3D Object Detection Midterm Project

Report by: **Nolan Lunscher**
Report Submitted: **2021-10-03**
Repo: https://github.com/nlunscher/nd013-c2-fusion-starter (branch: devel-nl)

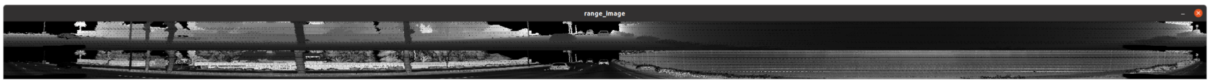
Introduction

In this project, we explore the use of BEV based 3D object detection for use with 3D lidar sensor data. We utilize algorithms adapted from image based object detection, and show how they can be applied to 3D data.

Data Visualization

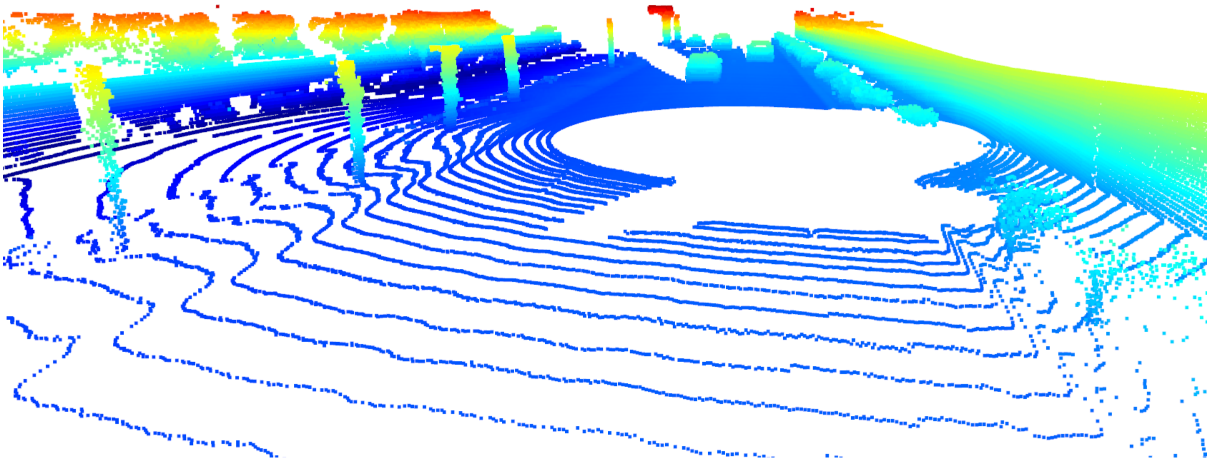
ID_S1_EX1

The output image created using show_range_image:



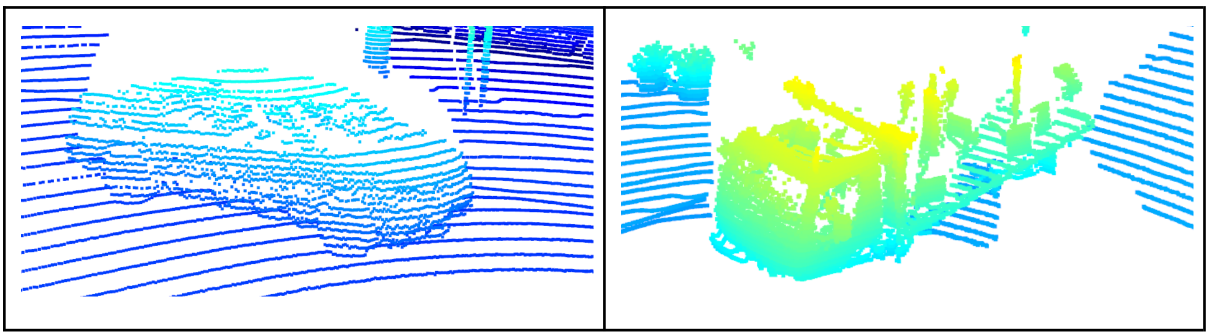
ID_S1_EX2

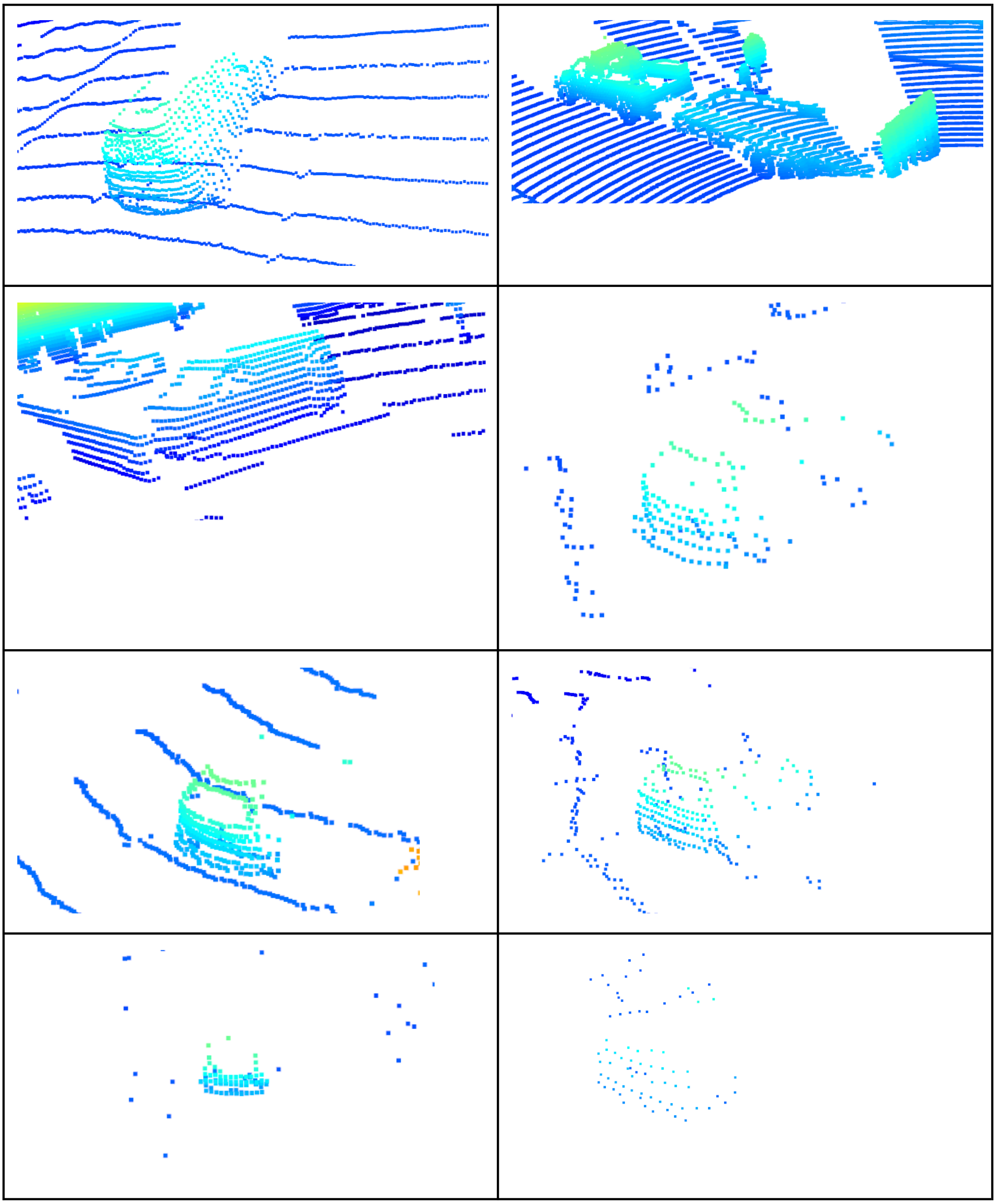
The output point cloud using show_pcl:



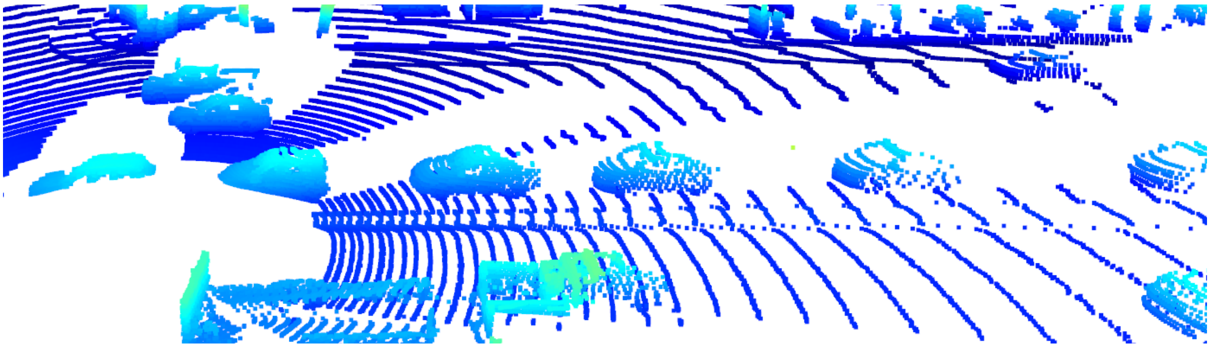
Viewing Vehicles with Varying Visibility

Below are 10 examples of vehicles that have varying degrees of visibility. Images are ordered roughly in descending order from most distinctly visible to least. Visibility largely seems to be a function of distance to the sensor, where the point cloud becomes more and more sparse. In addition to this, visibility can be affected by occluding objects creating shadows in the [point cloud.



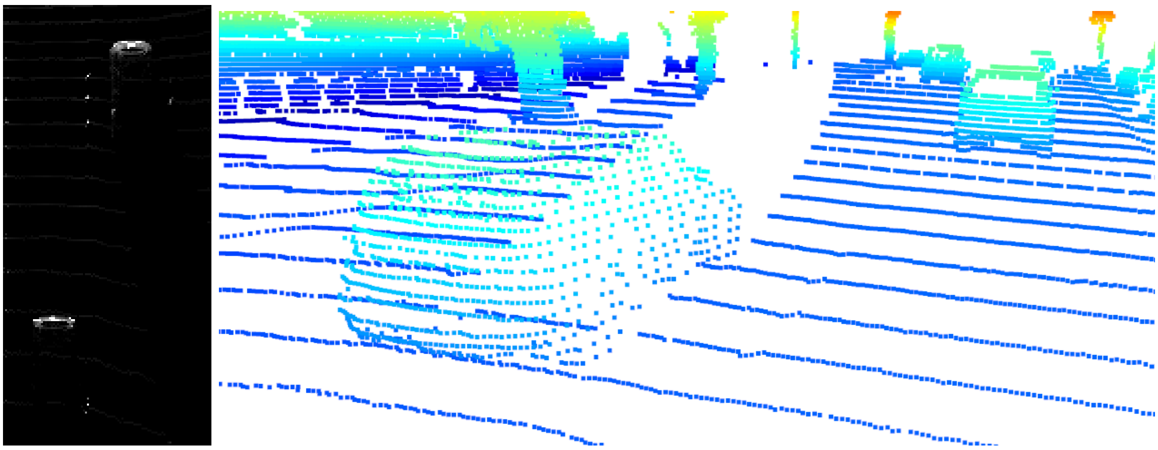


This example from sequence 3 shows a variety of vehicles with visibility decreasing with distance.



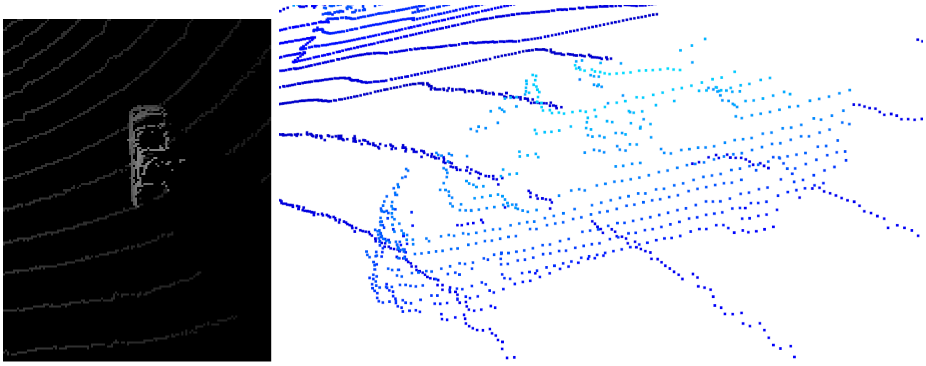
Identifying Features

In the provided images, a couple of identifying features stand out as stable features that a detection algorithm could latch on to. One of which is the car bumper surface and head/tail lights (front or back). This feature is often the main visible surface, especially for vehicles directly in front of the lidar. When inline with vehicles, these surfaces should also have a high incidence angle, resulting in stronger reflections.. These lights often tend to contain reflective materials that may cause higher intensity return signals. Examples of this can be seen below in the intensity image and pointcloud.

