UDACITY

Logout

‹ Return to "Self-Driving Car Engineer" in the classroom

DISCUSS ON STUDENT HUB

Finding Lane Lines on the Road

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Hello,

Good work overall, you did a good job with your pipeline meeting all the requirements in just one go!! Keep this up 😄
Congratulations, on meeting all of the specifications and good luck with your Self Driving Car Nanodegree. 😄

Required Files

The project submission includes all required files:

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

Lane Finding Pipeline

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

Well done providing an annotated output video!

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 right on target. ✅

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

- Single solid line ✅
- On target ✅
  The left and right lane lines are accurately annotated throughout most of the video. Well done!!

SUGGESTIONS

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.

- `Low_threshold` ~ 50
- `high_threshold` ~150
- Decreasing the `kernel-size` in the Gaussian Filter might also help, as this will remove the noise making the image less blurry.
- 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.

For further reading and references:

link 1
link 2
link 3
link 4 for optional video.
link 5
Link 6

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.

- Pipeline description ✅
- shortcomings ✅
- Improvements ✅

This pipeline chalks out straight lines, hence it works best on straight roads. Using it on curved roads and also on roads on a downhill or sometimes also on a uphill wouldn't result in satisfactory results as it would best fit the curve with a straight line.

The shortcoming of the curved lane could be solved by using a quadratic fit, instead of a linear fit, where we'd have more coefficients and we could create the curved lane accordingly. The problem of ending one lane could be solved by creating a temporary parallel lane for the vanished lane allowing the car to change lanes until it finds well defined lanes. The problem of cracks on the roads leading to edges in the region of interest could be solved by creating a regions of interest which is a combination of offsetted trapezium, with parallel edges, one sitting inside and one outside. This would only keep the lanes in the region and as long as a lanes are defined, we'll have our lines.

One improvement can be using a different color space. HSL will be much more efficient in finding the yellow lane line and in case of shadows.The HSL color mask in this sceneario performs better than the RGB because it is able to consistently filters smooth yellow pixels than RGB color mask, despite it sometimes picks up unwanted noises such as yellow road sign and the lawn on the right. However, these minor imperfections of HLS mask should not be a problem, as later we can apply `region of interest` to specifically crop the non-road regions.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review