Mapping the Static Parts of Dynamic Scenes from 3D LiDAR Point Clouds Exploiting Ground Segmentation

Mehul Arora Louis Wiesmann Xieyuanli Chen Cyrill Stachniss

2021 European Conference on Mobile Robots (ECMR) | 978-1-6654-1213-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ECMR50962.2021.9568799

***Abstract*— Dynamic objects are an inherent part of our world, but their presence deteriorates the performance of various localization, navigation, and SLAM 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. 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 of the generated static map. To this end, we propose a novel 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. The evaluation results show that our method outperforms the baseline methods in both tasks and achieves good performance in generating clean maps.**

I. 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 [9], stereo images [18] and LiDAR scans [26], [2], [22]. In this paper, we address the problem of detecting and removing dynamic measurements in 3D LiDAR data and generate static point cloud maps in the end.

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 [26], [2] suffer from such dynamic objects and generate maps with so-called "flying 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, one can classify them into two main types, (i) removing dynamic objects while the construction of map [13], [25], [6] and (ii) removing dynamic objects after map generation [19], [14]. 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.

All authors are with the University of Bonn, Germany.

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

Fig. 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 KITTI Odometry sequence 05 frame 904 to 944.

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 [2], 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 [13] 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 [13] 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 (k-NN) 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 static point cloud maps, (ii) in a probabilistic and neighborhood-aware fashion, with (iii) a novel pre-ground segmentation algorithm to better preserve the static environment. These claims are backed up by the paper

describing our approach and our experimental evaluation.

II. 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.

A. Ground Segmentation

Ground segmentation is important for autonomous mobile systems to perform traversable analysis and navigation. Instead of exploiting deep learning-based dense semantic segmentation methods [17], [15], there are also a large number of non-learning-based methods proposed [8], [4], [16]. The learning-based methods perform well in the trained environments but usually cannot generalize well in different environments or with different sensors. Therefore, we focus more on non-learning-based approaches.

Fischler et al. [10] propose RANSAC, which focuses on detecting inliers and separating them from the outliers. This approach can be used to detect the inlier points for ground and segment the ground by fitting a plane that could accommodate those ground points [11]. The approach proposed by Thrun et al. [21] uses probabilistic methods for ground detection. It predicts the movable area by dividing it into smaller grids and predicting the binary classes for ground and non-ground. Carl et al. [24] use Markov random fields for ground segmentation, along with some other spatial constraints based on smooth ground and class continuity. While some approaches target computing the ground points for complete point clouds at once, others focus on breaking the region into smaller parts [16] and then applying techniques like RANSAC [10] and PCA [23], [7].

Our proposed ground segmentation combines a heightmap, with an edge detection algorithm [5] for segmentation. Our segmentation approach works scan-wise and does not need to divide and afterwards fuse the areas. Also, the proposed approach does not involve any learning.

B. Static Map Generation

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

Point cloud segmentation has been a very popular topic and different techniques have been proposed to tackle the problem. Most recently, more and more methods exploit learning-based neural networks to achieve dense full-class segmentation [17], [15]. Based on the full class semantic segmentation, one could directly remove all movable classes like vehicles and humans to clean the map. Instead of removing all potential moving objects, Chen et al. [6] propose a LiDAR-based semantic SLAM method that combines both semantic and geometric information to detect and remove the moving objects on the fly, while leaving the static objects on the map. Even though the full-class semantic segmentation-based methods work well, the training labels are not always available. On the other hand, there are also lots of non-learning-based point cloud segmentation methods [3], [25]. In this work, instead of semantically segmenting the whole point cloud, we only distinguish between ground and non-ground parts and do not require any labels.

Visibility-based approaches check for the query to map associations and are based on the fact that if the query point in consideration is detected beyond an already existing point then the considered query point is deemed as dynamic [19], [14], [16]. These kinds of methods are often corrupted by motion ambiguities and can cause errors. Kim et al. [14] provide an offline approach and clean maps using multiresolution range images, in which the finer resolution was used to remove the dynamic points, while the coarser resolution was used to revert the wrongly classified static points. Another different approach that used the complete map as input is given by Lim et al. [16] where the comparison was made using descriptors. They divide a single point cloud into smaller sectors, and the descriptors are then calculated for each sector. The calculated descriptors are matched with the descriptors calculated for the region in the map. Lim et al. [16] denotes the importance of ground segmentation and exploited the fact that most of the dynamic objects are connected to the ground.

