**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 simultaneous localization and mapping (SLAM) [7, 8, 9], recent advancements in 3D light detection and ranging mapping have been reported to counter an ambiguous environment. Most autonomous robotic systems always assume that all observations are accessible, however excluding dynamic items as outliers and relying solely on static objects might prevent confusion and increase resilience.

During the process of SLAM utilizing LiDAR in urban environments containing mobile items such as vehicles, pedestrians, etc. [14], we encounter non-static points with limited interpretability in raw scan data. The changes in maps between two registered scans correspond to the difference of point clouds, and existing algorithms [10, 11, 12, 13] also employ this principle to get rid of dynamic objects. However, the gap between maps is divided into two categories: a) moving objects and b) residuals caused by the restricted view. The former can be reflected as the ghost trail effect [10, 13] by sequential accumulations of scan data, whereas the latter may be misinterpreted as false dynamic points due to inconsistent observations in local maps. To address this problem, we present a segment-based dynamic filter that concentrates on processing a batch of measurements from a single-session SLAM. Our primary objective is to generate a static instance map of a particular length without regard to processing speed and to ensure that actual dynamic points are vigorously removed, even if some false negatives occur (i.e., wrongly erased points on natural static objects).

In this paper, the procedure is divided into two stages: a) instance segmentation assisted by intensity, and b) dynamic filtering at object level. Notably, all scan data used in the post-processing are derived from [3], and we deem the process flaws resulting from imprecise pose estimate neglectable. Firstly, we build the pre-processed point cloud into the coordinate system based on curved-voxel and perform clustering segmentation. Compare with [16], we additionally calibrate sensor intensity to refining the outcome and classifying instances by geometric verification. The assistance of intensity compensates for the lack of spatial clustering and groups the occlusion-caused distant, dispersed point clouds from the same object. Secondly, based on semantics in geometric distinguishing, we selectively conduct dynamic identification on possibly mobile objects (e.g., elements contacted with the ground and unknown instances). Furthermore, the two mechanisms are tightly coupled in object tracking to correct imperfect cluster and enhance the accuracy of dynamic removal. Therefore, the multi-frame data is integrated to compensate for the insufficient observation of objects in different screenings.

Above all, we propose a novel post-processing LiDAR reconstruction methodology, called SegDF, a segment-based dynamic filter in the 3D point cloud. Our contributions are threefold:

• An instance segmentation based on curved-voxel clustering assisted with intensity and geometric classification. (§III-B, §III-C)

• A dynamic object-level filter utilizing curved-voxel occupancy searching and instance tracking. (§III-D)

• An innovative approach for instance map initialization and a tight coupling framework for instance refining and dynamic removal. (§III-D) (效果不明显，要不要去掉？？并且处理速度慢)

|  |
| --- |
|  |
| 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. |

**Ⅱ. RELATED WORK**

*A. Clustering segmentation*

According to [17], panoptic LiDAR segmentation is a combination of semantic segmentation for static stuffs and instance segmentation for countable objects. Despite the fact that many researches [18, 19] plunge right into deep learning solutions, point cloud clustering is a well-established research area that has the potential to contribute to the panoptic challenge.

**Clustering with Euclidean Distance**. Using the Euclidean distance to cluster points is a straightforward idea. A novel ground segmenting algorithm was proposed in [22], which clusters other non-ground points with voxelized Euclidean neighbors. In [23], researchers provided a probabilistic framework to incorporate Euclidean spatial and temporal information from consecutive frames. Though it considers the drawbacks of European clustering and combines the continuous frame information to improve the result, it is still based on the redundant point-to-point search.

**Clustering with Supervoxels or Superpoints**. Inspired by the concept of superpixels from traditional image processing [24], some researchers are interested in finding super voxels or super points [25, 26] in Euclidean space. However, unlike superpixels using color or texture due to the sensor type, it is challenging for point cloud to provide manual descriptor with extra information beyond Euclidean distance. In [16], the algorithm attains fine discriminations by considering three important aspects for clustering 3D lidar points: distance from the sensor, directional resolutions, and rarity of points to deal with sparse 3D point clouds. This method is similar to ours, but it still relies on searching in Euclidean space, and its results are highly resolution-dependent.

