Finding Lane Lines on the Road

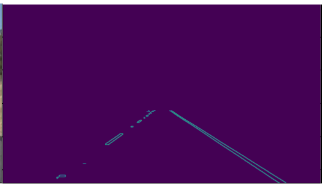
The goals / steps of this project are the following:

- Make pipeline that finds lane lines on the road
- Reflect on the work in a written report

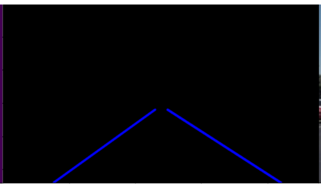
Reflection

The pipeline included 5 steps. First I converted the images into grayscale after which canny transformation was applied to the grayscale images. Once I got the canny transformed image i extracted the region of interest from the image, which in my project was a polygon with 4 predefined vertices. To fine the lane lines from the canny edges I applied hough transform which returned the lines in the image. In order to draw a single line on the left and right lanes, I first segregated the (x1 points of the hough lines) to either left or right based on their position from the midpoint of the x-axis. Now that I had pairs of points on either left side or right side, I applied the polyfit() function from the numpy library to find the best fitting line for the left and right side. With 2 equations for left and right lane line, I found 4 points (xi,yi) where yi were the bottom and top of the polygon. Finally the lanelines were plotted back onto the original image.

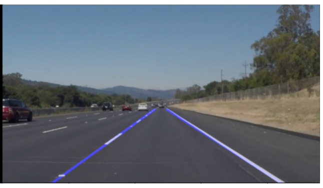
Original Image



Canny transform



draw_lines()



Final Image

Potential shortcomings of the current pipeline

- The current model dosent deal with turns in the lane lines properly
- Other cars in the region of interest can disturb the canny transform and in such cases the lane lines won’t be identified correctly
- Markings on the road such as zebra crossings can cause errors

Possible improvements to the pipeline

- Instead of using a single large polygon for region of interest , we can use multiple smaller polygons which can effectively handle turns and road markings.