Instead of requiring the complete map of the journey as the input, ray tracing methods distinct between free and occupied space frame-wise and maintain the map in a probabilistic fashion. OctoMap proposed by Hornung et al. [13] used LiDAR scans along with the corresponding poses as input and created a probabilistic occupancy grid map of the environment formed in an Octree representation, which can be used to distinguish between static and dynamic points. Schauer et al. [20] proposed a method based on the traversal of voxel grid along the line of sight of sensor and a point to remove dynamic objects from the LiDAR scan.

The approach presented in this paper deals with the generation of static maps based on OctoMap [13]. It exploits a novel heightmap-based ground segmentation algorithm and combines it with OctoMap along with a voting scheme to produce a static map free from dynamics.

III. Our Approach

In this paper, we propose a point cloud map cleaning method as shown in Fig. 2. First, we preprocess the raw point cloud using a novel heightmap-based ground segmentation approach, which separates the ground from the non-ground points. The points are then fed to OctoMap [13], while marking the ground points as occupied, to distinguish between static and dynamic points. In the end, we exploit a k-NN-based voting scheme to further decide the labels of uncertain estimations generated by OctoMap.

A. heightmap-based Ground Segmentation

There are two reasons why we use ground segmentation for cleaning the point cloud map. First, a large proportion

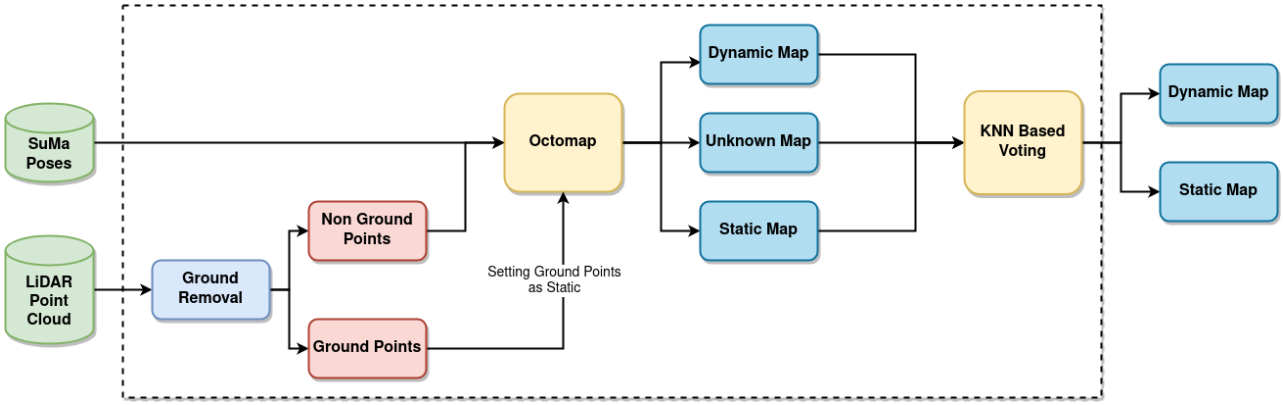


Fig. 2: The figure shows a complete pipeline of our approach. The input of our method is the LiDAR scans together with the poses estimated by SLAM. In the preprocessing module, our method uses a heightmap and canny edge detector to provide ground and non-ground points. After feeding all points to OctoMap, 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 a k-NN-based voting.

