Finding Lane Lines on Road

Amol Sahasrabudhe

June 1, 2017

Abstract

The goals of the project are the following:

- Make a pipeline that finds lane lines on road
- Reflect on what worked and what needs improvement

1 Reflection

My pipeline consisted of 6 steps.

- I converted the image to grayscale
- I applied a gaussian blur to the image. The reason for gaussian blur is that the next step of canny edge detection picks up only the genuine edges and not noise.
- I used canny edge detection with parameters that had given me good performance in the earlier quiz.
- I picked the region of interest. I hard coded the numbers here which means I failed the challenge video. In the next version I will pick them based on image dimensions and should fix one part of the challenge video.
- I apply the hough transform to create lines from edges. I am not sure whether hough transform should be done before or after region selection but I get good results when I do it in this order. It allows me to extrapolate/avg lines only in the interesting region.
- Take a length weighted avg of lines and extrapolate them to region boudary.
- Take weighted avg of lane lines image and original image to get the final image.

In order to draw a single line on the left and right lanes, I modified the $draw_lines()$ function by partitioning lines detected by hough transform into ones with high positive vs. high negative slope. I chose the slope threshold so that lots of other lines that are at an angle not likely to be lane lines get filtered

out. Once I get left and right lines, I extrapolate them almost (but not fully)all the way to the region of interest boundary and take the mean of lines. There are few strategies possible here that I wanted to try but ran out of time.

- Redefine lines into (position,slope) and take avg of (position,slope)
- Take a length weighted mean. (Lengths are calculated before extrapolation)
- Filter outliers before taking mean

I tried 2 and 3 and 3 seems to give better answers in images but there is no discernable difference in videos. I could not try 1.

2 Potential problems with the method and solutions

- I fail the challenge video due to hard coding of region of interest parameters(apart from other reasons I might be failing)
- Curves are also going to be a problem since I detect only lines. I am also likely to lose all lane lines if the car happens to take a wrong turn since now the slope of lane line might not be in the permissible range for my filter.
- I also seem to have problem with the yellow line video where detection of lines in certain frames is failing. I have circumvented the issue by returning the original frame in such cases and as long as there are enough frames where detection works, the video looks good enough. However, I am not sure what impact can losing lines even for one frame can have on car control. A better solution for detection loss in a few frames can be handled by taking the line detection from previous frame and using that for current frame since, in a video it is unlikely that there is a drastic change from one frame to another.
- The lines seem to flicker a lot in the video. Blending current frame lines with previous frame lines is a potential fix for this issue. It would mean lines can only move slowly which is much more likely.