Week 3 Report

Doğaç Güzelkokar, Niklas Bubeck

June 2022

1 Week 3 Goals

The goals of this week included getting first results on the behavior of sceneflow on the semantic kitti dataset, and make first evaluations on the performance. Similar, the semantic backbone should be evaluated.

2 SceneFlow

2.1 Downsampling via Voxelization

For the downsampling, voxelization is used. Therefore, for each label a separate pointcloud is extracted and downsampled by its own scaling factor. Because of this, we can emphasize points of the things classes while under-represent stuff classes also seen in fig. 1.

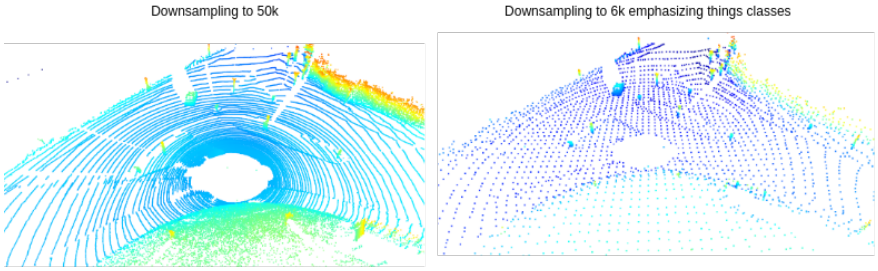

Figure 1: Comparison between two voxelization schemes.

2.2 SceneFlow Results

A short video with a higher point resolution (20k points) inferred on a RTX2080 can be found on LRZ-SyncShare. Otherwise, some pictures with the same color scheme inferred on the remote PCs are given below 3, 4. Those are down sampled to a representation of 6k points. 2 gives an overview of the used scenario.

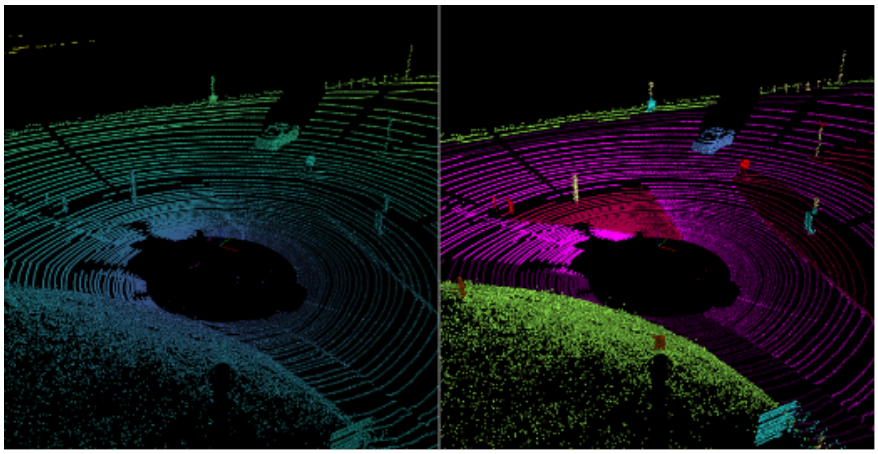

Figure 2: Overview of the scene. Visualization done via semantic-kitti-api [1]. Left: raw, Right: semantics

Here, the ego vehicle makes a right turn while another vehicle does a left turn, generating translation as well as rotation between frames.

As it can be seen in the given visualization the sceneflow operates quite good for cars, but struggles with signs and poles. We consider two possible reasons for this: (a) the number of points representing those objects is too low. (b) or more likely the training data is just not as representative for smaller objects.

3 Panoptic Segmentation

3.1 Models

We tried to setup and run DS-Net[2] and EfficientLPS[3] for 3D Panoptic Segmentation We tried to run a few panoptic segmentation networks. We were able to get inferences with efficientLPS. Unfortunately, DS-Net requires a depreceated package, spconv, which makes it very difficult to run despite our own modifications to the code to adapt it to newer versions of spconv. In the case of EfficientLPS, we had problems regarding GPU memory. EfficientLPS works by projecting 3D point cloud input to images, by downscaling the image sizes, we were able to overcome the GPU memory problem. However, the model performance is likely to decrease over this modification.