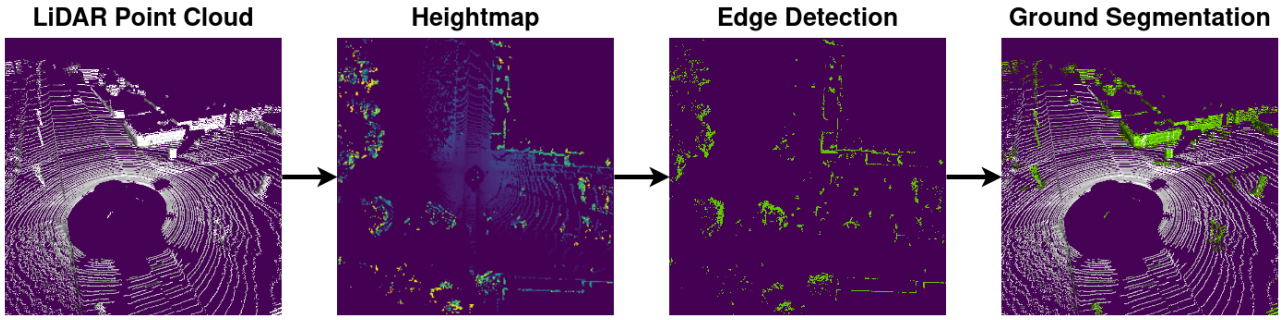


Fig. 3: The figure demonstrates the procedure of our ground removal method, a heightmap is generated using the 3D LiDAR scan, which is then passed through the Canny Edge Detector, which as a result provides us with the edges (non-Ground points). On the right-hand side, we visualize the segmentation result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

of 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.

In this work, we propose a novel multiresolution height-map-based ground segmentation algorithm to solve this issue. 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, visualization in Fig. 3). First, we define the area of interest and a certain grid resolution around the scan. Afterwards, 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. 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 [5] 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 z value above a certain threshold as non-ground. The same algorithm 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 detect the boundary-edges in the grid. Due to the sparsity of the heightmap many points appear as edges of some sort. Here, the discrimination in strong and weak edges comes into play to discard the ground points (weak edges), leaving us with the non-ground (strong edges), which eventually helps in segmenting the ground.

B. OctoMap

OctoMap [13] 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 subsequentially be broken down into eight sub-volumes until a specific resolution is reached. The leaf nodes store an occupancy probability p 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). 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.

C. k-NN-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 the other 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 was 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 [13]. 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.

IV. Experimental Evaluation

The main focus of this work is to remove as many dynamic points as possible, but at the same time preserving the static parts of the map. We present our experiments to show the capabilities of our method and to 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 (ii) a novel pre-ground segmentation algorithm to better preserve the static environment.

A. Dataset

We evaluate our method using KITTI Odometry [12] with the labels from SemanticKITTI [1] 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 trated as static. We use sequence 08 for tuning the hyperparameters and sequence 00-07, 09-10 for evaluation. The labels are just used for the evaluation and are not required elsewhere.

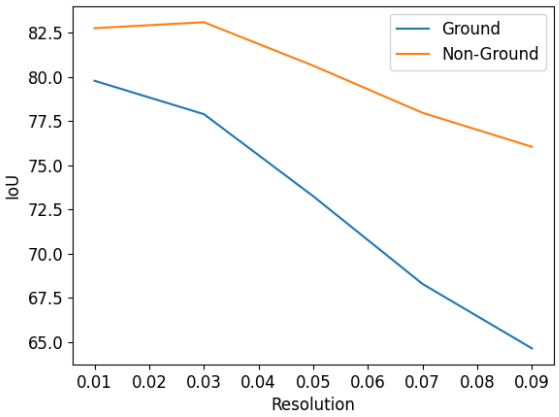


Fig. 4: We evaluate the Intersection over Union (IoU) for heightmaps with different resolutions. Our method is configured to use multiresolution for getting a better result than using a single resolution, so we used a set of resolutions, i.e. $[0.01m, 0.03m, 0.05m, 0.07m, 0.09m]$ before the IoU for ground and non-ground went below a certain level.

B. Data preparation and Hyperparameters

For the purpose of reproducibility, we list 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. The height threshold for the ground segmentation τ_h is set to be -1.5 m $< \tau_h <$ 2.0 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, and the strong edges are given as output. We use $k = 25$ in the k-NN voting for assigning labels to the uncertain points of OctoMap. We voxelized the final point cloud with a grid of 10 cm resolution.

