Self-driving Car Nanodegree Program

UDACITY

Project 1: Finding Lane Lines on the Road
Due: 12/11/2018

Duc Tran
ducth1903@gmail.com

I. Objectives

The goals of this project are:

- Design a pipeline that finds the lane lines on the road from images and videos (series of images)
- Document the process and findings along the project

II. Reflection

1. Pipeline

Using the provided helper functions, my pipeline contains of 6 steps, as shown in the flow chart below:

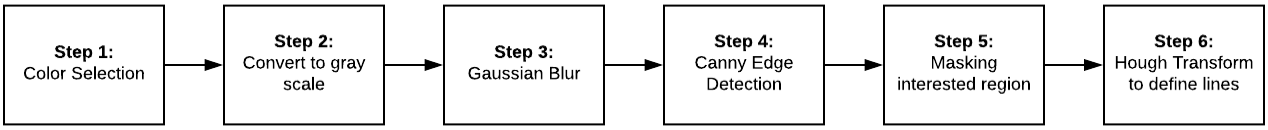

Figure 1: Pipeline for Project 1

In the final step (step 6), in addition to finding the lines using Hough Transform, we need to use the coordinates to draw lines on the image. Instead of plotting all the lines found by the Hough transform, it is desired to have a single solid line on each side to determine the left and right lane, respectively. To do this, I modified the *draw_lines()* function by first calculating the slope for all the lines found by the Hough Transform. Since the origin of the coordinate is at the top left corner of the image, *positive* slope corresponds to right lane while *negative* slope is for left lane. After separating lines for each lane side, I extrapolated by fitting a function with *np.polyfit()* (the function can be a line, a quadratic, a cubic or higher-order equation) through each side's pixels. This equation is the line we are looking for to draw a solid line on each side lane. Then I picked few points on this equation to draw the line (or the curve, depending on the chosen order for the equation).

The output at each step of the pipeline can be seen in the following flow chart:

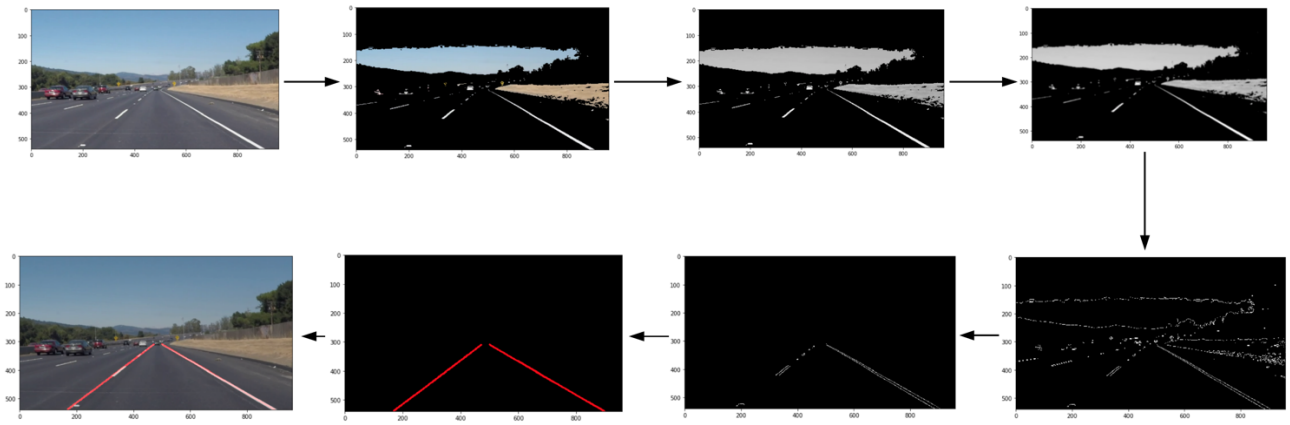

Figure 2: Output at each step in pipeline

Duc Tran
ducth1903@gmail.com

2. Potential shortcomings
    - One shortcoming for the current implementation of the pipeline is fitting the curvature
    - Another shortcoming could be taking care of “unwanted noise” such as the car head at the bottom of the image
    - Tuning parameters to work with the provided examples but may not perform well on new images/videos

3. Possible improvements to the pipeline
    - For the first problem of fitting the curve line, I think we could potentially fit a higher-order equation and introduce more points to make the extrapolation and fitting smoothly
    - For the second issue, we can pre-process the image by filtering out/color selecting (only select white and yellow as well as their associated variations)
    - Apply gradient threshold and color threshold to highlight the lanes and reduce edges of surroundings
    - Tuning parameters for Canny Edge Detection and Hough Transform methods
    - Better algorithms for lane finding