

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 6 steps. First, I converted the images to grayscale, then I apply Gaussian blur function on the image to make it suitable for next procedure. Third, I apply Canny Edge detection function to take the edges from the previous stage. Next, I mask the region of interest with trapezoid shape so that I can take only the edges from the lanes in front. In fifth stage, I generate Hough lines to mark the lines with edge in the masked region. At the end, I add it to original image and return the result.

In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function by line fitting(`numpy.polyfit`) the multiple edges in the masked region with two different kind of slopes(left, right). After the line fitting, generated lines have slopes and intercepts within the image. By taking the median of those parameters (mean is too easily influenced by extraordinary values and less stable than median), I can figure out the one line that suits the best among the other candidate lines available for marking. With chosen parameters with minimum and maximum x and y coordinate, I can draw line for each side that has maximum length with stable look.

2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when the road is very curvy, steep corner, or no line sign on the road, the line fitting would not work and fails dramatically like the challenge video.

Another shortcoming could be the processing time of the image. This should be a real time application and processing high resolution image within a very short moment of time(less than 100ms) may be a critical feature of this application. Modify the image, finding edges, masking it, line fitting it, and applying it to original images

are in fact whole lot of computing process that sometime cannot be done in a moment. Even though it can be done quite fast with powerful desktop or laptop computer, embedded system in car might not possible to handle all these tasks in real time.

3. Suggest possible improvements to your pipeline

A possible improvement would be to using curve fitting rather than line fitting so the it can handle more sophisticatedly shaped road, but still the computational time left as unsolved problem.