

[Return to Classroom](#)

# Finding Lane Lines on the Road

## REVIEW

## CODE REVIEW

## HISTORY

### Meets Specifications

Brilliant work done. This shows your dedication and commitment towards learning from this nanodegree. Wish you all the best for future projects.

Your parameters looks good, yet you may try one optimal possible set of parameters which might not create much difference your current output but will be helpful in certain conditions like curved lanes, shadows etc

- max\_line\_gap that defines the maximum distance between segments that will be connected to a single line.
- min\_line\_len that defines the minimum length of a line that will be created. Increasing min\_line\_len and max\_line\_gap (~100 and above) for Hough Transform will make your lines longer and will have less number of breaks. (this will make the solid annotated line longer in the output) Increasing max\_line\_gap will allow points that are farther away from each other to be connected with a single line.
- threshold increasing (~ 50-60) will rule out the spurious lines. (defines the minimum number of intersections in a given grid cell that are required to choose a line.)
- Decreasing the kernel-size in the Gaussian Filter might also help, as this will remove the noise making the image less blurry.
- Consider using rho value of 2 (rho, distance resolution of the Hough accumulator in pixels.)
- The detection of straight edges through Hough transform will induce some uncertainties because of the variations in the photograph conditions such as lighting, shadow, vibrations etc. This makes the calculations of the slopes and the end points fluctuate within a certain zone. In order to avoid this noise, a Kalman filter can also be used to smoothen out the fluctuations in the slope and end point estimation.

### Required Files

The project submission includes all required files:

- Ipython notebook with code
- A writeup report (either pdf or markdown)

Good work! All required files are present 🍌🍌

### Lane Finding Pipeline

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

Good work! Your output videos are properly annotated !!

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

Your pipeline is pretty good, with resulting lines centered on the target lane lines. The annotations were solid as well as the left and right lane lines were accurately annotated throughout the videos.

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

You did good enough as the resulting lane lines are a single solid line and are centered right on the actual lane lines throughout the videos.



## 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.

Good work describing your current pipeline and guring out some of its potential shortcomings and possible improvements. Whereas more possible improvements are:

1. Image from infrared camera.
2. Adding a outlier reduction approach like RANSAC on the hough lines.
3. Using curve fitting to plot the curve instead of straight lines

For further reference :

Ransac: [https://udacity-reviews-uploads.s3.us-west-2.amazonaws.com/\\_attachments/50281/1516262752/RANSAC.pdf](https://udacity-reviews-uploads.s3.us-west-2.amazonaws.com/_attachments/50281/1516262752/RANSAC.pdf)

RETURN TO PATH

Rate this review

START

---