**Clustering on Range Image**. To improve the clustering rate, many researchers explored more clues to find better criteria for separating neighboring points from distinct clusters. In [27, 28], the angle formed by two adjacent laser beams is considered to construct the discriminator. To make the algorithm fast enough for real-time applications, authors of [27] worked on the 2D range image representation of the lidar point cloud. However, it is still a depth-dominated clustering method that is strongly impacted by the viewing angle or scan line density.

*B. Dynamic object removal*

Dynamic removal is closely related to static map construction. We exclude online approaches in this paper since we are primarily concerned with constructing a high-quality static map without regard to processing speed.

**Conventional approaches**. The remote-sensing community has carried out extensive studies on dynamic point removal as a method for establishing sterile environments, which is essential for constructing or comprehending the surroundings. However, expensive, dense terrestrial laser scanning (TLS) point cloud data with precisely aligned poses are required for use in time-intensive voxel ray casting-based algorithms [11, 29, 30].

**Visibility-based approaches**. Visibility-based approaches have been proposed [31, 32] to alleviate the computational burden. This sort of approach associates a query point (or a cluster) with a map point within nearly the same narrow field of vision (FOV) (e.g., cone-shaped [15]), concluding that the occluded points at subsequent sites should be static.

**Point cloud segmentation-based approaches**. Segment-based methods are also included in the list of related topics. Tools for segmenting point clouds have been developed [33]. Given accurate segment points and dynamic labels, it is simple to construct a map by omitting them [34, 35]. Nonetheless, segment-based algorithms rely largely on supervised labels and are susceptible to human mistake or unknown classes [36]. This point's semantic label must be combined as a prerequisite with the visibility-based process.

**Ⅲ. METHODOLOGY**

The proposed is a post-processing method to reconstruct a static instance map. Assuming that the process errors due to imperfect pose estimation are tolerable, we examine batch data of a certain length in [14] to enhance the quality of the static map.

|  |
| --- |
|  |
|  |

*A. Problem definition*

Given a point cloud map generated from a set of raw lidar scans, we intend to remove dynamic objects in the map. Specifically, we consider the global unified map coordinate *M* and the local sensor query coordinate , where k is the frame index. We assume that the related SE(3) pose T (i.e., the transformation from to ) and the scan data in the local coordinate system are known. Additionally, we describe two groups in M, with D representing the dynamic group and S representing the static group. The mission thus corresponds to eliminating the dynamic objects in . The challenge stated above is phrased formally as

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

where the is the transform in SE(3).

Our methods consist of two phases: *Seg*(...) (i.e., instance segmentation) and *Df*(..., ...). (i.e., dynamic filter). A static instance local map equates to the combination of static instances after dynamic removal between two registered scans, hence the procedure can be stated as

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |
|  | |
| Fig. 3 A visualization of the product (i.e., static/dynamic map segregation) in the point-wise form on the SemanticKITTI [14] 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 [41] for every 2 m  and has many dynamic points from various types of instances (e.g., vehicles, pedestrians, etc.). Our method can successfully separate static instances and dynamic ones, even when urban environments contain a lot of noise. 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. | |

*B. Intensity-assisted curved-voxel clustering*

Sharing the analogous philosophy to the existing approach [16, 22], We perform clustering for segmentation based on curved-voxel and solve difficult-to-cluster split sections by combining intensity information

**Ground extraction**. According to [13, 22], The majority of each frame's raw data consists of ground points, which may affect the segmentation of other instances; consequently, it is necessary to separate the ground before clustering. Moreover, the ground is utterly immobile, thus motion detection should not be conducted on it. We choose the method described in [37] to extract ground from the point cloud, resolving the uneven ground scene by multi-region plane fitting.

**Intensity calibration**. The sensor intensity reading indicates the surface reflectance structure surrounding the sensor. The intensity channel is noisy because it is influenced not only by target surface properties (e.g., surface roughness, surface reflectance), but also by acquisition geometry (e.g., distance) and instrument effects (e.g., transmitted energy) [31]. Calibration is important to limit the influence of other elements, yet, it is impossible to totally recover the true value of the object's substance through the mathematical model. In order to use the intensity in clustering efficiently, ensuring the continuity of the intensity in the adjacent space, we guide or restrict the operation of curved-voxel neighbor.