4 Literature Review

We made another literature review and found out of Self-Supervised LiDAR Scene Flow and Motion Segmentation (SLIM) [4]. They also refer to the problems of point cloud size, correspondence assumption, robustness to outliers, and

introduce some RAFT-flow backbone and use a self-supervised learning manner. BUT: there is no publicly available code base.

Just Go with the Flow: Self-Supervised Scene Flow Estimation, also makes use of self-supervised learning and provide a code-base (tensorflow) as well as a pretrained models based on nuScenes and the kitti dataset [5].

[6] also introduce self-supervised learning on lidar-based 3d data, using a pyramidal cost layering scheme. They also offer a code base (pytorch) and pretrained models.

5 Next Steps

Next steps could include upsampling the pointcloud again. Thus, getting the label for each pixel, find out the voxel that it gets represented by, take the estimated sceneflow for the voxel and apply it to the corresponding points.

Based on the issues that we were facing, with networks that got trained on RGB-D data with restricted field of view, and transferring it to 360deg lidar data. We thought one might could make use of the semantic kitti-dataset in a self-supervised learning manner, thus, making use of a method proposed in the literature review.

In case of DS-Net, it could be worth to spend more time on it as it readily provides the workflow for our case. In this regard, the depreciated spconv can be tried to be built again.

6 Questions

- Some networks we employ need more computational resources than we have, i.e GPU memory. Is it possible to have access to more resources?
- When should we start looking into implementing our own architectures instead of trying to run/fix existing models?

References

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

[2] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu, "Lidar-based panoptic segmentation via dynamic shifting network," 2020. [Online]. Available: https://arxiv.org/abs/2011.11964

[3] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada, "Efficientlps: Efficient lidar panoptic segmentation," 2021. [Online]. Available: https://arxiv.org/abs/2102.08009

[4] S. A. Baur, D. J. Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger, "Slim: Self-supervised lidar scene flow and motion segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 126–13 136.

[5] H. Mittal, B. Okorn, and D. Held, "Just go with the flow: Self-supervised scene flow estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 177–11 185.

[6] W. Wu, Z. Wang, Z. Li, W. Liu, and L. Fuxin, "Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds," *arXiv preprint arXiv:1911.12408*, 2019.

