**SegDF: Segment-Based Dynamic Filter in 3D Point Cloud**

**Abstract**

Dynamic objects in lidar data frequently leave undesirable traces in 3D point clouds, degrading the quality of maps and the effectiveness of localization. We present a novel static reconstruction method, called SegDF, that focuses on filtering the potential-mobility elements contacted with the ground in urban environments. Our approach can not only reconstruct maps with low-semantic information, but also remove movable objects at a low cost. Given a pair of registered scans, we generate a curved-voxel map with the assistance of sensor intensities to cluster the scans into instances and populate semantics by geometric classification. To exclude instances that are non-static yet unstable due to inconsistent observations, we monitor volumetric states based on the occupancy change of curved voxels at object level. In addition, our method tightly couples instance segmentation with dynamic detection in order to refine imperfect clustering and enhance the removal precision. SegDF is validated on the KITTI dataset using SemanticKITTI as the ground truth, and it is demonstrated that the proposed can produce a high-quality static instance map.

**Ⅰ. INTRODUCTION**

Leveraging LiDAR odometry [1, 2, 3], location recognition [4, 5, 6], and long-term mapping [7, 8, 9], recent advancements in 3D LiDAR detection and mapping have been reported to encounter an ambiguous environment that contains diverse moving objects such as vehicles, pedestrians, and so forth [10, 11, 12, 13, 14]. Although most autonomous robotic systems always assume that all observations are accessible, excluding dynamic objects as outliers and relying solely on static objects might prevent confusion and increase reliability.

A scan data presents a snapshot of the surroundings, which invariably contains dynamic points with limited interpretability. Since a 3D point cloud map is the accumulation of sequential scan data, there might be traces of dynamic objects, or the ghost trail effect [10, 14], as shown in Fig.1 on the left. These undesirable traces act as fake static obstacles in map and thus impede mobile robots from performing well during localization and navigation.

Generally, removing dynamic points is always considered as detecting instantaneous changes in several registered maps. Conventional methods to tackle this problem are mainly divided into two categories: a) rejecting dynamic points timely in map generation [15, 16] and b) post-removing dynamic points with a global map [13, 14]. The proposed focuses on the latter, which could be separated into three classes: a) segment-based, b) ray tracing-based, and c) visibility-based methods.

Segment-based methods [15, 17, 18, 19, 20] are straightforward, because it only needs to implement dynamic detection on the objects estimated to be dynamic. To distinguish dynamic and static objects, there are methods based on clustering segmentation [17, 18] or learning-based segmentation [19, 20]. Additional features are required to determine the mobility of objects in clustering methods; while, due to the inconsistent observations caused by restricted viewpoint, the same object might have unstable characteristics in adjacent scans. Learning-based methods can directly discard objects labeled as dynamic, but they rely heavily on supervised labels.

Ray tracing-based methods [10, 11, 12, 14, 16] deem that dynamic points just exist momently, only actually static points are always restored in grids. However, being carried out on scans with few dynamic points, these methods need to traverse each point to occupy grids, which leads to unnecessary computational cost.

Visibility-based methods [7, 13, 21] 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. But because of the registration errors and occlusions, the above judgment often makes mistakes to remove dynamic points incompletely.

|  |
| --- |
|  |
| Fig. 1 Before (the left) and after (the right) the application of the proposed called SegDF on sequence 05 of SemanticKITTI [14] (top view). The task of our method is to remove dynamic objects highlighted in red in the original map, and build a static instance map including ground, buildings, vegetation and the other stationary objects, which are respectively marked in yellow, brown, green and random color. |

To overcome the aforementioned limitations, we proposed a novel scheme for dynamic removal, called SegDF, which aims to reconstruct a static instance map, as shown in Fig.1 on the right. Our contributions are threefold:

1) We proposed a clustering segmentation based on curved voxel with intensity and design a low-cost recognition strategy utilizing geometric characteristics. Taking [22] as the baseline, we add LiDAR intensity to constrain the curved-voxel searching and optimize the outcomes through iterative growth. Then the geometric recognition distinguishes potential-mobility objects, such as vehicles, pedestrians, being inevitably in contact with the ground (§III-B, §III-C).

2) We proposed a fast object-level dynamic filter to track potential-mobility objects by detecting the occupancy of curved-voxel. The structure of curved voxel adapts the spherical emission of 3D LiDAR to reduce the effect of registration errors on dynamic detection; and the instance tracking establishes object connection in adjacent scans to remove weakly associated objects, which lessens computation (§III-D).

