Removing human figures on LIDAR

Abstract

The use of light detection and ranging (LIDAR) sensors has increased in autonomous robots and vehicles. Therefore, algorithms capable of processing the resulting 3D point cloud data are necessary. In this project, we used Point Transformer V3 and the Waymo Open Dataset to develop a segmentation model that specifically targets humans. The ultimate goal is to next remove human figures from data captured by an Ouster LIDAR. This task is part of a broader development of a student-designed autonomous rover for the European Rover Challenge. During the competition, referees move freely, often interfering with obstacle detection systems.

1 Introduction

In the domain of mobile robotics and autonomous navigation, the use of LIDAR has become widespread due to their numerous advantages. The main one is their flexibility, as they can work properly under any lightning conditions and offer a longer detection range compared to cameras. However, LIDAR sensors generate point cloud data, which presents significant challenges for object detection and segmentation because of their sparse nature. Despite this problem, the advantages are too significant to avoid using LIDAR. To take advantage of the strengths of both technologies, many complex systems adopt a hybrid approach, combining LIDAR and cameras. Even in such setups, algorithms capable of processing point cloud data for object detection are essential. The crucial step in this process is to segment the data into groups of points based on their association with specific objects. This segmentation process forms the core focus of this work. After reviewing state-of-the-art methods, we employ deep learning method with the novel Point Transformer V3 (PTv3) [14] to develop a 3D point cloud segmentation model trained on the Waymo Open Dataset [31] and generalized for use with Ouster LIDAR data.
The broader context of this research is to improve object detection on a student rover project at EPFL, Xplore, competing in the European Rover Challenge (ERC). The primary objective is to develop an efficient and effective segmentation algorithm specifically to identify human figures. This is motivated by the need to address an issue observed during the competition: referees moving near the rover were detected as obstacles, interfering with path planning. The final goal is to remove human figures from the LIDAR data and implement the algorithm in real time on the rover. The human removing part will be addressed in this work but the on-board implementation will be addressed in future research.

2 Related Work

The task of human segmentation in LIDAR data is a complex task that intersects multiple domains, including 3D point cloud processing, semantic segmentation and human detection. Sub-tasks within this field have a wide range of applications such as autonomous driving, autonomous robots and computer vision.

2.1 Image Processing Generalization

In semantic segmentation, analysis is typically divided into three levels: scene-level, object-level, and part-level segmentation. For this work, we focus on scene-level segmentation, as the point clouds used come from general outdoors scenes. More specialized techniques, such as panoptic segmentation [2, 15, 18], which differentiate multiple instances of the same class, have been developed specifically for autonomous vehicles due to their similarity in the environment.
Recent solutions use deep learning techniques, achieving notable success. However, challenges persist when working with 3D point clouds due to their sparse and unordered nature, which contrasts with the structured format of regular images. While adapting classical image segmentation techniques to 3D data is a logical approach, it is not straightforward for the reasons mentioned above. For example, convolution is a very useful tool in image processing but not directly applicable in 3D. KPConv [19] enables the use of convolution by first creating order within the data. SparseConvUnet [1] fully takes advantage of the 3D information and computes convolution on sparse data to be more efficient.
The strategy of adapting image-based methods to 3D data has gained traction recently, with models like the Segment Anything Model [7] (SAM) and latest version SAM2 [22]. Developped by Meta AI, SAM is a general and promptable image segmentation model. Its second version expand capacities to videos. Inspired by recent Large Language Models, SAM aims to create a foundation model for images where it could recognize objects it has never seen before. It also resolves prompt ambiguities, for example when pointing to a person, is the target the person or just their shirt, by working on multiple scales at the same time and giving simultaneously all possible masks as output.
Building on SAM, models like SAM3D [8], Segment-lidar [27] and Segment Anything in LIDAR (SAL) [3], extend its utility to 3D point clouds. SAM3D and Segment-lidar process RGB point clouds by projecting them into multiple 2D images, each processed using SAM. The obtained masks are then merged to recreate the segmented 3D scene.
SAL is designed to handle general 3D data without relying on RGB values. It is a text-promptable, zero-shot model and includes a pseudo-labeling engine to create more labeled data for training than with human labeling methods. Since model performance correlates with the quantity of training data, this enables to improve the results. SAL combines SAM image masks generation with CLIP [6], which links visual features and language. Using a calibrated sensory setup, the masks can be then transferred to the LIDAR data.

