

Finding Lane Lines on the Road

Reflection

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

My pipeline consisted of 5 steps. First, I converted the images to grayscale, then I used Gaussian smooth to blur the image. Afterwards, the Canny edge detection was applied to obtain the gradient image. In the gradient image, only the region of interest was selected for further analysis, and the region of interest was defined by a polygon with four vertices. Next, the line segments in the gradient image were detected by Hough transform.

In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function. First, any lines with an absolute slope less than 0.5 were abandoned. Then, if a line has a negative slope, its two end points were added to a list containing points of left lane, otherwise they were added to right lane list. Afterwards, linear regression was applied on left lane list and right lane list. Finally, only two end points were obtained by using the linear regression equation for each of left and right lane, and the line connecting these two points were draw on the original image.

Here is an example image and the image after being processed by the pipeline





2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be if the camera on the car is at a different position, the region of interest of the image will be changed and my pipeline may fail.

Another shortcoming could be if the lane has a very small turning radius, my pipeline may only be able to detect a relatively short length of the lane ahead.

3. Suggest possible improvements to your pipeline

A possible improvement would be to detect region of interest on the fly. Or we could use a very large region of interest and find out those most relevant lines.

Another potential improvement could be to use the curvature of current lane to predict the direction of next section of the lane.