

Static Map Generation from 3D LiDAR Point Clouds Exploiting Ground Segmentation

Mehul Arora Louis Wiesmann Xieyuanli Chen* Cyrill Stachniss

ARTICLE INFO

Keywords:

Map Cleaning
Ground Segmentation

ABSTRACT

A clean and reliable map of the environment is key for a variety of robotic tasks including localization, path planning, and navigation. Dynamic objects are an inherent part of our world, but their presence often deteriorates the performance of various mapping algorithms. This not only makes it important but necessary to remove these dynamic points from the map before they can be used for other tasks such as path planning. In this paper, we address the problem of building maps of the static aspects of the world by detecting and removing dynamic points from the source point clouds. We target a map cleaning approach that removes the dynamic points *and* maintains a high quality map of the static part of the world. To this end, we propose a novel **offline** ground segmentation method and integrate it into the OctoMap to better distinguish between the moving objects and static road backgrounds. We evaluate our approach using SemanticKITTI for both, dynamic object removal and ground segmentation algorithms as well as on the Apollo dataset. The evaluation results show that our method outperforms the baseline methods in both tasks and achieves good performance in generating clean maps over different datasets without any change in the parameters.

1. Introduction

Clean and reliable maps play an essential role in autonomous driving applications. The quality of the map can influence the performance of downstream tasks like pose estimation, localization, path planning, etc. Many different types of sensor data are used for generating maps, e.g. monocular images [16], stereo pairs [33], or LiDAR scans [5, 42, 47]. In this paper, we address the problem of detecting and eventually removing dynamic measurements in 3D LiDAR data and generate static point cloud maps.

In a typical driving environment, besides the static parts of the scene, there are usually many moving objects such as vehicles, pedestrians, or bicyclists. Traditional online simultaneous localization and mapping (SLAM) methods [5, 47] suffer from such dynamic objects and generate maps with so-called "ghost artifacts" as shown in Fig. 1, which makes the maps difficult for later use. Various approaches have been proposed to tackle the problem of dynamic point removal in LiDAR point cloud maps. Broadly speaking, one can classify them into two main types; removing dynamic objects while the construction of map [10, 12, 20, 46] and removing dynamic objects after map generation [21, 34]. The latter are offline methods that can leverage more information and, therefore, usually have better performance in detecting and removing dynamic objects in the point cloud map.

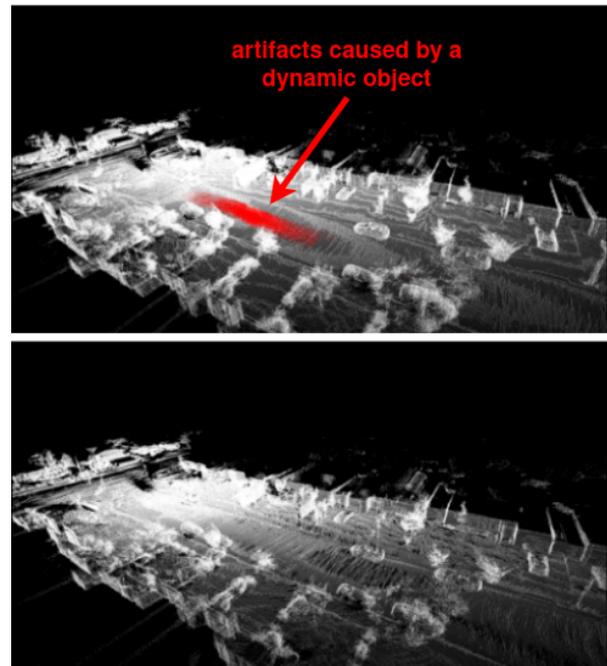


Figure 1: The figure represents the LiDAR point cloud before (above) and after (below) applying our approach. The red points denote the dynamic and the white points the static ones. The point cloud data is from the Apollo sequence 04 frames 1500 to 1780.

All authors are with the University of Bonn, Germany. Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

* corresponding author: xieyuanli.chen@igg.uni-bonn.de

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob, and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony).

ORCID(s):

The main contribution of this paper is a novel dynamic points detection and removal method to generate clean LiDAR point cloud maps. The input of our method is the raw point clouds together with the estimated odometry from a LiDAR SLAM method, SuMa [5], and the output is the point cloud map with voxel-wise binary labels, either static or dynamic. Our method first applies ground segmentation to distinguish between the ground and non-ground points

and then utilizes OctoMap [20] to distinguish between free and occupied space in a probabilistic fashion. The initially segmented ground points are then fed to OctoMap and marked as occupied. OctoMap [20] provides a static and a dynamic map, as well as some unknown points, which are added into the static or dynamic map based on a k-Nearest Neighbor (kNN) voting algorithm. Combining OctoMap with the proposed ground segmentation method, our approach removes most dynamic objects while, at the same time, keeping enough static parts to build a clean and complete map.

In sum, we make three key claims: Our approach is able to (i) generate clean point cloud maps by removing dynamics from the scene, by (ii) a probabilistic, neighborhood aware segmentation, with a novel pre-ground segmentation algorithm to better preserve the static environment, which (iii) generalizes well over different datasets obtained from different environments. These claims are backed up by the paper describing our approach and our experimental evaluation.

The source code of our approach can be accessed at: <https://github.com/PRBonn/dynamic-point-removal>.

2. Related Work

Various approaches have been proposed to remove dynamic objects and clean the maps. In this work, we focus on generating a static map using only LiDAR point clouds. Since the proposed method has two steps, ground segmentation, and point cloud cleaning, we, therefore, discuss the related work twofold.

