Project Proposal:
Sample #4 Autonomous Driving

Team members:
Murali Komaraolu, 1000564490
Hongman Cho, 1000216250
Vibhavi Peiris, 1000597687

The overall goal of the project is to understand images of the roads given the points from two cameras. This is an important problem because it paves the way (pun intended) for fully automated cars to navigate the road safely. Both parts, road detection and object detection, are required for the cameras to successfully map the road.

How are we going to tackle the project?

PART 1:

We will start by downloading the left & right color images, and calibration & training labels from the following websites

- http://www.cvlibs.net/datasets/kitti/eval_road.php
- http://www.cvlibs.net/datasets/kitti/

To computer disparity between two images, we will use MATLAB's api 'disparity()' function

- https://www.mathworks.com/help/vision/ref/disparity.html?requestedDomain=www.mathworks.com

We will refer to the parallel stereo camera technique from lecture 12 (slides from 35) for calculating depth of each pixel. First we assume a baseline is given (x distance between the left and right cameras). Also we have calculated x positions of left and right image points. Then we can calculate the depth of the pixel using similar triangles equation. For the features, we will find the interest points in 2d and 3d to match from the two images. Then for the training, we will most likely use a convolutional network from

- https://arxiv.org/pdf/1608.07711.pdf.

Now to use the depth of the pixels to fit a plane on the road, first we need to filter out the pixels only for road. Then we need to write an algorithm that can compute a plane (possibly using RANSAC in lecture 8ii) that can be projected to fit the road, using the depth of each pixel that we found in c).

- http://web.mit.edu/mobility/publications/Iagnemma_IVSS_10.pdf

Finally, we will use MATLAB's API 'showPointCloud()' to plot computed 3D pixels of the road.

PART 2:

We will be downloading the images, training data and camera calibration matrices from this site.

- http://www.cvlibs.net/datasets/kitti/eval_object.php

To detect cars in the image we will use the pre trained model recommended.

- kitti.is.tue.mpg.de/kitti/models_lsvm.zip

Then, we will use a supervised learning algorithm to train the data(split training data into training and validation DO NOT use Test set) so that it will be able to recognize the car from many different angles. We will use precision, recall, and other error measurements to make sure the algorithm is working correctly. The following websites might be useful for this part.

- https://www.mathworks.com/help/stats/classificationlearner-app.html
- http://www.cvlibs.net/publications/Fritsch2013ITSC.pdf
- http://www.cvlibs.net/publications/Menze2015CVPR.pdf

Once we have the model from the train data in the step above we can use it to detect the car and its viewpoints. Using that, we can use the angle & coordinates for the car returned above and draw an arrow and box around it. Lecture 12 can help with that. Once we have the box and the ground plane we can follow what is described in the paper below using RANSAC method to create a 3d box for it.

- Page 6,7 of this paper http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.145.3574&rep=rep1&type=pdf

PART 3:

Use the recommend library to record our algorithm running on the test set.

How is it going to be split up?

We have three members in our team with two parts in the project. Since both parts are about detection, we can split up each part into three and do it that way. So each person will work on a third from each part. But some of the individual sections from each part are dependent on previous sections. In that case, we will use test parameters to work with. i.e. for the training part instead of waiting for the disparity and depth part to be finished. We get examples from google and use it to work on the training part of the project.

Murali will find disparity and depth of the pixels from part 1 and detect the cars from part 2
Hongman will write the classifiers for both parts
Vibhavi will create the 3d plane and point cloud from part 1 and 3d bounding boxes from part 2