*3D Object Detection*

Author: Joao Silva, joao.p.silva@mercedes-benz.com
Date: 2022-12-12

## Project Scope

The goal of this project is to design, implement and evaluate a deep-learning approach to detect vehicles using a LiDAR (Light Detection And Ranging) sensor. The LiDAR sensor point cloud data is processed, cropped and transformed into a BEV (birds-eye view) perspective. The BEV result is fed into an object detection deep-learning model, which output detection characteristics such as object type, pose and dimensions. The objects detected are then compared with the Ground Truth labeled objects, and an evaluation is performed showing KPIs such as precision, recall and distance standard deviations.

## Compute LiDAR Point-Cloud from Range Image

The first step in the project is to compute a LiDAR point cloud, i.e., a set of Point measures in the 3D space with associated attributes, for example intensity. In order to do so, one has to transform the LiDAR sensor measured data, and transform it into the 3D World.

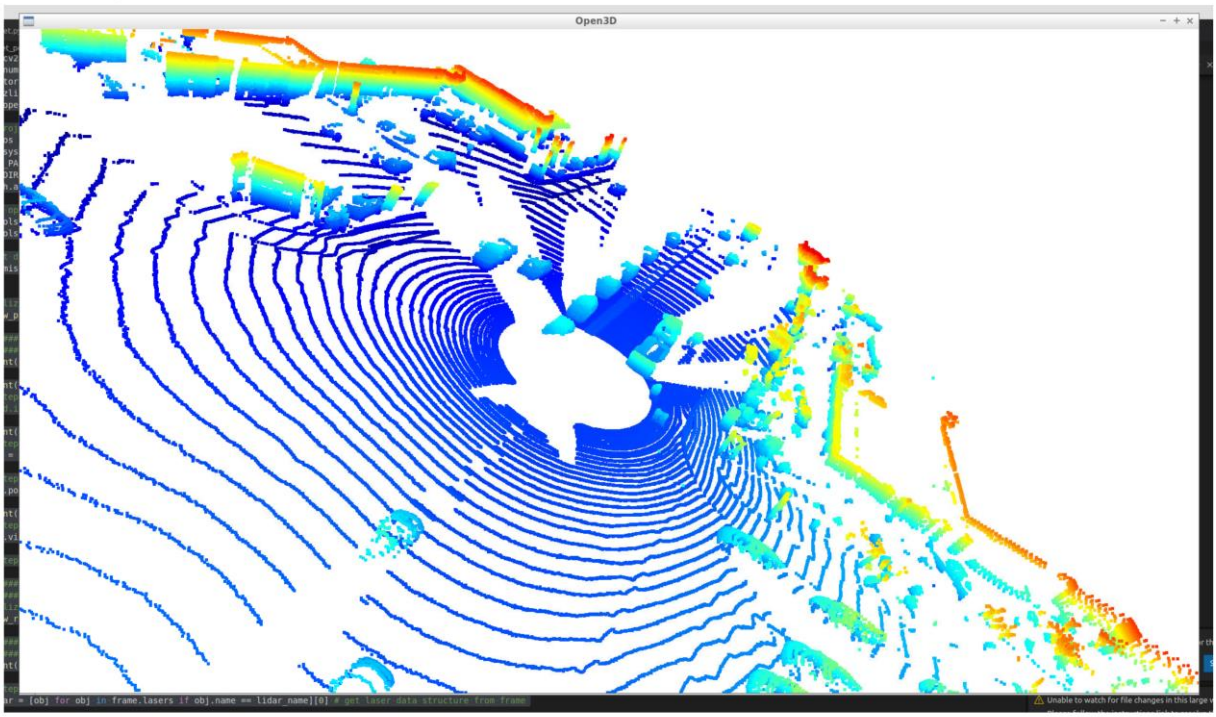
The LiDAR sensor we use is a scanning one, meaning that it consists of a transmitter/receptor pair array, spawning a vertical field of view, that swipes a given horizontal field of view. The pair emits and receives light (laser). The light goes, hits a target and returns. Measuring the properties of the return light we can identify some attributes such ToF (converted to range) or the intensity.

The following Figure shows such a scanning structure with range (top) and intensity (down) showed in a grayscale. The Figure represents the total horizontal FoV (360deg), and the vertical FoV is not up to scale.

