Deep Learning for Autonomous Driving I Project report

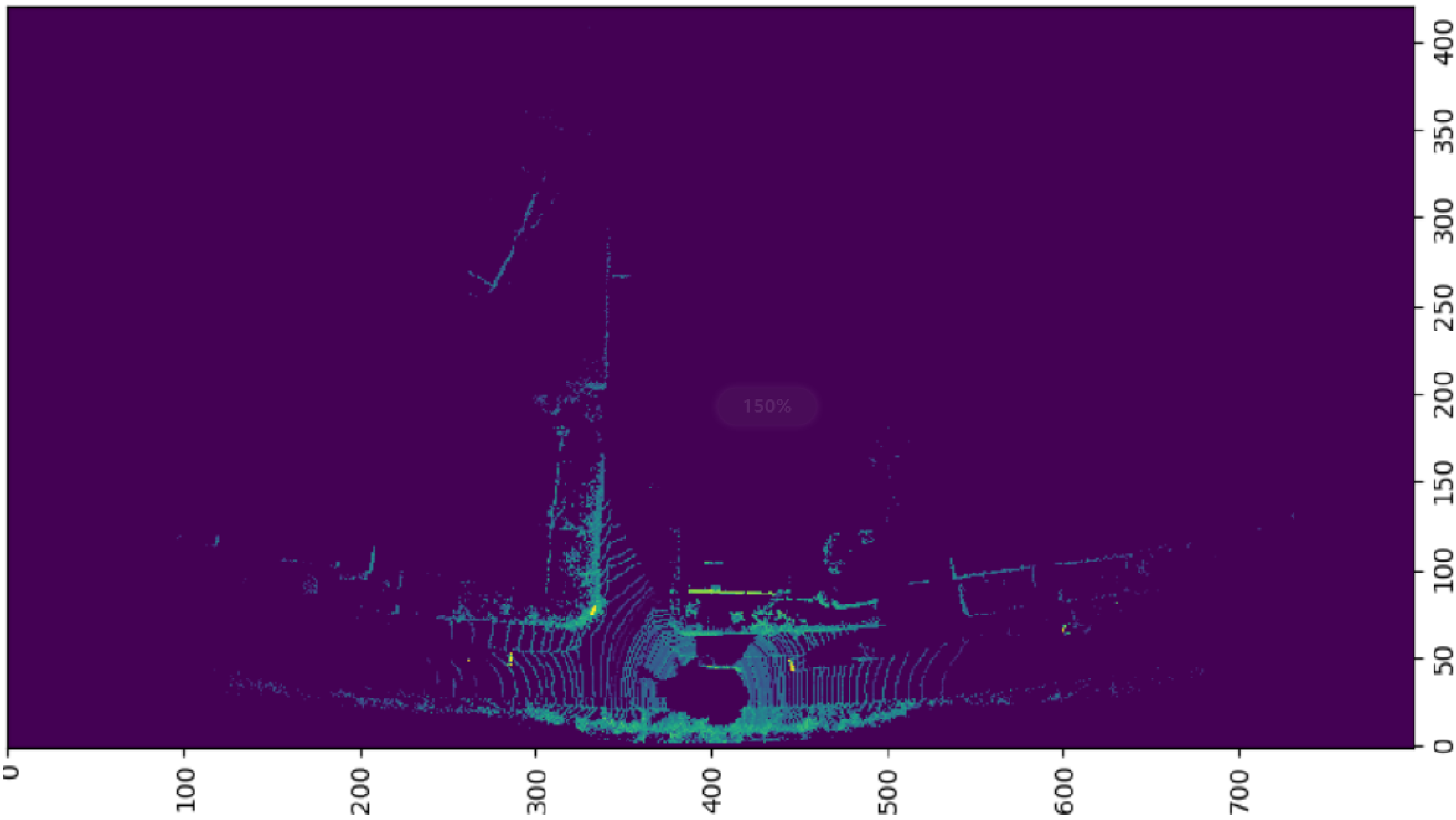
Project 01 - Understanding Multimodal Driving Data

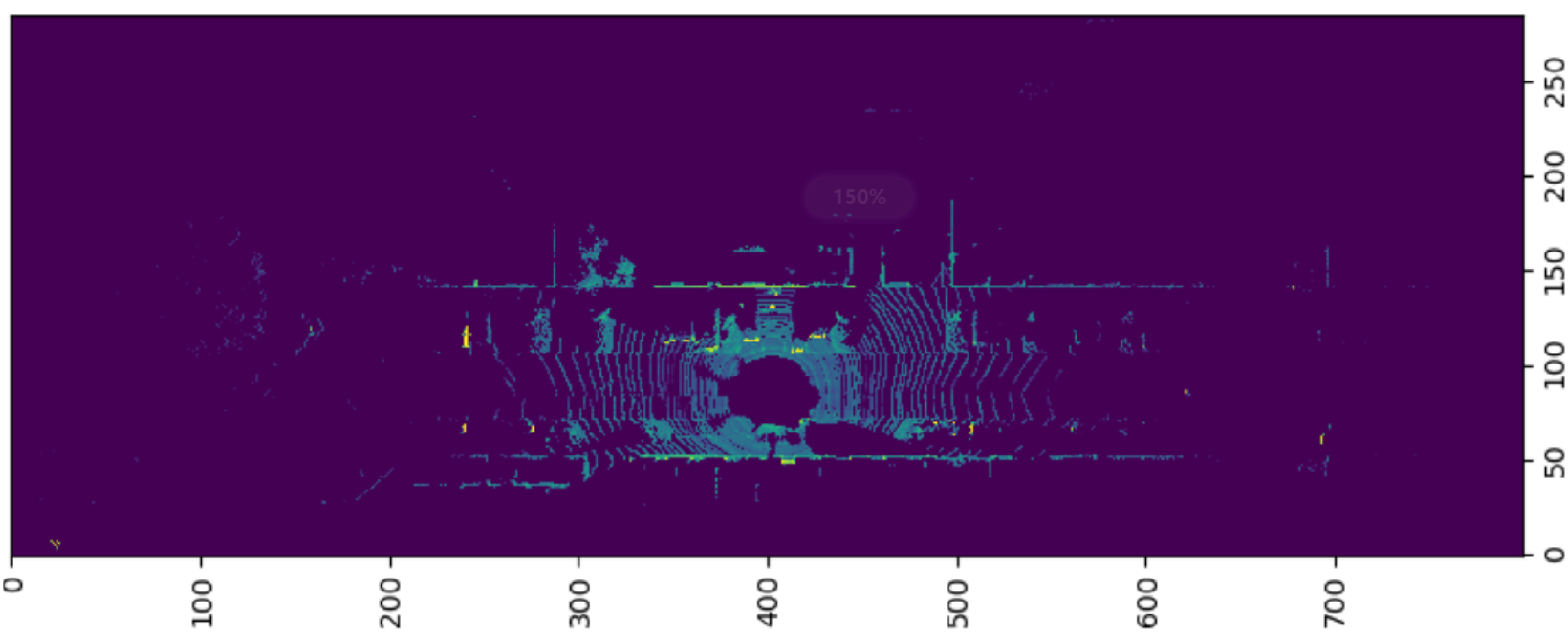
LIU Yifei I Universität Zürich

March 24, 2021

Task 1, Bird's eye view

Content: display the BEV image of the given scene with pixel intensities corresponding to the points' respective reflectance values.
Code: The code of this part can be reached either through *main.py* or *task1.py*.
Solution:





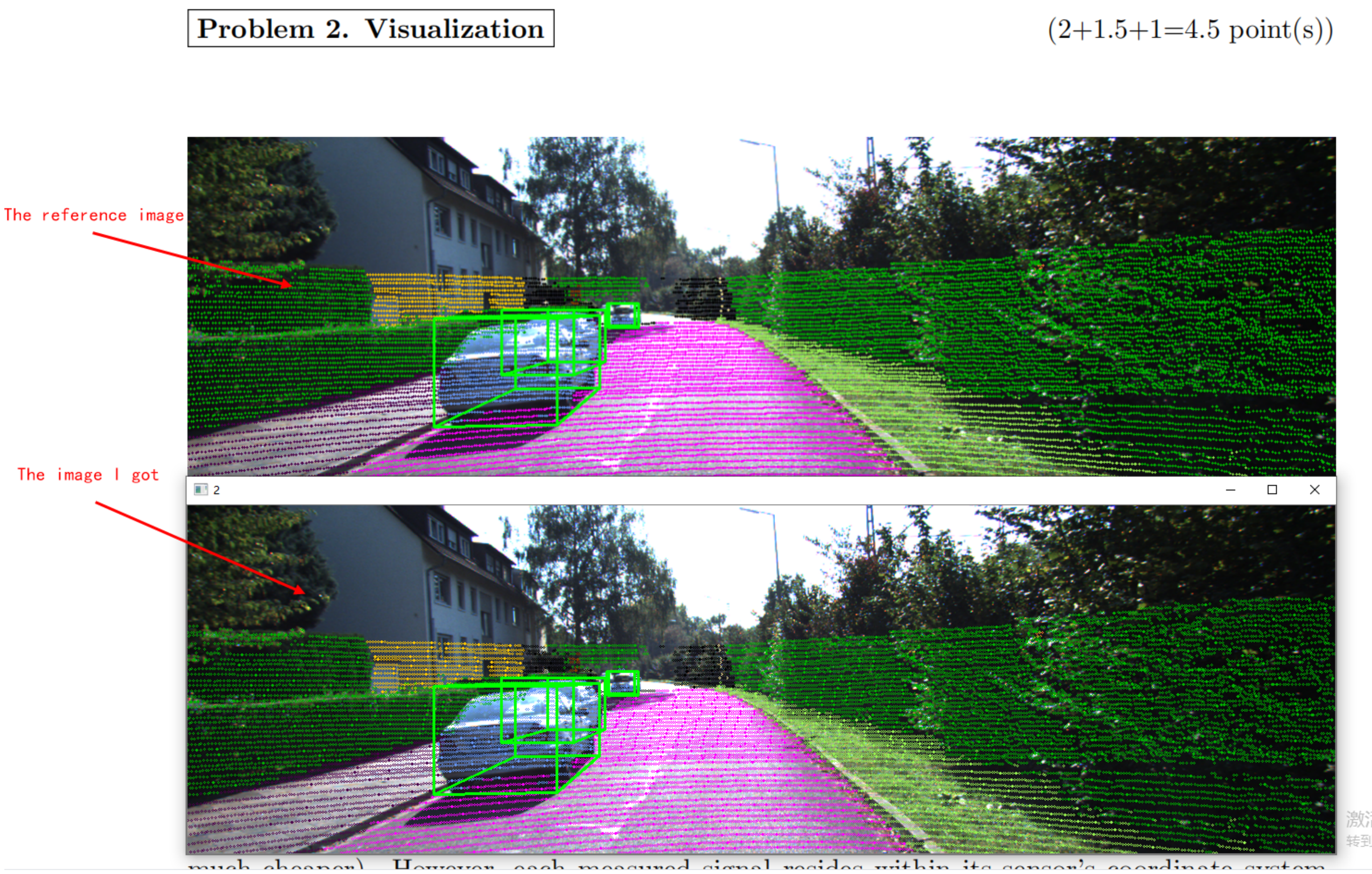
Task1. Bird's Eye View, demo(up) and data(down)

Task 2.1&2.2, 2D Visualization

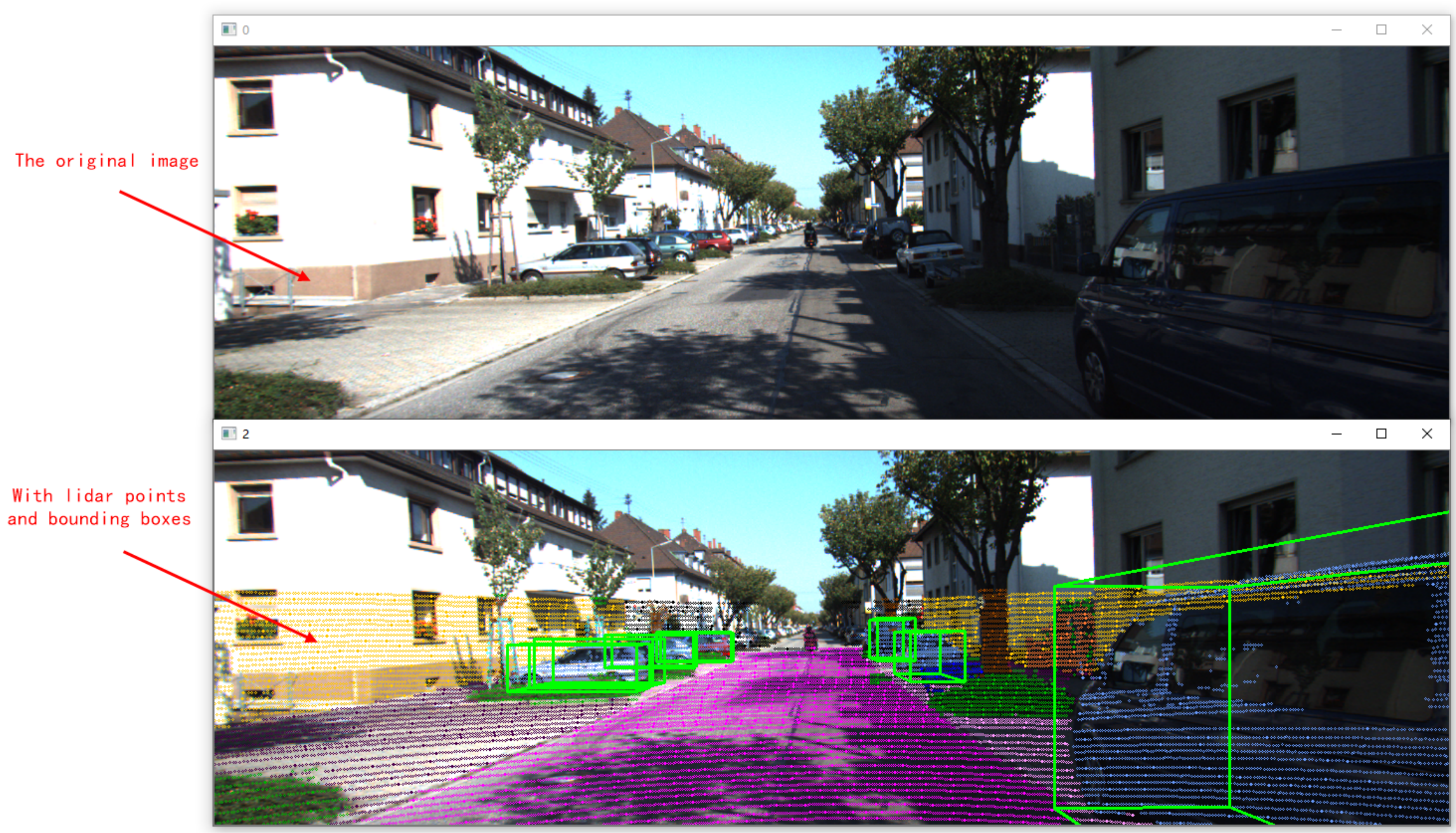
Content: project the lidar point cloud onto the image plane of camera2. Then draw the bounding boxes.
Main idea:
1.First, filter out all the lidar points whose x cooerdinate < 0
2.Second, use the perspective projection matrix to project the lidar points onto the image
3.Finally, filter out all the points that do not locate within the image size.
4.For the bounding boxes, first compute their 3d locations and then project them onto the image.

