the dynamic points of huge dynamic objects once static points are not observed behind the dynamic points (Fig. 2(c)).

In this paper, we propose a novel static map building method, referred to as Egocentric RAtio of pSeudo Occupancy-based dynamic object Removal (ERASOR), to overcome the limitations of the aforementioned methods. Our method is a visibility-free approach and leverages our proposed representation of points organized in vertical columns called *pseudo occupancy*. The contribution of this paper is fourfold:

- We propose a fast and robust method called *Scan Ratio Test* (SRT) to fetch bins which contain dynamic points based on the nature of most dynamic objects in urban environments, such as terrestrial vehicles and pedestrians, being inevitably in contact with the ground.
- After fetching the bins, we apply *Region-wise Ground Plane Fitting* (R-GPF), a novel static point retrieval method with low computational load. R-GPF overcomes the potential limitations of ray tracing-based methods and visibility-based methods, which are presented in Fig. 2 with details.
- Validations against other state-of-the-art methods are conducted. To the best of our knowledge, it is the first attempt to conduct quantitative comparisons with other state-of-the-art methods on SemanticKITTI [5]. In particular, we point out the limitation of precision/recall metrics and present alternative metrics suitable for static map building tasks: *Preservation Rate* and *Rejection Rate*.
- Our proposed method shows a promising performance over state-of-the-art methods. In particular, ERASOR removes dynamic points with the least loss of static

points and it is at least ten times faster than state-of-the-art methods.

The rest of the paper is organized as follows: Section II explains the concept of pseudo occupancy and our proposed method. Section III describes the experiments and novel error metrics, and the experimental results are discussed in Section IV. Finally, Section V summarizes our contributions and describes future works.

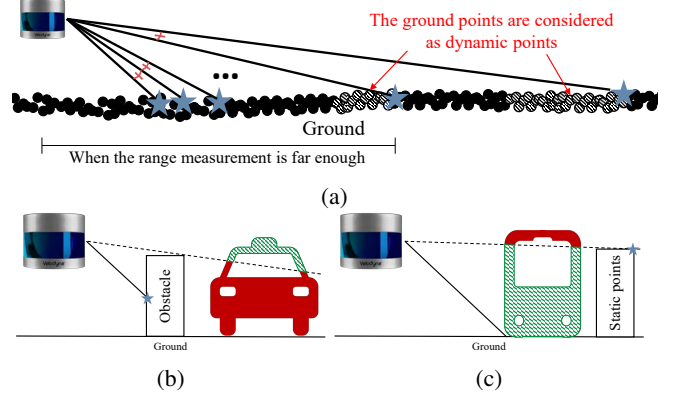## II. STATIC MAP BUILDING VIA ERASOR

Fig. 2: Potential limitations of the ray tracing-based methods and visibility-based methods. (a) Incidence angle ambiguity (b) Occlusion (c) Irremovable dynamic points when there are no static points behind them. The highlighted parts in green are erasable by previous methods, and parts in red are not (best viewed in color).

The schematic diagram of our proposed method on static map building is illustrated in Fig. 3. The following paragraphs highlight the problem definition and the reasoning behind each module of ERASOR. Note that this paper mainly focuses on instance-level dynamic object rejection for static map building. That is, large-scale changes in urban environments, such as the redevelopment or restoration of buildings, are beyond the scope of this work.

A. Problem Definition

Let $\mathcal{P}_t = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ be a set of point cloud that contains n points at time step t, where each point is expressed as $\mathbf{p}_k = \{x_k, y_k, z_k\}$ in Cartesian coordinates. Let $\mathcal{P}_t^Q$ be the query point cloud on the query frame and $_Q^W\mathbf{T}_t$ be SE(3) pose associated with $\mathcal{P}_t^Q$. In this paper, it is assumed that the set of poses are given after optimization or correction through registration. By letting $_Q^W\mathbf{T}_t * \mathcal{P}_t^Q$ be transformed representation of $\mathcal{P}_t^Q$ in the world frame W, the prior map built with a set of raw LiDAR scans, $\mathcal{M}$, can be formulated as follows:

$$\mathcal{M} = \bigcup_{t \in [T]} {}_Q^W\mathbf{T}_t * \mathcal{P}_t^Q \tag{1}$$

where T is the total time step and $[T]$ is equal to $\{1, 2, \ldots, T-1, T\}$. Note that $\mathcal{M}$ is on the world frame and contains all measured dynamic points.