3) Considering the inconsistent observations in adjacent scans, we tightly couple segmentation and dynamic removal. In the process of instance tracking, multi-frame segmentation information are combined to compensate the segmentation error, thereby improving the accuracy of dynamic removal (§III-E).

**Ⅱ. RELATED WORK**

Dynamic removal is closely related to static map construction. Online methods are excluded in this paper, since we are primarily concerned with constructing a high-quality static instance map without regard to processing speed.

**Segmentation-based methods**. After obtaining the motion properties of each object, it is straightforward to perform dynamic detection. A dynamic selecting method based on Viewpoint Feature Histogram (VFH) is proposed [17], but registration errors make it difficult to achieve constant features. [18] utilized feature matching to check false correspondences and then used them as seeds to extract clusters as dynamic objects, yet the residuals caused by restricted viewpoint might be mistaken as false matches. Given semantic points with dynamic labels by deep-learning, a map can be constructed by excluding the predicted dynamic points [19, 20]. However, these methods are vulnerable to labeling errors or dynamic objects of unlabeled classes [23].

**Ray tracing-based methods.** Typical methods using Ray tracing [11, 16] count hits and misses of scans in the grid map and remove points in low-occupancy grids. By extension, [12] proposed the removal of dynamic points by traversing a voxel occupancy grid and [10] suggested a combination of object detection and [11]. To reduce the computational burden, [14] fetches bins with potential-mobility by comparing the height difference, and then remove points on the fitting ground. However, it may fail in irregular environments, because it strictly depends on the spatial distribution.

**Visibility-based methods**. Ray tracing-based methods are computationally expensive, which led to the introduction of visibility-based methods [8, 9, 13]. The assumption that a dynamic point is often followed by a static point that is collinear with it, has incidence angle ambiguity with longer range measurement. To solve this problem, [8] uses normal and incidence angles to represent the state for each point. Furthermore, [13] proposed a pixel-to-window comparison method to take incidence angle ambiguity into account. However, all previous works struggled with the occlusion and registration errors.

**Ⅲ. METHODOLOGY**

The proposed is a post-processing method to reconstruct a static instance map. Assuming that the registration errors due to imperfect pose estimation are tolerable, we examine batch data of a certain length in [24] to validate the quality of the static map.

|  |
| --- |
|  |
| Fig. framework |

*A. Problem definition*

Given a point cloud map generated from a set of LiDAR frames {*P*}, we intend to remove dynamic objects in the map. Specifically, we consider the global coordinate *M* and the local sensor query coordinate , where k is the frame index. We assume that the related SE(3) pose (i.e., the transformation matrix from to ) and the frame data in the local coordinate system are known. Additionally, two groups in *M* are described, with *D* representing the dynamic group and *S* representing the static group, where . The mission thus corresponds to removing all the dynamic object , where , the index of dynamic object in . The challenge stated above is phrased formally as

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |

where the is the total frame indexes that is equal to .

|  |
| --- |
|  |
| Fig. 3 A visualization of the product (i.e., static/dynamic map segregation) in the point-wise form on the SemanticKITTI [24] sequence 02 (top view) from frame 1120 to frame 1200. The top point cloud map is constructed using scans and SE(3) poses estimated using SuMa [2] for every 2 m  and has many dynamic points from various types of instances (e.g., vehicles, pedestrians, etc.). As marked by the boxes in the same color, the effectiveness of our method can be verified by comparing the image frame with the point cloud map. |

Our methods consist of two phases: (i.e., instance segmentation) and. (i.e., dynamic filter between two adjacent frames). A static instance local map equates to the set of static instances after dynamic removal, as shown in Fig. 3, hence the procedure can be stated as

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |

where , which represents the transformation of segmented point cloud from to .

*B. Intensity-assisted curved-voxel clustering*

After pre-processing such as ground extraction and intensity rectification, intensity is added to constrain the curved-voxel clustering and refine the results by iterative growth.

**Ground extraction**. The majority of each frame's raw data consists of ground points, which may affect the segmentation of other instances [25]; consequently, it is necessary to separate the ground before clustering. Moreover, the ground is utterly immobile, thus dynamic detection should not be conducted on it. We extract ground from the point cloud by multi-region plane fitting that solves uneven ground scene [26].

