DLAD Project 1 Report

Kexin Shi, Zheng Shen

March 2021

1 Problem1

To get a BEV image with resolution (0.2,0.2)m, we divide the length and width of the image by 0.2m into several bin firstly. For each 3D point, We only care about its x and y coordinates and map it to its corresponding bin. If its reflectance value is larger than the existing value in the bin, we replace it. It will ensure that we take the highest intensity point if multiple points lie within the same bin.

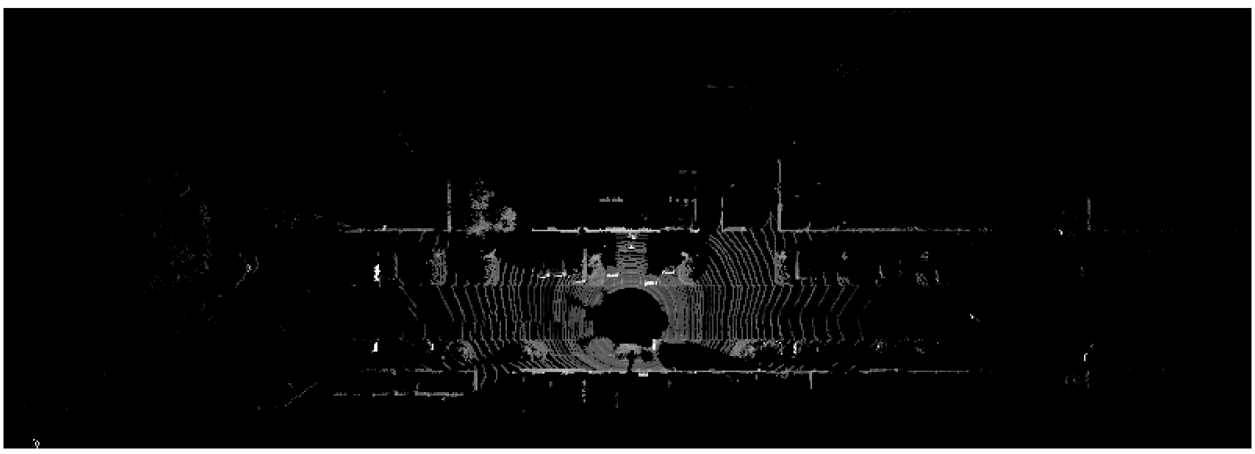

Figure 1: BEV image

The figure is rotated by 90°. The x-axis of this image is the negative y-axis in the LiDAR coordinate system and the y-axis of this image is the x-axis in the LiDAR coordinate system.

2 Problem2

To project point cloud onto the image of Cam 2, firstly we use *T_cam2_velo* to convert point cloud from velodyne coordinate system to Cam 2 coordinate system, then use *P_rect_20* to project 3D points onto image plane. For each point, we color it according to its semantic label which is stored in *sem_label*, and the color of each label is stored in *color_map*.

As for 3D bounding box, we compute the eight corners of each bounding box according to its height, width, length, the center of the bottom face and the rotation around Y-axis. Then we convert these eight corners from Cam0 to Cam2 using *P_rect_20*. Finally we draw lines between these corners.

When projecting points onto image plane, we need to filter out the 3D points whose X coordinate are smaller than 0, which means these points are behind camera. And we also need to filter out points outside the image plane.