C. 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)

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

$$DR = \frac{TD}{TD + FS} \tag{2}$$

TABLE I: Static Map Generation Results

	KITTI sequence	0	1	2	3	4	5	6	7	8	9	10	Avg.
Removert (R)	Accuracy [%]	79.73	79.83	84.76	**89.07**	**87.03**	77.41	76.35	77.73	78.80	**92.53**	64.27	80.68
	Recall [%]	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)	Accuracy [%]	75.46	71.76	80.83	85.18	82.13	69.81	72.75	75.05	78.56	87.83	61.59	76.50
	Recall [%]	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)	Accuracy [%]	79.94	72.59	80.13	76.58	72.21	79.80	76.67	77.92	76.02	78.48	72.35	76.63
	Recall [%]	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)	Accuracy [%]	80.09	73.22	80.37	76.90	72.42	79.95	76.96	78.06	76.09	78.64	72.47	76.73
	Recall [%]	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)	Accuracy [%]	80.45	73.13	80.26	76.52	72.36	80.52	76.78	77.90	75.62	78.79	72.35	76.78
	Recall [%]	**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	Accuracy [%]	**84.62**	**81.63**	**84.98**	84.54	71.36	**85.21**	**84.79**	**82.11**	**79.44**	79.74	**77.44**	**81.44**
	Recall [%]	92.12	81.78	93.83	95.02	77.07	92.58	88.39	89.15	85.11	85.67	78.92	87.24

TABLE I: The R&R represents the remove and revert version of Removert and R represents only the remove version of Removert whereas Occupied, Empty and k-NN represents the labeling of unknown points as dynamic, static, or based on its neighborhood respectively.

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 that a whole region is correctly classified. 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 the F1-Score.

D. Quantitative Results

The first experiment evaluates the performance of our approach and to support 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 [12] sequences ranging from 0 till 10 (refer Tab. I). Adding our kNN voting to OctoMap instead of either assigning the unknown points to static or dynamic increases the Accuracy and Recall. Combining this with our proposed ground removal increases the map quality by around 5% points. The Recall rate decreases at the same time due to wrongly assigning dynamic points as static. Our approach outperforms Removert in both accuracy and recall in most of the sequences.

E. Qualitative Results

In this section, we present some qualitative results to provide a deeper insight into the performance of our approach (see Fig. 5). We can see that Removert [14] can preserve more static points (points displayed in green color) than our approach but removes substantially fewer dynamic points (blue). The plain OctoMap [13] 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 color). Our approach can detect more static points while keeping high accuracy in removing the dynamics.

TABLE II: Ground Segmentation Results

#	RANSAC Based	PCA Based	Ours
IOU Ground	14.66	43.27	**78.46**
IOU Non Ground	60.55	68.22	**83.95**
Precision	14.88	45.76	**86.53**
Recall	**90.74**	88.84	89.38
F1 Score	25.58	60.41	**87.93**

F. Ground Segmentation

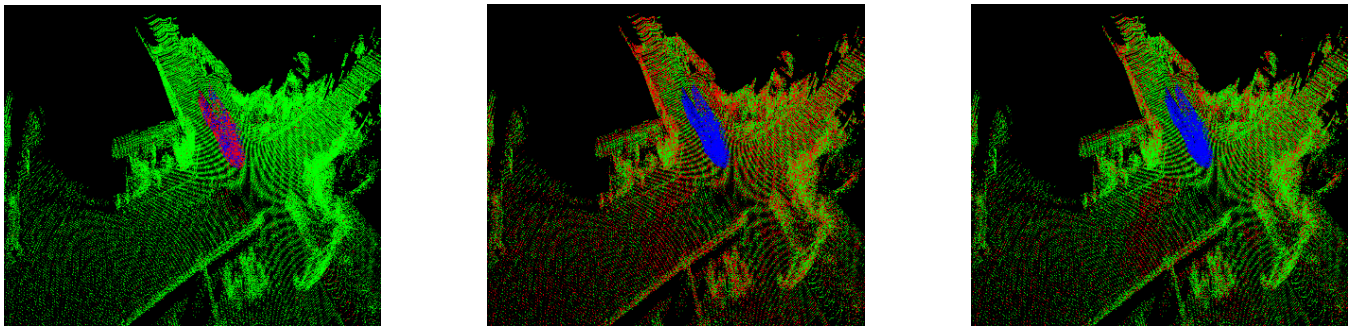
This experiment is presented to support the claim that our heightmap-based ground segmentation algorithm is well suited for separating the ground from non-ground points. We compare our method against the classical RANSAC and PCA-based methods. The results of the experiment are shown in Tab. II. Our approach is able to outperform the baselines in most metrics.

