GroundSeg3D: A Robust 3D Grid-Based Hybrid Approach for Ground Segmentation in Point Clouds Across Varied Terrains

Muhammad Haider Khan Lodhi and Christoph Hertzberg

***Abstract*— Ground segmentation in point cloud data is the process of separating the points into ground and non-ground points. It is a key component in the perception pipeline of an autonomous system. The state-of-the-art solutions for ground segmentation are often designed for use-cases from urban environments. The performance of such solutions is tightly linked to certain assumptions about the environment and the sensor configuration. Additionally, they require hand tuning of parameters. There is a need to have a robust, multi-terrain capable ground segmentation solution which does not require tuning of parameters. In this work, we provide such a solution for indoor and outdoor mobile robotics and autonomous driving applications. Our solution combines the strengths of different approaches as a single solution and produces good ground segmentation of point cloud data on varied terrains.**

I. Introduction

The perception system of an autonomous system utilizes multiple sensors to capture salient features of the environment. These features are used as inputs by various subsystems, e. g., navigation, control, learning etc. A core sensor of the perception system is the LiDAR and given the advancements in sensor design and reduced costs, it has become a standard sensor in autonomous systems [1]. The LiDAR uses bands of laser beams to scan the surrounding environment and the time-of-flight information is used to capture distance information at high resolution. The result of the single scan is a point cloud sample which in today's sensors may contain more than a million points. Evidently, the point cloud captures the points from the environment in the scanning region of interest. However, not all points in the region of interest are relevant for the downstream algorithms. For example, object and obstacle detection algorithms often detect ground points as false positives and removal of ground points improves performance and reduces computational burden [2]. Additionally, ground points can be used for traversability analysis, navigation, and static map generation which makes ground segmentation a necessary precursor step for extraction of useful information from point cloud data in autonomous systems [3]–[7].

In this work, our primary focus is to extract ground points with minimal parameter dependency, eliminating the need for parameter tuning, and ensure consistent performance across varied terrains and sensor configurations. Our algorithm was tested on real-world datasets, featuring different types of point cloud sensors in various configurations and environments.

Both authors are researchers at the Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI), Bremen, Germany. Email: Haider_Khan.Lodhi@dfki.de, Christoph.Hertzberg@dfki.de

Funded by the German Federal Ministry of Education and Research, grant number: 13N16537. All code will be published after acceptance.

II. Related Work

Existing ground segmentation methods typically rely on either traditional techniques or learning-based approaches [1].

A. Traditional Solutions

The traditional solutions are usually based on occupancy grids or elevation maps [8]–[10]. Some approaches extend the grid cell and analyze the points in neighboring cells for a better estimation of surface properties [11]. The modeling of the points in the grid cells using Gaussian process regression, plane fitting and line extraction is a common approach in many solutions [12]–[21]. Region growing and clustering is used to combine individual grid cells into a larger region of connected cells [22], [23]. Higher order inferences based on Conditional Random Field (CRF) and Markov Random Field (MRF) have also been used [24], [25].

B. Learning Solutions

The advancements in deep learning meant that some researchers used the existing CNN networks originally used for object detection for ground segmentation using range images generated from the point cloud data. The development of PointNet [26] and its successor PointNet++ [27] gave a unified solution for various applications, e. g., object and scene segmentation. Solutions like GATA [28], GndNet [29], PointPillars [30], JCP [31], SectorGSnet [32] show good results.

III. Proposed Solution

GroundSeg3D takes a single point cloud and outputs the ground and non-ground points. The segmentation process is divided into two distinct phases (Fig. 2). In the first phase of segmentation, a large height for the grid cells is used to capture as many non-ground structures as possible within each grid cell. However, the increased cell height

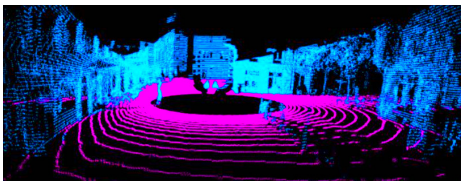

Fig. 1: GroundSeg3D: The point cloud is segmented into ground points (magenta) and non-ground points (blue).

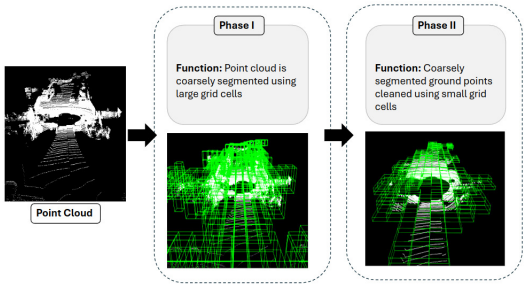

Fig. 2: Phase-I uses grid cells with large height to capture tall structures as non-ground points. The coarse ground points from Phase-I are subsequently processed in Phase-II using grid cells with small height.

