

LTM: A Fast and Error-sensitive Point Cloud Clustering Algorithm for LiDAR Panoptic Segmentation

Peng Wu¹²³, Zhengxi Hu¹²³, Qianyi Zhang¹²³, Jinzheng Guang¹²³, Jingtai Liu^{123*}

¹ Robotics and Automatic Information System, Nankai University, Tianjin, China

² Tianjin Key Laboratory of Intelligent Robotics

³ TBI center, Nankai University

{wupeng2022, guangjinzheng, zhangqianyi}@mail.nankai.edu.cn, 609100797@qq.com, liujt@nankai.edu.cn

Abstract—Point cloud clustering is a crucial approach to understand and analyze geometric information for both robotic systems and autonomous driving. To meet real-time requirements, several existing point cloud clustering algorithms are based on Connected Component Labeling. It means that classification decisions are made locally based on the geometric relationships among neighboring points. However, the existing neighborhood connection criterion lack robustness. This is because the difference in resolution between point clouds in the horizontal and vertical directions and various depth affect the distance between adjacent points, thereby affecting connectivity determination. In this paper, we propose a novel clustering algorithm LTM (includes Linearization, Trunk extraction and Merge). It processes the horizontal and vertical directions separately in two steps and innovatively utilizes three points to determine neighborhood connectivity. Thus, it can reduce the impact of these constraints on the effectiveness. We cascade the proposed clustering algorithm with existing semantic segmentation models and perform a panoptic segmentation task on the SemanticKITTI. Experimental results demonstrate that LTM achieved the best results among all published methods of the same category. Furthermore, the single-frame processing time for clustering is within 10 milliseconds.

Index Terms—LiDAR Panoptic Segmentation, Point Cloud Clustering, Connected Component Labeling

I. INTRODUCTION

Light detection and ranging (LiDAR) is widely utilized in fields like robotics and autonomous driving due to its capability to provide 3D information. To enhance a robot's perception of surrounding environment, it is necessary to extract valuable information from raw point cloud data. Point cloud clustering is an approach to understanding and analyzing geometric information from a traditional perspective. It is used to generate candidates for pedestrian detection tasks using classifiers such as AdaBoost [9]. It can also be applied as a part of advanced perception systems in fields such as semantic segmentation, instance segmentation, and more. Zhao et al. [19] combine traditional point cloud clustering methods [18] with deep learning semantic segmentation models [22] to accomplish the task of panoptic segmentation. It achieves state-

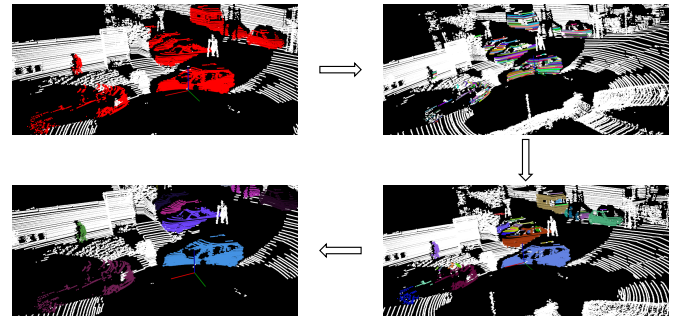


Fig. 1. The demonstration of the panoptic segmentation task and how LTM solves it. The red points in the first image correspond to those categorized as "thing" based on semantic segmentation. These points are the ones that the clustering algorithm needs to process. By performing linearization, trunk extraction and merge on semantic segmentation points, the panoptic segmentation task is completed.

of-the-art performance on the SemanticKITTI [1] panoptic segmentation benchmark.

Existing clustering algorithms can be primarily divided into two categories. The first category is based on 3D spatial point queries, where these algorithms mainly consider the distance relationships between point clouds. They often use optimizations like KD-Tree or OCT-Tree to enhance storage and improve clustering efficiency. However, even with these optimizations, such algorithms still incur significant time costs when dealing with a large number of point clouds. The second category is based on distance image point queries, where point clouds are transformed into distance images through spherical projection. This allows them to operate with linear complexity concerning the point cloud size.