Code: The code of this part can be reached either through *main.py* or *task2.py*.

Solution:



Task 2.1&2.2, demo (Exactly the same, aren't they?)



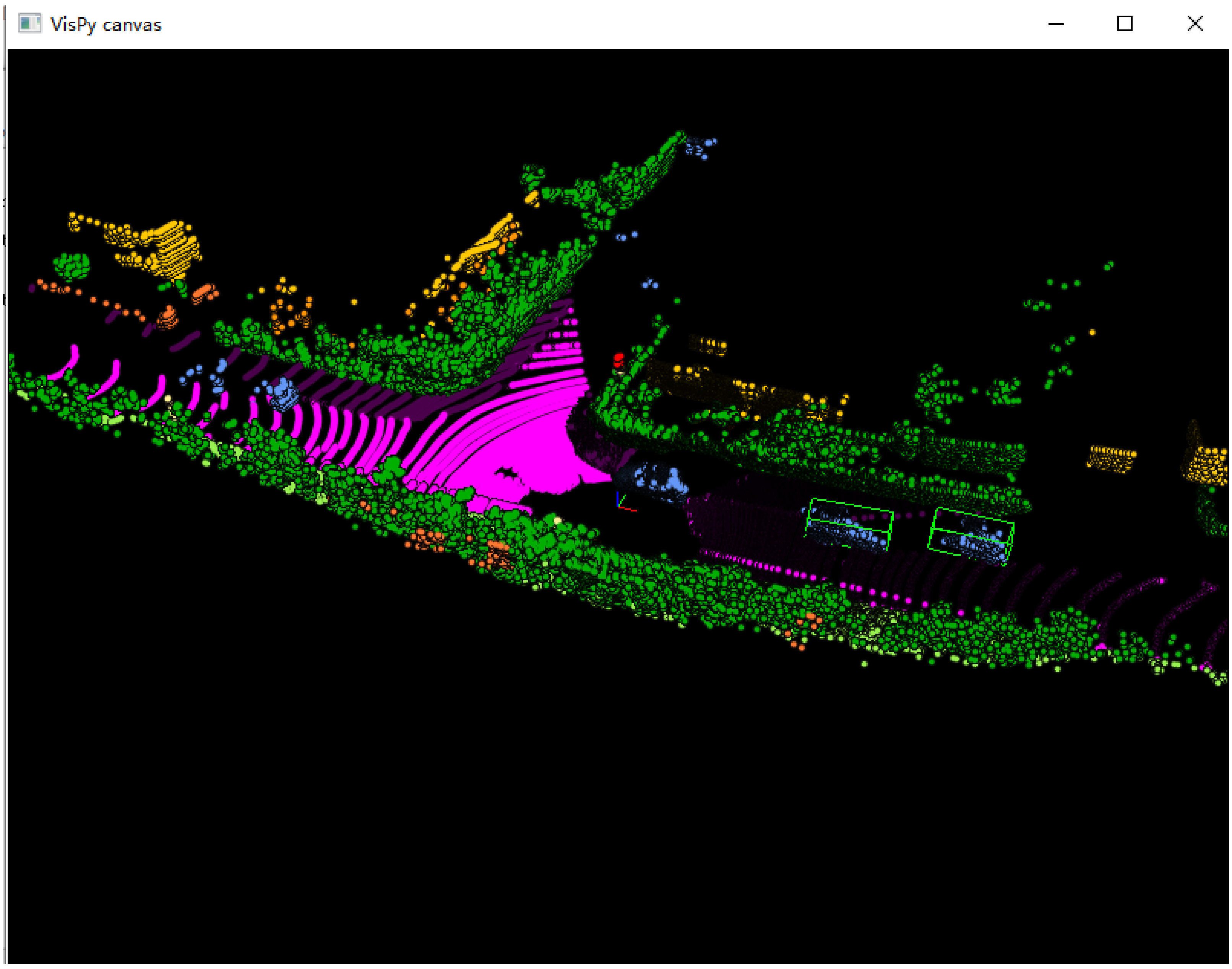
Task 2.1&2.2, data

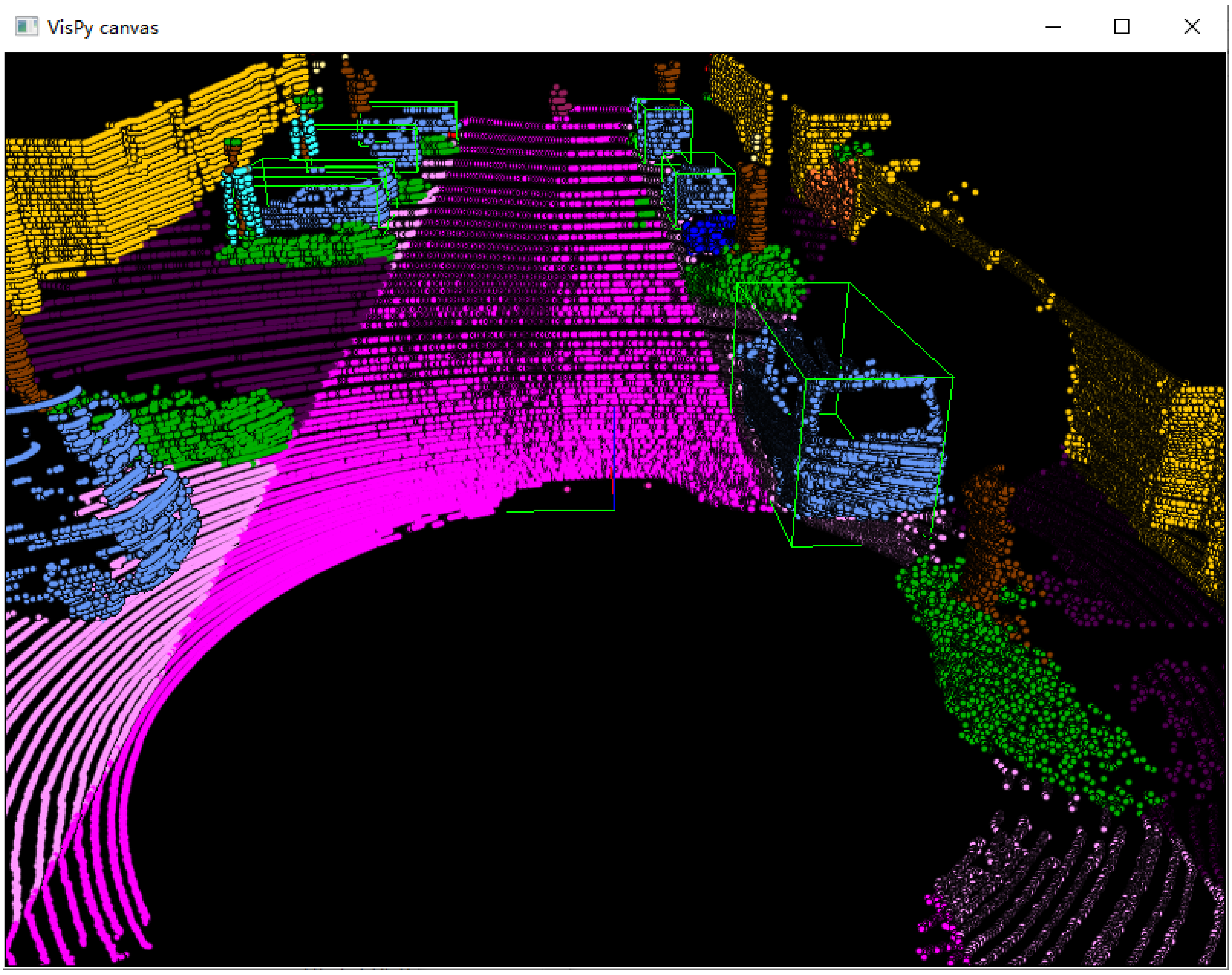
Task 2.3, 3D Visulization

Content: visualize the scene in 3D.

Code: The code of this part is only in *3dvis.py*. Run *3dvis.py* to see the results.

Solution:





Task 2.3, demo

