Week 2 Report

Doğaç Güzelkokar, Niklas Bubeck

June 2022

1 Week 2 Goals

The main goal of the second week was to get more confident with the networks we want to use. Furthermore we wanted to generate the first results to see on how the interface between the backbone, scene-flow and instantiation will look like. Also based on the reviewed networks we wanted to get to a common denominator regarding dependencies and versioning.

2 Rigid Transformation on SemanticKITTI Scenes

We looked into how point cloud alignment through rigid transformation looks on two consecutive SemanticKITTI scans.

Blue points belong to scan 0, red points belong to scan 1 and light green points are the points from scan 1 aligned with the points from scan 0.

In figure 4, it seems that rigid transformation is not doing too well for aligning the vehicle point cloud from scan 1. This could be a improvement point. Also, we now have a script that visualizes such alignment, and try to achieve extending it to include scene flow network. This way, we can compare rigid transformation with scene flow on SemanticKITTI data for any two consecutive scans.

3 Technical Difficulties

3.1 Dynamic Shifting Net:

3.2 FlowNet3D:

Regarding FlowNet3D, its initial implementation is given is Tensorflow, but DS-Net and other newer methods are given in PyTorch, which is also our prefered framework. Thus the initial implementation cannot be used. There is some PyTorch adapted version of the FlowNet3D given (here), but its very outdated and relying on PyTorchs THC class which was removed with newer versions. Also the pretrained model is given as a .t7 file which as of PyTorch 1.0 torch.utils.serialization is completely removed is not loadable anymore. We