The core challenge in these algorithms based on Connected Component Labeling (CCL), is how to determine whether adjacent points belong to the same object. In existing algorithms, SLR [18] employs the Euclidean distance between two points as the criterion. Depth Cluster [3] introduces a heuristic angle condition. However, the difference in the relative relationship between two adjacent points of the same object is influenced by multiple factors. 1) There is a difference in LiDAR res-

This work is supported in part by the National Natural Science Foundation of China under Grant 62173189.

olution in both the horizontal and vertical directions. This implies that applying the same operation uniformly to all points is not reasonable. 2) The distance between adjacent points varies with depth. This variation makes it difficult to obtain an accurate threshold for determining connectivity. 3) The incidence angle formed by LiDAR and objects varies greatly. This variation means that even at the same depth, the distance between adjacent points may not be consistent. The presence of all these factors introduces additional challenges in assessing the connectivity between adjacent points. Therefore, we aim to design a novel algorithm that can separately handle point cloud clustering in the horizontal and vertical directions while minimizing the impact of depth and angle of incidence on the clustering.

This paper has three major contributions: 1) We devise a novel criterion for point cloud connectivity. By introducing the constraint of three-point collinearity, we significantly reduce the likelihood of erroneous connectivity between different instances of point clouds. 2) Based on the characteristics of point clouds, we design a new clustering algorithm called LTM. This algorithm efficiently achieves point cloud segmentation, particularly improving the recognition capabilities of adjacent objects. 3) We apply LTM to panoptic segmentation. While ensuring efficiency, we have achieved the best results among similar algorithms.

II. RELATED WORK

Point Cloud Clustering. The traditional task of LiDAR point cloud clustering involves partitioning three-dimensional point cloud data into clusters or groups with similar properties. This task is of crucial importance in fields such as autonomous driving, robotic perception, environmental modeling, and map generation. Given the three-dimensional geometric information carried by point clouds, it is a natural idea to determine whether two points belong to the same cluster based on the distance between them. Euclidean clustering, as the most common and widely used clustering method, represents this category of methods. Radially bounded nearest neighbor (RBNN) [7] algorithm is proposed, and the introduction of normals enhances its ability to segment surfaces [8]. Yan et al.s [17] propose an adaptive threshold selection method to handle variations in LiDAR sparsity with distance. However, the large amount of point cloud data makes these clustering algorithms inadequate in terms of time complexity. To enhance the efficiency of clustering algorithms, current research is more focused on constructing neighborhood relationships to achieve linear complexity with respect to point cloud size. These methods largely draw inspiration from the ideas of CCL algorithms [5]. Adaptive angular threshold [3] is proposed to determine whether two adjacent points belong to the same object. On the other hand, Zermas et al.s [18] introduce a row-wise fast scanning algorithm based on point clouds or depth maps. Following the construction of surface voxels in spherical coordinates [13], rapid clustering is achieved based on neighborhood relationships. Moreover, these methods

exhibit a certain degree of adaptability to varying point cloud density at different distances.

Panoptic Segmentation. Panoptic segmentation is the fusion of semantic segmentation and instance segmentation, aiming to segment objects in a scene. This task was initially proposed for 2D images [6] and later extended to LiDAR [2]. Early panoptic segmentation was achieved by combining object detectors with semantic segmentation networks. The PointPillars [10] object detector was employed to obtain bounding boxes and classes for each object, and a combination of KPConv [15] and RangeNet++ [12] was deployed for instance segmentation of each class. Furthermore, Gasperini et al. [4] implement a LiDAR point cloud panoptic segmentation clustering method based on Q-learning. Additionally, Milioto et al. [11] used an end-to-end model to accomplish the panoptic segmentation task.

