Lane Finding Report @Udacity

Venkata Sai Kumar Gadde

June 2017

1 Introduction

Finding Lane lines is one of the primary tasks needed for a self driving car. This project has introduced me to the basics of image processing and processing videos.

2 Pipeline for the Project

The pipeline involves following steps from the raw input color image to the final image with lane boundaries in red color.

- Convertion of the color image to the gray scale image.
- Edge detection using Canny edge detector. Hyperparamters used are:
  - Threshold1: 50
  - Threshold2: 150
- Hough line Transform to identify the lines in the images.
- Overlapping of the image with lines and the original image.

3 ShortComings

There are shortcomings in this pipeline.

- All the cases used in the project doesn't have any kind of curve roads except for the optional case. As the lines need not be straight all the time in the journey, the basic approach of drawing line using the cv2 tools cannot be used in the final production systems.
- All the cases given in the project have lines on the both sides. But in real life scenarios, there may be no lines on the road. The car has to identify even in those kind of scenarios in which this pipeline doesn't work.
- When there is an obstruction in front like vehicles in front on the road, this algorithm doesn't yield any kind of results. So, the car can be unsure whether the road is straight or curvy ahead.

4 Possible Improvements

Improvements can be done to solve the problems specified above in the section 3.

- Curvy roads have to be identified. Small lines can be identified locally and a spline can be used to draw the curvy roads.
- When there are no lane lines, the pipeline should estimate the width of the road, and then should infer its half of the road.
- When there is an obstruction in front of the vehicle, many possible cases arise.
  - **Only single line can be seen :** We can estimate the other line by drawing a parallel line. May need more computation.
  - **Both lines cannot be seen :** Can estimate the line using the google maps. But this can very computationally expensive, as internet is needed to load data from the api. This is not practically feasible, as this approach doesn't work when the car is not connected to the internet in the remote areas. A second approach can be obtaining data from the car in the front. This approach can be used to construct the road ahead. In this approach, all the cars need to be connected. A third approach can be to take the boundaries of the vehicle ahead and estimate the lane lines.

- The existing pipeline can act unpredictably when it rains. So, possible improvement is required to make the pipeline robust to different environment conditions.
- Convolutional Neural Networks can be used to solve a few of the problems, as the convolutional neural networks retrieve the lines and other features from the images. A CNN can be used to identify the lines irrespective of the color, light intensity, and other obstructions.

- Better optimization techniques can be used to decrease the number of repetitive computations from frame to frame in a video (In production system). As the frames per second can be approximately 30 to 40, only the change from the previous frame can be identified and the repetitive computations can be eliminated.