

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 in an image and in a video
 - Create a written report to reflect on the work performed
-

Reflection

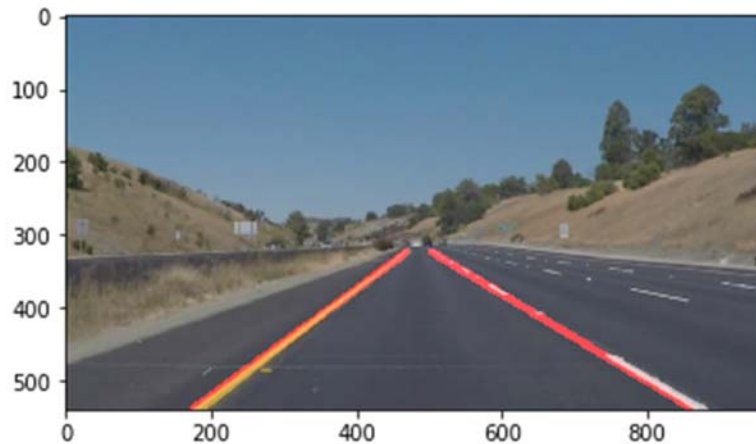
1. Description of Pipeline. As part of the description, explain how you modified the `draw_lines()` function.

My pipeline consisted of 6 steps. First, I converted the images to grayscale. Second, I applied Gaussian Blur to get rid of unnecessary noise. Third, I initiated a canny transform to turn my image into edges. Fourth, I masked my image so that I had a trapezoid showing my area of interest. Fifth, I applied Hough's principles to create an overlay on the image of the lane lines. Sixth, I applied that overlay to the original image.

In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function by initializing a starting line, comparing the slopes (positive or negative), then comparing the right hand and left hand x_1 , y_1 , x_2 , and y_2 values so that I could get an overall line. Using this overall line, I extrapolated where it would intersect with the bottom of the frame so that the line would go from the edge of the image to upper edge of my area of interest.

A challenge I faced when using this method is that, sometimes, the lines would cross, depending on how sensitive the image was. This translated into the video. However, by comparing the tips of the left and right lane lines, I could keep them from crossing.

Below is an image of the example "whiteCarLaneSwitch" with the extrapolated lane lines in red:



2. Potential Shortcomings

Shortcomings of my pipeline are visible when applied to a video. My thresholds and Hough lines parameters pick up minute changes and it was difficult to anticipate them when applied to a video.

These shortcomings can be summed up as a hardcoding failure.

3. Suggest possible improvements to your pipeline

A possible improvement to these shortcomings can be made by creating a debug, visualization tool that would allow me to see how a video or frame change reacts to Hough line parameters and Canny thresholds.