Finding Lane Lines on the Road

Finding Lane Lines on the Road

The goals / steps of this project are the following:

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

Reflection

1. Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.

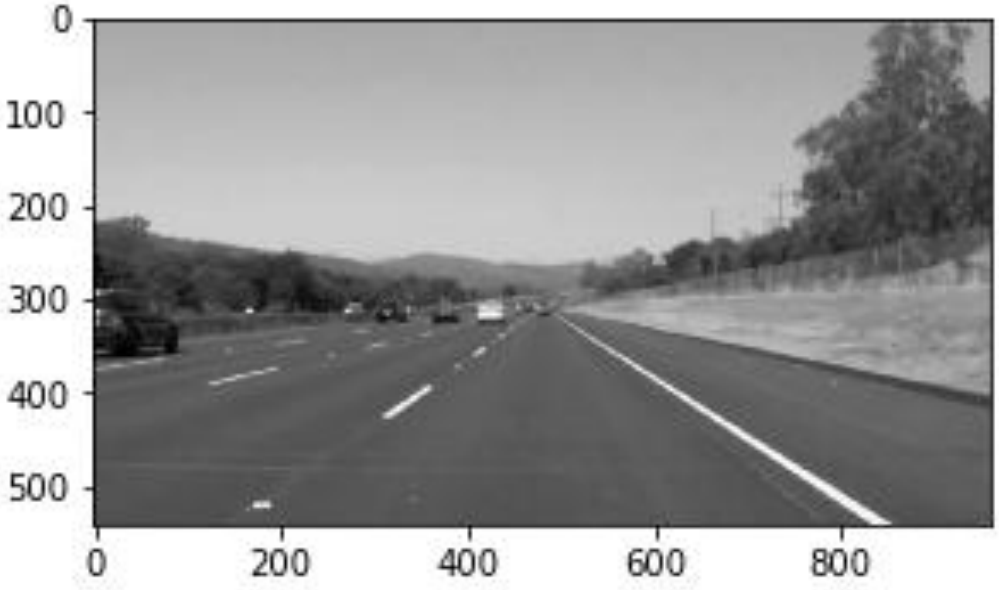
My pipeline consisted of the following steps:

1) Convert the images to grayscale.
2) Blur the image slightly to remove imperfections.
3) Do a canny edge detection to find the lines.
4) Do a dilation of the image to remove imperfections from the canny edge detection algorithm.
5) Do a region of interest selection where lanes should be found to limit the search region and avoid unnecessary line detection.
6) Do the hough transform to find lane lines in the left and right roi.

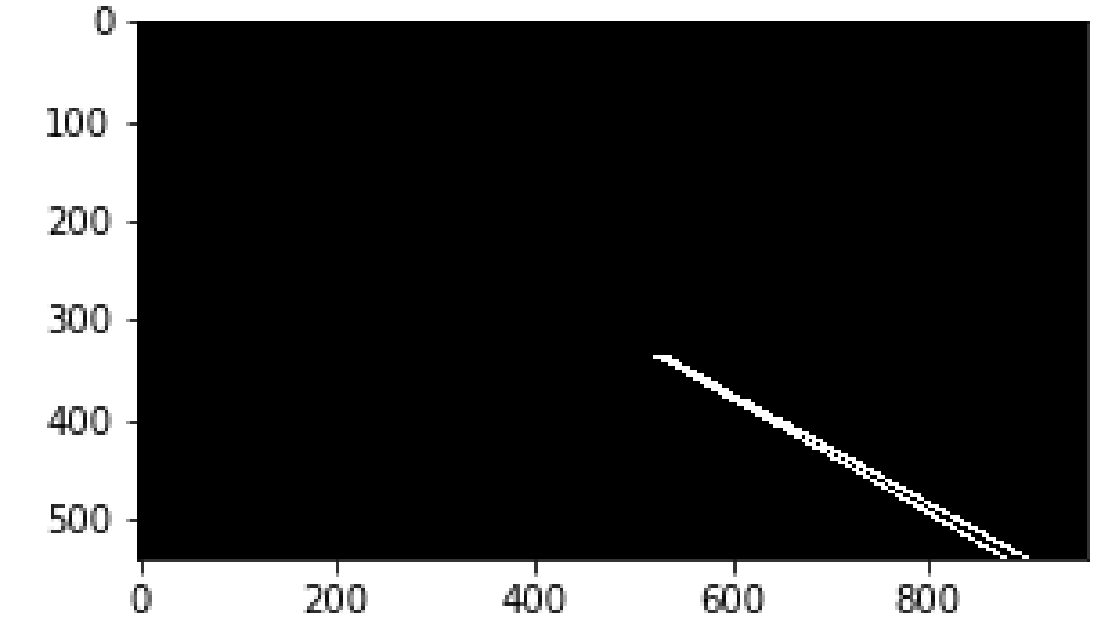
In order to draw a single line on the left and right lanes, I modified the draw_lines() function by finding the slope and by finding the minimum and maximum end points for the left and right line segments using a least squares solution.

The following images to show how the sequence of how the pipeline works:

The original image



The grayed image



Gray image with the right ROI, canny and Hough applied

The final result

2. Identify potential shortcomings with your current pipeline

Short comings of the current algorithm include lighting changes either by objects casting shadows or variations in the road surface itself. A tracking and filter algorithm will help smooth tracking performance from deviation from a straight line. Reading of a yaw sensor would improve tracking in curves.

3. Suggest possible improvements to your pipeline

Improvements would be to a filter to the output of the hough transform and to provide a tracking algorithm to smooth line tracking.

Another potential improvement could be from a yaw sensor but this would have disadvantage in being a lagging input and additional cost of a sensor.

https://github.com/raweaver00/CarND-LaneLines-P1-master