G. Ablation Study

In this experiment, we investigate the impact of the heightmap resolution r on the segmentation accuracy. In Fig. 4 we plot the IoU for ground and non-ground points in 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 resolution in the other experiments of the paper to $r = [0.01m, 0.03m, 0.05m, 0.07m, 0.09m]$ owing to the fact that after $r = 0.09m$ the value of IoU non-ground becomes to low to contribute.

V. Conclusion

In this paper, we presented a novel approach to generate clean maps of the static parts of a scene and proposed a novel ground segmentation algorithm used within the



(a) Removert [14] (b) OctoMap [13] (c) Our

Fig. 5: The figure shows a qualitative comparison between Removert [14], OctoMap [13] and our approach. The points displayed in green color 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.

mapping. Our approach operates on a ray tracing-based approach as used in OctoMap and exploits 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. We implemented and evaluated our approach using SemanticKITTI, provided comparisons to other existing techniques, and supported all claims made in this paper. The experiments suggest that our proposed approach can remove dynamic points while maintaining the quality of the static map hence generated.

REFERENCES

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[2] J. Behley and C. Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[3] I. Bogoslavskyi and C. Stachniss. Efficient online segmentation for sparse 3d laser scans. *Photogrammetrie – Fernerkundung – Geoinformation (PFG)*, pages 1–12, 2017.

[4] J. Byun, K.i. Na, B.s. Seo, and M. Roh. *Drivable Road Detection with 3D Point Clouds Based on the MRF for Intelligent Vehicle*, pages 49–60. Springer International Publishing, Cham, 2015.

[5] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679698, June 1986.

[6] X. Chen, A. Milioto, E. Palazzolo, P. Gigure, J. Behley, and C. Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[7] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss. Range Image-based LiDAR Localization for Autonomous Vehicles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[8] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3d lidar point clouds. In *2011 IEEE International Conference on Robotics and Automation*, pages 2798–2805, 2011.

[9] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 834–849, 2014.

[10] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, 1981.

[11] O. Gallo, R. Manduchi, and A. Rafii. Cc-ransac: Fitting planes in the presence of multiple surfaces in range data. *Pattern Recognition Letters*, 32(3):403–410, 2011.

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[13] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 34:189–206, 2013.

[14] G. Kim and A. Kim. 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)*, Las Vegas, Oct. 2020. Accepted. To appear.

[15] S. Li, X. Chen, Y. Liu, D. Dai, C. Stachniss, and J. Gall. Multi-scale interaction for real-time lidar data segmentation on an embedded platform. *arXiv preprint arXiv:2008.09162*, 2020.

[16] H. Lim, S. Hwang, and H. Myung. Erasor: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building. *IEEE Robotics and Automation Letters*, 6:2272–2279, 2021.

[17] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[18] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. Jacobo Berlles. S-PTAM: Stereo Parallel Tracking and Mapping. *Robotics and Autonomous Systems (RAS)*, 93:27 – 42, 2017.

[19] F. Pomerleau, P. Krüsiand, F. Colas, P. Furgale, and R. Siegwart. Long-term 3d map maintenance in dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[20] J. Schauer and A. Nchter. The peopleremoverremoving dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid. *IEEE Robotics and Automation Letters*, 3(3):1679–1686, 2018.

[21] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. *Stanley: The Robot That Won the DARPA Grand Challenge*, pages 1–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[22] I. Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss. Poisson Surface Reconstruction for LiDAR Odometry and Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[23] M. Wang and Y. Tseng. Incremental segmentation of lidar point clouds with an octreestructured voxel space. *The Photogrammetric Record*, 26:32 – 57, 03 2011.

[24] C. Wellington, A. Courville, and A. Stentz. Interacting markov random fields for simultaneous terrain modeling and obstacle detection. pages 1–8, 06 2005.

[25] D. Yoon, T. Tang, and T. Barfoot. Mapless online detection of dynamic objects in 3d lidar. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 113–120, 2019.

[26] J. Zhang and S. Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Proc. of Robotics: Science and Systems (RSS)*, 2014.