Next, let $\mathcal{P}_t^M$ be the submap of $\mathcal{M}$ represented with respect to the query point cloud coordinate, which means the submap of $\mathcal{M}$ is transformed via $_W^Q\mathbf{T}_t = {}_Q^W\mathbf{T}_t^{-1}$. Throughout this paper, the superscripts Q and M are used
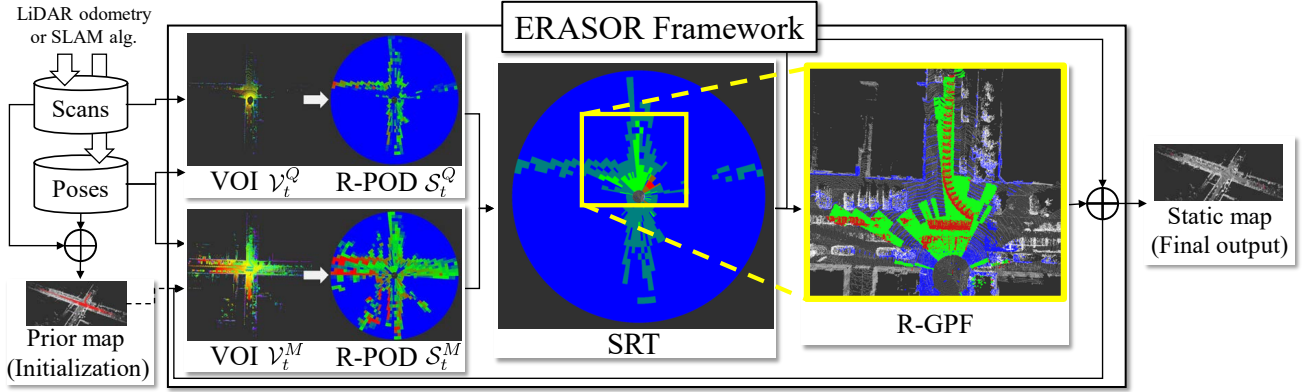
Fig. 3: Overview of our proposed method. A query cloud and a map cloud are each encoded into the *Volume of Interest* (VOI) followed by the *Region-wise Pseudo Occupancy Descriptor* (R-POD) to express the occupancies of unit spaces using *pseudo occupancy*. Then, the *Scan Ratio Test* (SRT) takes R-PODs from each cloud to discriminate spaces of varying occupancy. After fetching bins that potentially contain dynamic points (green on SRT), *Region-wise Ground Plane Fitting* (R-GPF) discriminates ground points, which are reverted into the map (green on R-GPF) while dynamic points above the reverted grounds are rejected (red on R-GPF). Best viewed in color.

to represent the query and map, respectively. By letting $\hat{\mathcal{M}}$ be the estimated static map, the problem of our interest is defined as follows:

$$\hat{\mathcal{M}} = \mathcal{M} - \bigcup_{t \in [T]} \hat{\mathcal{M}}_{dyn,t} \qquad (2)$$

where $\hat{\mathcal{M}}_{dyn,t}$ refers to the estimated dynamic points determined by discrepancies between $\mathcal{P}_t^Q$ and $\mathcal{P}_t^M$.

Before the introduction of our visibility-free method, explanations of the aspects of a unit space must be given. Our approach directs its attention to the nature of most dynamic objects in urban environments, such as terrestrial vehicles and pedestrians, being inevitably in contact with the ground. Based on this assumption, we may construct four possible cases:

- An object is on the ground in $\mathcal{P}_t^M$, while the ground of the same position is free from an object in $\mathcal{P}_t^Q$.
- The ground is free from an object in $\mathcal{P}_t^M$, while there is an object on the ground of the same position in $\mathcal{P}_t^Q$.
- An object is on the ground in both $\mathcal{P}_t^M$ and $\mathcal{P}_t^Q$.
- There are no objects on the ground in $\mathcal{P}_t^M$ and $\mathcal{P}_t^Q$.

Of these cases, points that represent dynamic objects will be categorized to the first case and the second case. However, our goal is to refine the map cloud, so the second case is neglected. For clarity, we define two terms; *potentially dynamic case* for the first case and *definitely static case* for the third and fourth cases.

Thus, our objective can be paraphrased to detect the unit spaces in $\mathcal{P}_t^M$ that satisfy potentially dynamic case, and to retrieve regions that include dynamic points (see Section II.*D*). The set of unit spaces can then be refined by removing the dynamic parts (see Section II.*E*).

B. Volume of Interest

Before $\mathcal{P}_t^M$ and $\mathcal{P}_t^Q$ are separated into unit spaces, physically significant domain of point space, or *Volume of Interest* (VOI, $\mathcal{V}_t$), is defined and formulated as follows:

$$\mathcal{V}_t = \{ \mathbf{p}_k \mid \mathbf{p}_k \in \mathcal{P}_t, \ \rho_k < L_{\max}, \ h_{\min} < z_k < h_{\max} \} \quad (3)$$

where $\rho_k = \sqrt{x_k^2 + y_k^2}$; $L_{\max}$, $h_{\min}$, and $h_{\max}$ are constant parameters that indicate the maximum radial boundary, the minimum height of the VOI, and the maximum height of the VOI with respect to the ground, respectively. Heights of dynamic objects of our interest, such as vehicles or pedestrians, usually fall into a reasonable range [6], [7]. Therefore, we may set $h_{\max} = 3.0$m, $h_{\min} = -1.0$m and $L_{\max} = 80.0$m. Note that the minimum value $h_{\min} = -1.0$m, which is lower than the ground, i.e. $h = 0$m, is set to cover sloped areas as well as the uncertainty of the detected height of the ground.

Note that $\mathcal{V}_t^M$ is extracted from $\mathcal{M}$ by neighbor search, i.e. K-D Tree [23], with respect to the position of the query frame followed by transformation via ${}_Q^W\mathbf{T}_t^{-1}$ to reduce computational cost. This is because $\mathcal{M}$ mostly consists of more than a million points; thus, it is not cost-effective to transform all points to the query frame and then extract $\mathcal{V}_t^M$.

C. Region-wise Pseudo Occupancy Descriptor

For region-wise dynamic object removal, the VOIs are encapsulated by the egocentric spatial occupancy descriptor called *Region-wise Pseudo Occupancy Descriptor* (R-POD). Its design is motivated by *Cells of Interest* (COI) [7] and *Scan Context* [6].