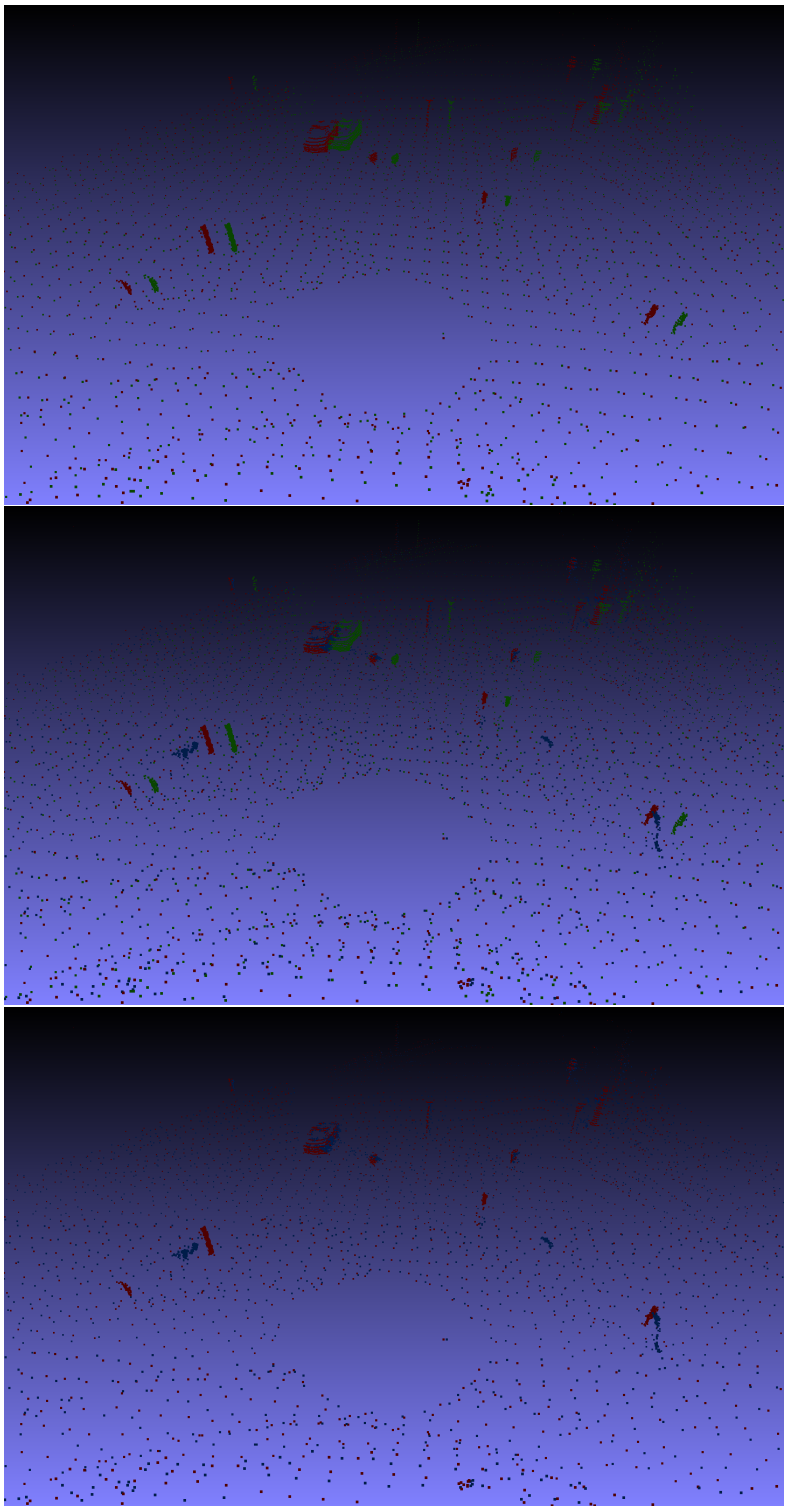

Figure 3: Green: Pointcloud 1, Red: Pointcloud 2, Blue: Pointcloud 1 + Scene-Flow of FLOT

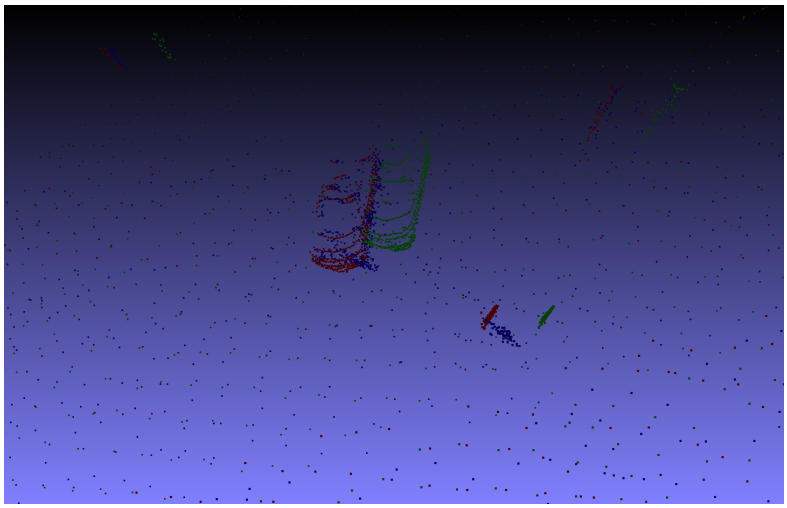

Figure 4: Closeup on the car from above