III. METHOD

A. Overview

We propose the LTM (includes Linearization, Trunk extraction and Merge), which addresses the computational cost associated with a large number of point clouds, controlling the time complexity within linear dependence on the number of point clouds.

LTM consists of three steps. In Fig. 2, we provide schematic diagrams for each step. Firstly, based on the novel idea of three-point collinearity, we introduce a new constraint to achieve linearization. Subsequently, the degree of overlap of the linear point set and the distance between the center points of the overlapping area are calculated to achieve instance trunk extraction. Finally, the merging of trunk and determination of the instance categories for the remaining points are accomplished based on the bounding boxes of the clusters.

B. Linearization

To extract linear point sets, we need to assess the connectivity of adjacent points in order to determine whether they belong to the same instance. Due to the sparsity of point clouds, the relative relationship between two adjacent points can vary under different depths or incident angles. Existing two-point connectivity constraints struggle to accommodate varying conditions simultaneously, as two points alone cannot provide sufficient geometric constraints. Therefore, we introduce a new criterion based on the assumption of three-point collinearity. This criterion is adaptable to different depths and incident angles, overcoming the limitations of existing two-point-based standards. Further details regarding the two-point criterion and four-point criterion will be discussed in Section C of Chapter IV.

Our criterion is based on an easily understood phenomenon: due to the sufficiently high resolution of the point cloud in the horizontal direction, three adjacent points from the same instance will approximately satisfy the condition of collinearity. This allows us to utilize the three-point collinearity to

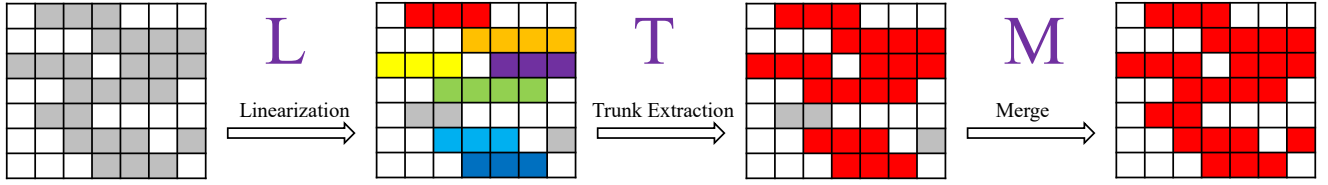


Fig. 2. Visualization of LTM clustering execution. Gray indicates unclassified points, while identical colors represent the same instance. After projecting the unordered point cloud into an ordered range image, the process proceeds with sequential horizontal linearization, vertical trunk extraction, and merge of remaining scattered points.

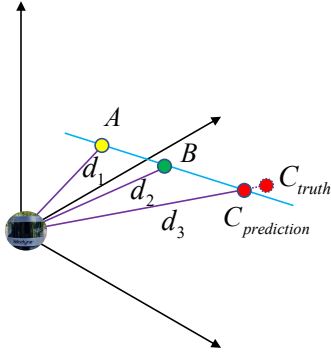


Fig. 3. How LTM utilizes adjacent three points to define heuristic conditions for checking whether points belong to the same object. A , B , C_{truth} are three adjacent points sharing the same polar angle. d_1 , d_2 and d_3 represent their respective depths. Point $C_{prediction}$ is collinear with line AB and shares the same azimuth as C_{truth} .

aggregate points from the same instance in the horizontal direction.

$$\delta = |d_1 + d_3 - \frac{2 * d_1 * d_3}{d_2}| \quad (1)$$

Based on geometric relationships and the condition that all horizontally adjacent points have a fixed and sufficiently small azimuth angle, we derive the criterion. As shown in Fig. 3, where d_1 , d_2 and d_3 respectively represent the distances from adjacent three points to the LiDAR. When adjacent points meet the three-point collinearity, $\delta = 0$. The partial derivative of δ with respect to d_3 approximates 1. When three points cannot strictly satisfy the collinearity, δ represents the distance between C_{truth} and the collinear-based $C_{prediction}$, as shown in Fig. 3. This enables the criterion to handle point clouds at varying depths. With the aid of this criterion, we can effortlessly accomplish the extraction of linear point sets in the horizontal direction. More details about linearization can refer to Algorithm 1.