Kim *et al.* [7] proposed COI to encode vertical information in a point cloud into binary bits. However, COI uses RGB format and is not LiDAR-friendly [3]. On the other hand, Scan Context [6], [24] takes the maximum height from each bin and arranges them into a 2D matrix in an egocentric way. It is efficient to check the scan-to-scan relationship, yet it may be risky to encode the absolute maximum of z-directional information to check the scan-to-map relationship. That is because z-directional information of poses from LiDAR odometry or LiDAR SLAM is usually more uncertain than x and y directional information [25].

To this end, our R-POD blends the best of both methods by defining a vertical bin that expresses occupancy with the boundary difference of vertical information, or *pseudo occupancy*, in an egocentric way. Similar to Scan Context, R-POD takes $\mathcal{V}_t$ and divides the volume over the regular

interval of azimuthal and radial directions, i.e. *sectors* and *rings*. Let N_r and N_θ be the numbers of rings and sectors. Then R-POD, which is denoted as $\mathcal{S}_t$, can be represented as follows:

$$\mathcal{S}_t = \bigcup_{i \in [N_r], j \in [N_\theta]} \mathcal{S}_{(i,j),t} \tag{4}$$

where $\mathcal{S}_{(i,j),t}$ denotes the (i,j)-th bin of R-POD at time step t. Let $\theta = \arctan 2(y, x)$. Then, each $\mathcal{S}_{(i,j),t}$ consists of the cloud points that satisfy the following condition:

$$\mathcal{S}_{(i,j),t} = \left\{ \mathbf{p}_k \mid \mathbf{p}_k \in \mathcal{V}_t, \; \frac{(i-1) \cdot L_{\max}}{N_r} \leq \rho_k < \frac{i \cdot L_{\max}}{N_r}, \right. \\ \left. \frac{(j-1) \cdot 2\pi}{N_\theta} - \pi \leq \theta_k < \frac{j \cdot 2\pi}{N_\theta} - \pi \right\}. \tag{5}$$

Thereafter, the unit space, i.e. each bin, assigns a single real value to describe pseudo occupancy, $\Delta h_{(i,j),t}$. Let $Z_{(i,j),t} = \{ z_k \in \mathbf{p}_k | \mathbf{p}_k \in \mathcal{S}_{(i,j),t} \}$. Then, the pseudo occupancy of each bin is encoded as follows:

$$\Delta h_{(i,j),t} = \sup \{ Z_{(i,j),t} \} - \inf \{ Z_{(i,j),t} \} \tag{6}$$

where $\sup$ and $\inf$ denote supremum and infimum, respectively.

D. Scan Ratio Test

Scan Ratio Test (SRT) is proposed to check whether discrepancy of pseudo occupancy occur given a pair of R-PODs of query and map point clouds, i.e. $\mathcal{S}_t^Q$ and $\mathcal{S}_t^M$. The SRT is motivated by Lowe's *Ratio Test* in Scale Invariant Feature Transform (SIFT) [26]. The ratio-based approach is shown to be more robust against the change of scenes over the global threshold-based method [26]. Due to this advantage and its generality, we may apply the concept of the ratio test to our methodology when comparing pseudo occupancy.
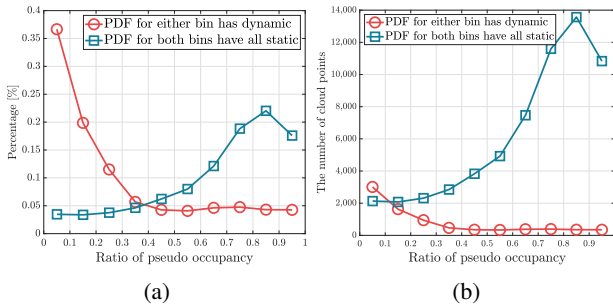

Fig. 4: (a) Probability Distribution Function (PDF) of the ratio of pseudo occupancy between two corresponding potentially dynamic bins (b) Histogram of the actual number of cloud points between two corresponding potentially dynamic bins.

Let the term *scan ratio* be the ratio of pseudo occupancy between each pair of bins in $\mathcal{S}_t^Q$ and $\mathcal{S}_t^M$, i.e. the ratio between $\Delta h_{(i,j),t}^Q$ and $\Delta h_{(i,j),t}^M$. The scan ratio is then used to categorize bins to the aforementioned cases and to select bins that belong to the potentially dynamic case where either bin includes dynamic points of our interest. Straightforwardly, the scan ratio must be close to 1 if there are no changes in both bins, i.e. the definitely static case, whereas the scan ratio that belongs to the potentially dynamic case must be

much smaller than 1, which is caused by the presence of a dynamic object on the ground as mentioned in Section II.*A*.

Therefore, we select bins with a scan ratio smaller than a ratio threshold as bins that may include dynamic objects, or *potentially dynamic bins*. Fig. 4(a) illustrates the Probability Distribution Function (PDF) of the scan ratio and Fig. 4(b) describes the histogram of the number of cloud points using the SemanticKITTI dataset (see Section III.*A*). A set of bins including dynamic points either in the query or in the map tends to have a small scan ratio. In contrast, a set of bins with only static points shows a scan ratio close to 1.