2.2 3D Point Cloud Segmentation

Looking now at solutions focused solely on semantic segmentation on 3D point cloud data, many models were proposed and improved continuously. Sarker et al. [4] provide a comprehensive review of deep learning methods for semantic segmentation and object classification in 3D point clouds.

Classical Methods For the classical methods, we can divide into three main categories: projection-based, discretization-based and hybrid methods. Projection-based methods project the 3D point cloud into multiple 2D views, performs semantic segmentation on each view, and then reassemble it into a 3D representation. Discretization-based methods transform the continuous data into a discrete representation, usually using a grid representation, which facilitates the application of convolution operations. The hybrid method combines the strength of the two other.

Learning-Based Methods Supervised learning methods for 3D point cloud segmentation can be divided into two main categories: feed-forward training and sequential training. Unsupervised methods, such as self-supervised or with model generation, exist but will not be the focus as results are less promising.
Feed-forward training processes individual points through multiple layers of a neural network allowing to uncover complex relationships. Three main methods are part of this sub category: Pointwise MLP, convolution-based and graph-based methods. Pointwise multi-layer perceptron (MLP) methods use fully connected layers processing each point independently in order to find local features then used to find global features with max-pooling. PointNet [11], its successor PointNet++ [23] and RandLA-Net [10] are good examples. PointNet++ introduces hierarchical neural network to capture features at different scales in local regions. RandLA-Net uses random points sampling to reduce computational and memory usage. Convolution-based methods adapt the convolution operator from classical image processing. KPConv [19] is a notable example, leveraging convolution to learn local and global

features effectively. Graph-based methods use the efficiency of graph representation for 3D points relationships, enabling graph convolutional networks to extract features, SPG [24] demonstrates strong results using this technique.
Sequential training orders the points before analyzing it, allowing outputs from previous steps as inputs for the next ones. Commonly apllied in recurrent neural networks (RNN) and transformer-based methods. RNN methods resemble MLP techniques but incorporate contextual information. 3P-RNN [25] combines multi-scale pooling with a bidirectional RNN to capture both local and global contexts. Transformer-based techniques uses Transformers [26], initially developed for natural language processing for its ability to capture long distance relationships. Their self-attention mechanism is particularly effective for weighing point-to-point interactions. Point Transformer [12] uses the self-attention to retrieve data in local neighborhoods. Point Transformer V2 [13] introduces grouped vector attention and partition-based pooling improving efficiency and scalability. Point Transformer V3 [14] further reduces memory usage, improves speed and performance. It simplifies the design by replacing KNN with serialized neighborhoods, uses simpler attention mechanisms and removes the need of relative positional encoding.

2.3 Human Removal

The task of human removal from LIDAR data has received limited attention due to its specificity. Schauer et al. [5] developed techniques to remove all moving objects from LIDAR scans using non-learning techniques. Similarly, Kim et al. [9] proposed an algorithm to create static laser maps useful for Simultaneous Localization And Mapping (SLAM) applications in autonomous robots. Using a combination of spatial and temporal data, Zhong et al. [21] and Zhang et al. [17] presented models to segment human figures on LIDAR.

3 Methodology

In order the segment humans on LIDAR, deep learning methods are preferred according to results found in the previous part. This section outlines the process of model selection, dataset preparation, and the evaluation metrics employed to assess model performance.

3.1 Model selection

Selecting an appropriate model is critical, as it must be compatible with the LIDAR data format and proven effective for human segmentation. Initial attempts included models built on SAM [7], but were not successful as SAL [3] code was unavailable, and Segment-lidar [27] required RGBD point clouds. Similarly, models removing all dynamic objects like Removert [9] were not suited as they do not specifically target human segmentation and lack proof of effectiveness.
To explore alternatives, the Open3D-ML library [16] was utilized, as it provides access to four established models: Point Transformer [12], RandLA-Net [10], SparseConvUnet [1], and KPConv [19]. While this allowed for initial experimentation, the installation process was complex, and these models did not achieve state-of-the-art results.
Ultimately, Point Transformer V3 (PTv3) [14] was selected due to its impressive performance and scalability. The model was installed via the Pointcept library [28] and trained using a robust configuration of four Nvidia H100 GPUs.

