LiDAR Point Cloud Visualization Tool
Instructor : Prof. Jaya Nair

Prajwal Agarwal (IMT2018056)

1 Abstract

This is my report for the Project elective CS 901, under the guidance of Professor Jaya Nair. This report explains the features of this tool in detail, contains the link to the code and demo video.

2 Introduction

Velodyne LiDAR systems are used in autonomous driving vehicles. These machines emit out light waves in all directions and light on reflection from different objects is received onto a sensor. These are very smart and powerful tools that are capable to recording large amount of data, received from the reflected light. Various parameters such as intensity, reflectance, position, distance etc. are recorded. This data is fed into a classification model, and the model tries to predict the object from these value. To visualize this data, I have developed this tool, so that researchers, students can visualize this dataset and try to figure out various interesting patterns from this. The aim to create this project was to develop a full-fledged tool that can displays high definition interactive visualizations from demonstration, exploratory analysis etc.

3 Intended Users

- Researchers.
- Students
- School and College Professors for demonstrations.

4 Tool Features

4.1 Widgets

1. **Central Canvas :** A canvas is used to display the visualization on the browser window using THREE.js . It covers most of the screen. Other smaller widgets are placed on top of this canvas.

2. **Controls GUI :** This is located towards right side of canvas. It gets activated as soon as the visualization file is loaded into the canvas. It shows all the classes present in classCode dictionary defined in metadata.json file. Every type of class contains options listed below :

   (a) **Show :** Checkbox button, which can be used to hide/unhide the point clouds of that particular class.

   (b) **Color :** Hex Value representing the color of the point cloud in the visualization. Value can be changed by dragging box over the color palette that is displayed on hovering over this hex value.

   (c) **NewColor :** Input box to change the color of the visualization. The value entered should be a valid hex color, else the color will not change and remain the same as before of that particular class of point cloud.

3. **Header :** This is present at the top centre of the canvas. It displays the current label name that is displayed in the visualization.

4. **Data Analytics :** This is present below the main canvas. It is used to display the bar chart class distribution and the confusion matrix of the displayed visualization.

4.2 Buttons

1. **Open File :** It is used to open a file. The file should be a csv file. An example file data.csv has been provided in the static folder. The file should be columns with names : x, y, z . These are used to plot the point on the canvas.

2. **Add New Label :** It is used add a new label using a file. This can only be used after some file has already been open in the browser, and

the user wants to add more labels into the same visualization. The new file can have many labels. However, the columns x, y, z should be the same as main file.

3. **Show Bar Chart :** After some file is loaded, only one label is displayed at a time. This button can be used to show the class distribution bar chart in Data Analytics widget.

4. **Show Confusion Matrix :** After some file is loaded, only one label is displayed at a time. For this, the user has to enter names of two labels. The label names has to be same as in the input file. This button can be used to show the confusion matrix between the input labels.

5. **Next Label :** This button can be used to see the next label, if present in the input file.

4.3 User Customization : Metadata.json

The purpose of this file is to provide some customization to the user using non-invasive methods.

1. **List : "labels"**
   This list represents the label names that can be present in label, input files. These are detected by the tool and are added to the visualization main canvas. Add new label names to this list if they are present in the file.

2. **Dictionary : "classNames"**
   This represents the mapping of all the classes to color value, that are present in the point clouds. To add more classes, update this list with the class name mapping to a new color value in hexadecimal.

3. **Dictionary : "classCode"**
   This represents the unique index of all the classes. To add a new class, update this dictionary with the next index in the list mapping to the class name.

4. **Constant : "lightMode"**
   Update the value of this variable to 0 for dark mode and 1 for light mode.

5 Code | Demo Video Links

The code is uploaded on Github. For evaluation, the demo video is uploaded on One Drive and the link has been provided here.
Code : https://github.com/Prajwal7842/LiDAR_Point_Cloud_Visualization_Tool
Demo Video : One Drive

6 Conclusion

This project is meant to be used by researchers to find classes that are predicted correctly, incorrectly by ML models, when the data from LiDAR device is collected and fetched to the model. I have tried to make the project using basic programming that I knew. I have not formally tested the code. This project has helped me learn a lot of concepts in development of software tools, process of requirements gathering, use of figma to draw layout, writing SRS docs etc.

7 References

1. ThreeJS Starter

2. Three Js Library

3. D3 js Library