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

**Abstract**

Dynamic objects in lidar data often leave unwanted traces in 3D point clouds, affecting the quality of maps and localization performance. We propose a novel static reconstruction method, called SegDF, for focusing on filtering the dynamic objects on the ground with high mobility in urban environment. Given pairs of registered scans, we build a curved-voxel map assisted by sensor intensity to cluster them into instances and obtain semantics through geometric verification. Meanwhile, we represent the occupancy state of the volume of space by curved-voxels to remove dynamic points in object level. Furthermore, our approach tightly couples the instance segmentation and dynamic detection to refine the imperfect clustering and improve the removal accuracy in the process of building a static instance map. We validate SegDF on the KITTI dataset using SemanticKITTI as ground truth and prove that it works well in real urban streets.

**Ⅰ. INTRODUCTION**

Recent advances in 3D light detection and ranging (LiDAR) mapping have been reported leveraging lidar odometry [1, 2, 3], place recognition [4, 5, 6], and simultaneous localization and mapping (SLAM) [7, 8, 9]. Most autonomous robotic systems always assume that all observations are available, but only relying on static objects while rejecting dynamic objects as outliers can avoid confusion and achieve robustness.

During a process of SLAM using lidar in urban sites containing various dynamic objects like vehicles, pedestrians and so forth[14], thus we encounter non-static points with low interpretability in a raw scan data. The changes of maps are equivalent to the diversity of point cloud between two registered scans, and existing algorithms [10, 11, 12, 13] also utilize this idea to remove dynamic objects. Unfortunately, the difference of maps are divided into two categories: a) moving object and b) residual caused by restricted view as in Fig.1(体现遮挡和动态残影), the former can be reflected as ghost trail effect [10, 13] by sequential accumulations of the scan data, but the latter might be misjudged as false dynamic points by the association errors in local maps. To tackle this problem, we propose a segment based dynamic filter, which processes a batch of measurements from one single-session SLAM and, unlike sequential Bayesian updates of [15], focuses on the post-processing role, because our main purpose is to construct a certain length of static object map without concerning the processing speed, and to ensure that actual dynamic objects are strongly erased even if some false negatives occur (i.e., points on actual static objects are wrongly erased).

In this paper, our method divides the process into two stages: a) instance segmentation assisted by intensity and b) dynamic filtering in object level. It is worth mentioning that all scan data used in the post-processing process are generated from [3], and we consider the process errors due to imperfect pose estimation is tolerable. Firstly, we fill the pre-processed original point cloud into the coordinate system built by curved-voxel and perform clustering segmentation, which is similar to [16], but the difference is that we utilize sensor intensity calibrated additionally to refine the result and classify instances through geometric verification. The assistance of intensity makes up for the deficiency of clustering in [16], and cluster the distant divisive point clouds from the same object caused by the occlusion problem. Secondly, according to the semantics, we selectively execute dynamic detection on the objects with high mobility (e.g., moving objects contacted to the ground and unknown instance after geometric distinguishing). Furthermore, the two mechanisms are coupled tightly in the process of object tracking to refine the imperfect clustering and improve the tracking and removal accuracy. Therefore, the multi-frame data is integrated to compensate for the insufficient observation of objects in different screening.

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

• An instance segmentation method based on curved-voxel clustering assisted by sensor intensity calibrated and geometric verification. (§III-B, §III-C)

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

• A novel initialization method for instance map and a tight coupling scheme for instance refine and dynamic removal. (§III-E)

**Ⅱ. RELATED WORK**

*A. Panoptic segmentation*

As reported in [17], lidar panoptic segmentation is an ensemble of both the semantic segmentation for static stuff and the instance segmentation for countable objects. Though many researches [18, 19] directly dive into the deep learning solutions, despite the semantic classification part, point cloud clustering is a long-existing research topic that also has a chance to contribute as part of the panoptic task.

**Clustering with Euclidean Distance**. Using the Euclidean distance to cluster points is a straightforward idea. A novel ground segmenting algorithm was proposed in [22], other non-ground points were clustered with voxelized Euclidean neighbors. In [23], researchers provided a probabilistic framework to incorporate not only the Euclidean spatial information but also the temporal information from consecutive frames. It firstly takes account of the shortcomings of European clustering and fuses the continuous frame information to improve the result, but it is still a point-to-point search, which makes it redundant.