2.1. Ground Segmentation

Ground segmentation is a key building block for many autonomous systems like traversable analysis and navigation [7, 29]. In this context, elevation maps have been widely used to monitor the surface [1] but also to distinguish between ground and non-ground. Some approaches investigated their height distribution [3, 15], or combined them with classical image segmentation methods [2]. A common heuristic for segmenting the ground is the assumption of a planar horizontal surface. Principal Component Analysis (PCA) [13, 43] and random sample consensus (RANSAC) [17] are common tools to estimate the plane, which can then be used to segment the points based on the distance. Due to slopes in the terrain, it has been beneficial to divide the scene into multiple areas and only assume local planarity [24, 30]. Markov random fields have been used to model the dependencies between cells on a grid level [7, 35, 40].

When dealing with point clouds from the LiDAR scanner, one can exploit additional sensor-specific information. Polar coordinates deal better with the distant dependent point distribution in the measurement process [9, 14, 19, 23, 24, 41]. A range image representation allows for fast neighborhood search [6, 22, 28].

Recently more and more deep learning-based dense semantic segmentation methods emerged. Convolutional

neural networks in the point cloud domain often operate either on sparse voxel grids [44, 38], range images [28, 22] or directly on point level [39]. These typically perform well in areas that are similar to the training set, but worse when facing different sensor modalities or setups. Additionally, they require a lot of training data which is usually expensive to acquire.

2.2. Static Map Generation

Related approaches which are focusing on static map generation can be mainly classified into three different types, namely, segmentation-, visibility-, and ray tracing-based methods.

Some segmentation algorithms provide a dense semantic segmentation [22, 28] for every single scan, which allows filtering the points already before integrating them into the map. For this, one can directly remove all points which belong to movable classes like vehicles and pedestrians. Instead of removing all potential moving objects, Chen et al. [12] propose a LiDAR-based semantic SLAM method that combines both semantic and geometric information to detect and remove the moving objects, while leaving the static objects on the map. Recently, there are also approaches focusing directly on predicting the moving objects [10, 32, 27, 37], which typically requires significantly less labeling effort for obtaining the training data. Non-learning-based methods typically cluster the point clouds and utilize multiple scans to distinguish between moving and static parts [11, 25, 45, 46].

Visibility-based approaches check for the query to map associations and are based on the assumption that if the query point in consideration is detected beyond an already existing point then the considered query point is deemed as dynamic [21, 23, 34]. This kind of methods can easily be corrupted by motion ambiguities which cause errors. Kim et al. [21] provide an offline approach and clean maps using multiresolution range images, in which the finer resolution is used to remove the dynamic points while the coarser resolution reverts the wrongly classified static points. Another different approach that used the complete map as input is given by Lim et al. [23] by comparing descriptors. They divide a single point cloud into smaller sectors and determine the descriptors respectively. The descriptors are matched with the one calculated for the region in the map. Lim et al. [23] denote the importance of ground segmentation and exploit that most of the dynamic objects are connected to the ground.

Ray tracing methods deal naturally with dynamic objects when building the map by updating the occupancy probability of the space along the ray. Space that has been occupied at some point by a dynamic object gets freed the more other rays pass through it. The relatively big memory demand of voxel grid-based methods [36] can be bypassed through octomaps [20, 31]. The drawback of ray tracing methods is the discretization which can lead to errors and usually removes also some static parts. Our approach builds on top of OctoMap [20] and tries to tackle the problem

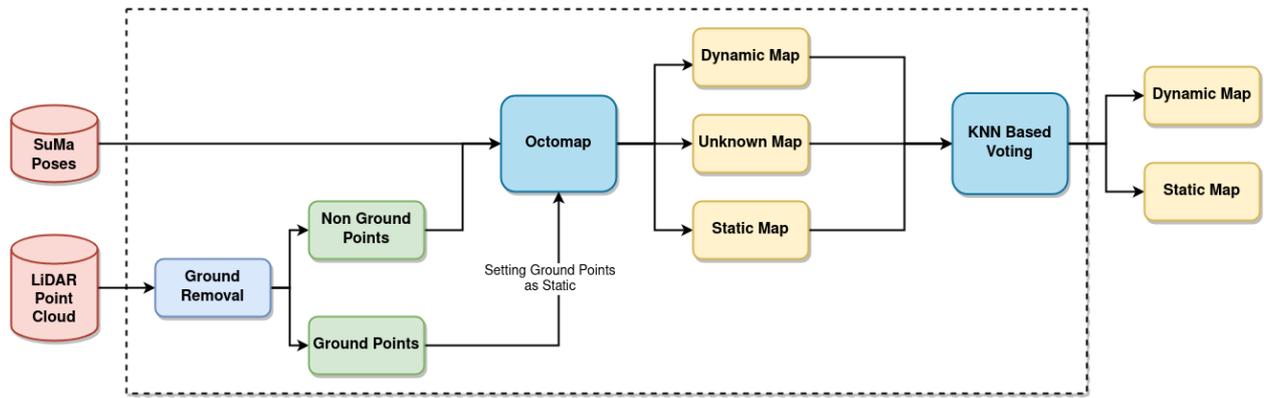


Figure 2: The figure shows a complete pipeline of our approach. The input of our method is the LiDAR scans together with the poses estimated by SLAM. In the preprocessing module i.e. the ground segmentation module, our method uses a heightmap and canny edge detector to provide ground and non-ground points. After feeding all points to OctoMap, we assign all octants with ground points as occupied. This results in static, dynamic, and unknown point cloud maps. The ambiguity of the unknown points is resolved by using kNN-based voting.

