

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 my work in a written report
-

Reflection

1. Pipeline description

My pipeline consists of 6 steps.

1. Converts the images to grayscale.
2. Applies Gaussian smoothing.
3. Applies the Canny transform.
4. Identifies interest region and applies an image mask.
5. Applies Hough transform.
6. Draws the line on the image.

In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function. First of all I divide lines in two groups based on their slope. Then I take all the start and end points in each group and set up linear regression respectively. Using the regression model, I draw the lines for each lanes.

The final result will be something like the image below:



2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when lanes are turning. In this case I think we can take a smaller interest region to avoid unnecessary noise. Also, we can take the advantage of frames before or after this one to take a smooth operation.

Another shortcoming could be computation speed. In order to process the videos in real time, we can downgrade the image quality. Which means if the image quality is too high, we don't need to compute all the pixels. Only parts of them can also achieve similar results.

3. Suggest possible improvements to your pipeline

As I mentioned about, I will try a smaller interest region, may be a dynamic region based on the pre-processing of the image. Obviously, static interest region may cause some problems during complicated cases, like the car is turning or changing the lanes. In this case we can use previous image frames to help us decide interest region. Also, take the info from previous frames can help us building the algorithm more robustic.

Another potential improvement could be sampling image before detection. This mainly helps us speeding up our detection algorithm.