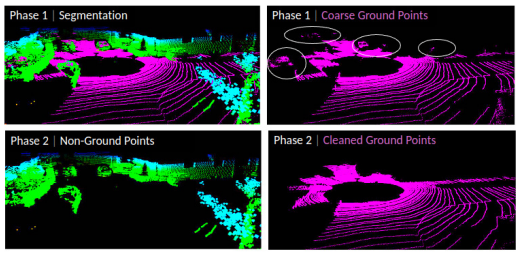

Fig. 3: Correction of over-segmentation of ground points based on dual-phase segmentation.

can lead to false segmentation of ground points, especially where the ground is obscured by overhead objects. These false detections are addressed in subsequent stages of the algorithm.

The second phase uses a smaller grid cell height to refine the segmentation results from the first phase. In this phase, false positive ground points from Phase-I are identified as obstacle points and false positive non-ground points are identified as ground points based on neighborhood correction (Fig. 3).

Table I shows the variation in the processing pipelines of in the two phases. The steps are described in the following sections.

TABLE I: Behavior of steps in the first and second phase of segmentation

Step	First Phase	Second Phase
Grid Representation	Grid cells have a large height	Grid cells have a small height
Local Eigen Analysis	Unchanged	Unchanged
Surface Gradient Analysis	Unchanged	Unchanged
Ground Region Growth	All 17 neighbors below and at the same level are expanded	A detailed neighborhood check is performed for all 26 neighboring cells during expansion
Final Ground Segmentation	Unchanged	Unchanged

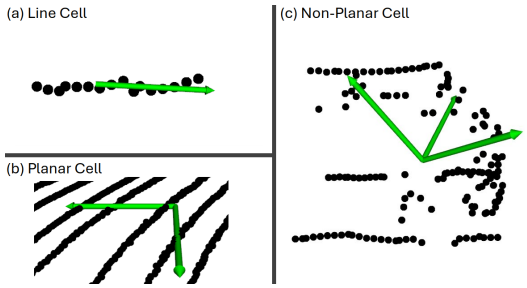

Fig. 4: Points (black) assigned to a grid cell and the dominating eigen vectors (green) are shown. The cell is classified as Line (a), Planar (b), or Non-Planar (c) based of the local eigen analysis.

A. Grid Representation

The points are assigned to cells within a 3D grid to model and detect empty spaces.

$$cell_{\{x,y,z\}} = \lfloor point_{\{x,y,z\}} / cellsize_{\{x,y,z\}} \rfloor, \quad (1)$$

where $cellsize$ is a parameter.

B. Local Eigen Analysis

We use local eigen analysis to estimate the local spatial distribution of points within each grid cell. For each grid cell, we calculate the eigenvalues of the covariance matrix of the points. Let $\lambda_1 > \lambda_2 > \lambda_3$ be the eigenvalues, we then compute the ratio:

$$\text{Ratio} = \frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3} \quad (2)$$

The ratio of the largest eigenvalue to the sum of all eigenvalues provides an indication of whether the points in a grid cell are more aligned with a line, a planar surface, or a non-planar structure.

1) Line Cell: If the largest eigenvalue is significantly larger than the other eigenvalues, the cell is categorized as a Line Cell. For these cells, we compute the angle between the largest eigenvector (depicted in green in Fig. 4a) and the z-axis.

$$CellType = \begin{cases} \text{Obstacle} & angle \geq 90° - Threshold \\ \text{Tentative Ground} & angle < 90° - Threshold \end{cases} \quad (3)$$

where $Threshold$ refers to the ground slope threshold.

The ground slope threshold is a parameter and is used to determine the classification of cells based on their angle of uprightness. For tentative ground cells we cannot be certain of their exact nature. As a result, such cells are marked as tentative ground until it is clarified in the region growth step (Section III-D).

2) Planar Cell: Cells that contain multiple scan lines are suitable candidates for plane fitting when the spatial distribution of the points exhibits two dominant eigenvalues, with the third eigenvalue being negligibly small or significantly smaller. The planar cells at this stage can not be considered as definitive ground. It might be the case that the cell exhibiting a planar distrubtion of points is a car roof top or the top of an obstacle.

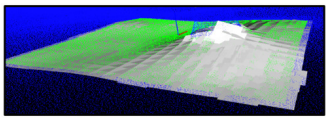

Fig. 5: Plane model fit for local surface slope estimation.

3) Non-planar Cell: If a cell has no dominant eigenvalue, the cell is classified as non-planar, suggesting that the points do not align well with a planar surface.

C. Surface Gradient Analysis

For planar cells, RANSAC plane fitting is applied with a given threshold, getting a plane and separating the points into inliers and outliers. We compute the slope of the fitted plane and apply a slope threshold to assess whether the cell meets the ground criteria. The resultant ground cells are used as candidates for initial seeds in the region-growing algorithm.

$$CellType = \begin{cases} \text{Tentative Ground}, & angle \leq Threshold \\ \text{Non-Ground}, & angle > Threshold \end{cases} \quad (4)$$

where $Threshold$ refers to the same ground slope threshold used in (Section III-B.1, equation 3).