Based on these observations, the ratio threshold is set to 0.2, which is empirically determined to be strict enough. Fig. 3 illustrates the result retrieved from SRT. The green bins on SRT are potentially dynamic bins, and the teal bins are those whose scan ratio is over 0.2. The red bins are deterministically static regions or already cleaned regions, i.e. $\Delta h_{(i,j),t}^M / \Delta h_{(i,j),t}^Q < 0.2$. Note that if either the map or query bin contains few points, SRT is skipped, where those are colored in blue on the SRT in Fig. 3.

E. Region-wise Ground Plane Fitting

According to the assumption in Section II.*A*, it is observed that many potentially dynamic bins solely consist of ground points and dynamic points. Therefore, we may think of the low-computationally demanding static point retrieval method.

Bin-wise estimation of the ground plane has advantages over the estimation of a single plane model of the whole map, because many urban areas include grounds with curbs or sloped regions that do not form perfect planes [27]. For this reason, bin-wise ground plane fitting is adopted, which is motivated by bin-wise line fitting to find ground points [28]. Because each bin is small relative to the whole map, it is safe to assume that the ground is planar within the bin. As a result, R-GPF can robustly select and revert more static points back to the map on the non-planar, urban specific regions (see Section IV.*B*).

Let the l-th bin, i.e. $\mathcal{S}_{l,t}^M$, be the candidate where the total L potentially dynamic bins are selected among $N_r \times N_\theta$ bins throughout the SRT. Because the points of each bin with the lowest heights are most likely to belong to the ground surface [28], the lowest-height seed points are selected from $\mathcal{S}_{l,t}^M$, and let $\bar{z}$ be the mean z value of the selected seed points. Then, the initial estimated ground points set $^0\mathcal{I}_{l,t}$ is obtained as follows:

$$^0\mathcal{I}_{l,t} = \left\{ \mathbf{p}_k \mid \mathbf{p}_k \in \mathcal{S}_{l,t}^M, \; z(\mathbf{p}_k) < \bar{z} + \tau_{\text{seed}} \right\} \tag{7}$$

where $z(\cdot)$ represents the z value of a point and τ_{seed} denotes height margin. Because our method is iterative, let the i-th inliers be $^i\mathcal{I}_{l,t}$, then the covariance matrix $^iC_{l,t}$ of $^i\mathcal{I}_{l,t}$ is calculated as follows:

$$^iC_{l,t} = \sum_{j=1:|^i\mathcal{I}_{l,t}|} \left(\mathbf{p}_j -^i \bar{\mathbf{p}}_{l,t} \right) \left(\mathbf{p}_j -^i \bar{\mathbf{p}}_{l,t} \right)^T \tag{8}$$

where $|\cdot|$ and $^i\bar{\mathbf{p}}_{l,t}$ denote the size of a set and mean position of $^i\mathcal{I}_{l,t}$, respectively.

Next, using Principal Component Analysis (PCA), three eigenvalues and the corresponding three eigenvectors are calculated, i.e. $^iC_{l,t}\vec{v}_m = \lambda_m \vec{v}_m$ where $m = 1, 2, 3$. Then, the eigenvector with the smallest eigenvalue is the most likely to represent the normal vector to the ground. Let the eigenvector be $^i\mathbf{n}_{l,t} = [^ia_{l,t}\ ^ib_{l,t}\ ^ic_{l,t}]^T$. Then, the plane coefficient can be calculated as $^id_{l,t} = -^i\mathbf{n}_{l,t}^T\, ^i\bar{\mathbf{p}}_{l,t}$, whose plane equation is $^ia_{l,t}x + ^ib_{l,t}y + ^ic_{l,t}z + ^id_{l,t} = 0$. Finally, our goal is to extract the potentially static points below the plane as follows:

$$^{i+1}\mathcal{I}_{l,t} = \left\{ \mathbf{p}_k \mid \mathbf{p}_k \in \mathcal{S}_{l,t}^M,\ ^id_{l,t} - ^i\hat{d}_k < \tau_g \right\} \qquad (9)$$

where $^i\hat{d}_k = -^i\mathbf{n}_{l,t}^T\mathbf{p}_k$ and τ_g denotes the distance margin of the plane. The procedure is repeated three times, after which the final $^3\mathcal{I}_{l,t}$ is reverted back into the map rejecting dynamic points. Therefore, the dynamic points rejected from our method can be directly retrieved as follows:

$$\hat{\mathcal{M}}_{dyn,t} = \bigcup_{l \in [L]} (\mathcal{S}_{l,t}^M - ^3 \mathcal{I}_{l,t}). \qquad (10)$$

Fig. 3 presents the procedure of R-GPF briefly. The red points on R-GPF represent dynamic points and the green points represent the ground points that are reverted back into the map cloud.

III. Experiments

A. Dataset

For thorough validation, we conducted experiments using the SemanticKITTI dataset [5], [29]. Because SemanticKITTI provides point-wise annotations, points annotated with selected classes (252, 253, 254, 255, 256, 257, and 259) are labeled as ground-truth dynamic points to be removed.

Next, we manually selected the top-5 time frames among the Semantic KITTI dataset that contain the largest number of appearances of dynamic objects to evaluate algorithms quantitatively. Of course, our proposed method is applicable to the whole map as shown in Fig. 1. However, in general, the number of static points is too much larger than those of dynamic points on the map, and most of the frames contain no or few dynamic points; thus, it is not distinguishable at a glance whether the dynamic points have been removed well or not.