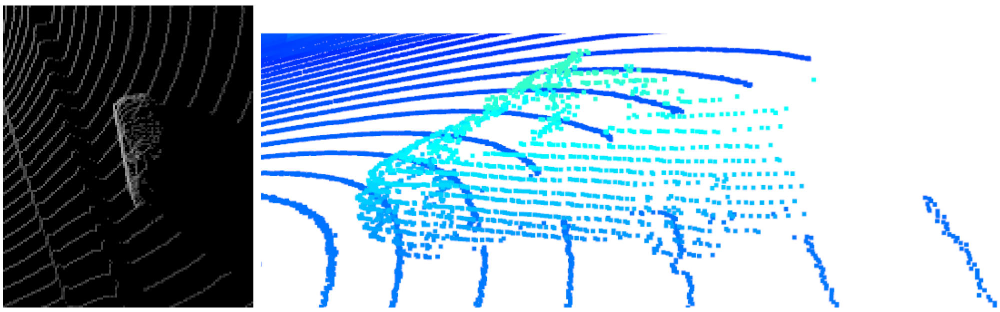


Here we can see that despite the point cloud containing a large number of vehicle points, the intensity image signal is strongest right on the rear of the vehicle where the lights and bumper are.

Another feature can be seen in the height map. That feature being the roof and the front or rear windshield of sedan cars. These can be distinct as the glass creates a distinctive hole in the points, with the roof being taller than the hood or trunk of the car. This feature is most distinct and is symmetrical in sedans, however similar features exist for other vehicles as well. An example of this can be seen below in the height map and point cloud.



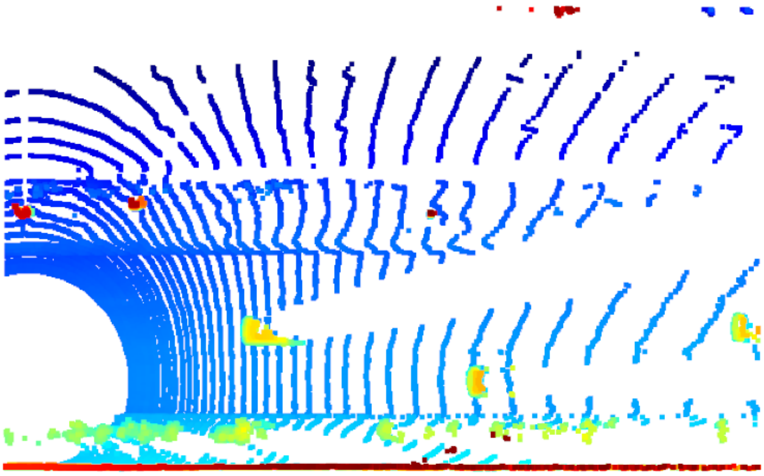
Looking at the density map, the most distinct feature is the vertical side surface of the vehicle body. This surface contains significantly more points than surrounding horizontal surfaces. This feature’s strength will depend on the body shape of the vehicle. The more “boxy” the vehicle the more point density its side will have when projected into a single BEV pixel, and the stronger this feature. An example of this can be seen in the below density map image.



