Semantic labeling of lidar point clouds for UAV applications

Maria Axelsson[*], Max Holmberg, Sabina Serra[†], Hannes Ovrén, Michael Tulldahl
Swedish Defence Research Agency (FOI), Linköping, Sweden

Abstract

Small Unmanned Aerial Vehicle (UAV) platforms equipped with compact laser scanners provides a low-cost option for many applications, including surveillance, mapping, and reconnaissance. For these applications, semantic segmentation or semantic labeling of each point in the lidar point cloud, is important for scene-understanding. In this work, we evaluate methods for semantic segmentation of three-dimensional (3D) point clouds of outdoor scenes measured with a laser scanner mounted on a small UAV. We compare the performance of four different semantic segmentation methods, which are all applied in a scan-by-scan fashion, on semi-sparse laser data. The best method achieves 95.3% on the three classes ground, vegetation, and vehicle in terms of mean intersection over union (mIoU) on a previously unseen scene from a different geographical area. The results demonstrate that it is possible to achieve good performance on the semantic segmentation task on data measured using a combination of a small UAV and a compact laser scanner.

1. Introduction

The technical development of Unmanned Aerial Vehicles (UAV) and a drive for miniaturization of advanced sensor technologies enables new sensing approaches. For example, a small UAV can be equipped with a scanning lidar and be used for short-range applications in surveillance, mapping, and reconnaissance. UAVs with scanning lidars are well suited for mapping three-dimensional (3D) environments and can provide accurate 3D measurements during both day and night conditions.

Scene-understanding is an important step in the data analysis of point clouds from scanning lidar. This paper addresses the problem of labeling each point in a 3D point cloud with the correct class in data from a scanning lidar mounted on a UAV. This labeling problem is denoted semantic segmentation and is well studied for lidar point clouds in computer vision applications like autonomous cars, where the annotated public dataset SemanticKITTI [1] is available. However, semantic segmentation is not well studied in the context of scanning lidar data from UAVs. Applications of point cloud semantic segmentation are also found in robotics and remote sensing.

In this work, we want to expand the applications of lidar semantic segmentation. We select top-performing methods for scan-by-scan semantic segmentation of 3D lidar data from ground based vehicles and evaluate on 3D lidar data from a UAV. We focus especially on small UAVs with compact lidar sensors. For the comparison, we have chosen recent works that are high ranking on the SemanticKITTI benchmark as the lidar sensor is similar. There are some differences between the autonomous car application and the UAV application. The most prominent difference is the view of the scene and the objects. We move from a ground-based view of objects to an elevated view when we mount the lidar on the UAV and fly over a scene. The data we use from the UAV is also more sparse than SemanticKITTI and we need to adapt the segmentation methods to this type of data and evaluate the performance. Methods trained only on SemanticKITTI can not be applied directly.

The methods we evaluate in this paper are all applied on data from single scans, i.e. single rotations of the lidar scanner. This approach is common in automotive applications, since real-time analysis is needed. An alternative approach, where data is collected and jointly registered before further analysis, is often used for data from terrestrial lidar or aerial lidar. When using registered point clouds, all data in each neighborhood in the imaged space is available at the time of analysis. In a scan-by-scan approach data is sparse and only few points may be available at the time of analysis for some object features in the imaged space. The semantic segmentation still needs to provide a classification of the points based on the point coordinates and intensity. An example of a single scan measured using a small UAV and the corresponding semantic segmentation result predicted from only this data is shown in Figure 1. A scan-by-scan approach enables segmentation of a moving scene and does not require high accuracy positioning of the platform as registration of all data to a joint reference frame in a larger scene often does.

[*]Corresponding author: maria.axelsson@foi.se
[†]Work performed while at FOI.