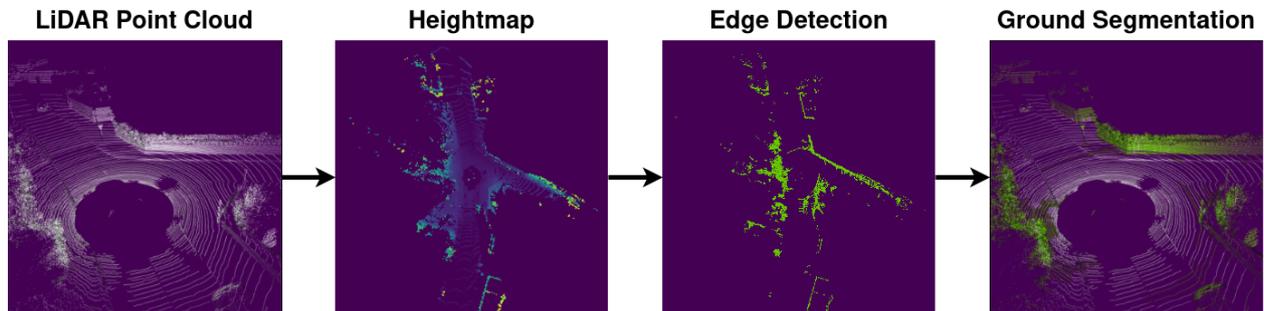


Figure 3: The figure demonstrates the procedure of our ground removal method, a heightmap is generated using the 3D LiDAR scan, which is then passed through the Canny Edge Detector, which as a result provides us with the edges (non-ground points). On the right-hand side, we visualize the segmentation result (ground points in white, non-ground in green).

of falsely classifying static points by combining it with segmentation-based techniques. We assume that the ground is static which allows us to utilize ground segmentation [24] to label the space of the ground points as occupied.

3. Our Approach

In this paper, we propose a point cloud map cleaning method for which the key steps are illustrated in Fig. 2. First, we preprocess the raw point cloud using ground segmentation approaches, which separates the ground from the non-ground points (see Sec. 3.1). The points are then fed to OctoMap [20], while marking the ground points as occupied, to distinguish between static and dynamic points (see Sec. 3.2). In the end, we use a kNN-based voting scheme to further decide the labels of uncertain estimations generated by OctoMap (see Sec. 3.3).

3.1. Ground Segmentation

We decided in favor of using a ground segmentation for cleaning the point cloud map based on two assumptions. First, a large proportion of the points belong to the ground where we can assume that it is not moving and thus static. Consequentially, we can pre-assign all the ground points

in advance as "static" to reduce the number of false dynamics. Second, most of the moving objects, e.g., vehicles, and humans are connected to the ground. A good ground segmentation can largely reduce the difficulty of dynamic object detection and removal in the following steps.

We propose a novel multiresolution height-map-based ground segmentation algorithm, which we will refer to as HMSeg in the following. A heightmap is a 2.5-dimensional representation that stores the height of the surface in a 2D grid (similar to RGB values in an image; see Fig. 3 for a visualization). First, we define the area of interest and a certain grid resolution around the scan. Afterward, we project each point in the scan onto the 2D grid and store its height in the corresponding cell. We restrict the points up to a certain height τ_h to stop the trees and other objects to overshadow the ground points and hence going undetected. The limit τ_h needs to be adjusted based on the sensor setup. In typical urban driving environments, the value of τ_h does not need to be adjusted for a specific surrounding environment, but it should be adjusted based on the height of the sensor, which for most cars will not change drastically once it has been mounted. Once the heightmap of the point cloud is generated, we apply an edge detection algorithm to find the

non-ground area. We use the Canny edge detector [8], which provides us with the flexibility to distinguish between strong and weak edges and allows the sensitivity to measure them accordingly. A height filter is then applied on the resultant ground points which label the points with a height above a certain threshold as non-ground. The same approach is applied using different resolutions, which provide probable ground points for each resolution respectively. The final label for each point is then achieved by voting through the probable ground points. The main motivation behind using an edge detection algorithm on the heightmap is that when viewing the point cloud from a bird's eye view, one can easily detect the boundary edges in the grid due to the sparsity of the heightmap. Here, the discrimination in strong and weak edges is used to discard the ground points (weak edges) and remain the non-ground (strong edges), which eventually helps in segmenting the ground.

Due to the highly modularized nature of our proposed framework, our method also works well with other ground segmentation methods. Our proposed ground segmentation algorithm does not assume the sensor or measuring process. Having additional information about the kind of sensor, one can use more sophisticated methods which exploit the sensor characteristics. To this end, we also test our method combined with the state-of-the-art ground segmentation method, Patchwork, proposed by Lim et al. [24], which is specially designed for LiDAR data.

The Patchwork method by Lim et al. [24] can be broken down into three parts, namely, concentric zone model, region-wise ground plane fitting, and ground likelihood estimation. The concentric zone model divides the point cloud, first into k zones depending on its vicinity to the LiDAR sensor and by this exploits the sparser point density at larger distances. Then, the region-wise ground plane fitting is applied to divide the point cloud further into n bins/regions and a plane is fitted to each of the bins by a PCA. The idea behind breaking the point cloud into different bins is that each bin is considered very small as compared to the whole point cloud and as a result, is safe to assume it to be planar. The region-wise ground plane fitting is an iterative method that detects a set of inlier points and constructs the covariance matrix using them. PCA is applied to them to find an initial estimate of the ground plane which is then again used to find the inlier points. This process is repeated three times to get the final estimate of the ground plane. All the points on this plane are then considered to be ground points. Once an estimate for the ground points is found, a ground likelihood estimation is conducted to further verify the ground segmentation results. It consists of three factors, uprightness, elevation, and flatness. Each of these factors takes care of different end-case conditions, which helps to filter out the wrongly classified ground points. Uprightness considers the angle of the normal plane with the XY-axis of the sensor. The elevation is designed to eliminate points belonging to surfaces like the rooftop of a car, while flatness helps with the case where surfaces have steep slopes.