3.2 Datasets

Comprehensive and high-quality datasets are essential for training deep learning models. Autonomous vehicle research datasets, which often include human figures, are particularly suitable. Three main datasets are used in the research: SemanticKITTI [29,30], Waymo Open Dataset [31] and nuScenes [32]. All having different formats but provide similar content. The Waymo dataset was chosen for the model training because results were published for PTv3 enabling to compare with our own.
A second dataset was needed to evaluate the performance under similar conditions as on the rover. The JRDB dataset [20] was first considered for its resemblance to the final data. Ultimately, a custom dataset was created for optimal evaluation. This dataset was collected using an Ouster OS0-32-Rev-05 LIDAR sensor capturing sequences with human figures in diverse environment and settings. All human figures were then manually labeled to asses the performance of the model.

3.3 Evaluation Metrics

Two metrics are mainly used to evaluate segmentation: Accuracy and Intersection over Union (IoU).

Accuracy This metric shows the proportion of correctly predicted outcomes. Where TP, TN, FP, FN respectively represent True Positive, True Negative, False Positive, False Negative. The mean accuracy (mAcc) and overall accuracy (OA) are also widely used to show the accuracy over all instances for OA and across multiple segmentation classes for mAcc.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Intersection over Union The IoU metric is used in segmentation evaluation, showing the ratio between the overlap and union of two sets, the predicted bounding box (A) and the ground-truth bounding box (B). The mean IoU (mIoU) is the calculated average across all classes.

$$\text{IoU Score(A,B)} = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

Relevance for Human Segmentation For our application, where only the human segmentation performance is evaluated, most of the metrics are similar and only the accuracy of the pedestrian class is important.

4 Results

As discussed in previous sections, the Point Transformer V3 model was selected and trained using the Waymo Open Dataset. To validate the model's installation and training configuration, its performance was compared against the published results from Pointcept under the same settings. Our implementation achieved a mIoU of 0.6971, which is close to the published value of 0.7120, confirming the correctness of our setup.
Next, the trained PTv3 model was applied to our custom dataset. The Waymo dataset contains 22 classes, but only the pedestrian class is relevant for our objective of human segmentation.

4.1 Dataset Preprocessing

In order to achieve accurate segmentation, our dataset had to be formatted similarly to the Waymo dataset, used for the training. Waymo stores point cloud data using mainly three NumPy files: one for the coordinates in 3D, one for the light intensity value of each point captured by the LIDAR and one for the ground truth labels. Apart from the files format and organization, the main difference comes from scaling the spatial values and normalization of the intensity values.
Our dataset contains sequences in three resolution (512, 1024 and 2048), representing the number of points captured per full rotation. Each frame consists of 20 rotations at various heights. Experiments were conducted to determine the optimal intensity clipping thresholds for different resolutions. Results (Fig. 1) showed that resolution influenced these thresholds, halving the value with each increase in resolution due to the extended capture time for each point. The same experiment was conducted for the optimum coordinates scaling in order to have humans of the same height as in the training data.

4.2 Segmentation Performance

Using the optimal preprocessing parameters for each resolution, segmentation outputs were analyzed across different detection ranges: 0-5 meters, 5-10 meters and 10+ meters.
The dataset includes nine outdoors at different resolutions and five indoors sequences captured in the 512 resolution. Table 1 summarizes the segmentation accuracy for the pedestrian class across resolutions, intensity thresholds, and detection ranges. The segmentation is accurate in the close range but not in the longer ones. The accuracy does not decrease when lowering the resolution, and still achieve overall results higher than 63% in their respective best parameters. The best result of 87.3% is achieve in the short range at the 1024 resolution.