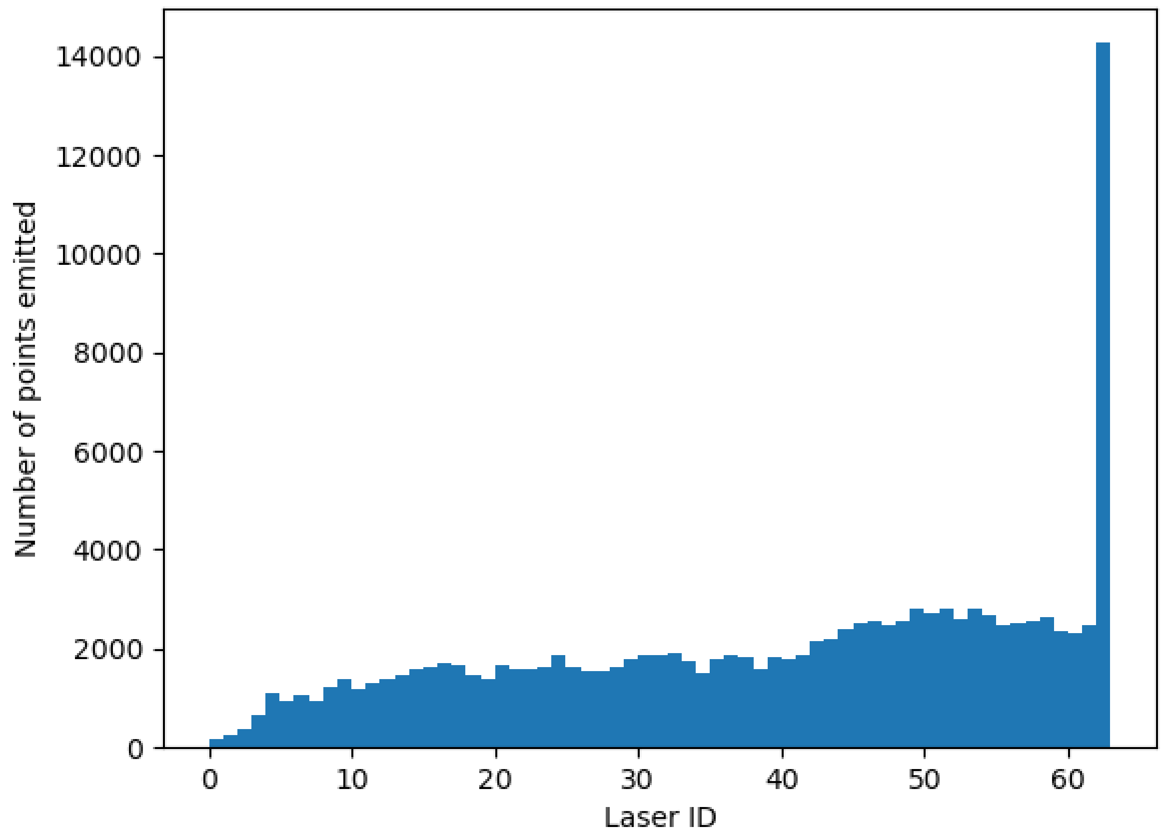
Task 2.3, data

Question: How many cars did our network fail to identify that lie within the field-of-view?

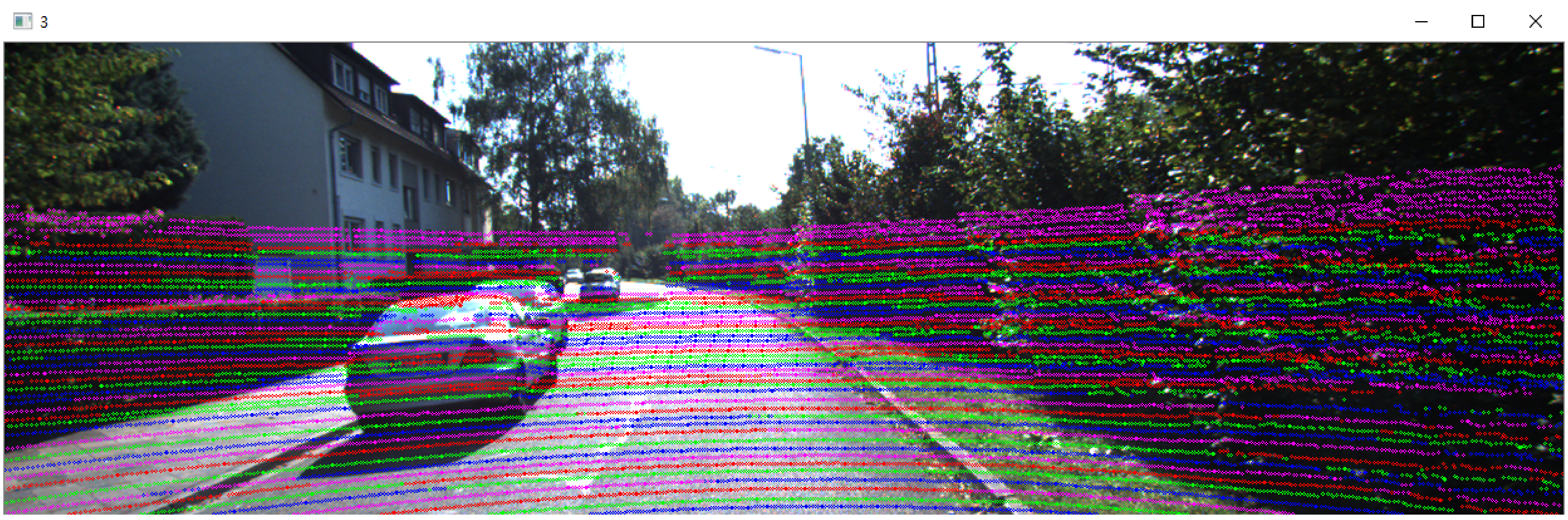
A:By interacting with the 3D data, I found all cars in the field-of-view successfully identified. But maybe only the front-most car on the right way is not identified, however, it has only very very few lidar points, so I choose ignore it.