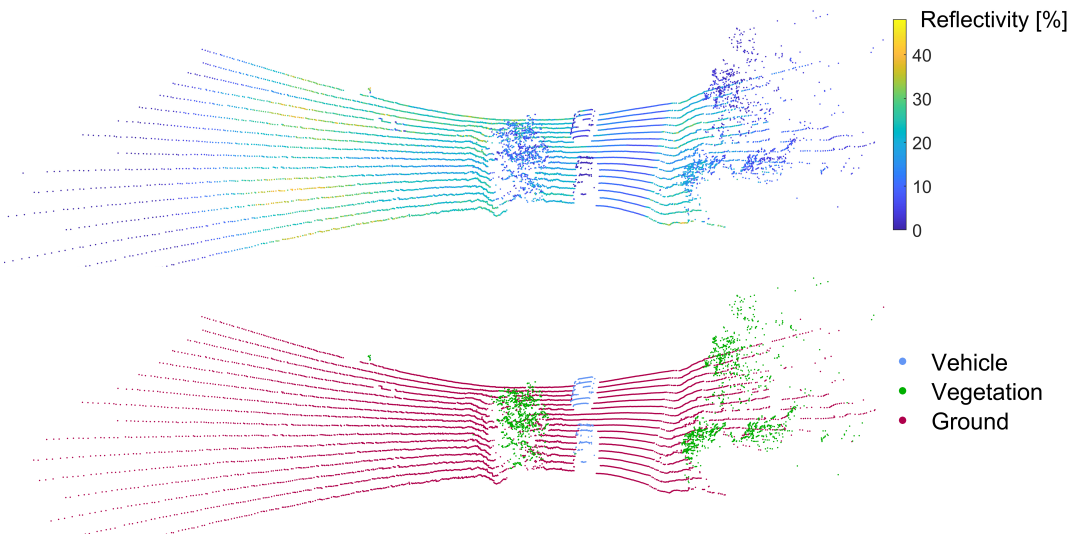

Figure 1: Point cloud from single scan of a VLP-16 lidar mounted on a UAV. Top: Colored by point intensity. Bottom: Colored by the three semantic labels predicted by the method SPVCNN. Best viewed in color.

In this paper, we present an evaluation of four methods [10, 19, 16, 20] for semantic segmentation of 3D point clouds from a small UAV with a compact scanning lidar. We evaluate the methods using annotated data with three classes: ground, vegetation, and vehicle. The methods are trained and validated on data from six different outdoor scenes and we evaluate on a separate outdoor scene from a different geographical area. We compare the methods using the parameters optimized for SemanticKITTI and apply adaptations of the batch size for the UAV data. We describe the methods for semantic segmentation in Section 3. The datasets and experiments with the results are presented in Section 4 and conclusions are drawn in Section 5.

2. Related work

Semantic segmentation of lidar point clouds has been studied for different types of applications. One field of research concerns semantic segmentation of larger point clouds from terrestrial laser scanning or aerial laser scanning. This type of research is often evaluated on datasets such as the Vaihingen dataset for semantic segmentation [14] for the aerial sensors and Semantic3D [7] for the terrestrial sensors. Recently, new datasets have been presented for aerial laser scanning [17, 8]. Typically the data is registered to a larger point cloud in a first step and sensor specific characteristics and sensor positioning are marginalized. When using registration, all data points near a specific world coordinate, or in a point neighborhood, are available in the analysis. Another field of research concerns data from compact scanning lidars mounted on moving platforms. The research topic which attracts most attention using this type of data is self-driving cars. If wanted, data can be registered to a common coordinate system, but often the semantic segmentation is intended for online analysis as input to the scene analysis in the autonomy. As data is only available up until a certain time-point a scan-by-scan approach is often applied in the analysis. For automotive lidar applications the large public dataset SemanticKITTI [1] has advanced research on semantic segmentation of point clouds in recent years, since methods can be directly compared and benchmarked. The dataset contain over 43 000 scans, with over 21 000 scans available for training and validation and the rest are withheld for testing on the benchmark. SemanticKITTI contains 19 different classes in the single-scan setup.

