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.

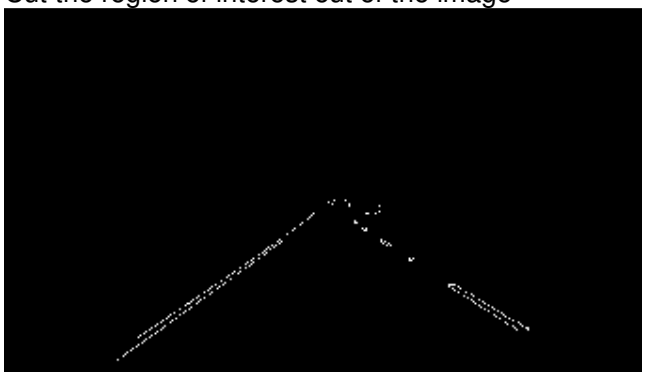
My pipeline consisted of the following steps:

1. Convert the images to grayscale

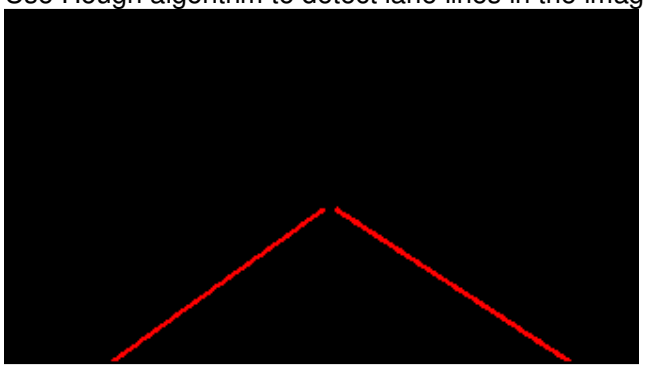
2. Apply Gaussian blur to smooth out noises in the image

3. Use Canny edge detection to detect edges

4. Cut the region of interest out of the image



5. Use Hough algorithm to detect lane lines in the image



6. Finally, combine these lines into the original image to get the final image showing lane lines

In order to draw a single line on the left and right lanes, I modified the draw_lines() function as follows:

1. Find slope of the lines
2. Based on the slope, determine if it belongs to left or right lane
3. Find out mean of slope and bias
4. Draw the lane lines using the mean values of slope and bias

2. Identify potential shortcomings with your current pipeline

As this is a basic lane detection pipeline, therefore the shortcomings are:

1. The program will fail to detect curves correctly, as it does not consider curvature while calculating slopes
2. Lane detection is at a fixed position on the screen
3. There may be difficulty detecting lane lines in images with bright sunlight or shadows

3. Suggest possible improvements to your pipeline

Possible improvements:

1. Consider using polygonal fit to overcome shortcoming #1
2. Instead of detecting lanes lines at a fixed position, try to find out locations of potential lines and then draw the lines there. This overcomes shortcoming #2
3. Fine tuning of parameters may help with shortcoming #3 but will not solve the problem in all cases