**Intensity rectification**. The LiDAR intensity indicates the surface properties (e.g., roughness, reflectance, etc.) on objects composed of various materials. However, the intensity channel is noisy because it is influenced not only by surface properties, but also by acquisition geometry (e.g., distance) and instrument effects (e.g., transmitted energy) [27]. Thus, Intensity rectification is necessary to reduce the influence of insignificant factors, yet, it is impossible to totally correct the true value of intensity through a mathematical model. In order to use intensity efficiently, the local connection on objects is established through the continuity of intensities in adjacent spaces, as shown in Fig. voxel内的方差.

We solely consider the effect of distance , and incidence angle on raw intensity during the scanning period. Assuming that is constant, hence the rectified intensity can be formed as

|  |  |
| --- | --- |
|  | (5) |

**Curved-voxel clustering assisted by intensity.** We transform cartesian coordinates to spherical ones, , which consists of the radial distance , the polar angle, and the azimuth angle [22]. Curved-voxel with intensity is redefined as *CVI*, a spatial unit comprised of three constituents: a) spherical coordinates, b) the average of intensity *AV*, which represents local reflectance, and c) the covariance of intensity *COV*, which represents credibility of the rectified values. Then the thresholds of average *threAV* and covariance *threCOV* are defined to respectively measure the intensity continuity in adjacent spaces, and the reliability of rectification. The fundamental components of is as follows

|  |  |
| --- | --- |
|  | (6) |

where , , and are unit size parameters for each spherical direction.

|  |
| --- |
|  |
| Fig. 4 Steps of the proposed, using a bird-eye view, where a LiDAR surrounded by sparse point cloud. In Step 1, we convert cartesian coordinate into spherical one and build a sparse CVI map that managed by hash table. In Step 2, CVIs in the same color represent good continuity on intensity map (bottom), which constrains the CVI search in clustering (top). In Step 3, A refinement based on intensity map has been performed through iterative growth. In Step 4, points in well related CVIs are combined into clusters. |

We describe our proposed segmentation method in Algorithm 1, as shown in Fig. 4. Firstly a preliminary procedure is implemented to extract ground [37] and rectify intensity (line 1~2). Then we convert the cartesian coordinates of no-ground points into spherical ones (line 3), and build sparse CVI map that managed by hash table (line 4). Notably, to maintain the sparse representation of the hash table, only CVIs with at least one point are stored. After establishing the CVI map, we combine each point with its neighbor CVIs, whose difference of AV between target CVI is less than the *threAV* (line 5~13). This method makes the intensity continuity in the adjacent spaces equal to the local consistency of one object. After obtaining an initial list of clusters, a novel iterative growth is adopted to refine the outcomes (line 14 ~ 19). We traverse each cluster to find its neighbor clusters through CVI search, and all the neighbors must satisfy *threAV* and *threCOV.*

|  |
| --- |
| Algorithm 1: **Curved-voxel clustering assisted by intensity** |
| **Input:** , {, }, and {*threAV*, *threCOV*} |
| **Output:** |
| 1: **Extract ground:** remove ground points in [26] |
| 2: R**ectify intensity:** in local region normal vector incident vector,  formula (6) |
| 3: Convert-to-Spherical ( |
| 4: hash-table Build-CVI (*X*, , , , *AV*, *COV*) |
| 5: **for** in **do** |
| 6: **if** is already included in a cluster **then** |
| 7: **continue** |
| 8: **end if** |
| 9: Neighbor () CVI-search **and**  < *threAV* |
| 10: Combine-clusters (, Neighbor ()) |
| 11: **if** is still not included in any cluster **then** |
| 12: assign-cluster (, Neighbor ()) into |
| 13: **end for** |
| 14: **for** cluster in **do** |
| 15: **for** in **do** |
| 16: Neighbors () CVI-Search  **and**  < *threAV*, *COV* < *threCOV* |
| 17: Neighbors () get-clusters (Neighbors ()) |
| 18: Combine-clusters (, Neighbors ()) |
| 19: **end for** |
| 20: **return** |

C. Geometric classification

Intensity-assisted curved-voxel clustering obtains multiple objects, but instance segmentation should identify each object. We use geometric features [6, 28, 29] to distinguish dynamic instances. The dimension of the geometric feature vector is 110, which consists of two parts:

1) Eigenvalue: This descriptor is merged into a 7-dimensional feature vector. In accordance with [6, 28], we compute the linearity, planarity, scattering, omni-variance, anisotropy, eigen-entropy, change of curvature measures, and primary orientation. It utilizes the aggregated global characteristics to precisely describe the geometric distribution of objects.