General advances in deep learning and advances in methods aimed for point clouds have pushed the performance also for semantic segmentation of point clouds from laser scanning. Recent overviews of semantic segmentation of point clouds can be found in [6, 18]. Methods for semantic segmentation of point clouds are now dominated by deep learning approaches. The first deep learning approaches for point clouds were introduced around 2015 using voxel grid [9] and multi-view [15] approaches. Later, in 2017, the first point-based methods were introduced [11, 12]. The development of the deep learning approaches is strongly dependent on the availability of annotated data. The release of SemanticKITTI in 2019 with the associated challenges has boosted research on data from scanning lidars and new methods are constantly improving the results on the leaderboard. There are different approaches for point cloud segmentation that performs well on the benchmark, regarding both network design and data representation. Some methods are projection based [5, 19, 10], where the point cloud

is projected to a 2D image with several bands and a 2D convolutional neural network is used for the semantic segmentation. Recently, networks using sparse convolution in 3D are improving on the benchmark [16, 4, 20, 2]. A combination of point-based and voxel-based approach is employed in [16] to support both attention to detail and possibility to scale up to large scenes. A cylindrical voxel grid and asymmetrical 3D convolution networks are used in combination with a point-based branch in [20]. An encoder-decoder CNN network which fuse the voxel-based and point-based learning with two attention blocks is used in [2].

3. Semantic segmentation of 3D UAV data

In this section, we provide more details on the methods we adapt and compare for lidar data collected using a UAV.

We compare the performance of RangeNet++[10], SqueezeSegV3 [19], SPVCNN [16], and Cylinder3D [20] on the task of semantic segmentation of single scans from a compact scanning lidar mounted on a small UAV. We have selected these methods since they have recently shown good results on scanning lidar data from a ground vehicle application on the SemanticKITTI benchmark. They also represent two different data representation approaches for point cloud semantic segmentation.

All evaluated methods are scan based approaches. They are fast compared to approaches that require joint registration of multiple point clouds before segmentation. Scan based approaches are desirable when it comes to onboard execution of the algorithms on a small UAV. Small UAVs have a limited processing capacity and algorithms should preferably be fast and lightweight.

RangeNet++ is a projection based method. The data is mapped from a point-cloud to an image of range values from the sensor with associated point coordinates (x, y, and z) and the intensity as additional channels using a spherical projection. On each projected image a fully convolutional semantic segmentation is applied in 2D. The data is mapped back to 3D and is post-processed using a k-Nearest-Neighbor (kNN) search. The network follows an encoder-decoder structure and makes use of a modified DarkNet backbone [13]. RangeNet++ comes in two main flavors, RangeNet21 and RangeNet53, that are based on two versions of the DarkNet backbone with 21 and 53 layers, respectively. In this work we use RangeNet21 in the comparison.

SqueezeSegV3 is also a projection based method. It builds on RangeNet and propose Spatially-Adaptive Convolution (SAC) which replace parts of the RangeNet network structure. It also comes in two main flavors, which are based on RangeNet21 and RangeNet53. The SAC process different parts of the image with different filters which also adapt to feature variations. The motivation for SAC is that an analysis of filter activations from RangeNet shows that specific filters are only activated in certain parts of the projected image. By adding SAC the network can be better utilized and better performance can be obtained. This is shown for SemanticKITTI in [19]. In this work we use SqueezeSegV3-21 with the same post-processing as for RangeNet21 in the comparison.

SPVCNN is an architecture which has recently achieved top-performing results on the SemanticKITTI dataset. The method combines voxel-based and point-based calculations. The voxel-based branch in the network use sparse 3D convolutions. The method employs a technique called Sparse Point-Voxel Convolution (SPVConv) where the point-based branch holds information about the high-resolution and the sparse voxel-based branch can gather information from a larger receptive field. There are also connections between the two branches. In [16], the authors also present a Neural Architecture Search (NAS) which is used to learn some of the hyperparameters in the network by training networks with different parameters and using Evolutionary Search. The NAS part is not used in our evaluation. We compare the backbone architecture called SPVCNN using the parameters that the authors have provided for SemanticKITTI.

Cylinder3D has also recently achieved top-performing results on the SemanticKITTI dataset. The method combines voxel-based and point-based calculations. Instead of a regular voxel grid, a cylindrical voxel grid is used in combination with asymmetrical 3D convolution networks to overcome difficulties with sparsity and varying density. Sparse 3D convolutions are used in this method. The point-based branch provides point-wise features to the cylindrical cells in the grid and it is used to enhance the output in a point-wise refinement module.