D. Ground Region Growth

The ground region growth selects seeds from tentative ground cells and recursively expands them until all cells are expanded based on neighborhood connectivity.

The seed cells are selected from the tentative ground cells based on the following conditions:

1) High Confidence of Ground Classification: The cell is high confidence when a large percentage of total points are inliers of the planar model fit.
2) Proximity to Sensor: Cells that are far from the sensor typically have sparse points with large gaps, making them unsuitable for initial seed selection due to limited connectivity with neighboring cells. Therefore, cells in close proximity to the sensor are selected.
3) Quadrant-Based Seed Selection: Seed cells are selected from each of the four quadrants in the Euclidean space. This approach helps mitigate issues related to large occlusions or gaps in the point cloud, which could inhibit region growth.
4) Selection of a Single Seed Cell: From each valid group, one seed cell is selected based on the following criteria:
   - The cell must have the highest number of tentative ground neighbors compared to other cells in the group.
   - It should be closest to the mean height of the group.
5) Final Seed Cells: After shortlisting seed cells in all quadrants, we select the lowest cell from the front (1st and 2nd quadrants) and the back (3rd and 4th quadrants) in GroundSeg3D and use all four seed cells in all quadrants for GroundSeg3D+ as shown in Table II. The seed cells are then used for region growing.

In Phase-I we recursively expand all ground neighbors of the seed cells in the grid. This region-wise growth process

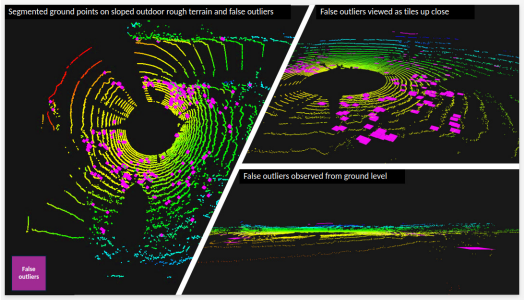

Fig. 6: False outliers (non-ground points) from RANSAC plane fitting on uneven terrain.

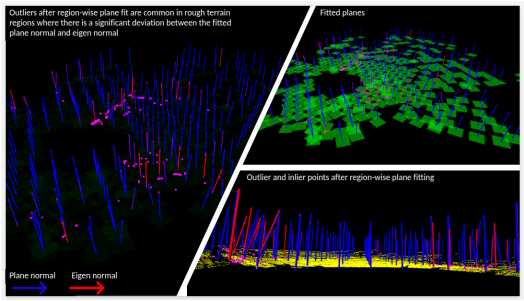

Fig. 7: Correction of false outliers using eigen normal (eigen vector of the smallest eigen value) instead of plane normal.

ensures that only actual ground cells are identified from the set of tentative ground cells.

In Phase-II we only expand cells if they are close to each other along the cell normal. This ensures continuity of the resulting ground cells.

E. Final Ground Segmentation

The final ground segmentation refines the ground and obstacle cells by correcting over- and under-segmentation.

1) Ground Points: As plane fitting on rough terrain produces false outliers, we iterate through the outliers of each ground cell, use KDTree to find the closest inlier point and compute the distance along the eigen normal direction (Fig. 7). If the distance is below the ground threshold, the point is considered a ground point.

2) Non-Ground Cells: Non-Ground cells often contain some ground cells, especially at boundaries of obstacles and ground surfaces. Similar to outliner check for ground points, we check every obstacle point, against the closest ground points in neighboring cells.

IV. Experiments

A ground segmentation algorithm must give robust and reliable performance without parameter tuning on varied terrains. The performance must be consistent across various types of point cloud sensors. For comparison, we selected state-of-the-art ground segmentation algorithms mentioned in Table II using their default parameters.

Some of the ground segmentation algorithms require the sensor height as a parameter. We believe that a constant sensor height can not be guaranteed for point cloud data in

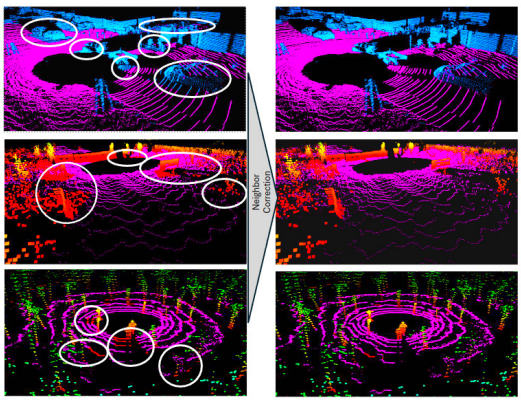

Fig. 8: Correction of under-segmentation of ground points based on neighboring cells. False detections are marked in white circles.

real world applications, e. g., handheld sensor, drones, legged robots which can change their height etc. On the contrary, GroundSeg3D does not depend on the sensor height. Furthermore, the state-of-the-art algorithms require extensive hand-tuning of many parameters while GroundSeg3D does not require any parameter tuning.