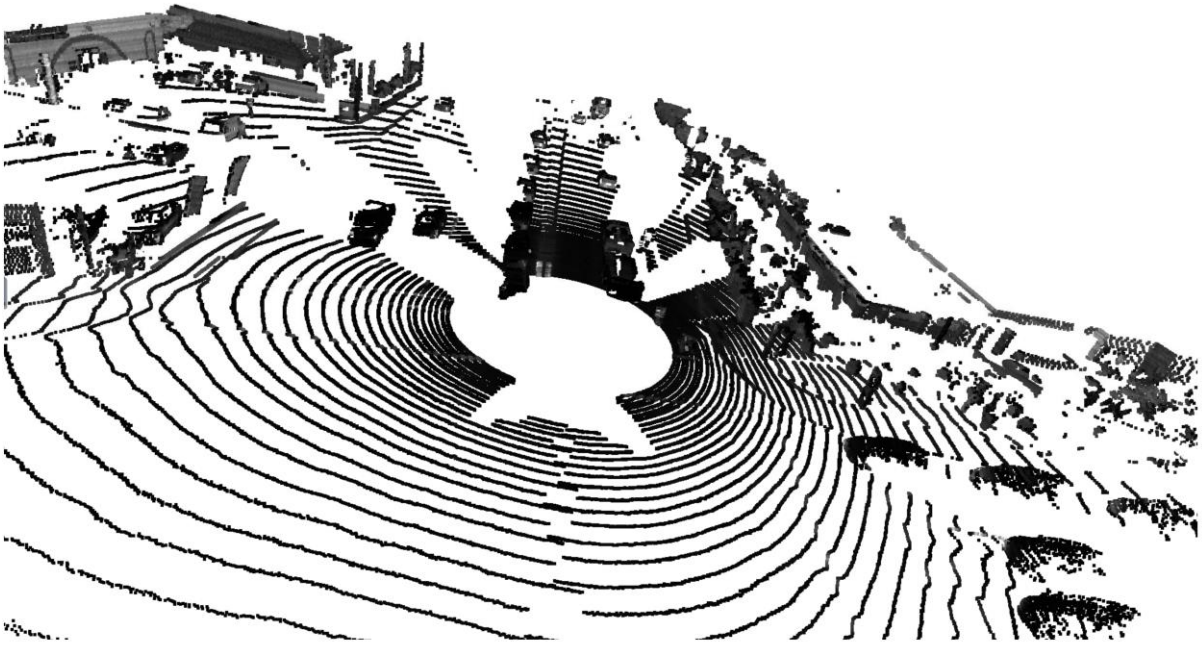
One may also summarize the transformation into something similar to