4. Evaluation and results

We evaluate the methods for semantic segmentation for the UAV application using annotated data with three classes. The data, experiments, and results are described in the following sections.

4.1. Dataset

A dataset was collected using a Velodyne VLP-16 mounted on a small UAV. The lidar has 16 channels (lasers) rotating 360° in the horizontal direction at 10 Hz and spanning a 30° Field of View (FoV) in the vertical direction (as seen in local sensor coordinates). The sensor is mounted underneath the UAV with the lasers imaging down and slightly forward to be able to measure the ground from above. Due to the mounting of the sensor, only about 180° of the sensor horizontal FoV provides interesting data. The measured point clouds are semi-dense with many points available from each laser in the horizontal direction and from only 16 channels in the vertical direction. At longer mea-

surement ranges objects are covered by only few horizontal lines due to the vertical sparsity.

There are some differences between this dataset and the commonly used SemanticKITTI dataset. SemanticKITTI data is collected using a Velodyne HDL-64E. The HDL-64E has four times as many channels as the VLP-16. In SemanticKITTI the sensor is mounted on a ground vehicle with the full 360° horizontal FoV and the UAV data has approximately 180°. The UAV data is both more sparse than the SemanticKITTI in the vertical direction, with only every fourth line available, and covers half of the horizontal FoV. A single scan of the VLP-16 sensor gives approximately 12 000 points in the UAV set-up. This is about 5-10% of the number of points in a single scan in SemanticKITTI. These numbers are approximate and depend on the content of the respective scenes.

The dataset contain ten flights over seven scenes in three different geographical areas. The data was collected in spring with clear to overcast weather conditions. All data show outdoor scenes from rural areas with one or two vehicles. The data includes examples of open areas with fields, gravel roads, asphalt, and grass, but also of trees and bushes. The flight altitude varies within each flight. The mean altitude is 15 m and the maximum altitude is 35 m. The distance to the vehicles varies mostly between 10 m and 25 m. In one flight the distance is up to 60 m. Vegetation and ground appear in all parts of the scene and distances are spread at both longer and shorter ranges.

The coordinates of the data points are given in a fixed sensor coordinate frame. The data was labeled with three classes: ground, vegetation, and vehicle. The ground class contains all types of ground and low vegetation that can not be separately annotated as vegetation. Example materials in the ground class include asphalt, gravel, grass, sand, rock, and soil. A small part of the data which belongs to other classes were marked as unlabeled. Also various outliers were removed from the training labels. The labeling tool released for SemanticKITTI data [1] was used for the labeling. To facilitate the labeling process, the scans were jointly registered to a common coordinate frame by combining the positioning information from the UAV and point cloud registration. The registration was performed by optimizing a pose graph consisting of pairwise ICP-registrations [3]. By using the registered point clouds in the labeling tool, a few hundred scans could be labeled together. If a very accurate registration is available, data from all flights over a static scene can be labeled together. However, this can be very difficult to achieve if expensive positioning equipment is not used onboard the UAV. The registered point clouds are only used in the labeling process and for visualization purposes in this work.

The dataset is divided into a training set, a validation set, and a test set. For training and validation, data from nine different flights showing six different scenes were annotated. All scenes contain all of the three classes ground, vegetation, and vehicle. For the vehicle class, there are cars, terrain cars, and a truck present in the data. For the test set, data from one flight from a different geographical area was annotated. This scene contains two cars. There is a total of 9931 annotated scans, i.e. single rotations of the lidar. 9361 of these are the training and validation sets and 570 are the test set. The total number of points in the training and validation sets is over 108 million and the number of points in the test set is over 6.6 million.

4.2. Experiments

In our experiments, we compare the methods directly as they are provided online with given parameters for SemanticKITTI, except for the batch size which is adapted to our available GPU-memory using results on the validation data. All networks are trained from scratch without pretraining using only the UAV data training set.