Task 3, Laser ID

Content: identify the laser ID of each point, and use four alternating colors to show the results.
Code: The code of this part can be reached either through *main.py* or *task3.py*
Analyse: First I reckon the vertical angle to be +2 to -24.9 degree and acoordingly compute out that if there are 64 lidars, then one lidar should cover 0.42 degree. Then I got the following:



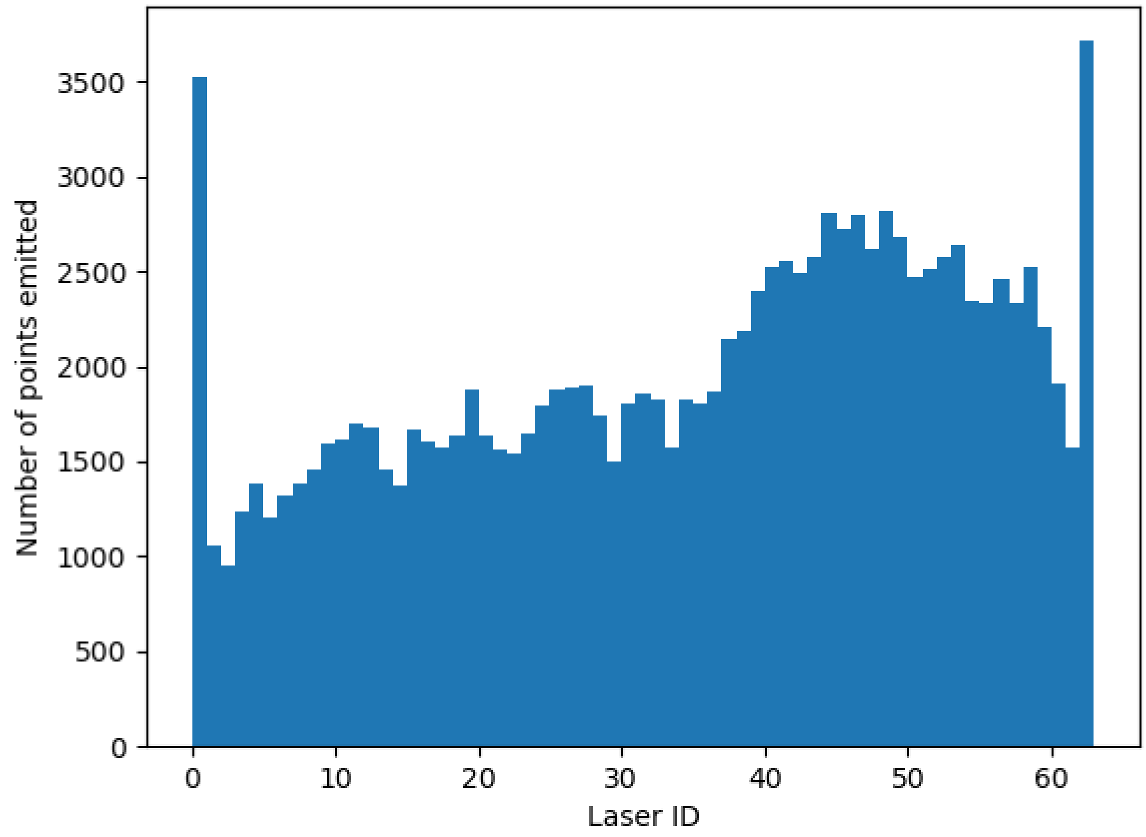
Task3, the 64th lidar shoots the most points



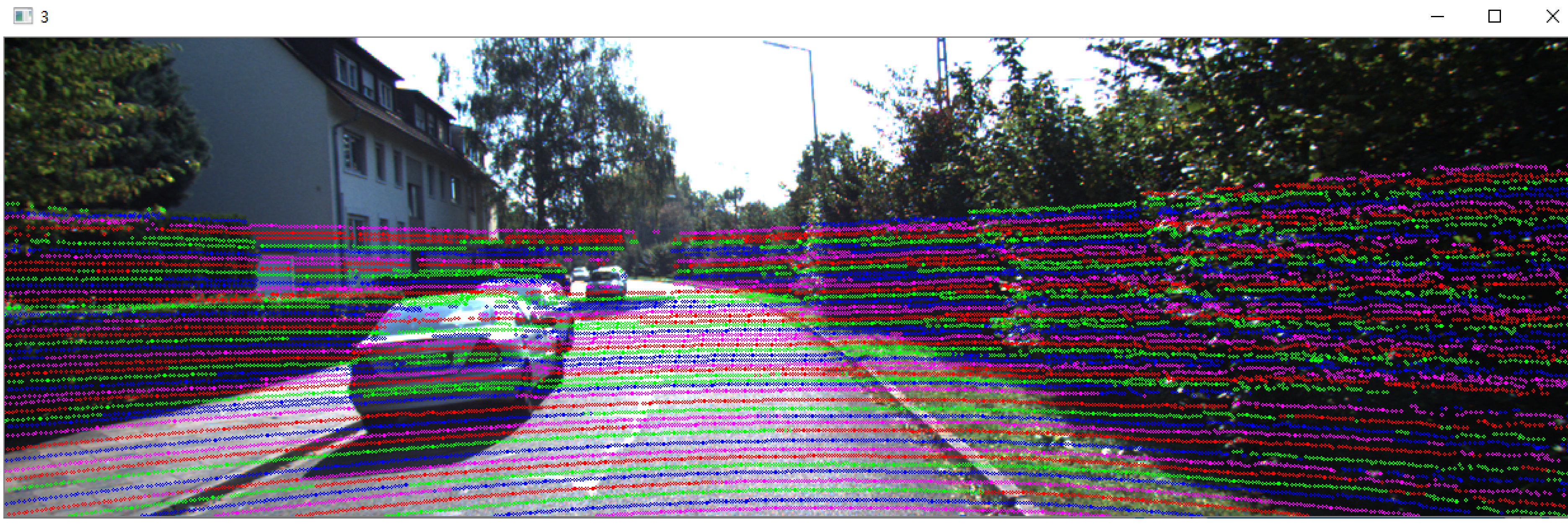
Task3, wrong laser ID identification

Then I introduced a *offset* variable to change the number of emitted points such that the 0th laser shoots roughly the same number of points as the 64th laser.