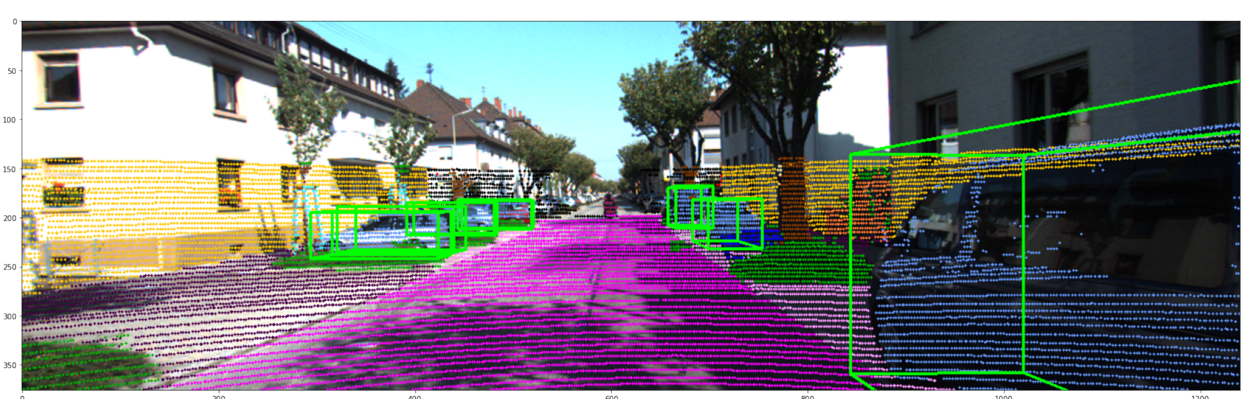

Figure 2: LiDAR point cloud with semantic label and bounding box

After modifying *3dvis.py*, we can visualize the scene in 3D. In this step, we don't need to filter out any points because point cloud is 360°. From the 3D visualization, we can find 7 cars. However, we can only identify 6 cars that lie within the camera field-of-view.