We evaluate the result in terms of intersection over union (IoU). It can be formulated as: $IoU_i = TP_i/(TP_i+FP_i+FN_i)$, where TP_i, FP_i, FN_i are the sum of true positive, false positive, and false negative predictions for class i and the mean intersection over union (mIoU) is the mean value of IoU_i over all classes. Each network is evaluated on the validation data during training. The network, which performs best on the validation data based on mIoU, is selected for the comparison of the four methods. We have trained with each parameter setup several times to get information on the stability of each method in training as well as on the validation result. In general, the result varies a couple of percent in mIoU between trainings and it is difficult to draw conclusions on network configurations or parameters when improvements are small since these can depend on the randomness of the training process.

In Table 1 we report our main result with the IoU for each class and the mean IoU for each method. The results change slightly between trainings and we report the IoU for the networks that got the best result on the validation data from a number of trainings with the same parameters. The method Cylinder3D performs best overall on the UAV data test set closely followed by SPVCNN, RangeNet++, and SqueezeSegV3. Cylinder3D performs best on all individual classes. Figure 2 shows a part of the test scene with the point intensities, the ground truth labels, and the predicted semantic segmentation using the four different networks. All scans, which cover the area, are visualized in the images. Figure 1 shows the result for a single scan using SPVCNN. It is clear that all four methods provide good results both qualitatively and quantitatively on the semantic segmentation task.

Figure 3 shows details on the cars for each method. The left column shows all scans colored by the prediction and the right column show only the misclassified points. Here

Method	mIoU	Ground	Vegetation	Vehicle
RangeNet++	0.8891	0.9844	0.8043	0.8785
SqueezeSegv3	0.8913	0.9861	0.8312	0.8566
SPVCNN	0.9352	0.9896	0.8657	0.9505
Cylinder3D	0.9526	0.9914	0.8875	0.9789

Table 1: IoU on the test set. Training on the UAV dataset for the three classes ground, vegetation, and vehicle.