For all experiments with GroundSeg3D, we use grids of sizes $(2, 2, 10)$ and $(2, 2, 0.5)$ meters in Phase-I and Phase-II respectively. The ground inlier threshold is 0.1 m and the ground slope threshold is 30°.

A. Datasets

We selected datasets from urban and natural environments. Each dataset uses a different point cloud sensor. The platforms used for data collection are Car, Mobile Robot and Handheld.

B. Performance Metrics

We use precision and recall to compare the algorithms.

$$\text{Precision} = \frac{\text{NTP}}{\text{NTP+NFP}}, \qquad \text{Recall} = \frac{\text{NTP}}{\text{NTP+NFN}}, \qquad (5)$$

where NTP, NFP, and NFN are the Number of True Positives (correctly predicted positives), False Positives (incorrectly predicted positives) and False Negatives (positives that were incorrectly predicted as negative).

Not all datasets provided labelled point clouds, therefore the performance metrics were collected only on the SemanticKITTI dataset. Visual analysis of ground segmentation results on other datasets was enough to compare the performance of GroundSeg3D and state-of-the-art solutions.

V. Results and Discussion

Figures 9 to 14 show consistent performance of GroundSeg3D in different datasets in natural and urban environments with different sensor configurations. We also successfully tested the algorithm on our own Robotics Test Track with very rough terrain (Fig. 15).

On SemanticKITTI sequences 00, 01, 04, and 09 the average precision of GroundSeg3D is generally above 94%, but due to some false positive ground points from vegetation

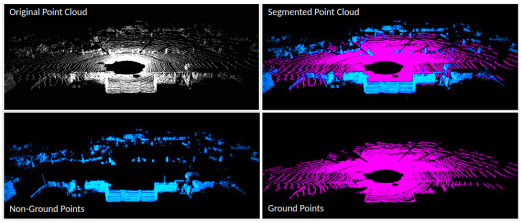

Fig. 9: GroundSeg3D results on the SemanticKITTI dataset.

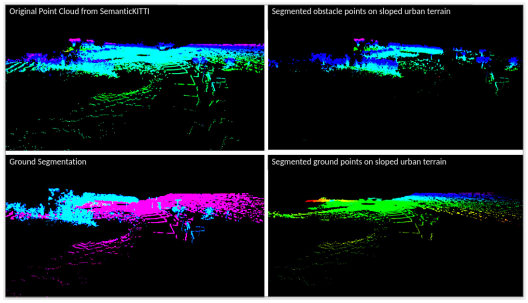

Fig. 10: GroundSeg3D results on sloped outdoor terrain in SemanticKITTI with color-coded height.

(Fig. 16), the precision drops to 92% in sequence 09. The average recall stays above 86%, except in sequence 01, due to gaps in the point cloud larger than grid cell size (Fig. 17). Increasing the number of seeds sells to 4 improves recall with significantly reduced standard deviation as shown in Table II.

While average precision and recall give a good estimate of performance, we can see in sequence 00 frame 20, 100, 140, 150 (Fig. 18 to 21) that GroundSeg3D shows consistent performance.

GroundSeg3D is slower in runtime but in terms of average precision and recall we are on-par with other state-of-the-art algorithms on SemanticKITTI (Table II) but outperform them

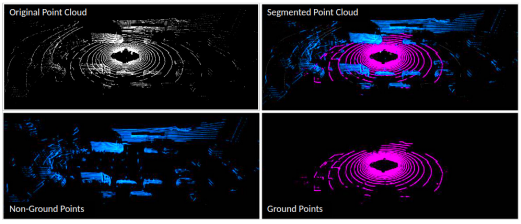

Fig. 11: GroundSeg3D results on the UrbanNav Tokyo dataset.

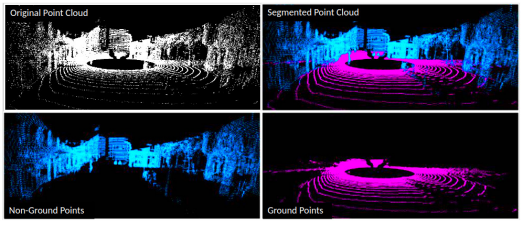

Fig. 12: GroundSeg3D results on the DOALS dataset.

TABLE II: Comparison of ground segmentation on SemanticKITTI sequences 00, 01, 04, and 09 with vegetation. As explained in section III-D, two and four seed cells are used for expansion in GroundSeg3D and GroundSeg3D+ respectively.