We solely evaluate the effect of distance , and incidence angle on raw intensity during the scanning period of a single keyframe. To satisfy the intensity we can suppose that is constant, hence the calibrated intensity can be formed as (fig关于邻域voxel内的方差)

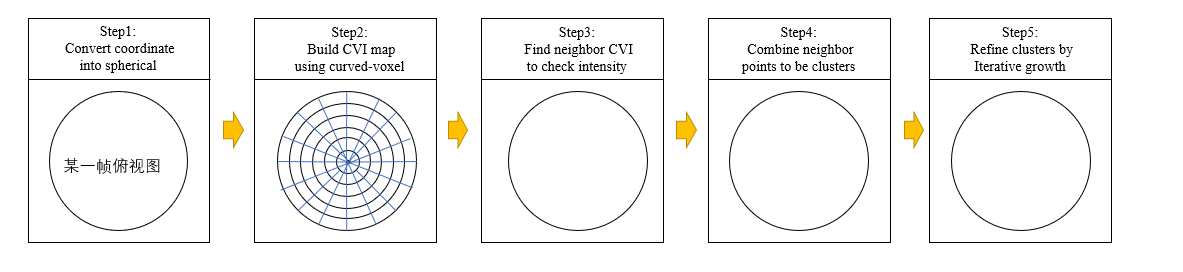
|  |  |
| --- | --- |
|  | (6) |

**Curved-voxel clustering assisted by intensity.** Referring to [16], we transform cartesian coordinates to spherical ones, X = [*ρ*, *θ*, *φ*]，which consists of the radial distance from the sensor , the polar angle, and the azimuth angle . We redefine curved-voxel assisted by intensity as *CVI*, a spatial unit comprised of three constituents: 1) spherical coordinates, 2) the average of intensity *AV*, which characterizes local reflectance, and 3) the covariance of intensity *COV*, which represents the credibility of the neighborhood. The fundamental components of is as follows:

|  |  |
| --- | --- |
|  | (7) |

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

Intensity辅助聚类图



We describe our proposed segmentation method in Algorithm 1. We first implement a preliminary procedure to extract ground by [37] and calibrate intensity (line 1~2). Then we convert the coordinate of no-ground points into spherical ones (line 3) and build a hash table that maps each voxel index to the indices of points within the *CVI* (line 4). To maintain sparse representations of the hash table, only *CVI*s with at least one point are stored. After establishing the hash table, we visit each point to locate neighbor points in 27 (=) voxels with tiny COV surrounding the target CVI, where the AV difference must be less than the threshold, and then group them in to a cluster (line 5~11). After obtaining an initial list of clusters, we adopt a novel iterative algorithm to refine the clustering outcomes. All the clusters are visited to find the neighbor clusters through the neighbor voxel searching, after that, the sets are merged if the strength condition is achieved. The entire procedure is a mode of iterative growth based on cluster (line 12~18).

Compared with [16], in the iterative clustering growth mode, the combination of intensities can merge incompletely observed residual clouds in the region, hence strengthening the consistency of the same instance across multiple scan frames. (fig)

|  |
| --- |
| Algorithm 1: **Curved-voxel clustering assisted by intensity** |
| **Input:** 3D point clouds from a scan data, and  curved-voxel size parameters , and , and  the threshold of average and covariance, *threAV* and *threCOV* |
| **Output:** |
| 1: **Extract ground:** use [37] to remove ground points in |
| 2: **Calibrate intensity:** in local region normal vector incident vector,  use the mathematical model (formula 6) |
| 3: Convert-to-Spherical ( |
| 4: hash-table Build-CVI (*X*, , , , *AV*, *COV*) |
| 5: **for** **do** |
| 6: **if** is already included in a cluster **then** |
| 7: **continue** |
| 8: **end if** |
| 9: Neighbor () Voxel-Searching **and** Satisfy (*threAV*) |
| 10: Initialize () Combine (Neighbor ()) |
| 11: **end for** |
| 12: **for** cluster **do** |
| 13:  **if** Noise-Remove () **then** |
| 14: **continue** |
| 15: **end if** |
| 13: **for** in **do** |
| 14: Neighbors () Voxel-Searching **and** Satisfy (*threAV*, *threCOV*) |
| 15: Neighbors () find cluster in Neighbors () |
| 16: Update () |
| 17: **end for** |
| 18: **return** |