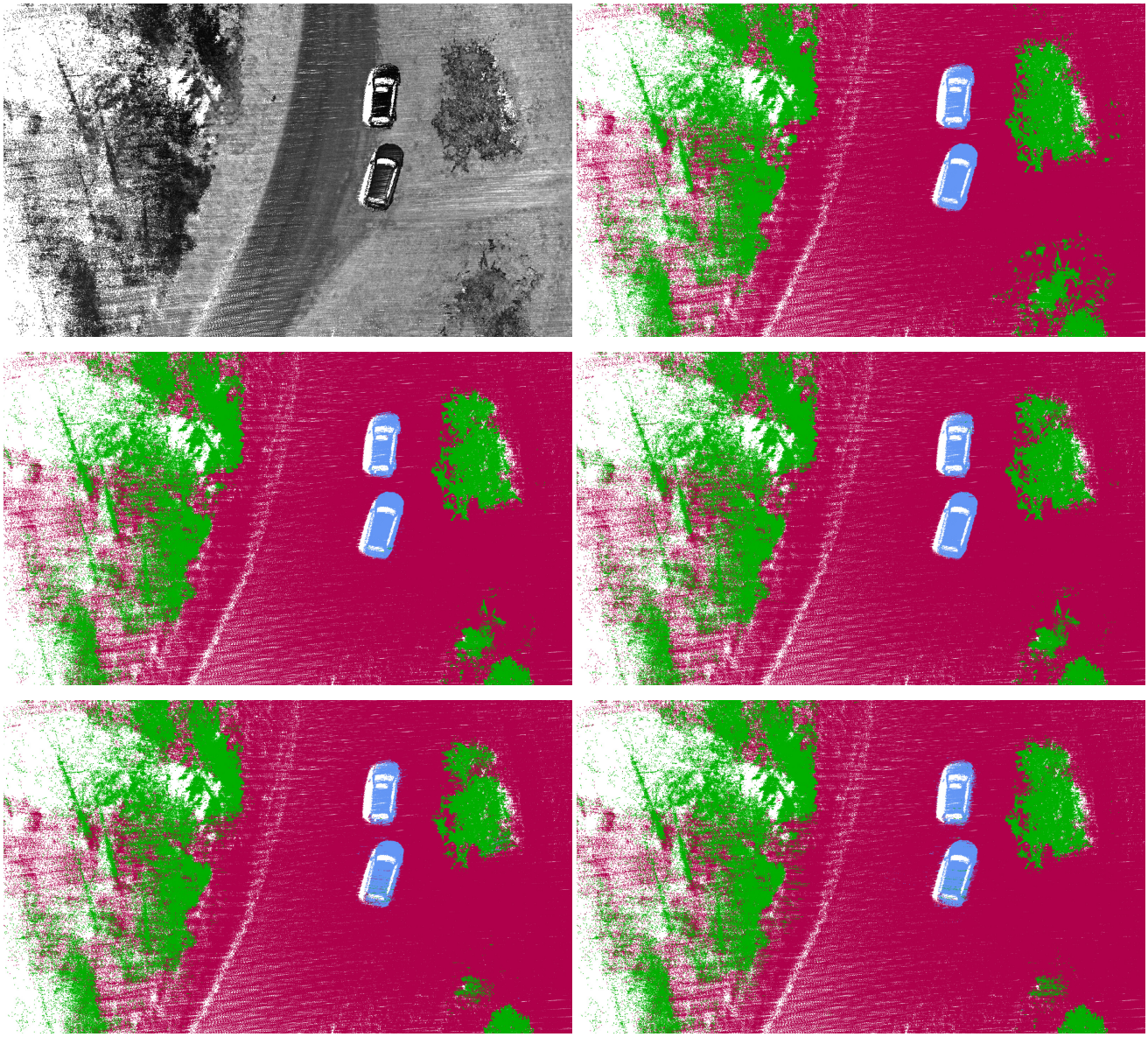

Figure 2: Point intensity and semantic segmentation result on the test data with the scans registered and visualized in a common coordinate system. Top left: Point intensity. Top right: Ground truth. Center left: Cylinder3D. Center right: SPVCNN. Bottom left: RangeNet++. Bottom right: SqueezeSegV3. Best viewed in color. Each color represents one of the three semantic classes: ground, vegetation, and vehicle.

we can see the difference in segmentation between the two voxel and point based methods and the two projection based methods. Cylinder3D erroneously predicts part of the vegetation and the cars as ground, but also part of the ground near the cars as vehicle. SPVCNN under-segments both the cars and the vegetation in this scene and erroneously predicts e.g. the wheels as ground and low vegetation as ground. RangeNet++ and SqueezeSegV3 also miss part of the wheels, and incorrectly predict part of the ground as the vehicle class and part of the car as ground or vegetation. In general, all four methods also misclassifies points on the borders between ground and vegetation. SPVCNN also misclassifies vegetation far up in the trees as ground. This is not the case for Cylinder3D or the two projection based methods even without kNN post-processing. In the visualizations, the scans are registered to a common coordinate system for illustrative purposes.

In our experiments, SqueezeSegV3 and RangeNet++ are trained using the same configuration for the backbone size and both use post-processing. They give very similar predictions on the data we have in our experiments. The post-processing using kNN improves the result for both RangeNet++ and SqueezeSegV3. The architecture design of SqueezeSegV3 was developed to utilize the network better than RangeNet++ for the scenes in the driving case. We obtain only a slightly better result using SqueezeSegV3, but the variation between trainings is larger than the difference between the results in mIoU. We apply the smaller backbone networks, RangeNet21 and SqueezeSegV3-21, on the UAV data. Experiments with the larger network, RangeNet53, for RangeNet++ did not improve the result.

The results show that the methods perform well using a relatively small amount of training data. However, it is difficult to make predictions on which method that would perform best if a large amount of annotated data was available. The UAV data is more sparse than the data in SemanticKITTI and has fewer classes. More data and more classes may change both the mIoU and the ranking of the methods for semantic segmentation. The segmentation result for the ground class is very high and the networks can be used for ground segmentation, which is useful for positioning and registration purposes.