C. Trunk Extraction

After completing the processing in the horizontal direction, we aggregate the linear point sets in the vertical direction to extract instance trunk. These linear point sets represent the geometric features of the instance at different heights. From a bird's-eye view, different linear point sets of the same instance are distributed near the center of the instance, with relatively

Algorithm 1: Linearization

Input: Number of point clouds N . Distance of point clouds d which has already completed ordered indexing using range images.

Output: linear point sets L

```

1: for each  $i \in [0, N - 1]$  do
2:    $con1 = d_i > 0$  and  $d_{i+1} > 0$  and  $d_{i+2} > 0$ ;
3:    $con2 = d_i + d_{i+2} - 2 * d_i * d_{i+2} / d_{i+1} < th_d$ ;
4:   if  $con1$  and  $con2$  then
5:     Save the indices of the points:
        $line.point\_index \leftarrow (i, i+1, i+2)$ ;
6:   end if
7:   if  $line.point\_index.size() > 2$  then
8:      $L \leftarrow line$ ;
9:   end if
10: end for

```

small distances between them. From a range view, different linear point sets of the same instance overlap in the vertical direction, which manifests as overlapping azimuth angles in spherical coordinates. Therefore, we use the degree of overlap in azimuth angles and the distance between overlapping regions in the top-down view as the criterion for extracting the main structure.

For all lines, we first calculate their respective centroids and construct a KD-Tree to accelerate the identification of all potential lines belonging to the same instance. For any two lines being processed, the first condition is the overlap ratio of their azimuth angles. This condition is satisfied when the overlap region occupies a proportion of the shorter line that exceeds a threshold th_{az} . Next, we extract the points from both lines within the overlap region and compute the center points of these points in bird's-eye view. The second condition is satisfied when the distance between the center points of the overlap region is less than the threshold th_d . If both conditions are met, the two lines are considered part of the same instance. Once all point sets are processed, the trunk extraction is completed.

D. Merge

We refer to points that do not satisfy the condition of three-point collinearity as isolated points. Since linearization only handles the processing of adjacent three points, isolated

points are not clustered into the main structure. Although these isolated points do not significantly affect the extraction of the main branches of the instance, they still need to be considered in the final clustering process. Given that these points are few in number and lack explicit geometric constraints, Euclidean clustering is sufficient to aggregate them into nearby main structures in a short time. This way, all foreground points are incorporated into their respective main structures.

The strict collinearity condition for three points may result in some instances being divided into multiple trunks. This is an expected outcome, as the merge of trunks is much less complex and more precise than point clustering. We calculate the bounding boxes of all main structures and use the intersection-over-union (IoU) to complete their aggregation. Additionally, using prior information from semantic segmentation, size constraints can be introduced for known categories to further improve accuracy.

IV. EXPERIMENTS

A. Dataset

The SemanticKITTI created by equipping the Velodyne HDL-64E S2 LiDAR sensor on the KITTI Vision Benchmark Suite is the first dataset to introduce the task of LiDAR panoptic segmentation. The dataset consists of 22 sequences, with ten LiDAR sequences for training, one sequence for validation, and 11 sequences for testing. There are 19 categories in the dataset, with eight categorized as thing and 11 as stuff. We selected two semantic segmentation models, Cylinder3D [22] and 2DPass [16], to provide semantic priors for better showcasing our clustering method.