$$P(x, y, z) \quad \text{from} \quad P(r, \psi, \theta)$$

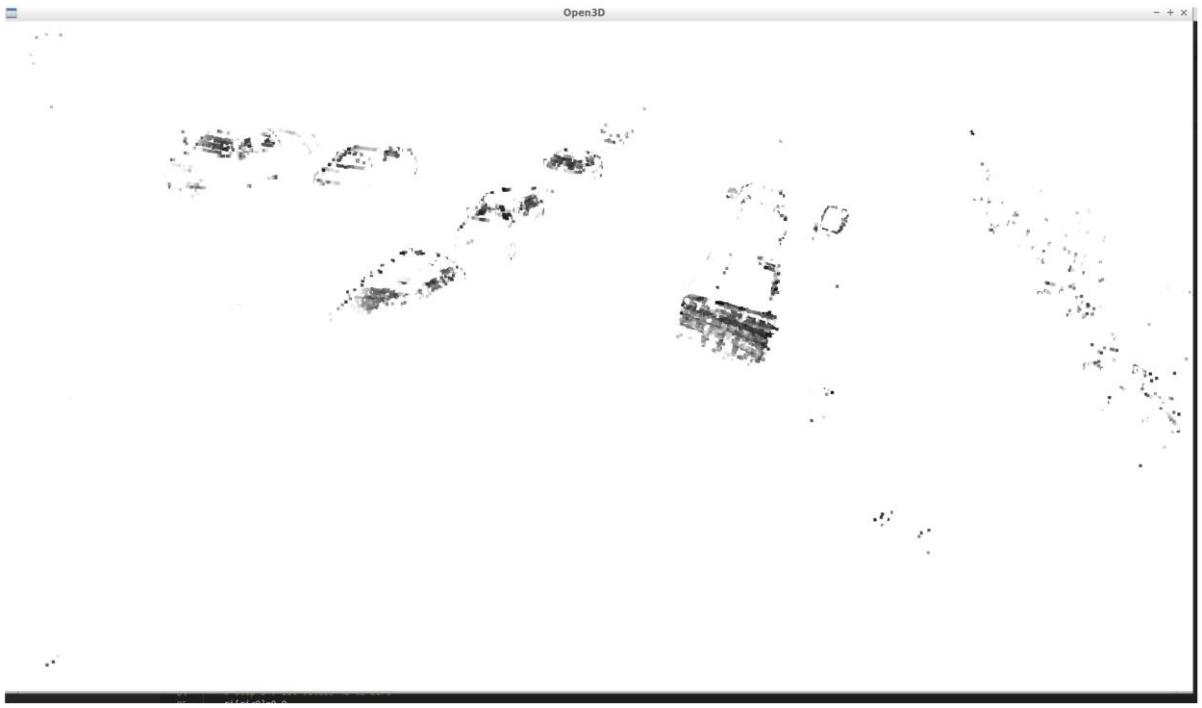
Using this conversion, one can transform raw data into 3D point cloud, which results in the following Figure




The same figure can also map the properties into colors for a better visualization. One example is to map the intensity, as displayed in the following Figure:



The intensity is shown here in absolute terms, and a better way is to scale it to show only distinguishable ranges of intensity. The following process can be seen in the following Figure:

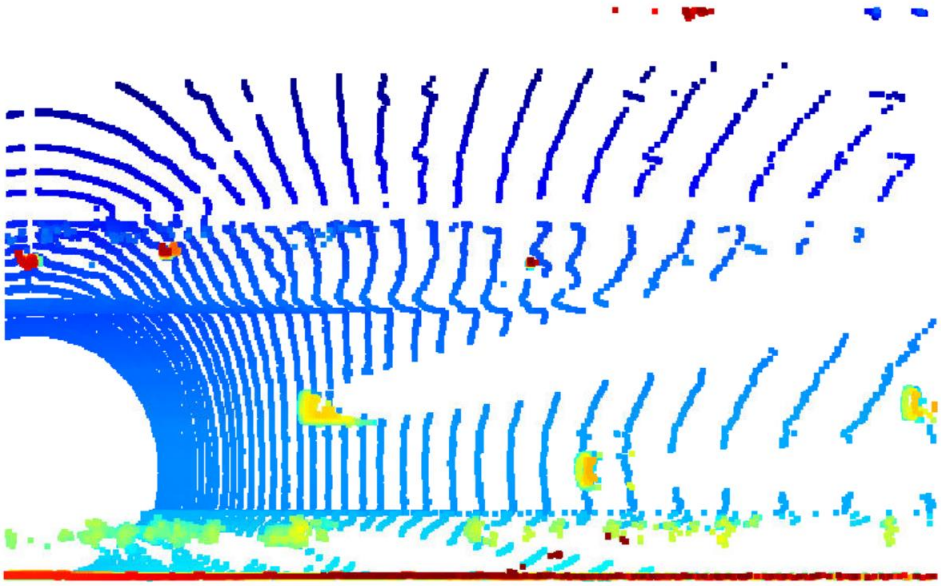



One is clearly able to distinguish certain characteristics of cars, such as the rear-bumper in the truck; windows of the cars, which clearly reflect; and edges of the vehicle. The same process can also be done in the raw intensity image, where one can, for instance, easily identify the license plates and head/tail lights, as in the following Figure, which, which is cropped to show only the forward horizontal FoV (-90deg to +90deg):