Algorithm	Average precision ± standard deviation (%)				Average recall ± standard deviation (%)				Avg. Time (ms)			
Sequence:	00	01	04	09	00	01	04	09	00	01	04	09
GPF [23]	94.9 ± 5.0	92.1 ± 8.5	95.4 ± 5.0	96.0 ± 6.2	79.0 ± 27.2	23.5 ± 21.2	72.6 ± 28.7	63.5 ± 29.5	13	11	12	12
R-GPF [15]	59.2 ± 11.6	89.7 ± 6.0	84.6 ± 9.0	71.4 ± 12.1	96.6 ± 1.8	89.7 ± 4.6	93.4 ± 1.7	95.4 ± 3.4	18	18	18	19
GPR [13]	81.0 ± 7.5	92.5 ± 4.5	95.0 ± 2.0	84.8 ± 7.0	95.9 ± 2.5	89.6 ± 4.9	94.4 ± 1.5	95.0 ± 3.1	11	12	13	13
CascadedSeg [17]	91.7 ± 9.0	97.1 ± 2.9	98.2 ± 2.4	94.8 ± 5.8	71.0 ± 10.4	73.4 ± 10.3	72.5 ± 3.7	70.1 ± 8.0	65	64	70	84
RANSAC [16]	87.5 ± 14.6	96.5 ± 2.5	96.0 ± 2.0	93.3 ± 7.8	93.3 ± 11.4	96.0 ± 3.8	97.0 ± 1.3	89.3 ± 6.2	64	64	42	64
LineFit [14]	98.1 ± 1.0	99.0 ± 1.5	99.5 ± 0.5	98.0 ± 1.3	83.1 ± 9.9	74.2 ± 9.5	80.1 ± 4.6	80.3 ± 9.0	7	6	6	6
Patchwork [18]	92.3 ± 3.3	96.0 ± 3.5	97.4 ± 1.4	92.7 ± 3.3	94.6 ± 3.4	89.1 ± 5.2	91.2 ± 2.8	92.2 ± 4.6	17	16	17	17
Patchwork++ [19]	94.6 ± 2.8	98.4 ± 0.7	98.2 ± 0.8	95.4 ± 3.4	98.7 ± 1.1	96.4 ± 2.9	97.2 ± 2.2	97.0 ± 3.3	11	12	13	13
GroundSeg3D	94.8 ± 5.1	94.8 ± 4.4	97.5 ± 2.1	92.8 ± 4.8	86.9 ± 11.5	78.4 ± 14.8	86.7 ± 14.2	88.8 ± 5.7	85	81	98	102
GroundSeg3D+	94.4 ± 5.0	94.4 ± 4.5	97.8 ± 1.6	92.2 ± 5.2	89.1 ± 6.3	80.4 ± 6.8	89.0 ± 2.6	89.3 ± 4.3	84	80	98	102

TABLE III: Varied datasets from natural and urban environments including average runtime of GroundSeg3D for the first 500 frames from a single sequence

Dataset	**Environment**	**Sensor**	**Year**	**Platform**	**Sensor Height**	**Time (ms)**
SemanticKITTI [33]	Urban	HDL-64E	2013	Car	1.72 m	86
UrbanNav [34]	Urban	HDL-32E	2021	Car	2.1 m	42
DOALS [35]	Urban	OS0 128	2021	Handheld	2.0 m	60
RELLIS-3D [36]	Natural	OS1 64	2022	Mob. Robot	1.16 m	82
BotanicGarden [37]	Natural	VLP-16	2024	Mob. Robot	1.16 m	21

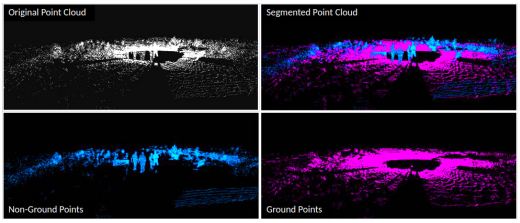

Fig. 13: GroundSeg3D results on the RELLIS-3D dataset.

on other datasets (Fig. 22 and 23).

VI. Conclusions

This paper introduces a hybrid multi-phase ground segmentation algorithm for 3D point clouds, designed to operate without the need for parameter tuning or dependence on specific sensor configurations, ensuring ease of deployment. Extensive experimental evaluations demonstrate that the proposed GroundSeg3D algorithm consistently delivers robust performance across diverse and challenging natural and urban environments.

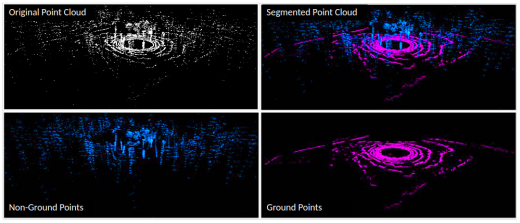

Fig. 14: GroundSeg3D results on the BotanicGarden dataset.