3.2. OctoMap

Once the ground segmentation results for the 3D LiDAR scans are obtained using the desired algorithm, we set those points as static in the Octree of the point cloud created inside the OctoMap [20]. OctoMap is a probabilistic 3D mapping framework based on an octree data structure. This hierarchical tree-based structure represents a cubical volume in each node, so-called octants. Each octant can subsequently be broken down into eight sub-volumes until a specific resolution is reached. The leaf nodes store an occupancy probability $p(x)$, which indicates whether the area is occupied, free, or unknown. While construction, it reduces the occupancy probability for each node along the ray of a measured point and therefore increases the probability of being free. The occupancy probability increases for the Octant in which the ray ends (namely at the actual position of the point). **When fusing multiple LiDAR observations, this process is repeated iteratively for each point in each scan.** This approach naturally deals with dynamic objects since areas that temporarily contain dynamics (and thus have a high occupancy value) will be lowered each time we traverse through it. After the construction, we can query the occupancy status of a certain point by traversing along the tree. Unseen areas and areas where the occupancy status is not clear (occupancy probabilities around $p \approx 0.5$) will be stated as unknown. Due to the relatively high pace in the automotive field, in practice, many nodes of the octree will just be updated a couple of times and thus have no clear occupancy status. These nodes could be either static or dynamic, and therefore, we propose a simple but effective way to deal with those ambiguities in the following section.

3.3. kNN-based Voting

To deal with the problem posed by the dilemma of unknown points, a series of experiments were conducted by either assigning the unknown nodes as static or dynamic but both assumptions lead to errors in one or another way. We propose a k-nearest neighbor-based voting scheme for each unknown point, where we have a majority vote on the labels of its k nearest points. We assign either static or dynamic to the unknown point based on whichever is in abundance. The motivation behind this approach is that dynamic and static objects often occur in clusters of points (e.g., as shown in Fig. 1). Using this, the k nearest points are selected from the static and dynamic map produced by the OctoMap [20]. From those points, a ratio of static and dynamic points is calculated and if it is above a certain threshold then the point is labeled as static and consequentially otherwise as dynamic. In this way, we label all points in the point cloud either dynamic or static, and use the static points to construct a clean map.

3.4. Implementation Details

For the purpose of reproducibility, besides releasing the source code of our approach, we also list and explain all the hyperparameters used in our approach. We preprocess the scans by filtering out all points over 50 m range to reduce the impact of errors in the pose estimation and sensor noise.

Methods	Metrics	KITTI Sequences											
		00	01	02	03	04	05	06	07	08	09	10	Avg.
Removert-R	MCA [%]	79.73	79.83	84.76	89.07	87.03	77.41	76.35	77.73	78.80	92.53	64.27	80.68
	DR [%]	61.24	61.97	70.73	81.26	77.22	56.70	55.58	57.51	62.74	86.67	30.59	64.16
Removert-R&R	MCA [%]	75.46	71.76	80.83	85.18	82.13	69.81	72.75	75.05	78.56	87.83	61.59	76.50
	DR [%]	52.27	45.74	62.80	73.14	66.79	41.00	47.54	51.64	58.98	77.08	24.59	54.78
Octomap-Empty	MCA [%]	79.94	72.59	80.13	76.58	72.21	79.80	76.67	77.92	76.02	78.48	72.35	76.63
	DR [%]	95.44	93.48	98.37	97.99	97.56	96.26	95.63	93.85	93.41	98.10	81.17	94.70
Octomap-Occupied	MCA [%]	80.09	73.22	80.37	76.90	72.42	79.95	76.96	78.06	76.09	78.64	72.47	76.73
	DR [%]	95.44	93.47	98.53	97.99	97.56	96.26	95.63	93.85	93.23	98.10	81.17	94.71
Octomap-kNN	MCA [%]	80.45	73.13	80.26	76.52	72.36	80.52	76.78	77.90	75.62	78.79	72.35	76.78
	DR [%]	96.79	94.12	98.56	97.85	97.69	97.96	96.24	93.81	95.38	98.39	81.21	95.27
Ours-HMSeg	MCA [%]	84.62	81.63	84.98	84.54	71.36	85.21	84.79	82.11	79.44	79.74	77.44	81.44
	DR [%]	92.12	81.78	93.83	95.02	77.07	92.58	88.39	89.15	85.11	85.67	78.92	87.24
Ours-Patchwork	MCA [%]	85.52	82.84	87.38	86.24	80.77	86.45	86.73	83.85	82.51	86.35	78.60	84.30
	DR [%]	93.20	88.27	96.85	95.11	95.90	94.67	93.38	91.96	91.17	97.83	80.17	92.59

Table 1

Quantitative map cleaning results on the KITTI dataset. The Removert-R&R represents the remove and revert version of Removert and Removert-R represents only the remove version of Removert. OctoMap-kNN combines OctoMap with kNN. Assigning the unknown points to static called Octomap-Occupied, and dynamic called Octomap-Empt. Ours-HMSeg means using our proposed ground segmentation while Ours-Patchwork includes the results of our method using the Patchwork ground segmentation method.