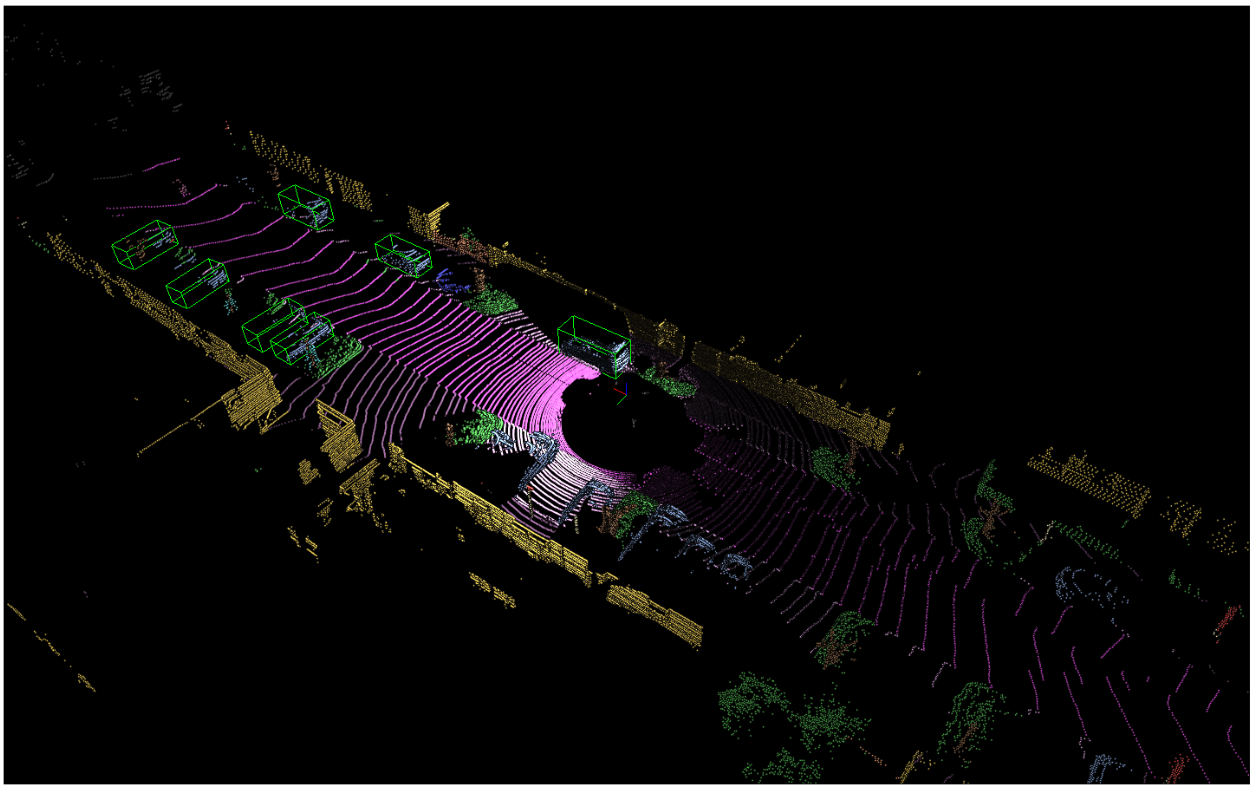

Figure 3: LiDAR point cloud visualized in 3D

3 Problem3

From the specifications about LiDAR given in handout, we know the sensor has 64 channels and the vertical field-of-view is +2.0° to -24.9°. Firstly we calculate the vertical angle θ of each point by:

$$\theta = \arctan \frac{z}{\sqrt{x^2 + y^2}}$$

We filter out points whose vertical angle are out of this range and then divide the stated FOV into 64 bins. After assigning each vertical angle of 3D point to one of 64 bins, we get the laser ID of each point. Finally we use four alternating colors to indicate the identified IDs.

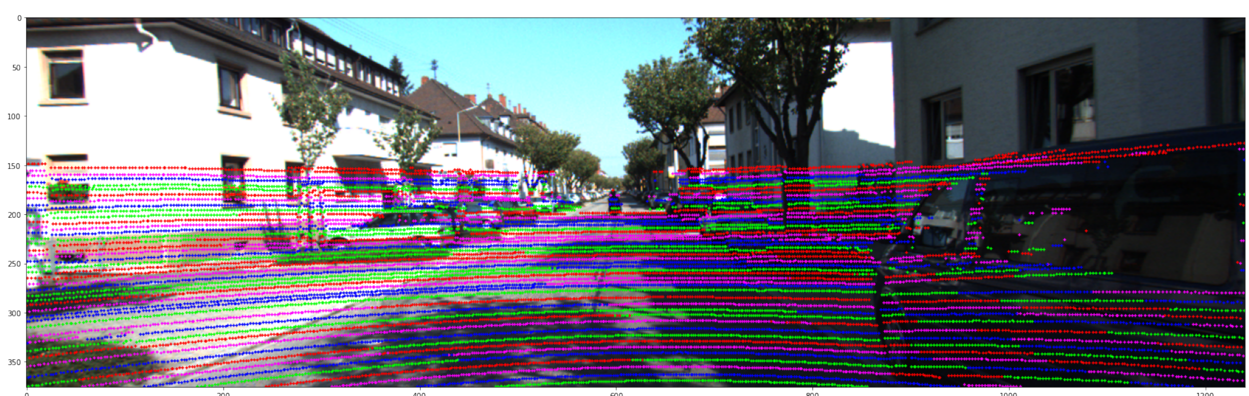

Figure 4: LiDAR point cloud with identified Laser IDs

4 Problem4

To correct the distortion, we need to transform each point into the frame of the LiDAR at the time exactly when the camera is triggered.

Firstly, we have to compute the time when each point is scanned relative to the trigger time. This is done by calculate the angle θ of each point relative to the x axis of the LiDAR frame, as y = 0 when the LiDAR is facing in the camera direction:

$$\theta = \arccos \frac{x}{\sqrt{x^2 + y^2}}$$

The sign of θ is determined by the value of the y coordinate of the point. Since the LiDAR rotates clockwisely, $\theta \in [-\pi, 0]$ if $y > 0$, and $\theta \in [0, \pi]$ if $y < 0$. Then the relative time could be calculated with

$$t = \frac{\theta}{LiDAR\_rotatation\_rate}$$

$$LiDAR\_rotatation\_rate = \frac{2\pi}{sweep\_end - sweep\_start}$$

Secondly, given the relative time, the relative transformation can be computed. The linear and angular speed can be originally obained with the IMU in the IMU frame, and can be transformed into the LiDAR frame easily. Since we assume planar movement, only angular velocity in yaw will be used. We denote the linear velocity in lidar frame with v, and angular velocity in yaw with w. The transformation can be calculated with