**Clustering with Supervoxels or Superpoints**. Inspired by the concept of superpixels from the traditional image processing [24], some researchers are interested in finding super voxels or super points [25, 26] in the Euclidean space. However, unlike superpixels using color or texture due to the sensor type, point cloud is hard to give these manual units more valuable information getting rid of 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 solely on geometric information and depends heavily on parameter setting.

**Clustering on Range Image**. To improve the rate of clustering, many researchers explored more clues aiming at finding better criteria to separate neighbor points belong to different clusters. Representatively, 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, especially when affected by the angle of view or the scanning line is sparse, it cannot remove dynamic points completely.

*B. Dynamic object removal*

Static map construction is closely related to dynamic removal. Between online and offline approaches, we exclude online in this paper because we are mainly interested in producing a high-quality static map without concerning the processing speed.

**Conventional approaches**. The remote-sensing community has widely investigated dynamic point removal as building up a pure environment’s structures is important to the construction or understanding of the environment. In general, however, it requires high-cost, dense terrestrial laser scanning (TLS) point cloud data with accurately aligned poses to apply in time-consuming voxel ray casting-based methods [11, 29, 30].

**Visibility-based approaches**. To alleviate the computational burden, visibility-based methods have been proposed [31, 32, 33]. This type of method associates a query point (or a cluster) and a map point within an almost same narrow field of view (FOV) (e.g., cone-shaped [15]), then checks which is located further away and concludes the occluded points at further sites should be static.

**Point cloud segmentation-based approaches**. With correct segment points with dynamic labels, constructing a map is straightforward by excluding them [34, 35]. However, segmentation-based approaches currently rely heavily on the supervised labels and are vulnerable to human error or unknown classes [36].

**Ⅲ. METHODOLOGY**

The proposed is a post-processing method to reconstruct a static instance map. Assuming the process errors due to imperfect pose estimation is tolerable, we examine batch data of a certain length as in [14] and the lidar odometry [3] to enhance the quality of the static map.

*A. Problem definition*

**Setup and notations**. Given a point cloud map constructed using a set of raw lidar scans, we aim to remove dynamic objects within the map. In doing so, we mainly consider two different coordinates, global unified map coordinate M and a local sensor query’s coordinate (k is the index of a frame). We assume the associated SE(3) pose (i.e., the transformation from to M) and a scan data in the local coordinate is known by [3]. We define two states in *M*, dynamic status is represented by *D* and static status is *S*. Thus, the mission is equivalent to remove the dynamic objects in . Formally, the aforementioned problem is expressed as

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

where is the transform in SE(3).

**Algorithms declaration**. Our algorithms are divided into two stages as (i.e., instance segmentation) and (i.e., dynamic filter). A static instance map is equivalent to combination of static instances after dynamic removal between two registered scans, so the process is expressed as

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

**Evaluation criteria**. In this paper, we refer 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. However, not all TP points are considered, so our goal is to the value of .

B. Curved-voxel clustering with intensity

Sharing the analogous philosophy to the existing approach [16, 22], we perform clustering for segmentation based on curved-voxel and solve split parts that are difficult to cluster by fusing intensity information.

**Ground extraction**. Inspired by [22, 13], the ground in the point cloud interferes with clustering and dynamic removal, we choose the method in [37] to extract ground from point cloud.

**Intensity calibration**. The intensity value affected by multiple factors with noise [31], so it is difficult to completely restore the true value through the mathematical model to reflect the material of the object precisely. In order to use the intensity in the clustering efficiently, we guide or restrict the clustering process of curved-voxel nearest neighbor searching by ensuring the continuity of the intensity in the adjacent space. The specific mathematical model is expressed as

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

where represent the raw value of intensity calibrated, the raw value of intensity, the reflected distance and the incident angle related to curvature.

**Curved-voxel clustering assisted by intensity.** Refer to [16], we convert cartesian coordinates into spherical ones, X = [*ρ*, *θ*, *φ*] (i.e., the radial distance from the sensor , the polar angle, the azimuth angle ,), we redefine curved-voxel assisted by intensity as CVI, a spatial unit consisting of three-dimensional spherical coordinates with the average (AV) and covariance (COV, represents the credibility of the intensity in the neighborhood) of intensity. The *i*, *j*, and *k*-*th* contains points in a spherically shaped voxel as follows:

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

where is in in spherical coordinate with the radial distance *ρ*, polar angle *θ*, average intensity *AV* and covariance *COV*. , and are unit size parameters for each spherical direction.