B. Metrics

The panoptic segmentation task involves not only assigning a semantic label to each point but also assigning an instance label (ID) to points belonging to the thing category. The evaluation metric for this task is typically measured using Panoptic Quality (PQ) [6], with the specific calculation formula as follows:

$$PQ_c = \frac{\sum_{(p,g) \in TP_c} IoU(p,g)}{|TP_c| + \frac{1}{2}|FN_c| + \frac{1}{2}|FP_c|} \quad (2)$$

This includes segmentation accuracy measured by SQ and detection accuracy measured by RQ. To obtain true positives (TP), it is commonly considered that a prediction matches the ground truth when the Intersection over Union (IoU) of two sets is greater than 0.5. False positives (FP) and false negatives (FN) can be obtained in a similar manner. PQ is calculated separately for each category, and the average is taken as the final metric.

C. Performance Evaluation

In Fig. 4, we demonstrate the performance of our panoptic segmentation. In dense traffic, we can accurately identify each instance, even when they are close together and share the same semantic information.

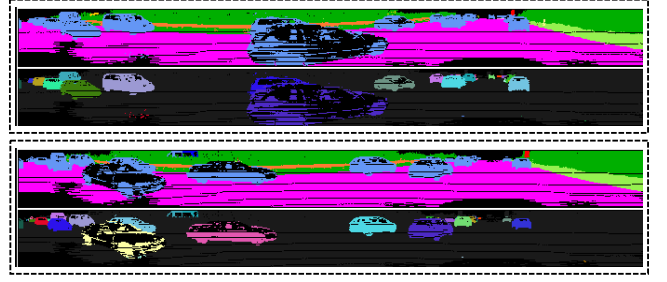


Fig. 4. The performance of LTM. The first and third images represent the input semantic information, while the second and fourth images depict the panoptic segmentation performance using LTM.

Compare with Cluster Methods. In TABLE I, we compare our method with similar panoptic segmentation approaches. We utilize 2DPass and Cylinder3D to provide semantic labels. In each comparative experiment, the same semantic information is used to validate the effectiveness of different clustering methods. We conduct validation on the SemanticKITTI (sequence 08). We do not process points with the semantic label stuff, hence the three metrics regarding stuff are essentially consistent. The points with the semantic label thing are the ones processed by the clustering algorithm. The three metrics regarding thing more accurately reflect the significance of the clustering algorithm. Experimental results demonstrate that our clustering method significantly outperforms previous methods in handling thing type instances, which aligns with the primary focus of clustering operations.

Compare with Other Panoptic Solutions. In TABLE II, we compare our algorithm with alternative panoptic segmentation strategies. It's evident that our algorithm achieves a higher PQ accuracy compared to end-to-end algorithms [14]. Additionally, due to the differences of the semantic segmentation network, metrics such as PQ^{St} , RQ^{St} , and SQ^{St} , which are unrelated to the clustering algorithm, do not achieve optimal performance compared to [19] [20]. We replicated using the same semantic labels provided by 2DPass to achieve a more fair comparison. In addition, as semantic segmentation networks continue to advance, our hybrid panoptic segmentation solution is poised to attain even higher accuracy levels.

Discussion on two or four points criterion. In the linearization process, once points from different instances are mistakenly clustered together, it is impossible to separate them. More critically, each judgment considers only the geometric relationships within a small neighborhood. If separation points are misjudged, the two-point criterion will inevitably merge points from two different instances. Erroneous linear point sets can affect the assessment of vertical overlap, ultimately leading to the incorrect clustering of two instances. In Figure 5, we show the results of different clustering algorithms applied to two closely positioned vehicles. It can be observed that the SLR algorithm failed to cluster correctly, and this failure was

TABLE I
PERFORMANCE COMPARISON ON SEMANTICKITTI VAL SET (SEQUENCE 08)

Methods	PQ	PQ [†]	RQ	SQ	PQ Th	RQ Th	SQ Th	PQ St	RQ St	SQ St	mIoU
Cylinder3D + SLR [19]	56.8	61.7	68.0	77.6	57.4	66.6	75.0	56.5	69.0	79.5	66.2
Cylinder3D + DMC [20]	57.1	62.0	68.3	77.5	58.0	67.4	74.8	56.5	69.0	79.5	66.3
Cylinder3D + LTM	59.2	64.1	69.7	78.7	62.9	70.6	77.6	56.6	69.0	79.5	66.9