Therefore, frames `00` (4,390 - 4,530), `01` (150 - 250), `02` (860 - 950), `05` (2,350 - 2,670), and `07` (630 - 820) were chosen as our static map construction benchmark where the numbers in parenthesis indicate the start and end frames. The selected frames include various environments such as countryside, highway, and intersections. Maps are constructed at the regular intervals with the poses provided by SuMa [4] which contains inherent uncertainty.

B. Error Metrics

In addition, novel static map-oriented quantitative metrics called *Rejection Rate (RR)* and *Preservation Rate (PR)* are proposed. An existing metric, namely, precision-recall [15],
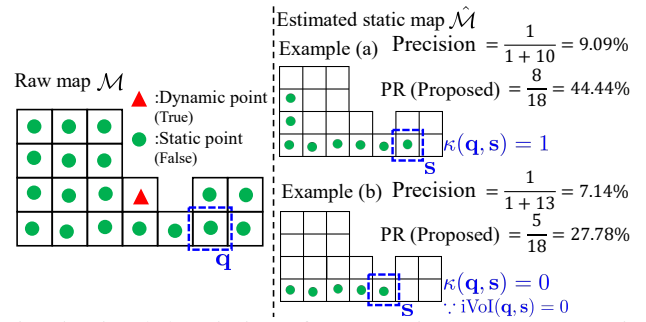

Fig. 5: Visual description of Preservation Rate. Preservation Rate is more appropriate to measure the performance of static-map building algorithms over Precision-Recall due to its higher sensitivity to the number of preserved static points.

is somewhat unsuitable for static map building tasks for the following reasons. First, as the number of static points to be erased (*false positives, FP*) is much larger than the number of dynamic points (*true positives, TP*), the precision is always too low. Thus it is not sensitive to relative changes as shown in Fig. 5. Second, use of different voxel sizes for different algorithms affects the number of FPs and TPs, i.e. precision-recall is a voxelization-variant metric.

The aforementioned metrics are defined as follows:

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

PR and RR are calculated voxel-wise. Here, as all different state-of-the-art methods use their own voxel size, we apply identical voxelization $\nu(\cdot)$ with a voxel size of 0.2 for static maps retrieved from the baseline models for fair comparison.

Let $\mathbf{q} \in \nu(\mathcal{M})$ and $\mathbf{s} \in \nu(\hat{\mathcal{M}})$ where $\mathbf{s}$ is the nearest point to $\mathbf{q}$. Then, the preservation function $\kappa(\cdot)$ is defined as follows:

$$\kappa(\mathbf{q}, \mathbf{s}) = \text{iVoI}(\mathbf{q}, \mathbf{s}) \wedge \psi(\mathbf{q}, \mathbf{s}) \qquad (11)$$

where iVoI means inter Voxel Inclusion and is a boolean function that returns **true** if two points are in the same voxel, and **false** otherwise. Likewise, $\psi(\cdot)$ returns **true** if both points are static or dynamic and **false** otherwise. Fig. 5 illustrates the examples of PR calculation. By measuring the joint frequency, our metrics reflect both the number of correct points and the inter-spatial relationships.

IV. Results and Discussion

A. Impact of the Ground Threshold

First of all, the impact of the ground threshold τ_g on the proposed metrics was examined. As can be shown in Fig. 6, increasing τ_g led to improvement of the Preservation Rate because it allowed more static points to be reverted. Unfortunately, higher τ_g also allowed more dynamic points that are closer to the ground to be classified as ground. As a result, higher τ_g yielded a lower Rejection Rate. In most cases, Preservation Rates showed sharper increments from 0.05 to 0.15, compared to that from 0.15 to 0.25, whereas Rejection Rates showed sharper decrements from 0.15 to 0.25, compared to that from 0.05 to 0.15. Therefore, we can conclude that the ground threshold value of 0.15 yielded
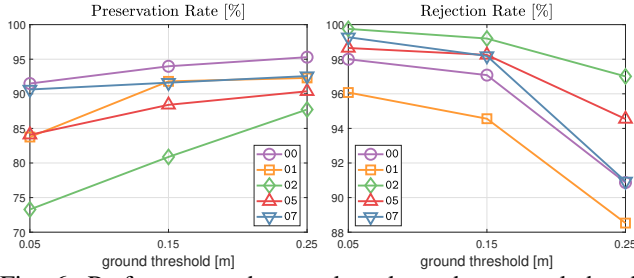
Fig. 6: Performance changes based on the ground threshold τ_g. The higher the Preservation Rate and Rejection Rate, the better.

the most reasonable plane-fitting performance throughout the experiment.

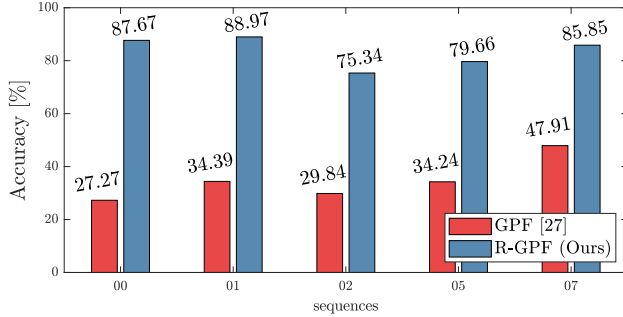### B. Comparison of R-GPF with Ground Plane Fitting

Fig. 7: Mean accuracy of static point retrieval.