Methods	Metrics	Apollo Sequences			
		01	02	03	04
Removert-R	MCA [%]	85.27	78.25	78.78	91.23
	DR [%]	70.81	56.64	57.72	82.76
Removert-R&R	MCA [%]	79.32	76.61	83.22	82.57
	DR [%]	58.87	53.37	66.58	65.3
Octomap-kNN	MCA [%]	67.84	73.71	75.55	76.58
	DR [%]	82.12	96.21	99.75	99.51
Ours-HMSeg	MCA [%]	70.96	81.21	82.11	84.48
	DR [%]	68.39	90.31	93.76	96.9
Ours-Patchwork	MCA [%]	76.61	83.46	84.43	85.59
	DR [%]	79.75	95.6	99.33	99.35

Table 2

Quantitative map cleaning results on the Apollo dataset. Our method is able to outperform Removert in 2 out of 4 sequences while it is able to increase the MCA (Mean Class Accuracy) of OctoMap with only a slight decrease in the DR (Dynamic Recall)

The height threshold for the ground segmentation τ_h is set to be $-1.5\text{ m} < \tau_h < 2.0\text{ m}$ in the local scanner frame. For the canny edge detector, we use 10 and 300 as the weak and strong thresholds. These thresholds divide the edges detected into two categories the weak edges and the strong edges, which provides us with the flexibility to increase or decrease the detection of points as ground and non-ground. In the experiments, only strong edges are given as output. We use $k = 25$ in the kNN voting for assigning labels to the uncertain points of OctoMap. We voxelized the final point cloud with a grid of 10 cm resolution.

4. Experimental Evaluation

The main focus of this work is to remove as many dynamic points as possible, but at the same time preserve the

static parts of the map. We present our experiments to show the capabilities of our method. The experiments support our key claims that our method is able to (i) generate clean point cloud maps by removing dynamics from the scene, by (ii) a probabilistic, neighborhood aware segmentation, with a novel pre-ground segmentation algorithm to better preserve the static environment, which (iii) generalizes well over different datasets obtained from different environments.

4.1. Datasets

We evaluate our method using KITTI Odometry [18] with the labels from SemanticKITTI [4] for the sequences 0-10. The KITTI Odometry dataset provides 3D LiDAR scans recorded from a Velodyne HDL-64E scanner mounted on a car, while the SemanticKITTI dataset provides corresponding point-wise semantic labels in 28 classes, where 6 classes are assigned the attribute moving or non-moving. In this work, we evaluate both ground segmentation and moving object detection. For evaluating the performance of ground segmentation, we treat the classes, "road", "sidewalk", "other-ground", "lane-marking", "vegetation" and "terrain" as ground, while the rest classes are non-ground. For moving object detection, we reorganize all moving object classes into one dynamic class, while the rest 22 classes are treated as static. We use sequence 08 for tuning the hyperparameters and sequence 00-07, 09-10 for evaluation. The labels are only used in the evaluation and are not needed elsewhere.

To verify the generalization ability of our method, we also evaluate the Apollo [26] dataset. The Apollo dataset provides 3D LiDAR scans recorded from Velodyne HDL-64E LiDAR mounted on the roof of a car. Different from the data of KITTI collected in German towns, the Apollo dataset was created in the US city, which typically has a quite different appearance. Since there is no ground

truth static/dynamic labels available for Apollo datasets, we manually label parts of the dataset. We used specific frames, i.e., 6500-7500 for sequence 01, 22300-24300 for sequence 02, 3100-3600 for sequence 03, and 1500-3100 for sequence 04 of the South Bay, Columbia Park as they contain a large number of dynamics after manually annotating the complete dataset for static and dynamic points. We will also release the static/dynamic labels of parts of Apollo datasets in our open-source repository for public research use.

4.2. Metrics

Our proposed approach aims at generating a clean static map, by removing as many dynamic points as possible, but at the same time preserving the static areas. Taking this aim into consideration, we use the following two metrics to evaluate the performance of our approach: (i) Mean Class Accuracy (MCA), and (ii) Dynamic Recall (DR)

$$MCA = \frac{1}{2} \left(\frac{TS}{TS + FD} + \frac{TD}{TD + FS} \right), \quad (1)$$

$$DR = \frac{TD}{TD + FS}, \quad (2)$$

where TS are the true static voxels, TD the true dynamic voxels, FS the dynamic voxels classified as static voxels, and FD the static voxels classified as dynamic voxels. We compute our metrics on the voxelized maps rather than on the point labels themselves to be more independent of the point density and to ensure a regional correct classification. [We assign the voxel the label of the point closest to the voxel center.](#) We use the MCA to quantify our map quality, which can be increased by correctly classifying the static and dynamic parts of the environment. This metric deals naturally with the imbalance of the classes. This is important due to the substantially higher number of static parts. The dynamic recall shows how many of the dynamics we remove, which is especially important for approaches that get deteriorated by those dynamic objects. We use the classical metrics for measuring the performance of ground removal: IoU of Ground, Recall, Precision, and F1-Score.

4.3. Static Map Generation Results

The first experiment evaluates the performance of our approach and supports the claim that it can generate a static map by removing dynamic objects, while at the same time, can preserve the quality of the static map (the absence of holes in the map). The approach was extensively tested on the KITTI [18] sequences ranging from 00 till 10 shown in Tab. 1. We compare several methods with different setups. The Removert-R&R represents the remove and revert version of Removert [21] while Removert-R uses only the remove part. Adding our kNN voting to OctoMap, named Octomap-kNN, instead of either assigning the unknown points to static (Octomap-Occupied) or dynamic (Octomap-Empty) increases the Accuracy and Recall. Combining Octomap-kNN with our proposed ground removal (Ours-HMSeg) increases the map quality by around 5% points.