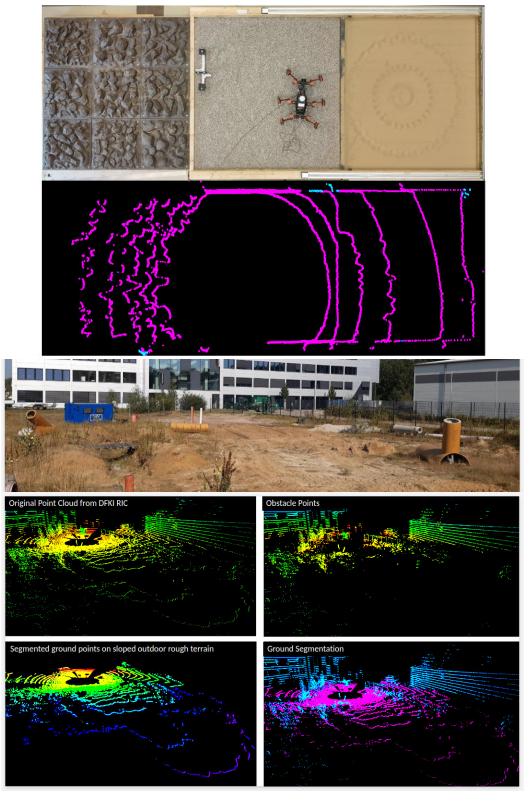

Fig. 15: GroundSeg3D results on rough indoor and outdoor terrain at the Robotics Test Track at DFKI RIC

In future research, we plan to improve the runtime and explore new ways for selection of the seed cells.

VII. Acknowledgment

We greatly acknowledge the open source Ground Segmentation Benchmark [18], [19] which we used for benchmarking. PCL [38] and Nanoflann [39] were used for processing the point clouds and parts of our algorithm were written with support of ChatGPT [40].

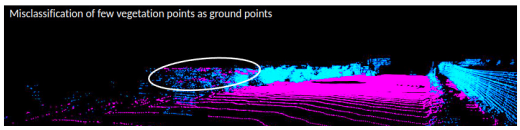

Fig. 16: Over-segmentation of some vegetation points.

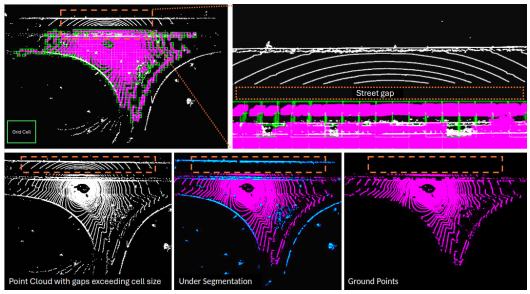

Fig. 17: Under-segmentation of ground points due to point cloud with gaps exceeding grid cell size.

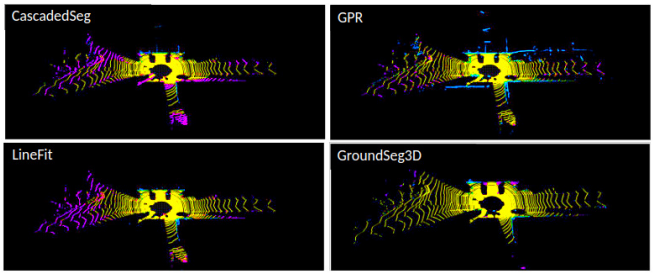

Fig. 18: Frame 20 of the 0th sequence in the SemanticKITTI.

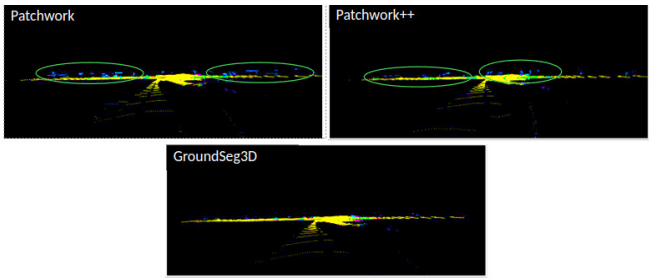

Fig. 19: Frame 100 of the 0th sequence in the SemanticKITTI. Notice the significantly greater number of false positives [blue] points in the Patchwork and Patchwork++ solutions.

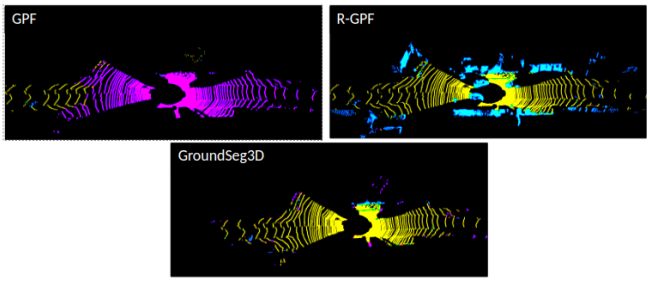

Fig. 20: Frame 140 of the 0th sequence in the SemanticKITTI.