2) Spatial scale: This descriptor consists of Minimum height, Maximum height, and scale in a 3-dimensional feature vector, which naturally represents the scale information and reduce the noisy from global changes.

To meet the requirements of low semantic information, we classify the clusters into three categories: a) vegetation (e.g., trees and grass), b) buildings (e.g., walls and fences), and c) potential-mobility objects (e.g., vehicles, pedestrians). In order to intuitively distinguish the three categories with limited information, we devised a hierarchical classification criterion. Due to buildings seem flat in most scans, we extract them by region growth and restrict them through planarity and change of curvature. Being quite complex, vegetation is difficult to be recognized, so we apply planarity, omni-variance, anisotropy and spatial scale to classify this category. Excluding buildings and vegetation, the rest of clusters contacted with ground, such as vehicles, pedestrians, billboards, and so forth, are possibly in moving, so they are all considered potential-mobility objects in environment. (fig)

D. *Dynamic removal in object tracking*

To remove dynamic instances, we apply an object-level tracking strategy. Only potential-mobility objects are identified throughout this process, which effectively eliminates the issue of inconsistent observation of the same objects in successive scans, and the voxel search optimizes the processing speed of a single frame. In addition, to boost the accuracy of object tracking, we select a specific amount of frames to initialize the instance results and tightly couple the tracking and instance segmentation processes.

We discuss our proposed dynamic removal strategy in Algorithm 2. After obtaining and , we assemble them into the same curved-voxel map and perform an object-level registration (line 1). Then, for each instance with potential mobility in , we find its neighbor instance by curved-voxel searching and calculate the occupancy to check the observation consistency (line 5~8).

Registering the segmentation results from both frames in the same curved-voxel map, we utilize the amount of instance coverage and voxel occupancy changes to characterize the nature of motion. Notably, the above procedure is entirely based on curved-voxel tracking, which may not only lower the cost but also modify the instance in a flexible manner. According to this criterion, the motion properties of objects in can be divided into four categories:

*1) High Dynamic (HD) Objects:* Due to the unpredictability of observations in , interpreting high-mobility objects is exceedingly difficult. We divide it into two types: a) cannot be covered, meaning there is no voxel occupancy link; and b) may be covered, however the voxel occupancy is variable. Objects with high dynamic qualities, such as vehicles, pedestrians, etc., tend to have fast translation speeds. (line 9 ~ 12).

*2) High Static (HS) Objects:* Non-moving objects, such as parked vehicles, crates, etc., could be properly tracked in , provided they occupy one or more instances and satisfy a specific occupancy ratio. (line 13 ~ 18).

(3，4点不确定需不需要留下)

3) Object Split: We only conduct dynamic detection for objects with high mobility. However, due to the inconsistent observation of the two adjacent frames, different types of objects will establish contact after geometric recognition. To solve this problem, the proposed splits the object in the previous frame according to the observation occupancy (line 15).

4) Object Fusion: Faced with a one-to-many search, if the attributes of multiple objects searched are the same, they will be fused (line 18).

(待做类似于segmatch上下比较的图，表明跟踪)

|  |
| --- |
| Algorithm 2: **Dynamic removal in object tracking** |
| **Input**: } and , and |
| the threshold of occupancy *threOccu* |
| **Output**: in , and |
| after compensation |
| 1: **Registration**: Inverse (, and |
|  |
| 2: **for** cluster in **do** |
| 3: Convert-to-Spherical () |
| 4: Neighbor () Build-CVI () |
| 5: **for** **do** |
| 6: **if** find in hash-table () **then** |
| 7: Neighbors () check cluster |
| 8: **end if** |
| 9: **if** Size (Neighbors ()) = 0 **then** |
| 10: add into |
| 11: **else if** Size (Neighbors ()) > 0 **and** Satisfy (*threOccu*)**then** |
| 12: add into |
| 13: **else if** Size (Neighbors ()) > 0 **and** no-Satisfy (*threOccu*)**then** |
| 14: add into |
| 15: Split (Neighbors ()) |
| 16: **else if** Size (Neighbors ()) > 0 **and** Satisfy (*threOccu*)**then** |
| 17: add into |
| 18: Fuse (Neighbors ()) |
| 19: **end if** |
| 20: **end for** |
| 21: **return** and |