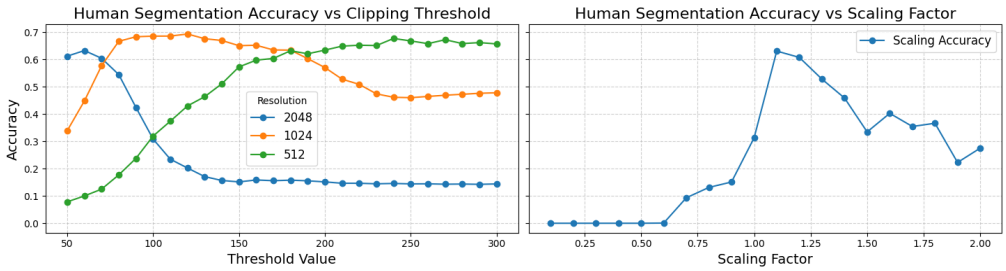
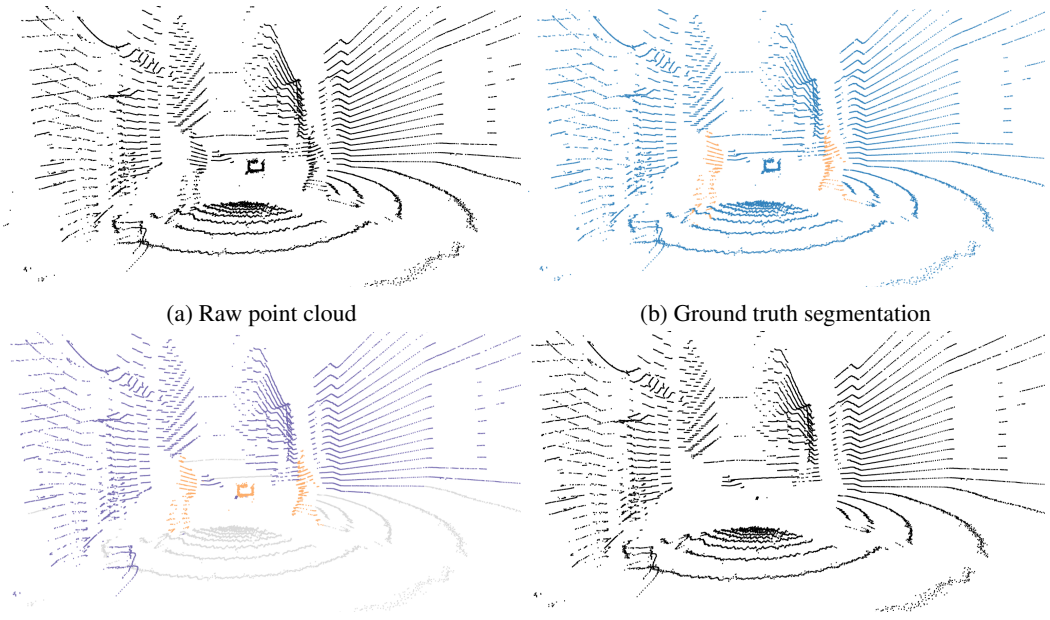


Figure 1: Human Segmentation Accuracy with changing parameters

Figure 2 illustrates the human segmentation pipeline, showing the key processing steps and resulting outputs.

	512	1024	2048	indoor	outdoor	all sequences	range: 0-5m	range: 5-10m	range: 10+m	512 range: 0-5m	1024 range: 0-5m	2048 range: 0-5m
Threshold Value: 60	0.2187	0.6689	0.6312	0.0634	0.5887	0.4207	0.6627	0.0	0.0	0.4456	0.8089	0.6760
Threshold Value: 120	0.5335	0.7665	0.4273	0.2042	0.5495	0.4383	0.5792	0.0	0.0	0.7198	0.8605	0.3841
Threshold Value: 240	0.6843	0.7917	0.2504	0.3577	0.4803	0.4408	0.5270	0.0219	0.0	0.7792	0.8726	0.2681

Table 1: Human Segmentation Accuracy at different Intensity Threshold values



(a) Raw point cloud

(b) Ground truth segmentation

(c) Model Segmentation

(d) Human Removed

Figure 2: Point cloud view at different steps

5 Discussion

5.1 Accuracy

The primary aspect to discuss is the model's effective operational range. While the model achieves high accuracy within a range of up to 5 meters, its performance diminishes for longer distances. This limitation is coming from the LIDAR characteristics therefore is not a problem because the obstacle detection algorithm is working in a similar range.
As illustrated in Fig. 2c, the model also segments objects into other classes. Because this was not the focus of this research, ground-truth annotations were limited to the human class, explaining the

difference of segmentation. This capability could offer potential utility in future applications.
The overall accuracy across each resolutions cannot be perfectly compared due to the strong dependence of detection success on the distance of the human. While the sequences were designed to be similar, they are not identical, leading to biases when humans are positioned farther away in certain sequences.
Indoor sequences presented additional challenges, with fewer full-body detections due to obstacles and occlusions. These sequences were only recorded in the 512 resolution format. The indoor model's performance limitations were not pursued as the conditions differ a lot from those in the competition. However, the model still demonstrates partial effectiveness in indoor environments.
Finally, human removal is done by simply excluding points classified as humans. This straightforward method is effective in our case because of its computational efficiency. Background reconstruction of the points occluded by the removed humans is unnecessary for this application and would increase processing time.