TABLE I: Performance comparison with different ground plane fitting algorithms on SemanticKITTI dataset with $\tau_g = 0.15$.

Seq.	Input method	$\mathbb{E}(N_{s,g})$	$\mathbb{E}(N_{d,g})$	$\mathbb{E}(\hat{N}_{s,g})$	$\mathbb{E}(\hat{N}_{d,g})$
00	GPF [27]	29.49k	0.42k	7.74k	**2.15**
	R-GPF (Ours)			**25.85k**	40.69
01	GPF [27]	12.25k	0.45k	3.93k	**14.59**
	R-GPF (Ours)			**10.90k**	44.18
02	GPF [27]	54.07k	1.04k	15.43k	**20.22**
	R-GPF (Ours)			**40.71k**	238.22
05	GPF [27]	32.55k	1.65k	10.09k	**29.79**
	R-GPF (Ours)			**25.71k**	109.16
07	GPF [27]	11.71k	1.58k	4.79k	**4.95**
	R-GPF (Ours)			**9.89k**	56.11

As shown in Table I and Fig. 7 where $\mathbb{E}(\cdot)$ denotes expectation; $N_{s,g}$ and $N_{d,g}$ denote the total numbers of static and dynamic points of all potentially dynamic bins; and $\hat{N}_{s,g}$ and $\hat{N}_{d,g}$ denote the total number of static and dynamic points selected by an algorithm, respectively, our R-GPF yielded higher performance in static point retrieval over GPF [27] in various urban environments. In particular, the difference in performance is the largest for the sequence 00, where many curbs are placed to form non-planar floors, from which we can infer that R-GPF leverages the advantage of bin-wise operation over the non-planar floor as explained

TABLE II: Comparison with state-of-the-art methods on the SemanticKITTI dataset. PR: Preservation Rate, RR: Rejection Rate (see Section III.*B*).

Seq.	Method	PR [%]	RR [%]	F_1 score
00	OctoMap - `0.05` [12]	76.731	99.124	0.865
	OctoMap - `0.2` [12]	34.568	**99.979**	0.514
	Peopleremover [14]	37.523	89.116	0.528
	Removert - `RM3` [15]	85.502	99.354	0.919
	Removert - `RM3+RV1` [15]	86.829	90.617	0.887
	ERASOR (Ours)	**93.980**	97.081	**0.955**
01	OctoMap - `0.05` [12]	53.163	99.663	0.693
	OctoMap - `0.2` [12]	20.777	**99.863**	0.344
	Peopleremover [14]	36.349	93.116	0.523
	Removert - `RM3` [15]	94.221	93.608	**0.939**
	Removert - `RM3+RV1` [15]	95.815	57.077	0.715
	ERASOR (Ours)	91.487	95.383	0.934
02	OctoMap - `0.05` [12]	54.112	98.769	0.699
	OctoMap - `0.2` [12]	23.746	**99.792**	0.384
	Peopleremover [14]	29.037	94.527	0.444
	Removert - `RM3` [15]	76.319	96.799	0.853
	Removert - `RM3+RV1` [15]	83.293	88.371	0.858
	ERASOR (Ours)	**87.731**	97.008	**0.921**
05	OctoMap - `0.05` [12]	76.341	96.785	0.854
	OctoMap - `0.2` [12]	33.904	**99.882**	0.506
	Peopleremover [14]	38.495	90.631	0.540
	Removert - `RM3` [15]	86.900	87.880	0.874
	Removert - `RM3+RV1` [15]	88.170	79.981	0.839
	ERASOR (Ours)	**88.730**	98.262	**0.933**
07	OctoMap - `0.05` [12]	77.838	96.938	0.863
	OctoMap - `0.2` [12]	38.183	**99.565**	0.552
	Peopleremover [14]	34.772	91.983	0.505
	Removert - `RM3` [15]	80.689	98.822	0.888
	Removert - `RM3+RV1` [15]	82.038	95.504	0.883
	ERASOR (Ours)	**90.624**	99.271	**0.948**

in Section II.*E*. Note that R-GPF reverts a few more dynamic points, i.e. higher $\mathbb{E}(\hat{N}_{d,g})$, because R-GPF is likely to consider the underside of dynamic points, such as the contact point of a vehicle wheel, as inliers. However, R-GPF was originally intended to return more static points to the map risking a few dynamic points, which is motivated by the Revert stage of Removert [15]. Besides, these dynamic points are so close to the ground that they have an insignificant impact on localization or navigation performance. Therefore, the effect of these undesirable points could be negligible.

C. Comparison with the State-of-the-Art Methods

ERASOR was quantitatively compared with state-of-the-art methods, namely, OctoMap[1] [12], Peopleremover[2] [14], and Removert[3] [15]. We leveraged the open source implementations for the experiment. In Table II, `0.05` and `0.2` denote the processing voxel sizes of OctoMap. `RM3` denotes the results after three Removal stages with per-pixel resolutions [15] of 0.4°, 0.45°, and 0.5°. `RM3-RV1` means the result of `RM3` followed by a Revert stage with the resolution per pixel of 0.55°.