***Ⅳ. EXPERIMENTAL RESULTS***

*A. Dataset*

We use the KITTI odometry dataset [30] and the SemanticKITTI dataset [24] to evaluate our static map construction performance qualitatively and quantitatively. The SemanticKITTI contains scan-wise labeled data and the related LiDAR SLAM-based SE(3) poses. Furthermore, the SemanticKITTI offers point-wise annotations, points labeled with specific class (252, 253, 254, 255, 256, 257, and 259) are defined as ground-truth dynamic points to be erased. We notice that we did not include unlabeled points, whose label index is zero, since if we did, dynamic points would arise and the original map could no longer function as a static ground truth map.

In general, static points in the map are too much more than dynamic points, and the majority of frames contain no or few dynamic points; hence, it is difficult to determine at a glance whether the dynamic points have been successfully removed. As shown in Fig. 3 and Fig. 4, we manually selected the frames from the SemanticKITTI [24] that have the maximum number of occurrences of dynamic objects in order to objectively evaluate the algorithms. Obviously, our proposed is applicable to the entirety of Fig. 1's map, which focuses on demonstrating the long-term effect of constructing a static instance map. Therefore, frames 00 (4320 - 4530), 01 (120 - 270), 02 (820 - 980), 05 (2350 - 2670), and 07 (650 - 840) were chosen as our static map construction benchmark where the numbers in parenthesis indicate the start and end frames. Maps are constructed at the regular intervals with the poses provided by SuMa [2] which contains inherent uncertainty.

*B. Evaluation Criteria*

The proposed refers to the static status as positive (P) and the dynamic status as negative (N). Then the estimates and are expressed as

|  |  |
| --- | --- |
| and | (5) |

where the TP, FP, TN, FN respectively mean correctly estimated static, falsely preserved dynamic (i.e., static that is actually dynamic), correctly eliminated dynamic and falsely removed static (i.e., static that should not be removed). Using this equation, we can redefine the problem as reducing

the number of FP and FN points to quantitatively analyze the quality of static maps.

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-recall is not sensitive to relative changes. Static map-oriented quantitative metrics called *Rejection Rate (RR)*, *Preservation Rate (PR)*, and *F1 score* are used [14], which are defined as follows:

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 with a voxel size of 0.2 m for static maps retrieved from the baseline models for fair comparison.

*C、Analysis of static map quality*

Fig. 4 shows our detailed quantitative results. Our method can successfully separate static instances and dynamic ones, even when urban environments contain a lot of noise. Most moving cars on the road can be removed completely. However, for the moving cars far from local coordinate, they are easily fitted in ground, which causes FP. And walls are too discrete to be clustered and recognized, so they are always judged as FN.

|  |
| --- |
|  |
| Fig. 4 The visualization of result using the evaluation criteria on SemanticKITTI [14] sequence 09 from frame 1310 to frame 1400. The TP, FP, TN, and FN points are indicated in green, pink, blue, and orange, respectively. Furthermore, we highlight FP and FN. |

*D、Comparison With the State-of-The-Art Methods*

Leveraging the open-source implementations for the experiment, SegDF was quantitatively compared with state-of-the-art methods, namely, Removert [13], ERASOR [14]. In Table Ⅰ, RM3 denotes the results after three Removal stages [13] with per-pixel resolutions of 1.5°, 2°, and 2.5°. RM3-RV2 means the result of RM3 followed by a Revert stage [13] with the resolution per pixel of 1.0° and 0.8°. RI3 denotes only three growth iterations are performed to refine clusters by intensity. TC means the result after tightly coupling without RI3. RI3-TC denotes the result of RI3 followed by TC.

The state-of-the-art methods give an exquisite static map, filtering out the most dynamic points.

However, they also remove a large number of static points excessively, which will result in a sparse and crippled map. While overcoming the shortcomings of segment-based, ray tracing-based and visibility approaches, our method can not only remove most dynamic points, but also persevere as many static points as possible.

The static map generation results by Removert [13] on the sequence 02 and 07 reported that there are still some remaining dynamic points on the top and bottom part of the bus, as shown in Fig. 5(b). 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 visibility. In contrast, our proposed method successfully removes the most of dynamic traces. Because our approach is based on segmentation, which uses curved voxel to search in space, so it does not require visibility to overcome the problem of object occlusion.