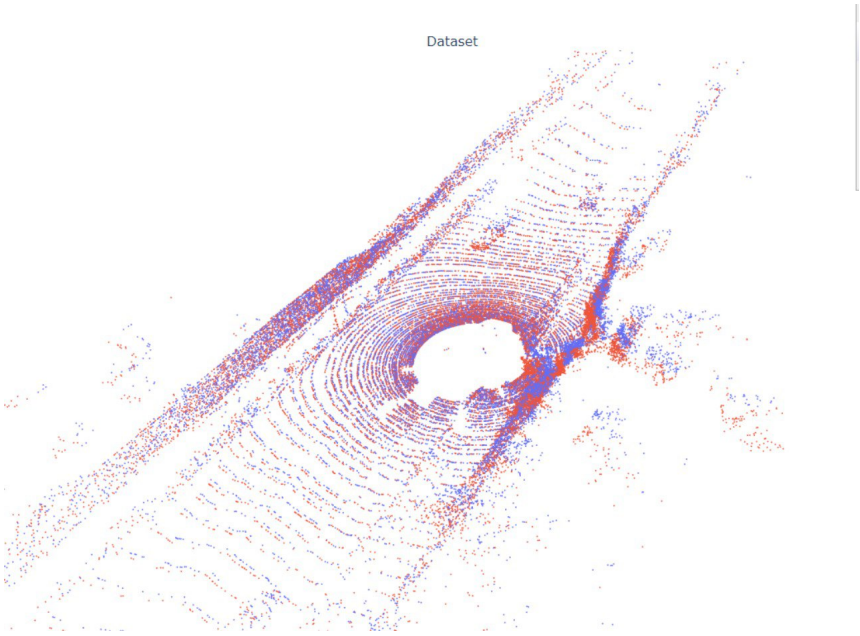

Figure 1: Scan 0 and Scan 1

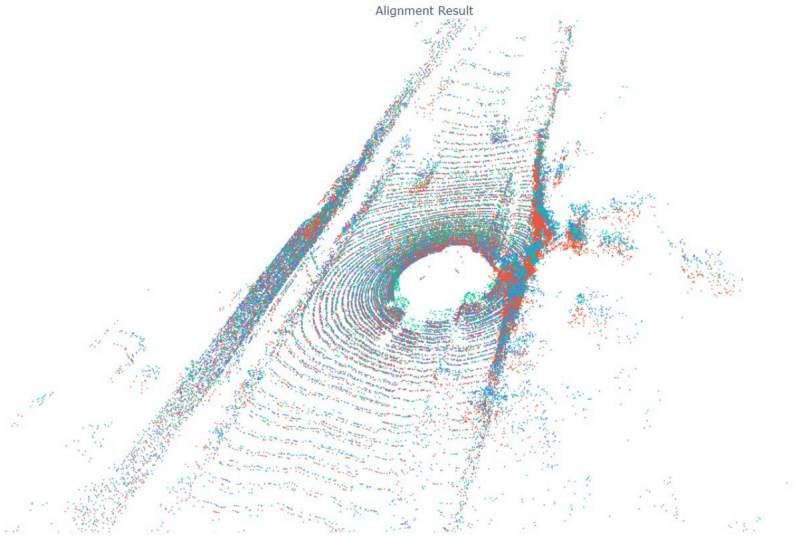

Figure 2: Scan 0, Scan 1 and transformed Scan 1

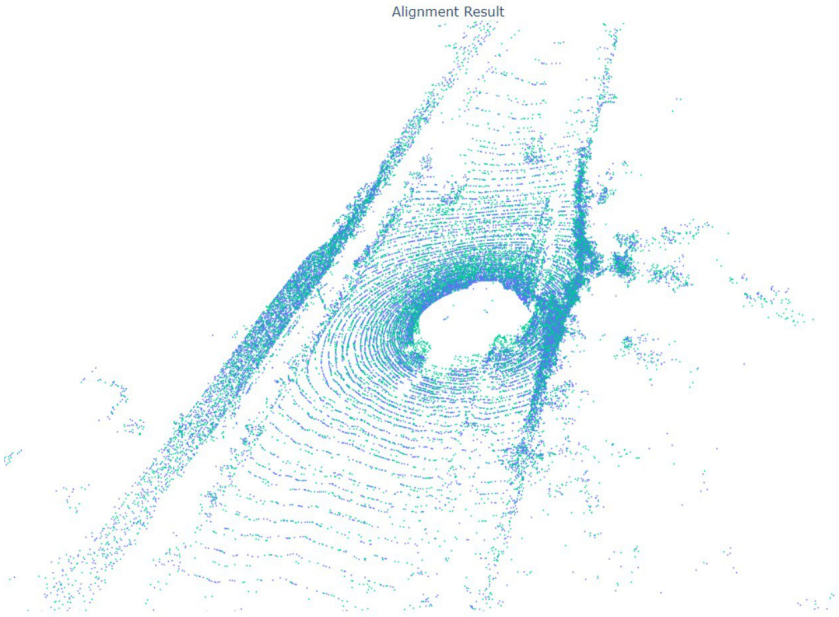

Figure 3: Scan 0 and transformed Scan 1

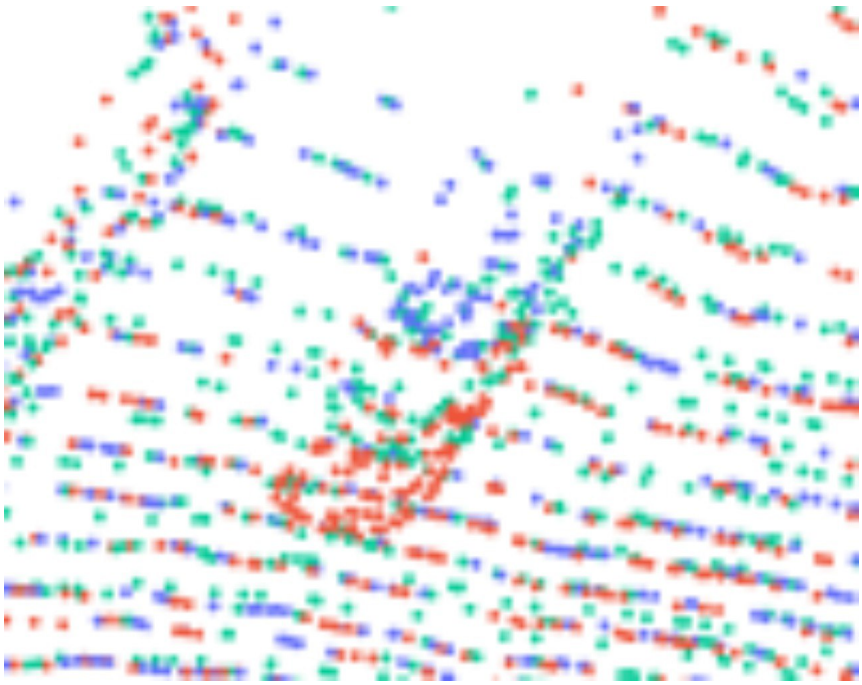

Figure 4: Close up on points belonging to a vehicle.

found a repository that seems to be able to make the pretrained model useable again, but couldnt try it out yet. If not, one would need to retrain the model. Also the pretrained model is trained on the FlyingThings dataset, if we had to retrain one could consider to directly train on the scene-flow Kitti dataset, which might be better suited to our task.

3.3 Occlusion Guided Scene Flow

We also had a look at Occlusion Guided Scene Flow, as it gathered our interest outputting an occlusion map next to the normal geometric flow. We thought that this might could help when having cases where objects leave the scene. But, it seems that the network needs some additional color values which it uses as part of the feature vector. We are not to sure if this is given considering our dataset.

3.4 FLOT: Scene Flow

We managed to get FLOT to work. However, during inference on two point clouds, it immediately outputs a memory allocation error due to the size of our point clouds (around 120k points). It is originally trained on point clouds with either 2048 or 8192 points.

4 Questions:

4.0.1 Should we include TLDR in our reports?

4.0.2 Is downsampling of pointclouds considerable regarding labeling?