TABLE II
PERFORMANCE COMPARISON ON SEMANTICKITTI TEST SET (SEQUENCE 11 TO 21)

Methods	PQ	PQ [†]	RQ	SQ	PQ Th	RQ Th	SQ Th	PQ St	RQ St	SQ St	mIoU
Panoster [4]	52.7	59.9	64.1	80.7	49.4	58.5	83.3	55.1	68.2	78.8	59.9
PolarNet [21]	54.1	60.7	66.0	81.4	53.3	60.6	87.2	54.8	68.1	77.2	59.5
Cylinder3D + SLR [19]	56.0	62.6	67.4	82.1	51.8	61.0	84.2	59.1	72.1	80.6	67.9
Cylinder3D + DMC [20]	56.5	63.1	67.9	82.3	52.9	62.1	84.8	59.1	72.1	80.6	68.2
EvLPSNet [14]	58.0	-	-	-	62.7	-	-	54.6	-	-	64.0
2DPass + SLR	55.1	61.9	66.4	81.8	50.7	59.6	83.8	58.4	71.4	80.3	66.6
2DPass + DMC	55.9	62.7	67.3	81.9	52.5	61.8	84.0	58.4	71.4	80.3	66.8
2DPass + LTM(ours)	58.7	65.4	69.7	83.2	58.9	67.4	87.0	58.5	71.4	80.4	67.4

entirely caused by the misclassification of just a few points. This highlights the importance of transitioning from a two-point criterion to a three-point criterion.

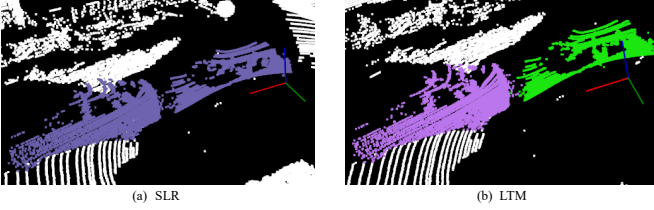


Fig. 5. LTM and SLR segmentation results on the SemanticKITTI. LTM accurately segments vehicles, even when they are in close proximity to each other.

TABLE III
PERFORMANCE COMPARISON ON SEMANTICKITTI VAL SET (SEQUENCE 08) FOR CLUSTERING METHODS BASED ON CCL

Methods	TN	FP	Significant error
Scan-line Run [18]	73725	1213	915
Depth Cluster [3]	72354	2584	1520
LTM	74460	478	194

We perform experiments on sequence 08 of the SemanticKITTI validation set to evaluate two two-point criterion algorithms and our algorithm. The thresholds for Scan-line Run(SLR) and Depth Cluster are referenced from [20]. We extract all pairs of neighboring points in the SemanticKITTI validation set that do not belong to the same instance. True negative(TN) indicates that the clustering algorithm successfully separates the points. False positive (FP) indicates failure to separate them. Significant error indicates that after a misjudgment, the algorithm fails to terminate in time and continued to explore. This can result in abnormal linear point sets, which affects trunk extraction. Detailed results are

shown in Table III. It is clear that, compared to previous pairwise point-based methods, our algorithm performs better in addressing these issues.

Compared to the two-point criterion, the three-point criterion introduces collinearity as a natural constraint. Similarly, criteria based on four or more points introduce multiple collinearity constraints. Taking the four-point criterion as an example, we first need to consider whether the relationship between these constraints is "and" or "or." Choosing the former ensures consistency with the standard established by the three-point linearization method, where more point constraints are equivalent to the three-point constraint. The latter relaxes the constraints, which may affect the accuracy of extracting linear point sets. In visualization experiments, we found that this relaxed method tends to skip over isolated points, and as discussed earlier, these isolated points are crucial for accurately defining the structure of the lines.

