Sexy Semantic Mapping.

Shounak Naik
Robotics Engineering Department,
Worcester Polytechnic Institute,
Worcester, MA, USA.
ssnaik@wpi.edu

I. Introduction

Point clouds are sets of points in 3D space that represent the surface characteristics of an environment. LiDar (Light Detection and Ranging) is a technology that uses laser beams to measure the distance to objects and generate high-resolution 3D point clouds of the environment. The objective of this project is to fuse RGB-based semantic segmentation and LiDar-based 3D point clouds. I will essentially classify and paint each point in the 3D point cloud. I will first semantically segment images, map these 2D points to the 3D point clouds and thus "classify" these point clouds.

II. Building the Map

I have used the Kitti-360 dataset for this project. This dataset has synchronized Lidar and Camera scans of the environment. The environment is mostly road data. This dataset was collected with a Camera and a Lidar mounted on a car. This dataset is very huge and thus I have only used one sequence only for this project. In that too, I have taken into account 50 frames. To begin this project, I had to convert the binary files from LiDar to Point Clouds using Open3D. The binary file had spatial information and intensity. Thus, there will be multiple point clouds and now I needed to align them and fuse them together. To do the same, I used Point-to-Point ICP implemented by Open3D. Point to Point ICP works as follows:

- ICP takes in 2 points clouds and essentially returns a transformation matrix that can be used to align these two point clouds. It is an iterative process that optimizes the transformation matrix based on the distance between corresponding points. This algorithm runs for a certain number of iterations.
- Among the multiple flavors of ICP algorithms, I have chosen the simplest- Point-to-Point ICP which minimizes the distance between points in point clouds.
- We see 2 point clouds as seen in Figure 1. These are non aligned and thus we must first align them. These figures are taken from Cyrill Stachnis's video.
- To find correspondences between these points clouds, we do a simple nearest neighbour approach to each point and thus we have correspondences as seen in Figure 2
- We then find the best transformation matrix using Equation 1.

$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \|\mathbf{p} - \mathbf{T}\mathbf{q}\|^2 \tag{1}$$

- In the above equation, p and q are point clouds and T is the transformation from q to p. Basically, we want the points as close to each other as possible and thus we optimize on Equation 1.

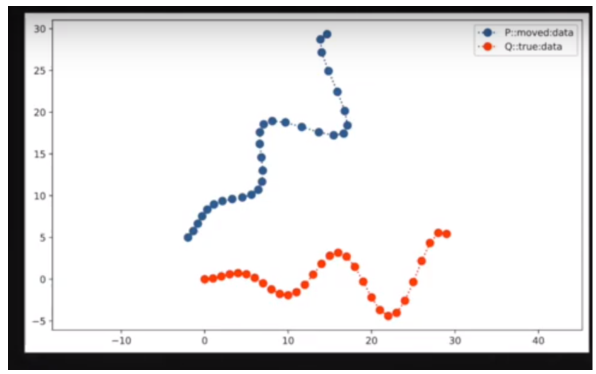

Fig. 1. Non-aligned Point Clouds

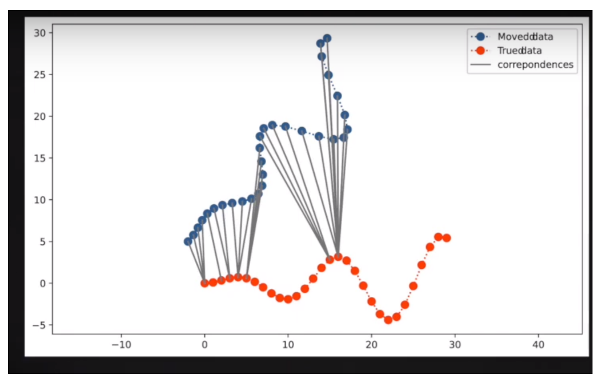

Fig. 2. Correspondences

The Point Cloud map after using this ICP pairwise on 50 such frames is seen in Figure 3.

III. Semantic Segmentation

To do Semantic Segmentation of the RGB images, I have used the DeepLabV3 framework by Google. I have chosen the resnet101 backbone for constructing this network. I have directy used the Github repository for doing inference on my images. The Github repo has pretrained weights on the 'Cityscapes' dataset which has similar classes as the KITTI-360 dataset. Thus, I chose this repository to run my

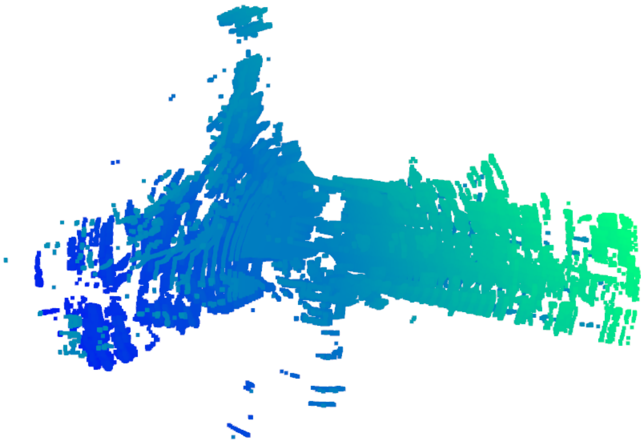

Fig. 3. Initial ICP done on Point Clouds

inference.
The model will give each pixel in the image a class label. Based on this class label, there is a colour associated with it. The output of this semantic segmentation can be seen in Figure 4.

Fig. 4. Semantic Segmentation

IV. Projection of Semantics onto Lidar

The Lidar and the Camera are seperated physically and thus the co-ordinate frames for them are different. But since we know the geometry between these 2 sensors, we know the Rotation Matrix R from the Camera to the Lidar and the Translation vector T from the Lidar to the Camera. Thus using Equation 2, I transformed the Lidar 3D points into the frame of reference of the camera.

$$X_{RGB} = R^{-1}.X_{lidar} - R^{-1}.T \tag{2}$$

Further we need to project these 3D world points onto the images using the Camera projection matrix using Equation 3.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

The Rotation matrix, translational vector and the Camera Projection matrix P is given in the KITTI360 dataset. The projection of the Lidar onto the image can be seen in the bottom figure of Figure 5.

Fig. 5. Lidar Projected on Image

Once we get these projections, we now have a direct map of the 3D to the 2D image points. We take the segmented colour of the 2D image point and use this colour back on the point cloud. Thus we get a Semantically Segmented point cloud. My semantically segmented point cloud can be seen in Figure 6. To get this map, I saved the point painted Point clouds of every frame and again used ICP to generate the map. The yellow part of the map is relating to the car.

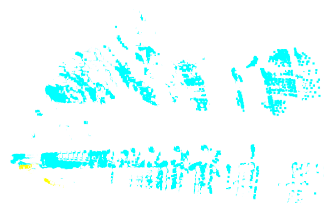

Fig. 6. Semantically Painted Point Cloud

V. Conclusions

This project gave hands on experience with Lidar point clouds and how to handle such data. This project also allowed me to explore the various models and datasets semantic segmentation is done.

REFERENCES

[1] https://github.com/VainF/DeepLabV3Plus-Pytorch
[2] https://github.com/naitri/PointPainting
[3] http://www.open3d.org/docs/release/tutorial/pipelines/icp$_{r}egistration.html$