We have also performed experiments with data augmentation including horizontal flip, vertical flip, and adjusting the range of all points in a single scan with a small constant value. However, we only obtain a small improvement on the validation data using this augmentation and it is unclear if it improves the result on the test data in the same way. SPVCNN could possibly benefit from the augmentation, but the results are within the variation obtained when running training multiple times. SPVCNN already include another type of data augmentation where the point cloud is rotated. Cylinder3D also includes a similar type of data augmentation using the default settings. The results for RangeNet++ and SqueezeSegV3 on the test data are not improved by the investigated augmentation. The results we report are obtained without this additional data augmentation.

The result depend on the quality of the annotation. Since different persons annotate different parts of the data there may be differences in how the interface between classes is annotated. Also, it is difficult to annotate some parts of each scene consistently when visual images are not available for each detail. Especially the interface between ground and vegetation is difficult to distinguish. Both of these issues may introduce slight differences between the data sequences on the border between classes.

The result may also depend on the division of the data set in training, validation, and test. We have few scenes, which result in few examples of each type of terrain and scene content. The terrain in the test data is similar to the scenes used in training and validation, but there are also differences. For example, there are no ditches or gravel road in the training or validation data. In our experiments, this may affect the absolute value of the results, but not the relative comparison. One training scene contain asphalt and five other scenes are off-road areas near forest edges. The methods perform well in our experiments, despite the small amount of data and relatively large within-class variation, but more data with more diverse appearance of each class and data for all types of interesting terrain is of course desired.

5. Conclusion

In this paper, we have addressed the problem of semantic segmentation of point clouds from UAV lidar scans of outdoor scenes. We have compared four different methods, which are all applied on single scans. The methods are evaluated on data with three different labels: ground, vegetation, and vehicle. The results show that semantic segmentation methods developed for other applications can be applied also to lidar scans from UAV with good performance, when training is applied for the UAV application. In our experiments, the method Cylinder3D performs best for this application with a mean IoU for the three classes of 95.3%. It is evident that the scan-by-scan approaches for 3D semantic segmentation is successful also on this type of data, which is more sparse and has a different view of the scene compared to data from street view scenes in automotive applications. The semantic segmentation result for the ground class is very good and this can be used for ground segmentation. The result raises an interest to investigate if it can be further improved in future work using more annotated data from more diverse scenes. It would also be interesting to add more classes in the annotations and data from different flight altitudes.