Converting from Point cloud to BEV Representation

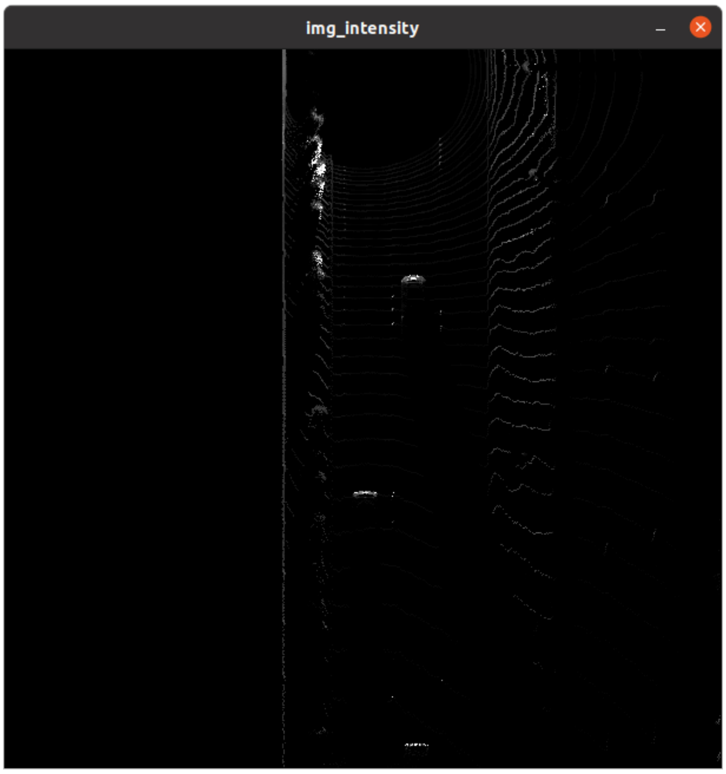
ID_S2_EX1

The output BEV point cloud using bev_from_pcl:



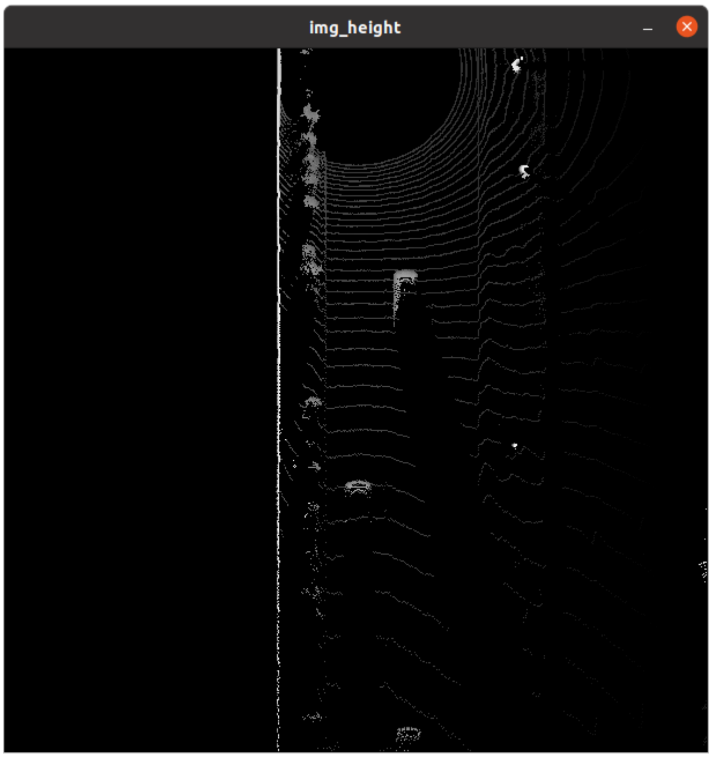
ID_S2_EX2

The output intensity map using bev_from_pcl:

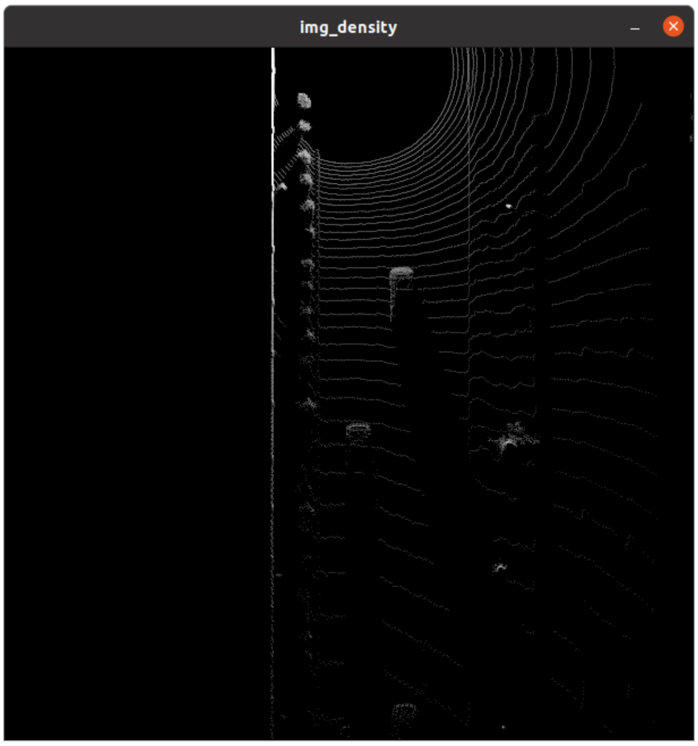


ID_S2_EX3

The output height map using bev_from_pcl:



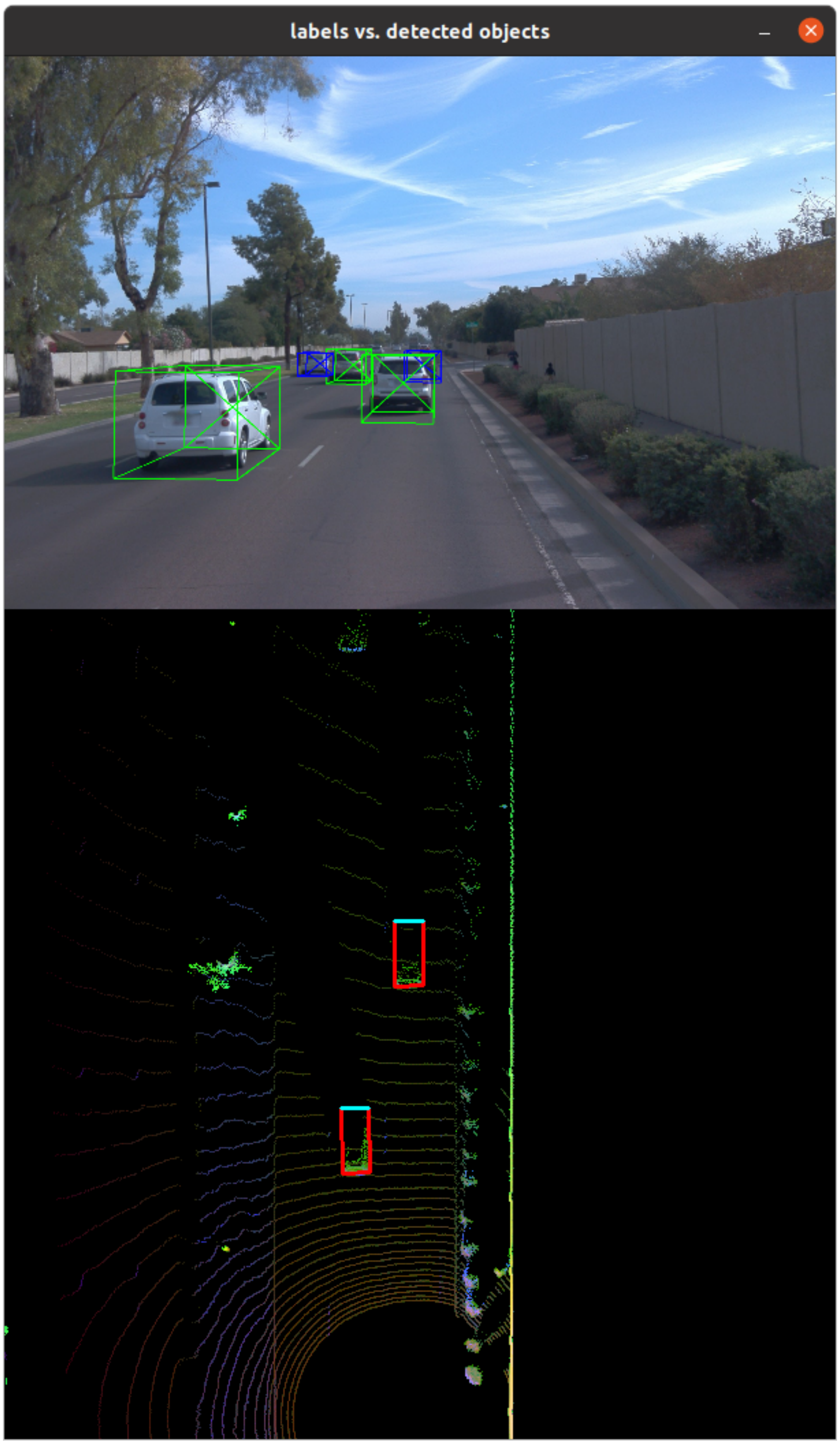
The output density map using bev_from_pcl:



Running Inference

ID_S3_EX1 and ID_S3_EX2

The fpn_resnet detections from detect_objects visualized:

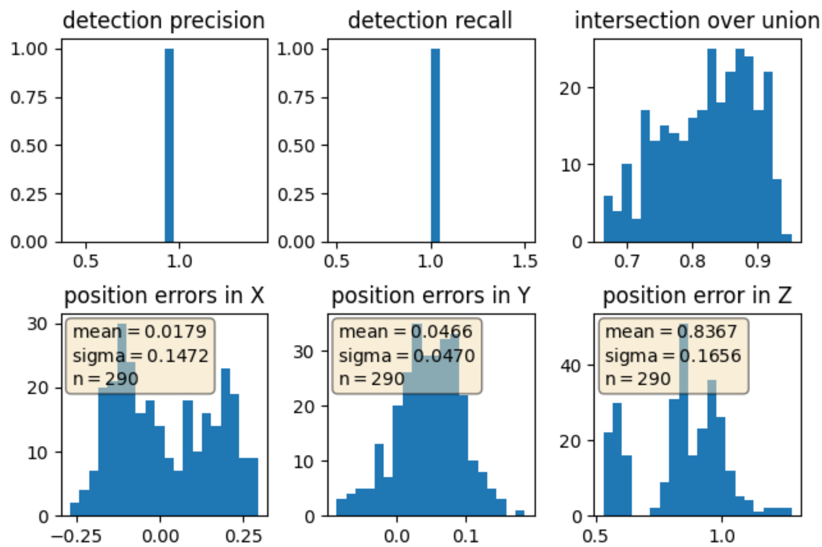


Measuring Performance

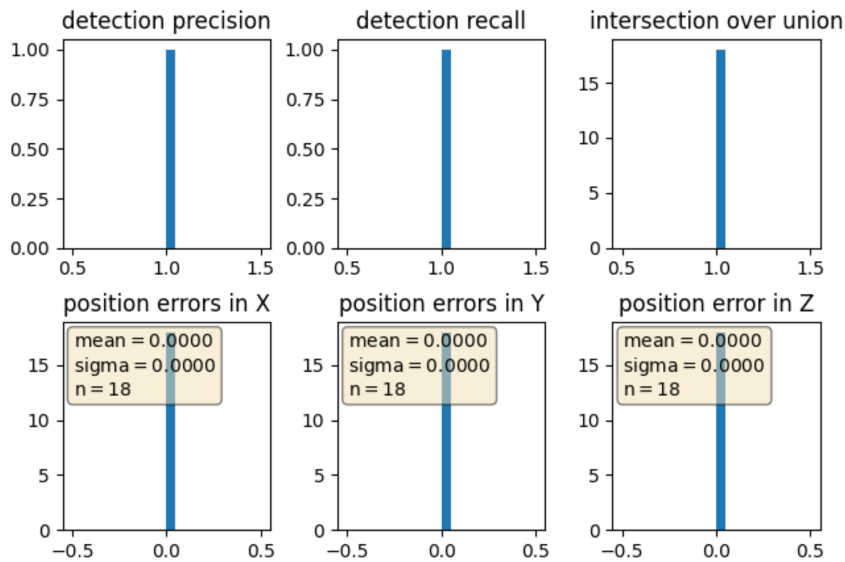
ID_S4_EX1, ID_S4_EX2 and ID_S4_EX3

Darknet

The performance metric plots when using the Darknet model:
precision = 0.9206349206349206, recall = 0.9477124183006536



The performance metrics plots when using the labels as the detected objects (perfect detector):
precision = 1.0, recall = 1.0



Fpn Resnet

The performance metric plots when using the Fpn_resnet model:
precision = 0.988929889298893, recall = 0.8758169934640523