ERASOR [14] can remove most dynamic points at a low cost, but it falsely removes large parts of static points, which results in a low PR as shown in Table I. ERAOSR utilizes the method of scan-to-map to fetch bins where there is a large height difference, and remove the points over the fitting ground. However, due to the registration errors and occlusions, it is unreliable to judge whether there is an object with height difference on the ground. In comparison, the proposed remove dynamic points in object-level tracking; thus it can integrate the information between multiple frames and improve the accuracy of dynamic object removal, persevering more actually static points.

As illustrated by in Table Ⅰ, our proposed method shows promising PR and F1 scores. The cause of lower PR in this frame is presumed to the removal of pedestrian or moving bicyclist in a low speed, because our object-level removal is based on the occupancy of curved voxels that is insensitive to slowly moving objects.

*E. Ablation*

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| (a) Original map | (b) Removert | (c) ERASOR | (d) proposed |
| Fig. 5 seq.02 07 | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TABLE Ⅰ  COMPARISION WITH STATE-OF-ART METHODS AND ABLATION ON THE SEMANTICKITTI DATASET.  PR: PRESERVATION RATE, RR: REJECTION RATE | | | | | |
| Seq. | Method | PR [%] | RR [%] | F1 score | Runtime/iteration (ms) |
| 00 | Removert - RM3 | 94.64 | 83.98 | 0.8899 | 611.04 |
| Removert - RM3+RV2 | 96.71 | 76.58 | 0.8574 | 804.53 |
| ERASOR | 89.07 | **96.74** | 0.9274 | **72.89** |
| Ours - RI3 | 96.88 | 90.97 | 0.9383 | 178.59 |
| Ours - TC | 96.86 | 89.10 | 0.9282 | 192.44 |
| Ours – RI3+TC | **98.81** | 94.53 | **0.9662** | 213.67 |
| 01 | Removert - RM3 | 98.70 | 53.23 | 0.6916 | 654.01 |
| Removert - RM3+RV2 | **99.03** | 46.99 | 0.6374 | 824.41 |
| ERASOR | 94.46 | 92.41 | 0.9343 | **88.23** |
| Ours - RI3 | 91.87 | 93.42 | 0.9263 | 153.50 |
| Ours - TC | 89.82 | 86.20 | 0.8797 | 169.28 |
| Ours – RI3+TC | 94.38 | **94.65** | **0.9453** | 179.66 |
| 02 | Removert - RM3 | 92.32 | 88.79 | 0.9052 | 548.76 |
| Removert - RM3+RV2 | 95.45 | 84.71 | 0.8976 | 686.35 |
| ERASOR | 75.14 | **99.58** | 0.8566 | **90.71** |
| Ours - RI3 | 96.13 | 92.98 | 0.9453 | 196.29 |
| Ours - TC | 96.09 | 90.48 | 0.9320 | 181.53 |
| Ours – RI3+TC | **98.52** | 96.12 | **0.9704** | 236.41 |
| 05 | Removert - RM3 | 94.52 | 72.52 | 0.8207 | 833.12 |
| Removert - RM3+RV2 | 97.67 | 66.90 | 0.7941 | 1087.33 |
| ERASOR | 88.62 | **98.09** | 0.9311 | **84.23** |
| Ours - RI3 | 96.51 | 91.42 | 0.9393 | 232.09 |
| Ours - TC | 96.45 | 90.09 | 0.9696 | 214.42 |
| Ours – RI3+TC | **98.97** | 96.67 | **0.9766** | 296.45 |
| 07 | Removert - RM3 | 99.68 | 77.48 | 0.8719 | 413.16 |
| Removert - RM3+RV2 | **99.76** | 75.93 | 0.8622 | 508.50 |
| ERASOR | 93.80 | 97.29 | 0.9551 | **65.96** |
| Ours - RI3 | 94.41 | 92.67 | 0.9357 | 121.27 |
| Ours - TC | 94.68 | 87.56 | 0.9082 | 118.68 |
| Ours – RI3+TC | 94.74 | **97.66** | **0.9618** | 150.24 |

*E. Limitation*

(实例分割错误，动态去除不完整)