The state-of-the-art methods give an exquisite static map, filtering out the most dynamic points. However, our ERASOR exhibits noticeable improvements over the aforemen-

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

[2] http://threedtk.de

[3] https://github.com/irapkaist/removert

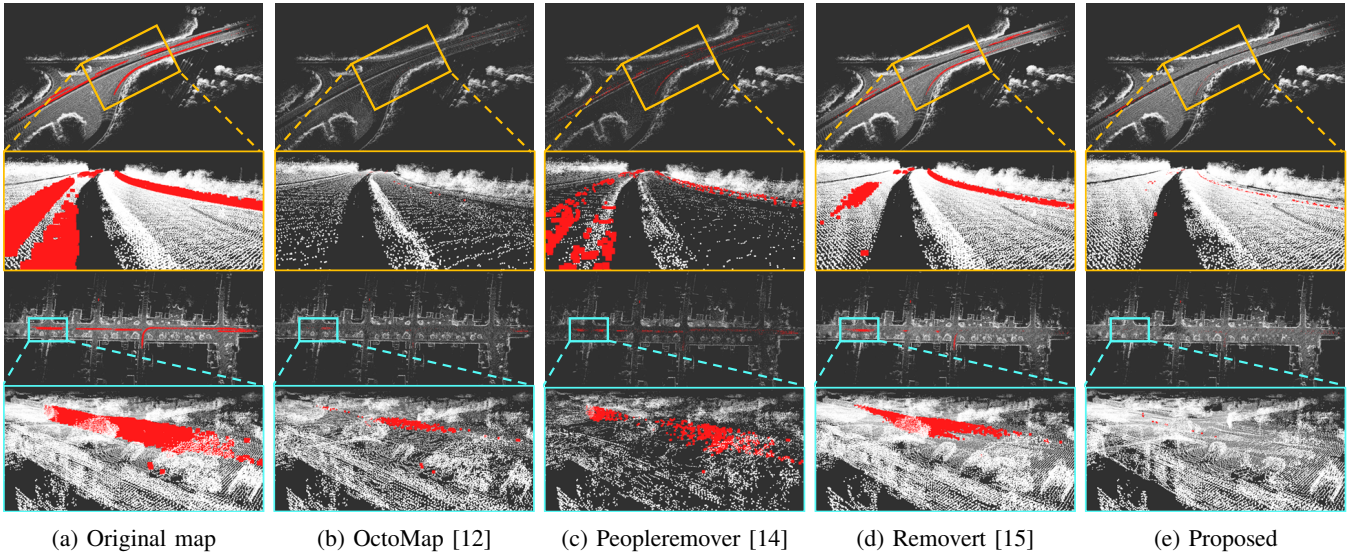(a) Original map   (b) OctoMap [12]   (c) Peopleremover [14]   (d) Removert [15]   (e) Proposed

Fig. 8: Comparison of static map generation results produced by the proposed method and state-of-the-arts methods on Semantic KITTI dataset (T-B): sequences `01` and `05`. Red cloud points indicate dynamic objects and the fewer red points, the better (best viewed in color).

(a) Original map   (b) OctoMap [12]   (c) Peopleremover [14]   (d) Removert [15]   (e) Proposed
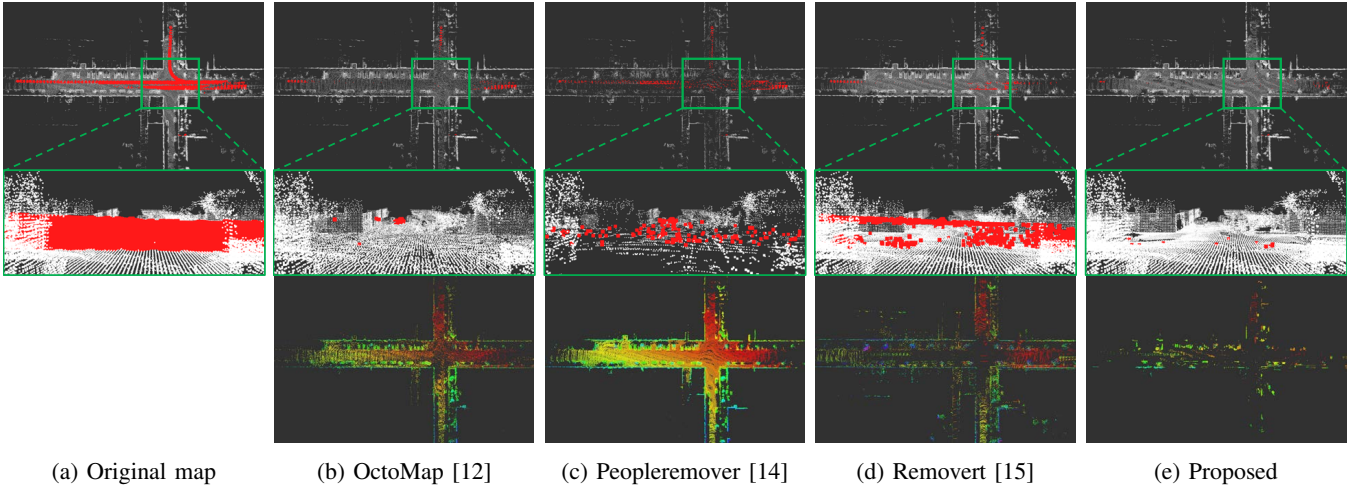
Fig. 9: Comparison of static map generation results obtained with the proposed method and state-of-the-arts on sequence `07` of Semantic KITTI dataset (T-B): results of static map and lost static points (*false positives*) during moving object removal. For both red cloud points and lost static points, the fewer the better (best viewed in color).

tioned limitations of both ray tracing-based and visibility approaches, as illustrated in Fig. 2. Note that our methodology is fundamentally free from visibility.