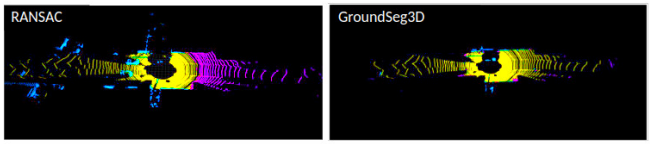

Fig. 21: Frame 150 of the 0th sequence in the SemanticKITTI.

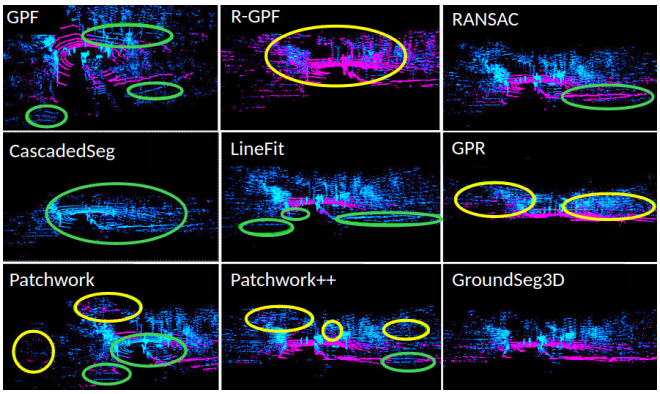

Fig. 22: Comparison on the Botanic Garden data set. Green circles highlight false obstacles, yellow circles highlight false ground points. Here CascadedSeg fails to detect most ground points and R-GPF fails to detect many non-ground points.

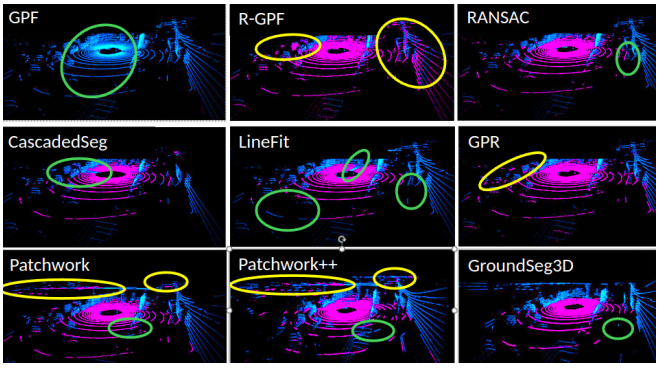

Fig. 23: Comparison on the Tokio UrbanNav data set. GPF fails to detect any ground point, R-GPF fails to detect many non-ground points. All algorithms at least make minor errors.

REFERENCES

[1] T. Gomes, D. Matias, A. Campos, L. Cunha, and R. Roriz, "A survey on ground segmentation methods for automotive lidar sensors," *Sensors*, vol. 23, no. 2, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/2/601

[2] E. Firkat, F. An, B. Peng, J. Zhang, T. Mijit, A. Ahat, J. Zhu, and A. Hamdulla, "FGSeg: Field-ground segmentation for agricultural robot based on LiDAR," *Computers and Electronics in Agriculture*, vol. 211, p. 107965, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169923003538

[3] H. Lim, M. Oh, S. Lee, S. Ahn, and H. Myung, "Similar but different: A survey of ground segmentation and traversability estimation for terrestrial robots," *International Journal of Control, Automation and Systems*, vol. 22, no. 2, pp. 347–359, 2024.

[4] T.-C. Tsai and C.-C. Peng, "Ground segmentation based point cloud feature extraction for 3d lidar slam enhancement," *Measurement*, vol. 236, p. 114890, 2024.

[5] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Static map generation from 3d LiDAR point clouds exploiting ground segmentation," *Robotics and Autonomous Systems*, vol. 159, p. 104287, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889022001762

[6] V. Jiménez, J. Godoy, A. Artuñedo, and J. Villagra, "Ground segmentation algorithm for sloped terrain and sparse LiDAR point cloud," *IEEE Access*, vol. 9, pp. 132 914–132 927, 2021.

[7] H. Lim, B. Kim, D. Kim, E. M. Lee, and H. Myung, "Quatro++: Robust global registration exploiting ground segmentation for loop closing in lidar slam," *The International Journal of Robotics Research*, vol. 43, no. 5, pp. 685–715, 2024. [Online]. Available: https://doi.org/10.1177/02783649231207654

[8] B. Douillard, J. Underwood, N. Melkumyan, S. Singh, S. Vasudevan, C. Brunner, and A. Quadros, "Hybrid elevation maps: 3d surface models for segmentation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1532–1538.