C. Geometric classification

Intensity-assisted curved-voxel segmentation clusters the point cloud into multiple items, but instance segmentation should identify each object. We use geometric features [20, 21] to distinguish clusters to obtain low-level semantics using feature vector, as opposed to learning methods that mainly rely on supervised labels. The dimension of the geometric feature vector is 120, which consists of three parts:

1) Eigenvalue: This segment descriptor is merged into a 17-dimensional feature vector. In accordance with [38], we compute the linearity, planarity, scattering, omni-variance, anisotropy, eigen-entropy, change of curvature measures, and primary orientation. It utilizes the aggregated data of local characteristics to precisely describe the geometric distribution of objects. Because to its susceptibility to noise interference, it is frequently used to compare instances within the same category.

2) Spatial scale: This descriptor consists of Minimum height, Maximum height, and scale in a 13-dimensional feature vector, naturally represents the instance's scale information and can discern global changes with precision.

3) Shape histograms: This 1x640-dimensional feature consists of 10 histograms encoding the shape functions D2, D3, and A3, as defined in [39]. The D2 shape function is a histogram of the distances between randomly selected point pairs, whereas the D3 form function encodes the region between chosen point triplets. The A3 shape function describes the angles between lines derived from triplets.

In comparison to the learning method, the categorization based on geometric characteristics is less robust. To meet the requirements of low semantic information, we classify the clustering conclusions into three categories: a) vegetation (e.g., trees and grass), b) buildings (e.g. walls and fences), and c) potential-mobility objects. In order to intuitively distinguish the three categories for point clouds with limited information, we devised a hierarchical classification criterion. Similar to the appearance of the ground, buildings seem flat in most scans, and due of the low complexity of multi-plane, we extract this category using region growth and restrict it using the planarity and change of curvature. The representation of vegetation is quite complex, and it is difficult to recognize it based solely on spatial scale; therefore, we added thresholds of planarity, omni-variance, and anisotropy to restrict this class. Owing to the inconsistency of observations and the existence of noise in the sweep, the majority of objects related to the ground, such as vehicles, pedestrians, billboards, etc., are possibly in mobility. To enhance the accuracy of dynamic removal, we categorize the aforementioned instances as potential-mobility objects, indicating that they are potential of motion. (fig)

D. *Dynamic removal in object tracking*

To remove dynamic instances, we apply an object tracking-based strategy. Only potential-mobility elements are identified throughout this process. This concept 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) 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).

这里需要一个类似于segmatch中的图进行上下比较，下图截图segmatch，将某个聚类物体分割、将多个聚类物体融合的图

|  |
| --- |
| 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 [40] and the SemanticKITTI dataset [14] 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, the number of static points in the map is too much larger than the number of 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 [14] 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.

|  |
| --- |
|  |
| Fig. 4 The visualization of result using the evaluation criteria on SemanticKITTI [14] sequence 09 from frame 1325 to frame 1375. 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). Furthermore, we annotate FP and FN to highlight them. |

*B. Parameter settings*

We used LiDAR SLAM-based poses predicted by SuMa [41] following the various sequences SemanticKITTI [14] to prove the feasibility of the proposed method in noisy situations and consider less accurate pose estimation.

In the process of front-end instance segmentation, the setting of parameters is very important. Trying to maintain more original points on the premise of ensuring the segmentation accuracy, we set the LiDAR visible minimum distance is 2 m and maximum distance is 50 m. Additionally, we set , , and the threshold , . During backend dynamic filtering, since two consecutive frames will cause inconsistent observations due to viewing angle problems, we set the minimum pose difference is 2 m and the smallest curved-voxel occupancy is 0.6.

*C. 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 TP, FP, TN, and FN represent true positive, false positive, true negative, and false negative point sets, respectively.

Using this equation, we can redefine the problem as reducing the number of FP and FN points within the static and dynamic estimates.

We used both SemanticKITTI’s ground-truth map and our resulted map voxel down-sampled with 0.05 m size cell. In the specific evaluation process, we adopt the method of point-to-point search from the ground truth to the dynamic and the static, and the search radius is 0.1m. For the sake of fairness, we will register the above three in the same coordinate system and use the same search radius, which means that the estimated value of each original point is determined by voting in the neighborhood. In Fig. 4, to quantitatively depict the results of dynamic filtering, the TP, FP, TN, and FN points are indicated in green, pink, blue, and orange, respectively.