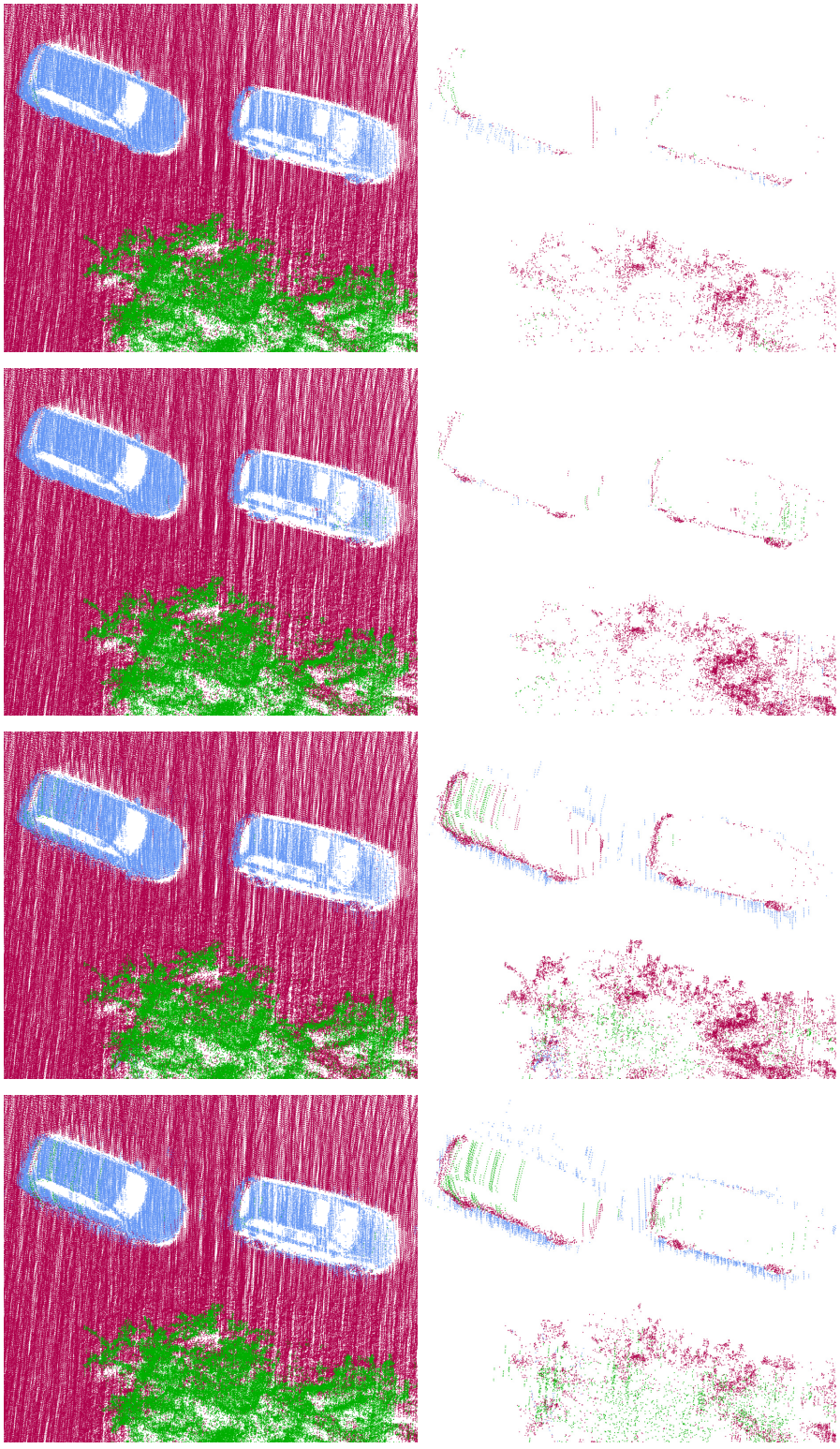

Figure 3: Semantic segmentation result showing details of the cars in the test scene for each method (left column) and the misclassified points colored by their respective predicted labels (right column). Top row: Cylinder3D. Second row: SPVCNN. Third row: RangeNet++. Bottom row: SqueezeSegV3. Best viewed in color. Each color represents one of the three semantic classes: ground, vegetation, and vehicle.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 1, 2, 4

[2] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (AF)2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 3

[3] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015. 4

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 3

[5] Tiago Cortinhal, George Tzelepi, and Eren Aksoy. Salsanext : Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving. In *Advances in Visual Computing : 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II*, volume 12510 of *Lecture Notes in Computer Science*, 2021. 2

[6] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3D point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[7] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3D. net: A new large-scale point cloud classification benchmark. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1-W1:91–98, 2017. 2

[8] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Towards semantic segmentation of urban-scale 3D point clouds: A dataset, benchmarks and challenges. *arXiv preprint arXiv:2009.03137*, 2020. 2

[9] Daniel Maturana and Sebastian Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In *Proceedings of (IROS) IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 922 – 928, September 2015. 2

[10] 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, 2019. 2, 3

[11] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 2

[12] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[13] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 3

[14] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sebastien Benitez, and Uwe Breitkopf. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, 1(1):293–298, 2012. 2

[15] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, 2015. 2

[16] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3D architectures with sparse point-voxel convolution. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 685–702, Cham, 2020. Springer International Publishing. 2, 3

[17] Nina Varney, Vijayan K Asari, and Quinn Graehling. DALES: a large-scale aerial LiDAR data set for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 186–187, 2020. 2

[18] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):38–59, 2020. 2

[19] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 1–19, Cham, 2020. Springer International Publishing. 2, 3

[20] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2, 3