In particular, the static map generation results by the state-of-the-arts on the sequence `05` and `07` reported that there are still some remaining dynamic points on the top part of the bus due to the limitation explained in Fig. 2(c), as shown in the last row of Fig. 8 and Fig. 9. That is, the object was too large and too close to the query coordinate; thus, the remaining dynamic points were in an invalid range to be checked for ray tracing or visibility. In contrast, our proposed method successfully removes most of the dynamic traces. Because our approach checks the scan ratio, even the points existing in the invalid range of visibility can be removed.

Although OctoMap rejects most of the dynamic points as shown in Fig. 8 and Fig. 9, it falsely removes large parts of static points, which results in the lowest PR as shown in Table II. The last row of Fig. 9 visualizes the lost static points, i.e. *false positives*, which implies that the OctoMap and Peopleremover suffer from motion ambiguity and unintentionally loses a relatively larger number of static points compared to our proposed method.

Therefore, our proposed method shows promising PR and F_1 scores, as shown in Table II, in all environments except for `01`. The cause of lower PR in this frame is presumed to be the removal of vegetation far from the highway because once vegetation is partially observed, and yields a small scan ratio.

D. Algorithm speed

In addition, our proposed method shows the fastest performance for one iteration, as reported in Table III. Although the time complexity of Removert [15] is $O(n)$ and that of our ERASOR is $O(n \log n)$, Removert requires multiple

TABLE III: Runtime per iteration on sequence `01` of SemanticKITTI dataset.

Method	Runtime/iteration [s]
OctoMap [12]	1.077
Peopleremover [14]	1,000
Removert [15]	0.8307
ERASOR (Ours)	**0.0732**

projections on the spherical image plane and assigns staticity to each point, while ERASOR is a bin-wise approach; thus, it removes the dynamic points with one-shot via the scan ratio. Therefore, our method is at least ten times faster than other methods.

V. CONCLUSION

In this study, a novel static map building method, ERASOR, has been proposed. Our method was tested on the SemanticKITTI dataset both quantitatively and qualitatively. As a result, our method was proved to overcome the limitations of ray tracing-based methods and visibility-based methods, and it shows promising results in various environments. Our method provides a static map with few dynamic objects, which helps improve the navigation and localization tasks of mobile robots. In future works, we plan to improve pseudo occupancy or to devise a spatial descriptor that is more robust to other kinds of noise and is suitable for more sophisticated dynamic object removal.

ACKNOWLEDGMENT

REFERENCES

[1] H. Lim, S. Hwang, S. Shin, and H. Myung, "Normal distributions transform is enough: Real-time 3D scan matching for pose correction of mobile robot under large odometry uncertainties," in *Proc. IEEE International Conference on Control, Automation and Systems (ICCAS)*, 2020, pp. 1155–1161.

[2] H. Kim, S. Song, and H. Myung, "GP-ICP: Ground plane ICP for mobile robots," *IEEE Access*, vol. 7, pp. 76 599–76 610, 2019.

[3] 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*, vol. 4, no. 2, pp. 1948–1955, 2019.

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

[5] 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. IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9297–9307.

[6] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.

[7] H. Kim, B. Liu, C. Y. Goh, S. Lee, and H. Myung, "Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large scale urban area," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1518–1524, 2017.

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

[9] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner, *et al.*, "The Strands Project: Long-term autonomy in everyday environments," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 146–156, 2017.

[10] N. Banerjee, D. Lisin, J. Briggs, M. Llofriu, and M. E. Munich, "Life-long mapping using adaptive local maps," in *Proc. IEEE European Conference on Mobile Robots (ECMR)*, 2019, pp. 1–8.

[11] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 716–12 725.

[12] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

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

[14] 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*, vol. 3, no. 3, pp. 1679–1686, 2018.

[15] G. Kim and A. Kim, "Remove, then Revert: Static point cloud map construction using multi-resolution range images," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, [Online]. Available: irap.kaist.ac.kr/publications/gskim-2020-iros.pdf.

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

[17] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[18] K. Litomisky and B. Bhanu, "Removing moving objects from point cloud scenes," in *Proc. International Workshop on Depth Image Analysis and Applications*, 2012, pp. 50–58.

[19] C. Yin, S. Yang, X. Yi, Z. Wang, Y. Wang, B. Zhang, and Y. Tang, "Removing dynamic 3D objects from point clouds of a moving RGB-D camera," in *Proc. IEEE International Conference on Information and Automation*, 2015, pp. 1600–1606.

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

[21] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett, "Recurrent-OctoMap: Learning state-based map refinement for long-term semantic mapping with 3-D-LiDAR data," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3749–3756, 2018.

[22] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying unknown instances for autonomous driving," in *Proc. Conference on Robot Learning*, 2020, pp. 384–393.

[23] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[24] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," *arXiv preprint arXiv:2003.05656*, 2020.

[25] J. Zhang and S. Singh, "LOAM: LiDAR Odometry and Mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.

[26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[27] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5067–5073.

[28] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," in *Proc. IEEE Intelligent Vehicles Symposium*, 2010, pp. 560–565.

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