Reference

|  |  |
| --- | --- |
| [1] | Zhang, Ji, and Sanjiv Singh. "LOAM: Lidar odometry and mapping in real-time." Robotics: Science and Systems. Vol. 2. No. 9. 2014. |
| [2] | Behley, Jens, and Cyrill Stachniss. "Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments." Robotics: Science and Systems. Vol. 2018. 2018. |
| [3] | Shan, Tixiao, et al. "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping." 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2020. |
| [4] | Kim, Giseop, and Ayoung Kim. "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018. |
| [5] | Wang, Han, Chen Wang, and Lihua Xie. "Intensity scan context: Coding intensity and geometry relations for loop closure detection." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020. |
| [6] | Dubé, Renaud, et al. "Segmatch: Segment based place recognition in 3d point clouds." 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017. |
| [7] | Pomerleau, François, et al. "Long-term 3D map maintenance in dynamic environments." 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014. |
| [8] | Hawes, Nick, et al. "The strands project: Long-term autonomy in everyday environments." IEEE Robotics & Automation Magazine 24.3 (2017): 146-156. |
| [9] | Banerjee, Nandan, et al. "Lifelong mapping using adaptive local maps." 2019 European Conference on Mobile Robots (ECMR). IEEE, 2019. |
| [10] | Pagad, Shishir, et al. "Robust method for removing dynamic objects from point clouds." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020. |
| [11] | Hornung, Armin, et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees." Autonomous robots 34 (2013): 189-206. |
| [12] | Schauer, Johannes, and Andreas Nüchter. "The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid." IEEE robotics and automation letters 3.3 (2018): 1679-1686. |
| [13] | Kim, Giseop, and Ayoung Kim. "Remove, then revert: Static point cloud map construction using multiresolution range images." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020. |
| [14] | Lim, Hyungtae, Sungwon Hwang, and Hyun Myung. "ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building." IEEE Robotics and Automation Letters 6.2 (2021): 2272-2279. |
| [15] | Yoon, David, Tim Tang, and Timothy Barfoot. "Mapless online detection of dynamic objects in 3d lidar." 2019 16th Conference on Computer and Robot Vision (CRV). IEEE, 2019. |
| [16] | Elfes, Alberto. "Using occupancy grids for mobile robot perception and navigation." Computer 22.6 (1989): 46-57. |
| [17] | Litomisky, Krystof, and Bir Bhanu. "Removing moving objects from point cloud scenes." Advances in Depth Image Analysis and Applications: International Workshop, WDIA 2012, Tsukuba, Japan, November 11, 2012, Revised Selected and Invited Papers. Springer Berlin Heidelberg, 2013. |
| [18] | Yin, Canben, et al. "Removing dynamic 3D objects from point clouds of a moving RGB-D camera." 2015 IEEE International Conference on Information and Automation. IEEE, 2015. |
| [19] | Ruchti, Philipp, and Wolfram Burgard. "Mapping with dynamic-object probabilities calculated from single 3d range scans." 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018. |
| [20] | Sun, Li, et al. "Recurrent-octomap: Learning state-based map refinement for long-term semantic mapping with 3-d-lidar data." IEEE Robotics and Automation Letters 3.4 (2018): 3749-3756. |
| [21] | Banerjee, Nandan, et al. "Lifelong mapping using adaptive local maps." 2019 European Conference on Mobile Robots (ECMR). IEEE, 2019. |
| [22] | Park, Seungcheol, et al. "Curved-voxel clustering for accurate segmentation of 3D LiDAR point clouds with real-time performance." 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019. |
| [23] | Wong, Kelvin, et al. "Identifying unknown instances for autonomous driving." Conference on Robot Learning. PMLR, 2020. |
| [24] | Behley, Jens, et al. "Semantickitti: A dataset for semantic scene understanding of lidar sequences." Proceedings of the IEEE/CVF international conference on computer vision. 2019. |
| [25] | Douillard, Bertrand, et al. "On the segmentation of 3D LIDAR point clouds." 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011. |
| [26] | Lim, Hyungtae, Minho Oh, and Hyun Myung. "Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3D LiDAR sensor." IEEE Robotics and Automation Letters 6.4 (2021): 6458-6465. |
| [27] | Kashani, Alireza G., et al. "A review of LiDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration." Sensors 15.11 (2015): 28099-28128. |
| [28] | Wohlkinger, Walter, and Markus Vincze. "Ensemble of shape functions for 3d object classification." 2011 IEEE international conference on robotics and biomimetics. IEEE, 2011. |
| [29] | Weinmann, Martin, Boris Jutzi, and Clément Mallet. "Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features." ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2.3 (2014): 181. |
| [30] | Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012. |