$$transformation = [R|T]$$

$$\alpha = w \cdot t$$

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = v \cdot t$$

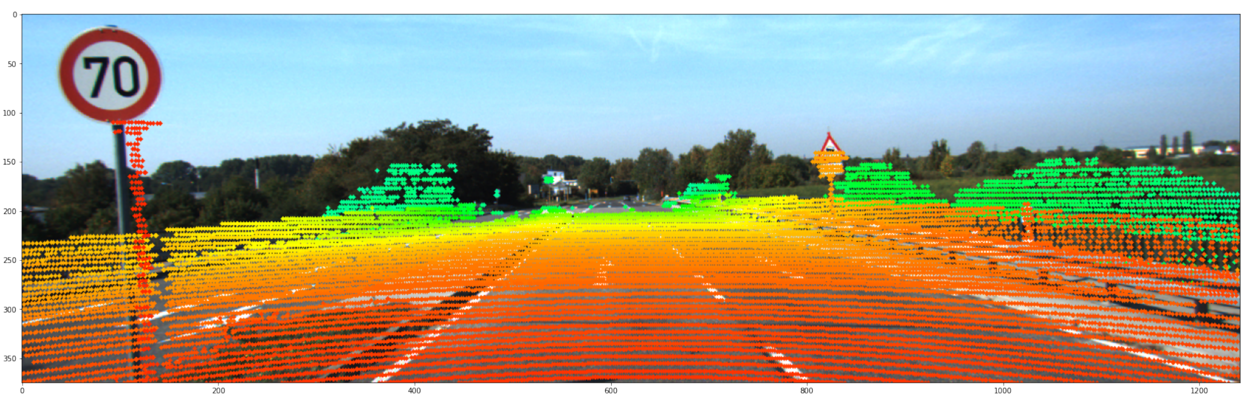

Figure 5: Direct projection of '0000000037' w/o removing motion distortion

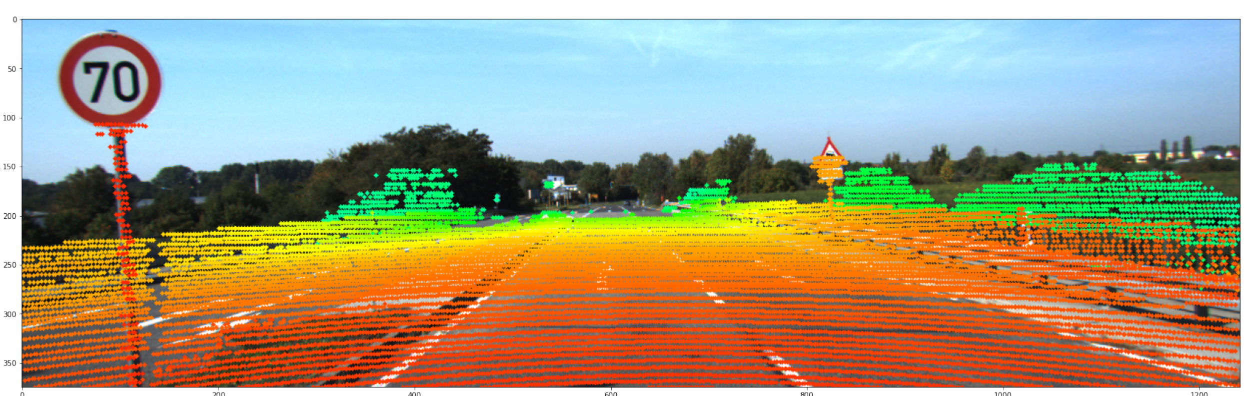

Figure 6: Projection of '0000000037' after motion distortion removed

5 Problem5

1. Because the closer the eyes to the sensor, the higher the intensify of the laser, which is more dangerous.

2. For camera, wet roads may create reflections, which could be misunderstood as real objects.

For LiDAR, wet roads reflect the laser instead of diffusing as what dry roads do. As the result, LiDAR receives back less light that is reflected back from roads.

3. Some LiDAR points are filtered, because their projections lie outside of the field-of-view of the camera. The more distant the two sensors are from each, the more LiDAR points are wasted during the projection.