Finding lane lines on the Road

Saikumar
13th Mar 2017

Contents

Image processing pipeline for highlighting lane lines:

The pipeline has in total 8 steps, starting from loading data till the results are saved.
Here is the detailed description of all the steps:

- **Load** image using mpimg.read function.
- Convert the image to **grayscale** to avoid noise(as color is not very important in many cases), reduce code complexity, speed, color is complex and to make the image simpler.
- **Gaussian smoothing** for suppressing noise and spurious gradients.
- For detecting the lanes, finding out gradient can help. **Canny edge** algorithm will be used to detect strong edges and fill the pixels between strong edges. Low and high threshold are set to 50 and 150 respectively.
- **Region of Interest** will be used to fit the polygon of interest and mask the other parts of the images.
- **Hough Transform** to find the lanes based on the parameters passed to function.

  Drawlines function will impose lines on a blank image. Multiple modifications have been made. Initially started with many assumptions, soon figured out slope and constant could help. Although, they have done good performance, certainly results are not good.

  Then the simplest thought would be do a **simple linear regression**. In this case, model is built with Y as input and X as output. Once the model is built, for 540 and 340 values of Y Coordinates, X coordinates will be predicted.

  There are two parts in this, loop part and the regression part.

  Loop does the following:

  1. Select each line
  2. Select x,y coordinates of the line
  3. Find the slope
  4. If less than 0, it is left
     else it should be right

  Now we have x and y points separated for left and right lanes.

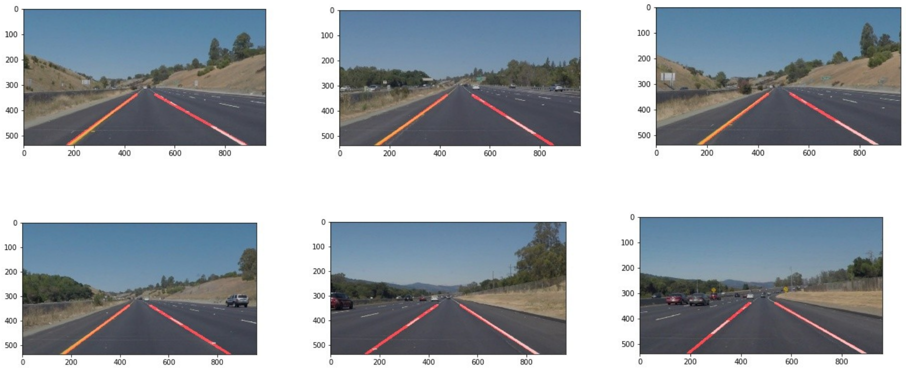
  Following steps are performed:

  1. reshape x and y axis to feed into a regression model
  2. Instantiate a regression model
  3. fit the model with data

4. Now predict what will be the values of x when Y is 540 and 340
5. Plot it in blank image
6. Repeat the steps for right lanes

- Now using **weighted_image** function original image and line image will be combined to superimpose and highlight the lane lines on road.
- **Save image** in to a file using plt.savefig method.

Output images snapshot:



Potential shortcomings with current pipeline:

a. It might not be able to trace curved lanes.

Improvements

a. Instead of simple linear regression, a complex regression should be used to handle curved lanes.