**Project Report**

**Autonomous Driving**

**Tianshu Zhu, Haoran Fang**

Introduction:

Our problems is to use neural network to detect road and vehicles in the real life situation. As the development of machine learning through these years, autonomous driving is no longer a unpractical idea. Some big companies like Uber, Google, Apple are working on developing their own auto-driving system. The core tech of autonomous driving is road detection and car detection. Through CSC420, we learned how to detect features and find depth. In order to use Stereo to calculate the depth, we read the paper “ Efficient joint segmentation, occlusion labeling, stereo and flow estimation” . Also we reference another article “ Feature extraction for Vehicle Detection using HOG+” to find a efficient way to extract feature for training.

Methods:

Q2(b) is about to calculate the disparity of stereo images. In this part, we the reuse the code “spsstereo” from TTIC Chicago to compute the disparity between left and right image.

Then we reuse the code from assignment 4 to calculate the depth of each pixel.

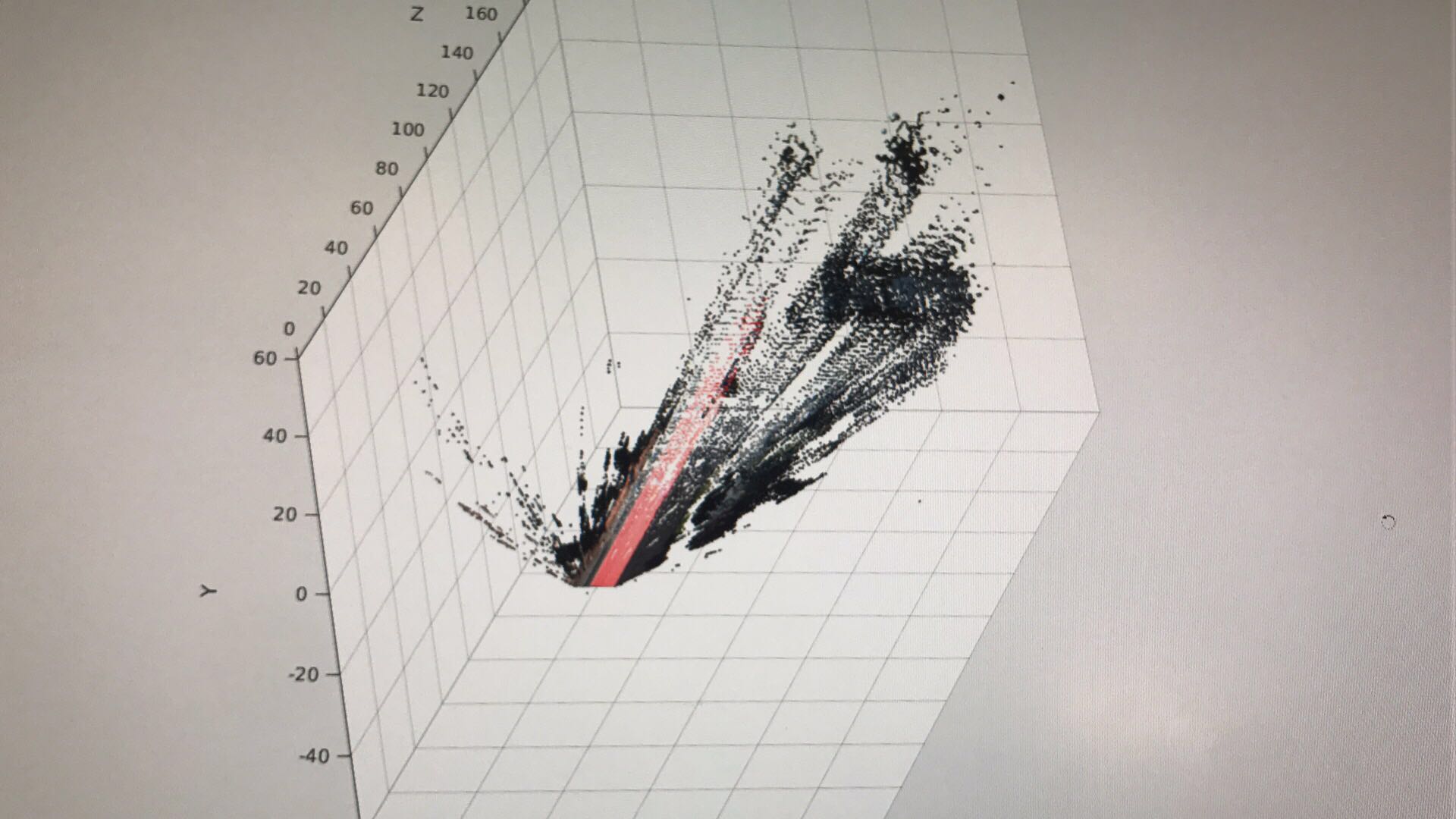
 In the data set provided by KITTI, the information of Camera has given out. Camera calibration will be used to calculate the depth along with the disparity.