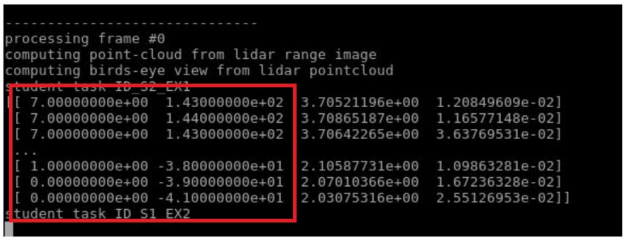

## Birds-Eye View

After analyzing the Point Cloud data one can can feed it to a Model in order to learn and extract the features that we have previously identify using object detection and classification. Many of the networks that perform such algorithms work on 2D models from Computer Vision, and one can use similar frameworks even in the 3D world. To do so, one can project the 3D into 2D from the top down view (Birds-Eye View), and use attributes for a given 2D point that result of properties of a Point Cloud (e.g. intensity) and compressions into vertical points (e.g. density and height) to replace the usual RGB scale.

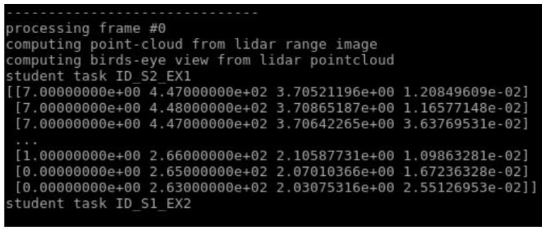
In the following Figure one can see the result of a simple Point Cloud BEV:



The PC can also be discretized. This is done in order to achieve the fixed size input that the object detection pipeline takes. In our case, we convert a PC into a 608x608 pixel frame. We also crop the PC for +/-25m lateral and 0m to 50m longitudinal. One must be careful to do so, one example of such is the conversion of meters to pixels, consisting of unsigned integers. Such thigs can be displayed in the following Figure:

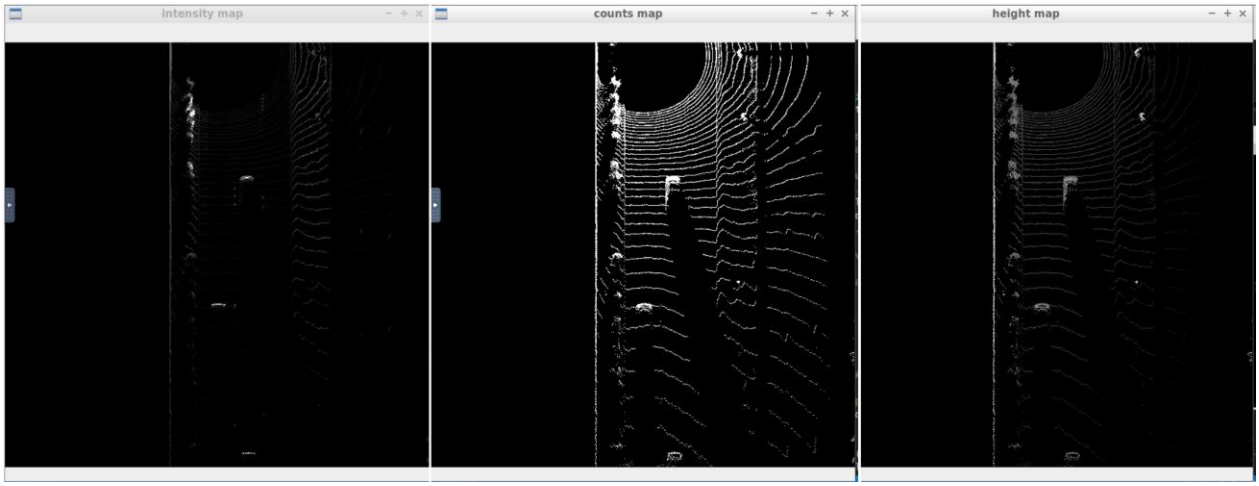



integers



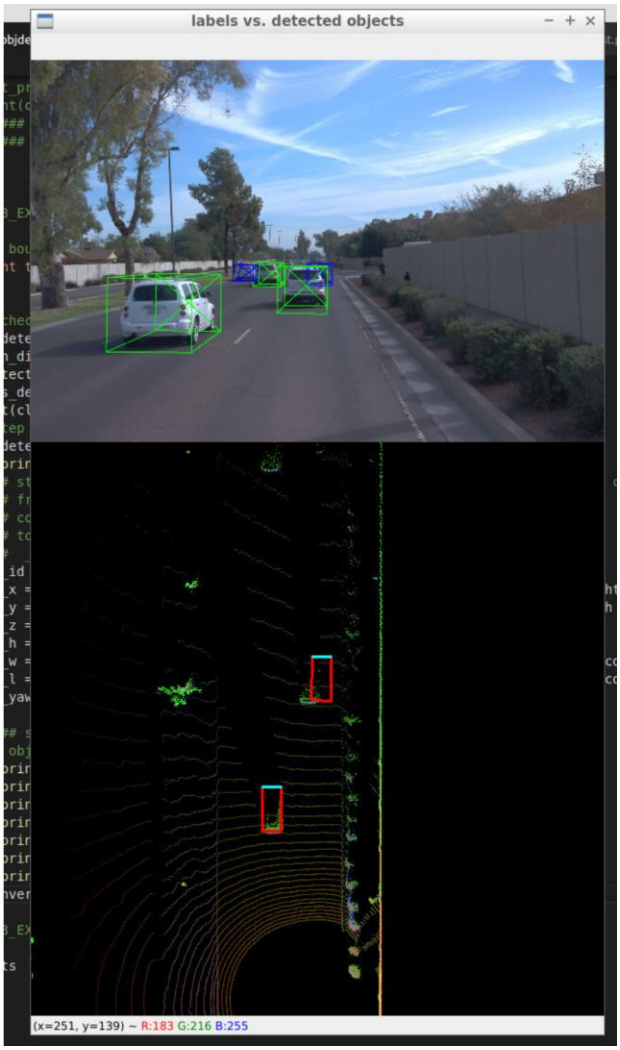

unsigned integers

Finally, we obtain an Intensity, Density (counts) and Height 2D PC image, seen in the following Figure:



## Model-based Object Detection in BEV Image

Now that the BEV Image is available in the format of the chosen model, we can feed it to the model which would output the detection and classifications. In this project, the Super Fast and Accurate 3D Object Detection based on 3D LiDAR Point Clouds is used, short with SFA3D. The repository was cloned and integrated into the project. In this stage, and since we chose to use the direct integration, some imports where temporarily disabled. As an example, one can see the result on a frame of the correctly 3D labeled GT data and the result of the detection (first without detection, then with) in the following Figure:

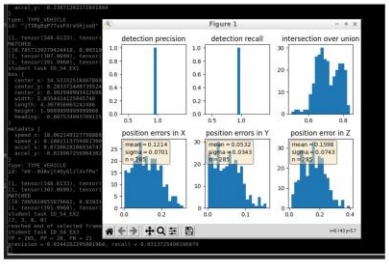




The overlap seems to be quite good, namely for the first vehicle, but not perfect, has it seems for the middle vehicle. For the last vehicle, it is not detected because it is already out of reach of the 50m.

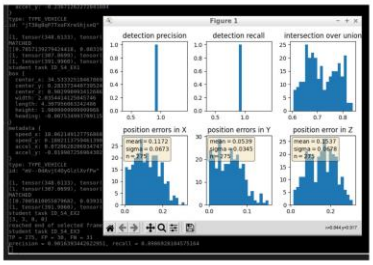
# Performance Evaluation for Object Detection

Finally, we can evaluate the performance of the Object detection algorithm. To do so, a good metric for a TP detection is the Intersection over Union (IoU) ratio of the GT object and the detected object. We implemented such metric, by comparing both the area IoU and the volume (including z) IoU. We can then compute the precision and accuracy of the model, using the TPs, FPs and FNs resulting from the evaluation, as well as the standard deviations of distances (x,y,z).

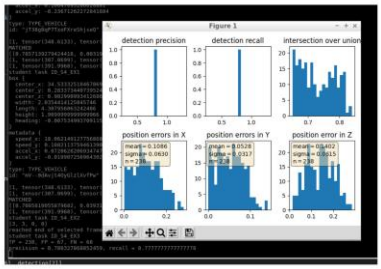
As an example, for the volume IoU, one obtain the following values, using several IoU TP ratios:




0.5




0.6




0.65

One can see the influence of IoU TP ratio over the performance. Furthermore, one can see that standard deviations are usually <20cm, and most importantly the lateral one, below 5cm.