



## PROJECT

## Finding Lane Lines on the Road

A part of the Self Driving Car Engineer Nanodegree Program

## PROJECT REVIEW

## NOTES

SHARE YOUR ACCOMPLISHMENT!  

## Meets Specifications

Really nice job with this submission. Your project meets all the specifications. Congratulations on passing the first project in the Self Driving Car Nanodegree on your very first attempt and best of lucks in the ones ahead.

Cheers and Keep up the good work !!!

Here are some links for further improvements:

<https://www.youtube.com/watch?v=hnXkCIM2RSg&feature=youtu.be>

<https://www.python.org/dev/peps/pep-0008/>

<http://stackoverflow.com/questions/36598897/python-and-opencv-improving-my-lane-detection-algorithm>

## Required Files

The project submission includes all required files

## Lane Finding Pipeline

The output video is an annotated version of the input video.

- Your code outputs video with annotations. Nice job !

In a rough sense, the left and right lane lines are accurately annotated throughout almost all of the video. Annotations can be segmented or solid lines

- The left and right lanes are accurately annotated on the videos.
- Good job !

## Suggestions &amp; Comments

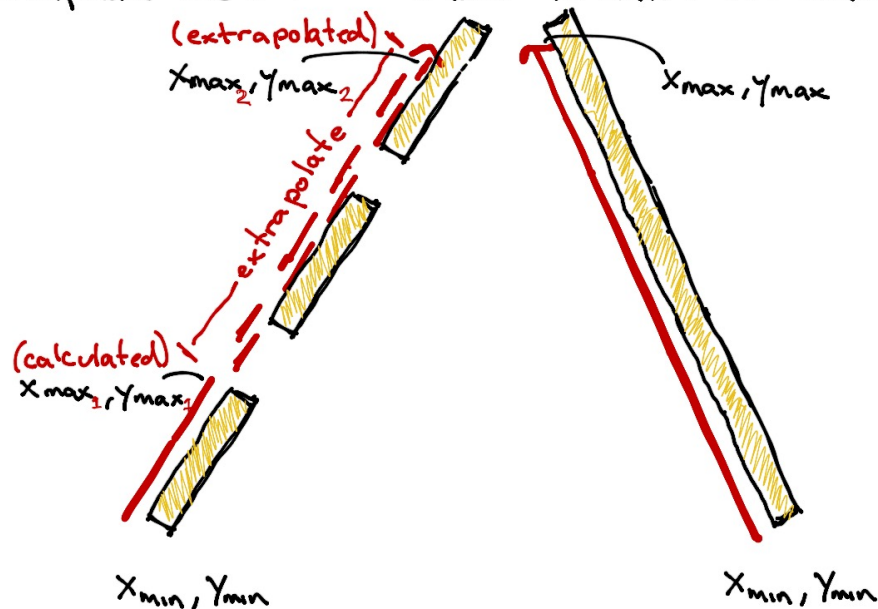
- To center even more your annotations on the actual lane lines, I recommend you try tuning the values of your parameters especially the `threshold`, `min_line_length` and `max_line_gap`.
- Examine the parameters by modifying them separately. That will allow you to identify how each parameter affects the lane line. For example:
  - `max_line_gap` defines the maximum distance between segments that will be connected to a single line.
  - `min_line_len` defines the minimum length of a line that will be created.
- Increasing these parameters will create smoother and longer lines
  - `threshold` defines the minimum number of intersections in a given grid cell that are required to choose a line.
- Increasing this parameter, the filter will choose longer lines and ignore short lines.
- These resources ( [1](#) & [2](#) & [3](#) ) might provide further intuition on the subject here

Visually, the left and right lane lines are accurately annotated by solid lines throughout most of the video.

- Both left and right lines are each a single solid line (i.e. not segmented) generally centered on the actual lane lines
- Here's an algorithm of the process

### Extrapolation Algorithm:

- Anchor the bottom left (of the left lane), and bottom right (of the right)
- Use the slope of both lines as a guide for extrapolation
- Determine which lane has the highest point (closest to horizon)
- Extrapolate the other lane line to match the alternate side



### Left-Lane Extrapolation

$$X_{min}: X_{min}$$

$$Y_{min}: Y_{min}$$

$$X_{max}: \frac{Y_{max} - Y\text{-intercept}}{\text{slope}}$$

$$Y_{max}: \max(Y_{max1}, Y_{max \text{ right-lane}})$$

### Right-Lane Extrapolation

$$X_{min}: X_{min}$$

$$Y_{min}: Y_{min}$$

$$X_{max}: \frac{Y_{max} - Y\text{-intercept}}{\text{slope}}$$

$$Y_{max}: \max(Y_{max1}, Y_{max \text{ left-lane}})$$

### Reflection

Reflection describes the current pipeline, identifies its potential shortcomings and suggests possible improvements. There is no minimum length. Writing in English is preferred but you may use any language.

- Current pipeline is well described
- Possible shortcomings and potential solutions have been provided.
- Good job !

## Suggestion

- As far as curves are concerned on the challenge video, I will encourage you to read this documentation:  
<http://airccj.org/CSCP/vol5/csit53211.pdf>

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

---

[Student FAQ](#)

[Reviewer Agreement](#)