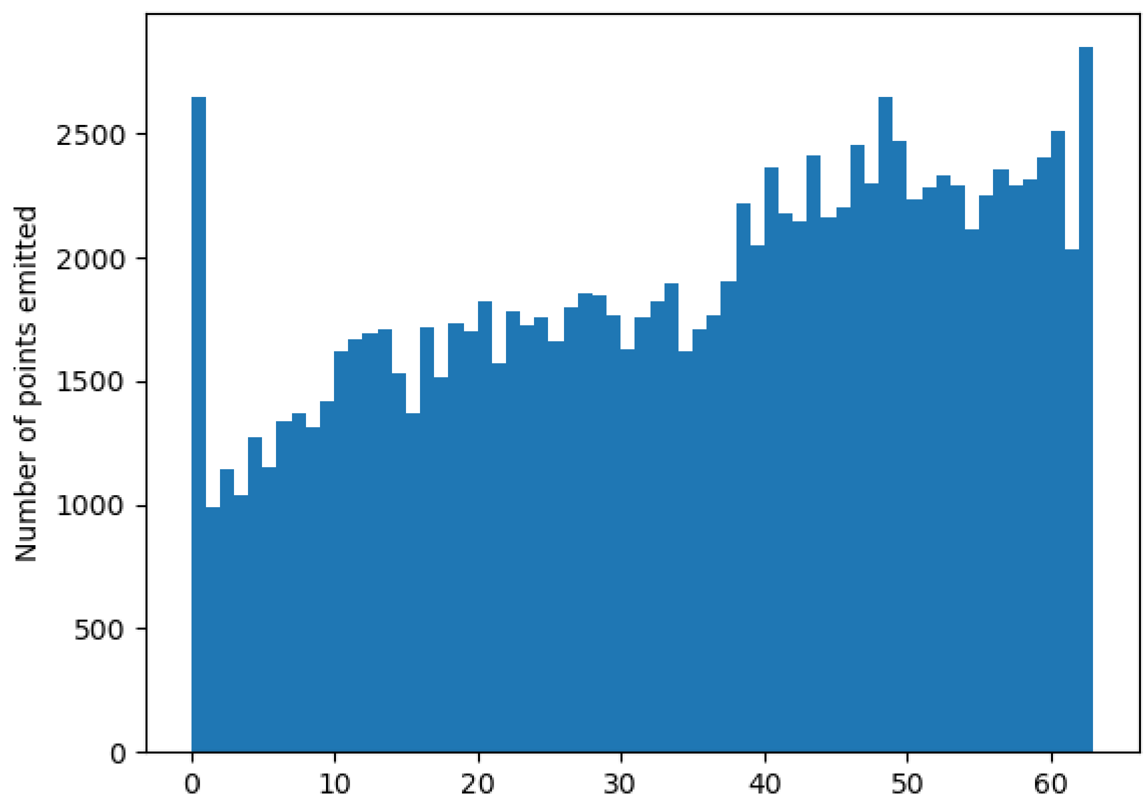
Solution:



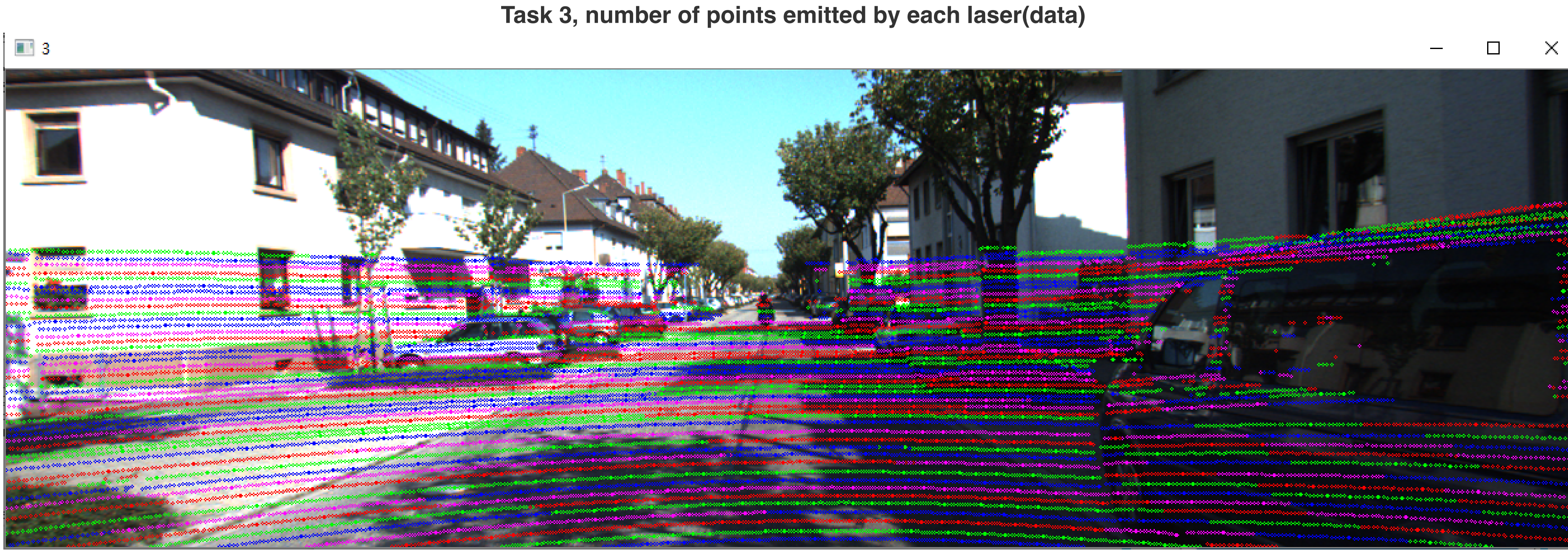
Task 3, number of points emitted by each laser(demo)



Task 3, demo



Task 3, number of points emitted by each laser(data)