We describe our proposed segmentation method in Algorithm 1. We first execute a pre-processing process to extract ground by [37] and calibrate intensity (line 1~2). Then we convert the coordinate of no-ground points into spherical one which consists of *ρ*, *θ*, *φ* (line 3) and build a hash table that maps each curved-voxel index to indices of points inside the CVI and calculate average and covariance of intensity (line 4). Note that we maintain sparse representations of the hash table, storing only voxels that contain at least a point, to generate a space-efficient hash table that contains information of only non-empty CVIs. After building the hash table, we visit each point to find neighbor points in 27 (=) voxels surrounding the target CVI, while the difference of neighbor CVIs’ AV from target CVI must be smaller than the threshold of average values, and combine them as a cluster to update the list of clusters (line 5~11). After getting an initial list of clusters, we design a novel method to solve a long-distance non-clustering problem. We visit each cluster that passes the noise selection, use its CVIs to find neighbor CVIs in a larger size of 48 (4) which satisfy the average difference and covariance conditions, checking labels in these neighbors to get the neighbor clusters of target cluster and combine them as a new cluster to get the final list of clusters (line 12~18).

Compared with [16], with the assistance of intensity, the proposed can cluster parts of the same object separated due to occlusion as in Fig (图片IOU).

|  |
| --- |
| Algorithm 1 **Curved-voxel clustering assisted by intensity** |
| **Input:** 3D point clouds *PC* from a scan data, and  curved-voxel size parameters , and , and  the threshold of average and covariance of intensity *threAV*, *threCOV* |
| **Output:** list of clusters of points |
| 1: **Extract ground:** use patchwork to remove ground in the raw point cloud |
| 2: **Calibrate intensity:** number n of no-ground points search neighbors by kdtree,  in local region normal vector incident vector,  use mathematical model (formula 6) to correct values |
| 3: convert-to-spherical |
| 4: **hash-table** build-hash-table {X, , , , *AV*, *COV*} |
| 5: **for** every point **do** |
| 6: **if** is already included in a cluster **then** |
| 7: **continue** |
| 8: **end if** |
| 9: CVI() neighbors(*k*) find in hash-table  **and** | CVI()**.***AV -* CVI()**.***AV | threAV* |
| 10: list of clusters combine-clusters (, in neighbors (*k*)) to update |
| 11: **end for** |
| 12: **for** every cluster C in list **do** |
| 13:  **if** NoiseRemove(C) **then** |
| 14: **continue** |
| 15: **end if** |
| 13: **for** every CVI in cluster **do** |
| 14: CVI() neighbors (*k*) | CVI()**.***AV -* CVI()**.***AV | threAV*  **and**CVI()**.** *COV threCOV* |
| 15: cluster C neighbors (*k*) CVI() neighbors (*k*) contains |
| 16: list of clusters combine-clusters (*C*, in neighbors (k)) to update |
| 17: **end for** |
| 18: **return** list of clusters |

C. Geometric verification

Curved-voxel clustering assisted by intensity just cluster the point cloud into several objects, but instance segmentation needs to label each point. Unlike learning methods that rely heavily on the supervised labels, after §III-B, we use geometric features [20, 21] to distinguish clusters to obtain a low-level semantics by feature vector ()

Eigenvalue based: This descriptor of the segment’s point cloud are computed and combined in a feature vector of dimension . We compute the linearity, planarity, scattering, omnivariance, anisotropy, eigenentropy and change of curvature measures as proposed in [38].

Space information: This part is composed by Minimum height, maximum height and scale in a feature vector of dimension .

Ensemble of shape histograms: This feature of dimension 1x640 is made of 10 histograms which encode the shape functions D2, D3 and A3 as described in [39]. The D2 shape function is a histogram of the distances between randomly selected point pairs while D3 encodes the area between randomly selected point triplets. The A3 shape function describes the angles between two lines which are obtained from these triplets. (图片？)

Given , we classify the example results into buildings, trees and highly dynamic objects

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] | Xiao, Wen, et al. "Street environment change detection from mobile laser scanning point clouds." ISPRS Journal of Photogrammetry and Remote Sensing 107 (2015): 38-49. |
| [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] |  |