5.2 Computation Time

Computation time is a critical point for the further development of the project, particularly for the real-time, on-board deployment. The segmentation time for a single frame was measured using an Nvidia V100 GPU. As shown in Table 2, the processing time scales modestly with resolution (i.e., the number of points per frame).
While the current processing time exceeds the real-time requirement for a 20 Hz sensor, it is not too far, making real-time performance potentially achievable with algorithmic optimizations, reduced refresh rate or hardware acceleration.

Resolution	**512**	**1024**	**2048**
Time	60-70ms	60-80ms	70-90ms

Table 2: Single frame computation time

6 Conclusion

This study successfully implemented a human segmentation model for LIDAR data to address challenges in obstacle detection. The Point Tranformer V3 model demonstrated strong performance for this task. The generalization of the Waymo Open Dataset, used during the model training, to Ouster LIDAR data has been made possible through preprocessing and parameter tuning.
The results show that the model is effective for close-range segmentation, aligning with the obstacle detection limitation. Additionally, the straightforward human removal process is computationally efficient and suitable for real-time applications.
Future work will focus on optimizing the model for on-board real-time deployment on the rover working along with the path planning systems.

References

[1] Shi, S., Wang, Z., Shi, J., Wang, X., & Li, H. (2020) From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *arXiv preprint*, arXiv:1907.03670. Retrieved from https://arxiv.org/abs/1907.03670.

[2] Aygün, M., Ošep, A., Weber, M., Maximov, M., Stachniss, C., Behley, J., & Leal-Taixé, L. (2021) 4D Panoptic LiDAR Segmentation. *arXiv preprint*, arXiv:2102.12472. Retrieved from https://arxiv.org/abs/2102.12472.

[3] Ošep, A., Meinhardt, T., Ferroni, F., Peri, N., Ramanan, D., & Leal-Taixé, L. (2024) Better Call SAL: Towards Learning to Segment Anything in Lidar. *arXiv preprint*, arXiv:2403.13129. Retrieved from https://arxiv.org/abs/2403.13129.

[4] Sarker, S., Sarker, P., Stone, G., Gorman, R., Tavakkoli, A., Bebis, G., & Sattarvand, J. (2024) A comprehensive overview of deep learning techniques for 3D point cloud classification and semantic segmentation. *Machine Vision and Applications*, 35(4). doi:10.1007/s00138-024-01543-1.

[5] Schauer, J. & Nüchter, A. (2018) Removing non-static objects from 3D laser scan data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143:15-38. doi:10.1016/j.isprsjprs.2018.05.019. Retrieved from https://www.sciencedirect.com/science/article/pii/S0924271618301527.

[6] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021) Learning Transferable Visual Models From Natural Language Supervision. Retrieved from https://arxiv.org/abs/2103.00020.

[7] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023) Segment Anything. *arXiv preprint*, arXiv:2304.02643. Retrieved from https://arxiv.org/abs/2304.02643.

[8] Yang, Y., Wu, X., He, T., Zhao, H., & Liu, X. (2023) SAM3D: Segment Anything in 3D Scenes. *arXiv preprint*, arXiv:2306.03908. Retrieved from https://arxiv.org/abs/2306.03908.

[9] Kim, G. & Kim, A. (2020) Remove, then Revert: Static Point Cloud Map Construction using Multiresolution Range Images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10758-10765. doi:10.1109/IROS45743.2020.9340856.

[10] Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2020) RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *arXiv preprint*, arXiv:1911.11236. Retrieved from https://arxiv.org/abs/1911.11236.

[11] Qi, C.R., Su, H., Mo, K., & Guibas, L.J. (2017) PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv preprint*, arXiv:1612.00593. Retrieved from https://arxiv.org/abs/1612.00593.