[9] B. Anand, M. Senapati, V. Barsaiyan, and P. Rajalakshmi, "Lidar-ins/gnss-based real-time ground removal, segmentation, and georeferencing framework for smart transportation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[10] N. Steinke, D. Goehring, and R. Rojas, "Groundgrid: Lidar point cloud ground segmentation and terrain estimation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 420–426, 2024.

[11] D. Guo, G. Yang, B. Qi, and C. Wang, "A fast ground segmentation method of lidar point cloud from coarse-to-fine," *IEEE Sensors Journal*, vol. 23, no. 2, pp. 1357–1367, 2023.

[12] Y. An, W. Liu, Y. Cui, J. Wang, X. Li, and H. Hu, "Multilevel ground segmentation for 3-d point clouds of outdoor scenes based on shape analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[13] T. Chen, B. Dai, R. Wang *et al.*, "Gaussian-process-based real-time ground segmentation for autonomous land vehicles[j]," in *Journal of Intelligent and Robotic Systems*, 2014, pp. 76(3–4):563–582.

[14] M. Himmelsbach, F. v. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 560–565.

[15] 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*, vol. 6, no. 2, pp. 2272–2279, 2021.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in Computer Vision*, M. A. Fischler and O. Firschein, Eds. San Francisco (CA): Morgan Kaufmann, 1987, pp. 726–740. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080515816500702

[17] P. Narksri, E. Takeuchi, Y. Ninomiya, Y. Morales, N. Akai, and N. Kawaguchi, "A slope-robust cascaded ground segmentation in 3d point cloud for autonomous vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 497–504.

[18] H. Lim, M. Oh, and H. Myung, "Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3d lidar sensor," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6458–6465, 2021.

[19] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 276–13 283.

[20] W. Deng, X. Chen, and J. Jiang, "A staged real-time ground segmentation algorithm of 3d lidar point cloud," *Electronics*, vol. 13, no. 5, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/5/841

[21] Y. Qian, X. Wang, Z. Chen, C. Wang, and M. Yang, "Hy-Seg: A hybrid method for ground segmentation using point clouds," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1597–1606, 2023.

[22] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *2009 IEEE Intelligent Vehicles Symposium*, 2009, pp. 215–220.

[23] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5067–5073.

[24] W. Huang, H. Liang, L. Lin, Z. Wang, S. Wang, B. Yu, and R. Niu, "A fast point cloud ground segmentation approach based on coarse-to-fine markov random field," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7841–7854, 2022.

[25] L. Rummelhard, A. Paigwar, A. Nègre, and C. Laugier, "Ground estimation and point cloud segmentation using spatiotemporal conditional random field," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1105–1110.

[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[28] I. d. Pino, A. Santamaria-Navarro, A. Garrell Zulueta, F. Torres, and J. Andrade-Cetto, "Probabilistic graph-based real-time ground segmentation for urban robotics," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 5, pp. 4989–5002, 2024.

[29] A. Paigwar, Ö. Erkent, D. Sierra-Gonzalez, and C. Laugier, "GndNet: Fast ground plane estimation and point cloud segmentation for autonomous vehicles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2150–2156.

[30] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[31] Z. Shen, H. Liang, L. Lin, Z. Wang, W. Huang, and J. Yu, "Fast ground segmentation for 3d lidar point cloud based on jump-convolution-process," *Remote Sensing*, vol. 13, no. 16, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/16/3239

[32] D. He, F. Abid, Y.-M. Kim, and J.-H. Kim, "Sectorgsnet: Sector learning for efficient ground segmentation of outdoor lidar point clouds," *IEEE Access*, vol. 10, pp. 11 938–11 946, 2022.

[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[34] L.-T. Hsu, F. Huang, H.-F. Ng, G. Zhang, Y. Zhong, X. Bai, and W. Wen, "Hong kong urbannav: An open-source multisensory dataset for benchmarking urban navigation algorithms," *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 4, 2023. [Online]. Available: https://navi.ion.org/content/70/4/navi.602

[35] P. Pfreundschuh, H. F. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart, and A. Cramariuc, "Dynamic object aware lidar slam based on automatic generation of training data," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 641–11 647.

[36] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1110–1116.

[37] Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Xu, F. Chen, B. Goossens, P. Z. Sun, H. Yu, C. Liu, L. Chen, W. Tao, and H. Zhao, "Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2798–2805, 2024.

[38] R. B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.
[39] J. L. Blanco and P. K. Rai, “nanoflann: a c++ header-only fork of flann, a library for nearest neighbor (nn) with kd-trees,” *GitHub Repository*, 2014.
[40] OpenAI, “ChatGPT (GPT-4),” 2023. [Online]. Available: https://www.openai.com/chatgpt