此处有TP/FP/FN/FP的图片、局部动态剔除的图、MulRan的图

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] | Shan, Tixiao, and Brendan Englot. "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 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] | Durrant-Whyte, Hugh, and Tim Bailey. "Simultaneous localization and mapping: part I." IEEE robotics & automation magazine 13.2 (2006): 99-110. |
| [8] | Bailey, Tim, and Hugh Durrant-Whyte. "Simultaneous localization and mapping (SLAM): Part II." IEEE robotics & automation magazine 13.3 (2006): 108-117. |
| [9] | Cadena, Cesar, et al. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age." IEEE Transactions on robotics 32.6 (2016): 1309-1332. |
| [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] | 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. |
| [12] | 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. |
| [13] | 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. |
| [14] | 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. |
| [15] | Pomerleau, François, et al. "Long-term 3D map maintenance in dynamic environments." 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014. |
| [16] | 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. |
| [17] | Zhao, Yiming, Xiao Zhang, and Xinming Huang. "A technical survey and evaluation of traditional point cloud clustering methods for lidar panoptic segmentation." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021. |
| [18] | Gasperini, Stefano, et al. "Panoster: End-to-end panoptic segmentation of lidar point clouds." IEEE Robotics and Automation Letters 6.2 (2021): 3216-3223. |
| [19] | Hong, Fangzhou, et al. "Lidar-based panoptic segmentation via dynamic shifting network." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. |
| [20] | Wohlkinger, Walter, and Markus Vincze. "Ensemble of shape functions for 3d object classification." 2011 IEEE international conference on robotics and biomimetics. IEEE, 2011. |
| [21] | 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. |
| [22] | Douillard, Bertrand, et al. "On the segmentation of 3D LIDAR point clouds." 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011. |
| [23] | Held, David, et al. "A Probabilistic Framework for Real-time 3D Segmentation using Spatial, Temporal, and Semantic Cues." Robotics: Science and Systems. Vol. 12. 2016. |
| [24] | Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." IEEE transactions on pattern analysis and machine intelligence 34.11 (2012): 2274-2282. |
| [25] | Ben-Shabat, Yizhak, et al. "Graph based over-segmentation methods for 3d point clouds." Computer Vision and Image Understanding 174 (2018): 12-23. |
| [26] | Landrieu, Loic, and Mohamed Boussaha. "Point cloud oversegmentation with graph-structured deep metric learning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. |
| [27] | Bogoslavskyi, Igor, and Cyrill Stachniss. "Fast range image-based segmentation of sparse 3D laser scans for online operation." 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016. |
| [28] | Yuan, Xia, Yangyukun Mao, and Chunxia Zhao. "Unsupervised segmentation of urban 3d point cloud based on lidar-image." 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2019. |
| [29] | Chen, Chi, and Bisheng Yang. "Dynamic occlusion detection and inpainting of in situ captured terrestrial laser scanning point clouds sequence." ISPRS Journal of Photogrammetry and Remote Sensing 119 (2016): 90-107. |
| [30] | J. Gehrung, M. Hebel, M. Arens, and U. Stilla, “An approach to extract moving objects from mls data using a volumetric background representation,” ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 4, p. 107, 2017. |
| [31] | 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. |
| [32] | Ambruş, Rareş, et al. "Meta-rooms: Building and maintaining long term spatial models in a dynamic world." 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014. |
| [33] | Milioto, Andres, et al. "Rangenet++: Fast and accurate lidar semantic segmentation." 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2019. |
| [34] | 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. |
| [35] | 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. |
| [36] | Wong, Kelvin, et al. "Identifying unknown instances for autonomous driving." Conference on Robot Learning. PMLR, 2020. |
| [37] | 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. |
| [38] | Wohlkinger, Walter, and Markus Vincze. "Ensemble of shape functions for 3d object classification." 2011 IEEE international conference on robotics and biomimetics. IEEE, 2011. |
| [39] | 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. |
| [40] | 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. |
| [41] | Behley, Jens, and Cyrill Stachniss. "Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments." Robotics: Science and Systems. Vol. 2018. 2018. |
| [42] | G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, “MulRan: Multimodal Range Dataset for Urban Place Recognition,” in Proc. IEEE Intl. Conf. on Robot. and Automat., Paris, May 2020. |