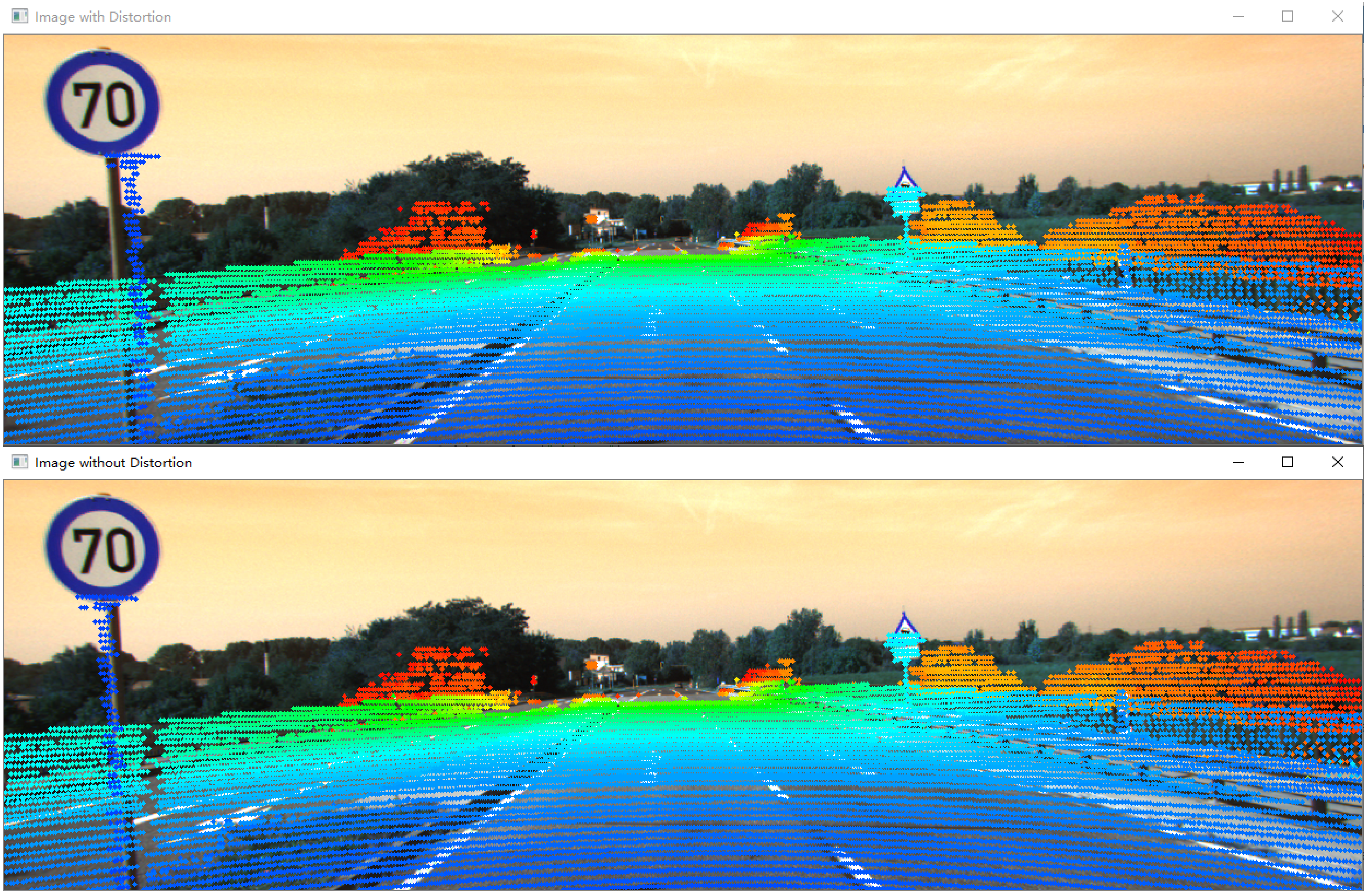


Task 3, data

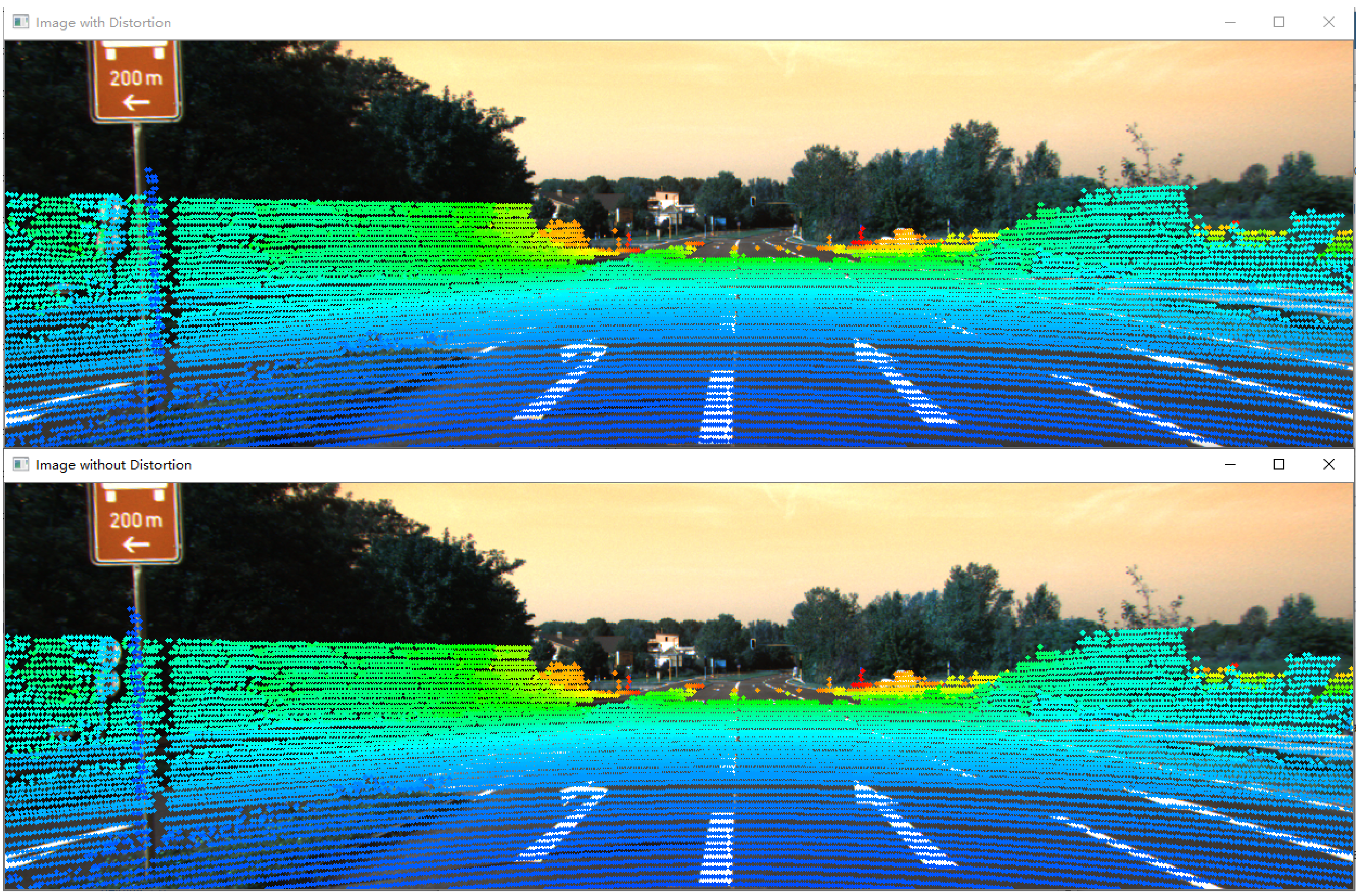
Explanation: Why do the 0th and 64th laser shoot more points than other lasers? I think this is because when the scanner is at the 0th and 64th laser, it needs to turn around and go into the opposite direction, so it has to stay there longer and consequently emitting more points.

Task 4, Remove Motion Distortion

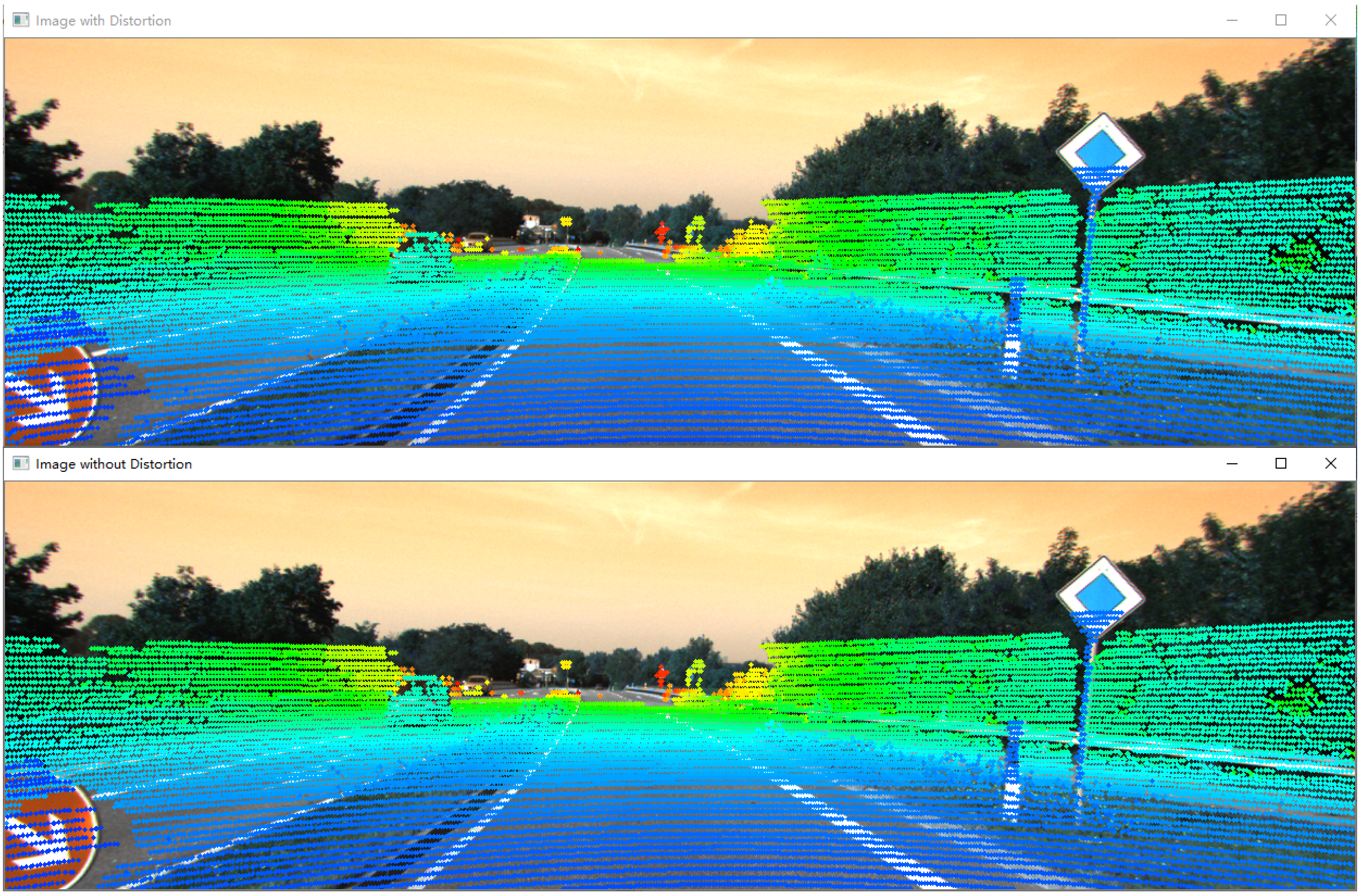
Content: correct the motion disttortion and show the lidar points
Code: The code of this part can be reached either through *main.py* or *task4.py*
Solution:



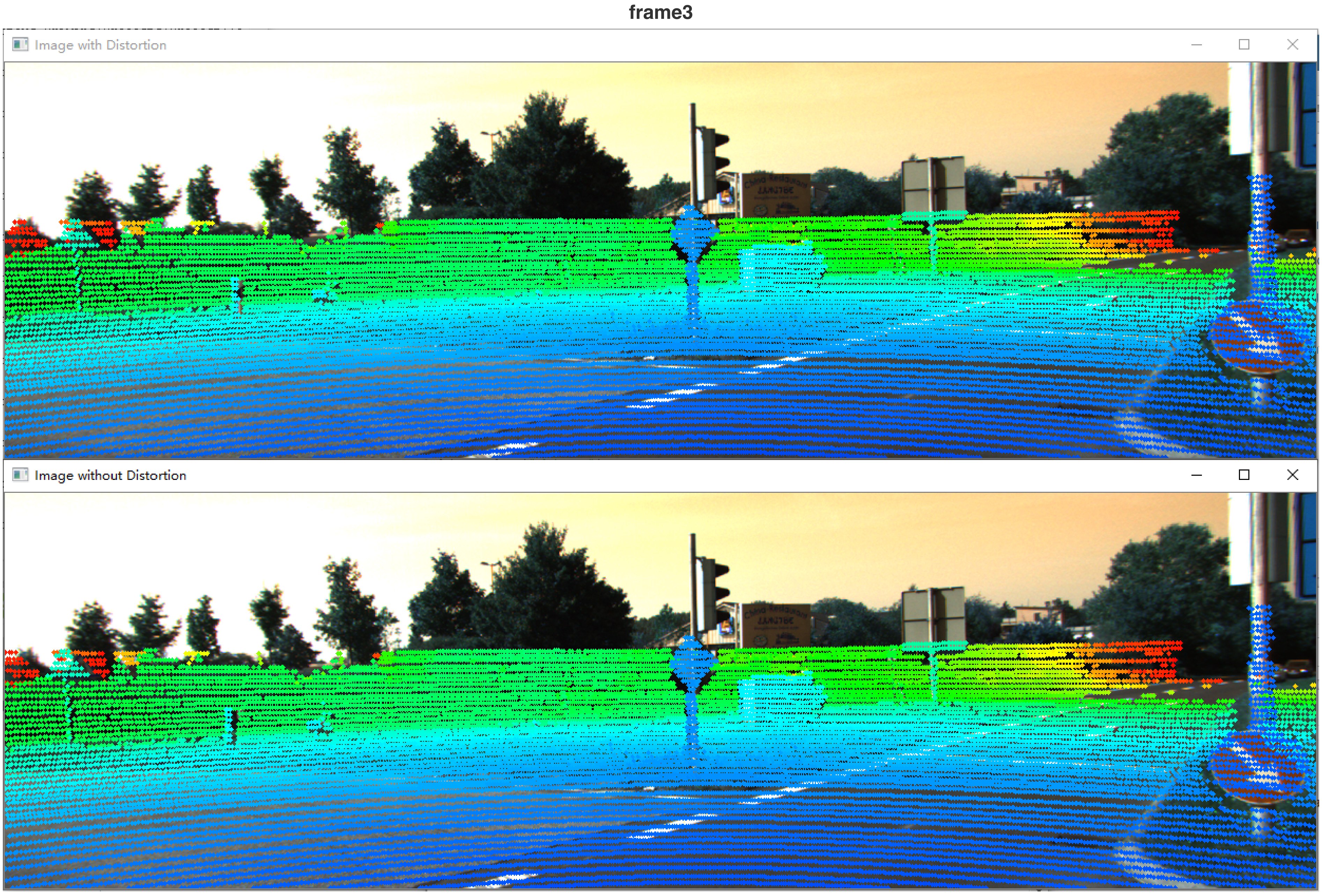
frame37



frame77



frame3



frame312

By experiments, lidar time suits objects on the left side, and camera time suits objects on the right side. My final decision is 0.35x lidar_ time + 0.65x camera_time, and it corrects the distortion for most of the situations.

Task 5, Questions

1. Eye safety: It is not safe for the human eye to look at a spinning LiDAR when it is too close. Why is the risk higher when we are closer to the sensor?
   Assuming the lidar shoots lasers in a constant field of view, when the eye is closer to the lidar, the more lasers it will receive, thus it will take more damage.

2. Wet roads pose challenges for both cameras and LiDAR. What are these challenges and why?
   1.The rain may drop on the camera lens and the LiDAR and directly obscure the image.
   2.Wet roads may have the mirror effect, so the laser may be reflected off and the receiver can receive less laser, so the range is smaller, and some part of the laser may be missing.
   3.The wet roads can reflect lights and produce mirrored images, which makes it difficult for recognition system to recognize the object's true postion.

3. In this exercise, you have projected LiDAR points onto images. In the setup in Fig.1, the LiDAR sensor and the cameras are non-cocentered – it can never be exactly cocentered. What problem this may cause for the data projection between the two sensors (LiDAR and Cam2 for instance)? Do you think this problem will be more severe or less severe when the two sensors are more distant from each other?
   The different coordinate systems have different origins, this means in order to convert the coordinates from one system to another system, we need ont only rotation but also translation. Basically, the translation and rotation matrix are measured somehow, but there would be error in it. I would expect the problem to be less severe when the two sensors are more distant from each other, because the error would be relatively smaller compared with the distance. In extremely exaggerated situations, like if the distance is 1km, the measurement error can be totally ignored.