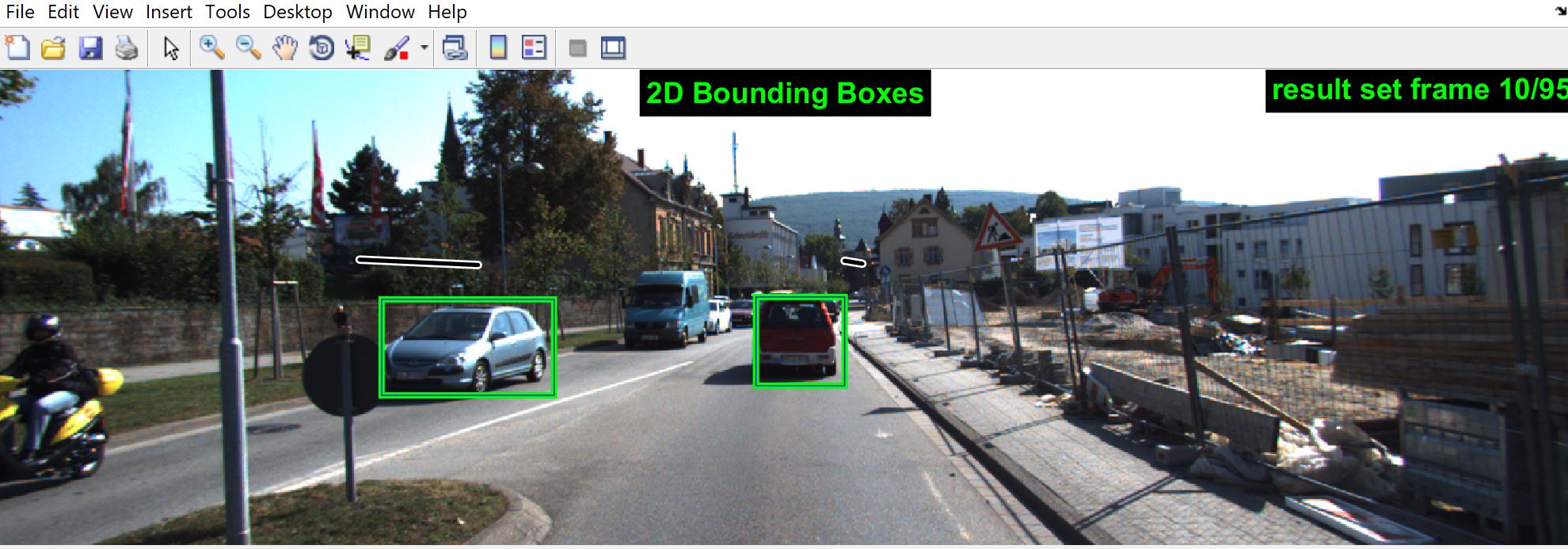
Q1 (f) by using the detected road picture, the road can be detected by using filter. A threshold is set for detect road. On the detected image, we marked the road as 255 and the rest as 0. So, we can extract the road part from the image. pointCloud() is used to plot the road in 3d space. Since we already have the each pixels’ 3d location, its easy to create a 3d plan of the detected road.

Q1(g)Here we use the threshold to find the road on the original pic. When found a pixel has non-zero value on the detected image, we mark the pixel on the corresponding location of original image. So at the original image we will have a marked road appear on the graph. Use pointCloud(), we can plot the original image with road marks to the 3d place

Q2(d) In this part, we download the development kit form KITTI. By using the information of lables, we can get information about the 2d bounded box’s location. Then we write line on the graph to form a rectangle box around each detected car. Since we store the rotation-y in each object, we can use the rotation-y to calculate the orientation of the car. Finally we project this orientation into the image plane.

Q3 We treated each image as one frame of film and set the framerate as 4 to make a film of detected road

Results: 



Yamaguchi, Koichiro, David McAllester, and Raquel Urtasun. ”Efficient joint segmentation, occlusion labeling, stereo and flow estimation.” European Conference on Computer Vision. Springer International Publishing, 2014 Paper and code.

Mohan Karthik. “Feature extraction for Vehicle Detection using HOG+”.https://medium.com/@mohankarthik/feature-extraction-for-vehicle-detection-using-hog-d99354a84d10