D. Runtime

The time consumption of the LTM algorithm consists of three components. The time cost of the linearization part is linearly related to the number of points in the point cloud. The time cost of the main body extraction, as well as the processing and merging of remaining points, depends on the scene. Based on our testing experience on SemanticKITTI, where the line count typically ranges from 100 to 1000 and the remaining point count falls between 500 and 1000, we conducted tests on the SemanticKITTI test set sequences (sequence 11-21) using an i7-12700H @ 4.7GHz processor. The average processing time for a single frame is around 10ms.

V. CONCLUSIONS

For the LiDAR panoptic segmentation problem, we propose a novel clustering algorithm called LTM. To address the issue of uneven point cloud density caused by factors such as

depth and normal vectors, we surpass the existing two-point-based criterion and explore the significance of the connectivity criteria using more pairs of points. Based on the simple assumption of the three-point collinearity, we propose a new connectivity criterion for the point cloud. This significantly reduces false positives (FP) and error rates. In addition, by combining existing semantic segmentation networks with the LTM clustering algorithm, we achieve panoptic segmentation tasks on the SemanticKITTI. We believe that the contributions of this paper will contribute to academic research as well as various service robot applications, including perception, localization, and map construction.

REFERENCES

- [1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019.
- [2] Jens Behley, Andres Milioto, and Cyrill Stachniss. A benchmark for lidar-based panoptic segmentation based on kitti. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13596–13603. IEEE, 2021.
- [3] Igor Bogoslavskyi and Cyrill Stachniss. Fast range image-based segmentation of sparse 3d laser scans for online operation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 163–169. IEEE, 2016.
- [4] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. Panoster: End-to-end panoptic segmentation of lidar point clouds. *IEEE Robotics and Automation Letters*, 6(2):3216–3223, 2021.
- [5] Frederik Hasecke, Lukas Hahn, and Anton Kummert. Flic: Fast lidar image clustering. *arXiv preprint arXiv:2003.00575*, 2020.
- [6] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.
- [7] Klaas Klasing, Dirk Wollherr, and Martin Buss. A clustering method for efficient segmentation of 3d laser data. In *2008 IEEE international conference on robotics and automation*, pages 4043–4048. IEEE, 2008.
- [8] Klaas Klasing, Dirk Wollherr, and Martin Buss. Realtime segmentation of range data using continuous nearest neighbors. In *2009 IEEE International Conference on Robotics and Automation*, pages 2431–2436. IEEE, 2009.
- [9] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2):1729881419841532, 2019.
- [10] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.
- [11] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. Lidar panoptic segmentation for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8505–8512. IEEE, 2020.
- [12] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019.
- [13] Seungcheol Park, Shuyu Wang, Hunjung Lim, and U Kang. Curved-voxel clustering for accurate segmentation of 3d lidar point clouds with real-time performance. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6459–6464. IEEE, 2019.
- [14] Kshitij Sirohi, Sajad Marvi, Daniel Büscher, and Wolfram Burgard. Uncertainty-aware lidar panoptic segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8277–8283. IEEE, 2023.
- [15] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotequi, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.
- [16] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *European Conference on Computer Vision*, pages 677–695. Springer, 2022.
- [17] Zhi Yan, Tom Duckett, and Nicola Bellotto. Online learning for 3d lidar-based human detection: Experimental analysis of point cloud clustering and classification methods. *Autonomous Robots*, 44:147–164, 2020.
- [18] Dimitris Zermas, Izzat Izzat, and Nikolaos 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)*, pages 5067–5073. IEEE, 2017.
- [19] Yiming Zhao, Xiao Zhang, and Xinming Huang. A technical survey and evaluation of traditional point cloud clustering methods for lidar panoptic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2464–2473, 2021.
- [20] Yiming Zhao, Xiao Zhang, and Xinming Huang. A divide-and-merge point cloud clustering algorithm for lidar panoptic segmentation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7029–7035. IEEE, 2022.
- [21] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021.
- [22] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021.