The ground segmentation can only mark points as static, therefore we can not increase the recall since this would require marking points correctly as dynamic. The upper bound of the recall when using the ground segmentation is therefore OctoMap. Consequentially, when improving the ground segmentation (Ours-Patchwork) we increase the dynamic recall. Our approach outperforms Removert in both accuracies and recalls in most of the sequences.

Fig. 4 presents some qualitative results to provide a deeper insight into the performance of our approach (see Fig. 4). We can see that Removert [21] can preserve more static points (points displayed in green) than our approach but removes substantially fewer dynamic points (blue). The plain OctoMap [20] approach has a very high detection rate of the dynamic points (blue) but it also deteriorates the static map quality by falsely classifying the static points as dynamic (red). Our approach can detect more static points while keeping high accuracy in removing the dynamics.

4.4. Generalization Ability

This experiment supports our claim that our approach generalizes well in a different dataset i.e. Apollo dataset. As shown in Tab. 2, when testing on the Apollo dataset, our method retains the same performance as achieved on the KITTI dataset. We also integrate another ground segmentation method, Patchwork [24], into our approach to show that our proposed method works well with different segmentation methods. Moreover, a better ground segmentation improves the results even further. The modularity of our approach allows us to easily incorporate new and better segmentation methods to boost performance.

According to Tab. 1 and Tab. 2, the improvement by using Patchwork as seen in the results is significant, especially when considering the Apollo Dataset. The accuracy is increased by at least 1.2% for all four sequences, and an increase of as big as 6% is observed in sequence 01. While the increase of overall accuracy is approximately 3% and an average increase in the recall is around 6%. The maximum increase of approximately 9% is noted in sequence 04. There is a similar trend in the KITTI dataset too, as the average increase in accuracy is 3% and an average increase in the recall by approximately 6%. A big increase in the recall of 19% is observed in the KITTI sequence 04. Also, our method achieves approximately 99% recall while maintaining accuracy above 84% for two sequences out of four in the Apollo Dataset.

4.5. Ground Segmentation

This experiment supports the claim that our relatively simple heightmap-based ground segmentation algorithm is well suited for separating the ground from non-ground points. We compare our method against the classical PCA and RANSAC-based methods as well as Patchwork [24] the new state of the art for LiDAR. The results of the experiment are shown in Tab. 3. Our approach is able to outperform the classical baselines. The more sophisticated Patchwork outperforms our method of HM-Seg in all the metrics by exploiting the characteristics of LiDAR scanners. This is

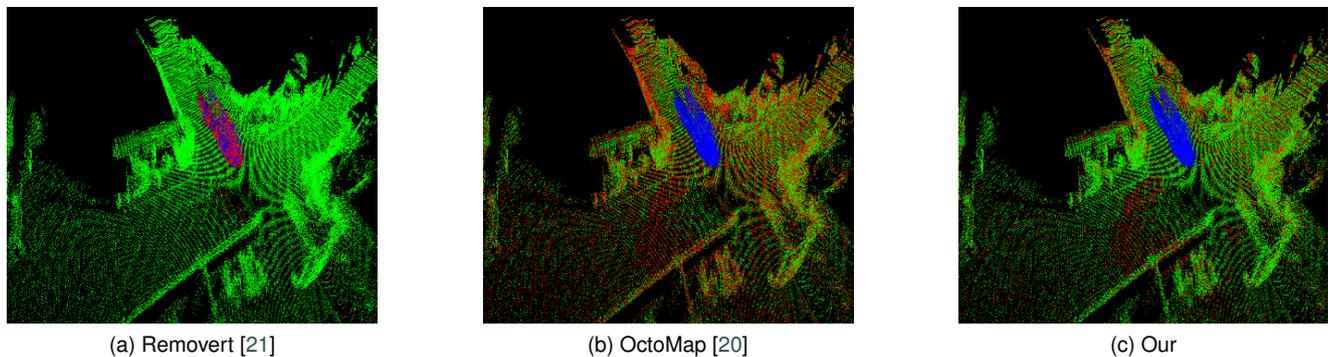


Figure 4: The figure shows a qualitative comparison between Removert [21], OctoMap [20] and our approach. The points displayed in green are the correctly classified static points, blue-colored points represent correctly classified dynamic points and red-colored points denote the wrongly classified points. These results are from KITTI sequence 10.

#	RANSAC Based	PCA Based	Ours	Patch -work
IOU Ground [%]	14.66	43.27	78.46	85.45
IOU Non Ground [%]	60.55	68.22	83.95	87.75
Precision [%]	14.88	45.76	86.53	86.88
Recall [%]	90.74	88.84	89.38	98.10
F1 Score [%]	25.58	60.41	87.93	92.15

Table 3
Ground Segmentation Results

also evident by the results produced in our map cleaning approach and supports our hypothesis that with a better ground segmentation algorithm, the map cleaning results of our method can be further improved. [The proposed ground segmentation method, i.e., HMSeg, is an offline approach with a runtime of approximately 2.33 s per scan, while Patchwork is an online method that has a runtime of 0.022 s per scan.](#)

4.6. Ablation Study

In this experiment, we investigate the impact of the heightmap resolution r on the segmentation accuracy. In Fig. 5, we plot the IoU for ground and non-ground points with dependence on the grid resolution r . The finer the resolution, the better we can classify the non-ground and ground points. As our approach utilizes multiple resolutions to achieve a better segmentation, we have set the list of resolutions in the other experiments of the paper to $r = [0.01m, 0.03m, 0.05m, 0.07m, 0.09m]$. Resolutions over $r > 0.1m$ have a low contribution due to the voxelization of the output.