[12] Zhao, H., Jiang, L., Jia, J., Torr, P., & Koltun, V. (2021) Point Transformer. *arXiv preprint*, arXiv:2012.09164. Retrieved from https://arxiv.org/abs/2012.09164.

[13] Wu, X., Lao, Y., Jiang, L., Liu, X., & Zhao, H. (2022) Point Transformer V2: Grouped vector attention and partition-based pooling. *arXiv preprint*, arXiv:2210.05666. Retrieved from https://arxiv.org/abs/2210.05666.

[14] Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., & Zhao, H. (2024) Point Transformer V3: Simpler, Faster, Stronger. *arXiv preprint*, arXiv:2312.10035. Retrieved from https://arxiv.org/abs/2312.10035.

[15] Gasperini, S., Nikouei Mahani, M.-A., Marcos-Ramiro, A., Navab, N., & Tombari, F. (2021) Panoster: End-to-End Panoptic Segmentation of LiDAR Point Clouds. *IEEE Robotics and Automation Letters*, 6(2):3216–3223. doi:10.1109/LRA.2021.3060405.

[16] Zhou, Q.-Y., Park, J., & Koltun, V. (2018) Open3D: A Modern Library for 3D Data Processing. *arXiv preprint*, arXiv:1801.09847. Retrieved from https://arxiv.org/abs/1801.09847.

[17] Zhang, T. & Nakamura, Y. (2018) Moving Humans Removal for Dynamic Environment Reconstruction from Slow-Scanning LiDAR Data. In *Proceedings of the 2018 15th International Conference on Ubiquitous Robots (UR)*, pp. 449–454. doi:10.1109/URAI.2018.8441778.

[18] Milioto, A., Behley, J., McCool, C., & Stachniss, C. (2020) LiDAR Panoptic Segmentation for Autonomous Driving. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8505–8512. doi:10.1109/IROS45743.2020.9340837.

[19] Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L.J. (2019) KPConv: Flexible and Deformable Convolution for Point Clouds. *arXiv preprint*, arXiv:1904.08889. Retrieved from https://arxiv.org/abs/1904.08889.

[20] Martín-Martín, R., Patel, M., Rezatofighi, H., Shenoi, A., Gwak, J., Frankel, E., Sadeghian, A., & Savarese, S. (2023) JRDB: A Dataset and Benchmark of Egocentric Robot Visual Perception of Humans in Built Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6748–6765. doi:10.1109/TPAMI.2021.3070543. Retrieved from https://doi.org/10.1109/TPAMI.2021.3070543.

[21] Zhong, T., Kim, W., Tanaka, M., & Okutomi, M. (2021) Human Segmentation with Dynamic LiDAR Data. In *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 1166–1172. doi:10.1109/ICPR48806.2021.9413014.

[22] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024) SAM 2: Segment Anything in Images and Videos. Retrieved from https://arxiv.org/abs/2408.00714.

[23] Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017) PointNet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint*, arXiv:1706.02413. Retrieved from https://arxiv.org/abs/1706.02413.

[24] Landrieu, L., & Simonovsky, M. (2018) Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567. doi:10.1109/CVPR.2018.00479.

[25] Ye, X., Li, J., Huang, H., Du, L., & Zhang, X. (2018) 3D recurrent neural networks with context fusion for point cloud semantic segmentation. *Computer Vision – ECCV 2018*, 11207:415–430. Springer International Publishing. doi:10.1007/978-3-030-01234-2-25.

[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2023) Attention is all you need. *arXiv preprint*, arXiv:1706.03762. Retrieved from https://arxiv.org/abs/1706.03762.

[27] Yarroudh, A. (2023) LiDAR automatic unsupervised segmentation using Segment-Anything Model (SAM) from Meta AI. *GitHub Repository*. Retrieved from https://github.com/Yarroudh/segment-lidar.

[28] Pointcept Contributors (2023) Pointcept: A codebase for point cloud perception research. *GitHub Repository*. Retrieved from https://github.com/Pointcept/Pointcept.

[29] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., & Gall, J. (2019). SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[30] Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3354–3361).

[31] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhao, S., Cheng, S., Zhang, Y., Shlens, J., Chen, Z., & Anguelov, D. (2020). Scalability in perception for autonomous driving: Waymo open dataset. arXiv preprint arXiv:1912.04838. https://arxiv.org/abs/1912.04838

[32] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).