5. Conclusion

In this paper, we presented a novel approach to remove dynamics and to generate clean maps of the static parts of a scene and proposed a novel ground segmentation algorithm used within mapping. Our approach operates on a ray tracing-based approach as used in OctoMap and exploits

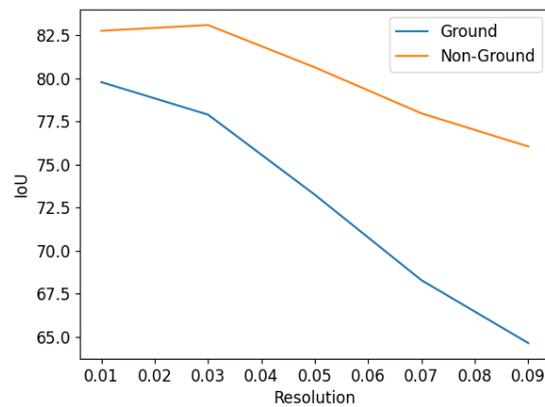


Figure 5: We evaluate the Intersection over Union (IoU) for heightmaps with different resolutions. Our method is configured to use multiresolution for getting a better result than using a single resolution, so we used a set of resolutions, i.e. $[0.01m, 0.03m, 0.05m, 0.07m, 0.09m]$ before the IoU for ground and non-ground went below a certain level.

the assumption of a static ground by segmenting it out in advance. Our heightmap-based segmentation algorithm does not assume a plane ground and is, therefore, better suited to deal with the problems caused by the slope and alleviation. This allows us to successfully generate static maps that are free from most dynamic objects. Our approach performs well on different datasets which speak about its generalisability. We implemented and evaluated our approach using SemanticKITTI and Apollo datasets, provided comparisons to other existing techniques, and supported all claims made in this paper. We also performed experiments to show the performance of the proposed approach on the accuracy of ground segmentation. The experiments suggest that our proposed approach can remove dynamic points while maintaining the quality of the static map hence generated, and better ground segmentation methods can further improve the map cleaning results.

Despite these encouraging results, there is space for further improvements. The main drawback is time consumption. Though our method produces good results but is at the moment a completely offline approach. There are multiple ways to make our method work online. One of those is to train a semantic segmentation model on our output which could infer the results in real-time and would eliminate the need for any manual annotation as our method works equally well on different datasets.

References

- [1] Armbrust, C., Braun, T., Föhst, T., Proetzsch, M., Renner, A., Schäfer, B.H., Berns, K., 2011. RAVON: The robust autonomous vehicle for off-road navigation, in: *Using Robots in Hazardous Environments*, pp. 353–396.
- [2] Arora, M., Wiesmann, L., Chen, X., Stachniss, C., 2021. Mapping the Static Parts of Dynamic Scenes from 3D LiDAR Point Clouds Exploiting Ground Segmentation, in: *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*.
- [3] Asvadi, A., Peixoto, P., Nunes, U., 2015. Detection and Tracking of Moving Objects Using 2.5D Motion Grids, in: *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems*, pp. 788–793.
- [4] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J., 2019. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences, in: *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*.
- [5] Behley, J., Stachniss, C., 2018. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments, in: *Proc. of Robotics: Science and Systems (RSS)*.
- [6] Bogoslavskyi, I., Stachniss, C., 2017. Efficient online segmentation for sparse 3d laser scans. *Photogrammetrie – Fernerkundung – Geoinformation (PGF)*, 1–12.
- [7] Byun, J., Na, K.i., Seo, B.s., Roh, M., 2015. Drivable Road Detection with 3D Point Clouds Based on the MRF for Intelligent Vehicle, in: *Field and Service Robotics*, pp. 49–6.
- [8] Canny, J., 1986. A Computational Approach to Edge Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 8, 679–698. doi:10.1109/TPAMI.1986.4767851.
- [9] Chen, T., Dai, B., Liu, D., Zhang, B., Liu, Q., 2011. 3D LiDAR-based ground segmentation, in: *Proc. of the IEEE Asian. Conf. on Pattern Recognition*, pp. 446–450.
- [10] Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J., Stachniss, C., 2021a. Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters (RA-L)* 6, 6529–6536. doi:10.1109/LRA.2021.3093567.
- [11] Chen, X., Mersch, B., Nunes, L., Marcuzzi, R., Vizzo, I., Behley, J., Stachniss, C., 2022. Automatic Labeling to Generate Training Data for Online LiDAR-based Moving Object Segmentation. *arXiv preprint arXiv:2201.04501*.
- [12] Chen, X., Milioto, A., Palazzolo, E., Giguère, P., Behley, J., Stachniss, C., 2019. SuMa++: Efficient LiDAR-based Semantic SLAM, in: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- [13] Chen, X., Vizzo, I., Läbe, T., Behley, J., Stachniss, C., 2021b. Range Image-based LiDAR Localization for Autonomous Vehicles, in: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- [14] Cheng, J., He, D., Lee, C., 2020. A simple ground segmentation method for lidar 3d point clouds, in: *Proc. of the IEEE Intl. Conf. on Advances in Computer Technology, Information Science and Communications*, pp. 171–175.
- [15] Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A., 2011. On the segmentation of 3d lidar point clouds, in: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pp. 2798–2805.
- [16] Engel, J., Schöps, T., Cremers, D., 2014. LSD-SLAM: Large-scale direct monocular SLAM, in: *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pp. 834–849.
- [17] Fischler, M., Bolles, R., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 381–395.
- [18] Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite, in: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361.
- [19] Himmelsbach, M., Hundelshausen, F., Wuensche, H., 2010. Fast segmentation of 3d point clouds for ground vehicles, in: *Proc. of the IEEE Vehicles Symposium (IV)*, pp. 560–565.
- [20] Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C., Burgard, W., 2013. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots* 34, 189–206.
- [21] Kim, G., Kim, A., 2020. Remove, then revert: Static point cloud map construction using multiresolution range images, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [22] Li, S., Chen, X., Liu, Y., Dai, D., Stachniss, C., Gall, J., 2022. Multi-scale Interaction for Real-time LiDAR Data Segmentation on an Embedded Platform. *IEEE Robotics and Automation Letters (RA-L)* 7, 738–745. doi:10.1109/LRA.2021.3132059.
- [23] Lim, H., Hwang, S., Myung, H., 2021a. Eraser: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building. *IEEE Robotics and Automation Letters* 6, 2272–2279.
- [24] Lim, H., Oh, M., Myung, H., 2021b. Patchwork: concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3D LiDAR sensor. *IEEE Robotics and Automation Letters (RA-L)* 6, 6458–6465.
- [25] Litomisky, K., Bhanu, B., 2012. Removing moving objects from point cloud scenes, in: *Intl. Workshop on Depth Image Analysis and Applications*, pp. 50–58.
- [26] Lu, W., Zhou, Y., Wan, G., Hou, S., Song, S., 2019. L3-Net: Towards Learning Based LiDAR Localization for Autonomous Driving, in: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- [27] Mersch, B., Chen, X., Vizzo, I., Nunes, L., Behley, J., Stachniss, C., 2022. Receding Moving Object Segmentation in 3D LiDAR Data Using Sparse 4D Convolutions. *IEEE Robotics and Automation Letters (RA-L)* 7, 7503–7510. doi:10.1109/LRA.2022.3183245.
- [28] Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, in: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- [29] Na, K., Park, B., Seo, B., 2016. Drivable space expansion from the ground base for complex structured roads, in: *Proc. of the IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC)*.
- [30] Narksri, P., Takeuchi, E., Ninomiya, Y., Morales, Y., Akai, N., Kawaguchi, N., 2018. A slope-robust cascaded ground segmentation in 3D point cloud for autonomous vehicles, in: *IEEE Trans. on Intelligent Transportation Systems (ITS)*, pp. 497–504.
- [31] Pagad, S., Agarwal, D., Narayanan, S., Rangan, K., Kim, H., Yalla, G., 2020. Robust method for removing dynamic objects from point clouds, in: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pp. 10765–10771.
- [32] Patil, P., Biradar, K., Dudhane, A., Murala, S., 2020. An end-to-end edge aggregation network for moving object segmentation, in: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- [33] Pire, T., Fischer, T., Castro, G., De Cristóforis, P., Civera, J., Jacobo Berles, J., 2017. S-PTAM: Stereo Parallel Tracking and Mapping. *Robotics and Autonomous Systems (RAS)* 93, 27 – 42. doi:10.1016/j.robot.2017.03.019.
- [34] Pomerleau, F., Krüsiand, P., Colas, F., Furgale, P., Siegwart, R., 2014. Long-term 3d map maintenance in dynamic environments, in: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.

- [35] Rummelhard, L., Paigwar, A., Nègre, A., Laugier, C., 2017. Ground estimation and point cloud segmentation using spatiotemporal conditional random field, in: *IEEE Trans. on Intelligent Vehicles*, pp. 1105–1110.
- [36] Schauer, J., Nüchter, A., 2018. The peopleremover-removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid. *IEEE Robotics and Automation Letters* 3, 1679–1686. doi:10.1109/LRA.2018.2801797.
- [37] Sun, J., Dai, Y., Zhang, X., Xu, J., Ai, R., Gu, W., Chen, X., 2022. Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation, in: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, IEEE.
- [38] Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S., 2020. Searching efficient 3d architectures with sparse point-voxel convolution, in: *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pp. 685–702.
- [39] Thomas, H., Qi, C., Deschaud, J., Marcotegui, B., Goulette, F., Guibas, L., 2019. KPConv: Flexible and Deformable Convolution for Point Clouds, in: *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*.
- [40] Tse, R., Ahmed, N., Campbell, M., 2012. Unified mixture-model based terrain estimation with markov random fields, in: *Proc. of the IEEE Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 238–243. doi:10.1109/MFI.2012.6343027.
- [41] Velas, M., Spanel, M., Hradis, M., Herout, A., 2018. CNN for Very Fast Ground Segmentation in Velodyne LiDAR Data, in: *Proc. of the IEEE Intl. Conf. on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 97–103.
- [42] Vizzo, I., Chen, X., Chebrolu, N., Behley, J., Stachniss, C., 2021. Poisson Surface Reconstruction for LiDAR Odometry and Mapping, in: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- [43] Wang, M., Tseng, Y., 2011. Incremental segmentation of lidar point clouds with an octree-structured voxel space. *The Photogrammetric Record* 26, 32 – 57. doi:10.1111/j.1477-9730.2011.00624.x.
- [44] Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., Pu, S., 2021. RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation. *arXiv preprint arXiv:2103.12978*.
- [45] Yin, C., Yang, S., Yi, X., Wang, Z., Wang, Y., Zhang, B., Tang, Y., 2015. Removing dynamic 3D objects from point clouds of a moving RGB-D camera, in: *Proc. of the IEEE Intl. Conf. on Information and Automation*, pp. 1600–1606.
- [46] Yoon, D., Tang, T., Barfoot, T., 2019. Mapless online detection of dynamic objects in 3d lidar, in: *2019 16th Conference on Computer and Robot Vision (CRV)*, pp. 113–120. doi:10.1109/CRV.2019.00023.
- [47] Zhang, J., Singh, S., 2014. LOAM: Lidar Odometry and Mapping in Real-time, in: *Proc. of Robotics: Science and Systems (RSS)*.