UDACITY

Advanced Lane Finding

REVIEW
CODE REVIEW
HISTORY

Meets Specifications

Dear Student
Congratulations for passing this project.👏👏

Writeup / README

The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.
README includes statements and supporting figures / images that explain how each rubric item was addressed.✅

Camera Calibration

OpenCV functions or other methods were used to calculate the correct camera matrix and distortion coefficients using the calibration chessboard images provided in the repository (note these are 9x6 chessboard images, unlike the 8x6 images used in the lesson). The distortion matrix should be used to

un-distort one of the calibration images provided as a demonstration that the calibration is correct. Example of undistorted calibration image is Included in the writeup (or saved to a folder).

OpenCV functions were properly used to calculate both the camera matrix and distortion coefficients. Un-distortion demonstration on the test image looks perfect, great job!

Can check more info about this process in documentation if you are interested:
http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html?
http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html
http://docs.opencv.org/3.1.0/dc/dbb/tutorial_py_calibration.html
http://research.microsoft.com/en-us/um/people/zhang/calib/

Pipeline (test images)

Distortion correction that was calculated via camera calibration has been correctly applied to each image. An example of a distortion corrected image should be included in the writeup (or saved to a folder) and submitted with the project.

All images/frames are properly un-distorted using the calibration and distortion matrices which were calculated in a previous step, excellent job!

A method or combination of methods (i.e., color transforms, gradients) has been used to create a binary image containing likely lane pixels. There is no "ground truth" here, just visual verification that the pixels identified as part of the lane lines are, in fact, part of the lines. Example binary images should be included in the writeup (or saved to a folder) and submitted with the project.

Well done!!

- The Sobel operator and color threshold(HLS) were applied to produce binary images for finding likely lane pixels.

SUGGESTION:

- Applying color threshold to the B(range:145-200) in LAB for shading & brightness changes and R in RGB in final pipeline can also help in detecting the yellow lanes.
- And thresholding L (range: 215-255) of Luv for whites.

OpenCV function or other method has been used to correctly rectify each image to a "birds-eye view". Transformed images should be included in the writeup (or saved to a folder) and submitted with the project.

The binary image is correctly rectified (warped) to obtain a "birds-eye view" prior starting with the line detection procedures.

Methods have been used to identify lane line pixels in the rectified binary image. The left and right line have been identified and fit with a curved functional form (e.g., spine or polynomial). Example images with line pixels identified and a fit overplotted should be included in the writeup (or saved to a folder) and submitted with the project.

Left and right lane lines are perfectly identified from the rectified binary image, by means of a sliding window and a second order polynomial fitting, a very good job here as well!

Here are some good links for other methods described:
https://www.researchgate.net/publication/257291768_A_Much_Advanced_and_Efficient_Lane_Detection_Algorithm_for_Intelligent_Highway_Safety
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5017478/

Algorithm based on current project:
https://chatbotslife.com/robust-lane-finding-using-advanced-computer-vision-techniques-46875bb3c8aa#.l2uxq26sn

Here the idea is to take the measurements of where the lane lines are and estimate how much the road is curving and where the vehicle is located with respect to the center of the lane. The radius of curvature may be given in meters assuming the curve of the road follows a circle. For the position of the vehicle, you may assume the camera is mounted at the center of the car and the deviation of the midpoint of the lane from the center of the image is the offset you're looking for. As with the polynomial fitting, convert from pixels to meters.

Well done!! Now the annotated values for the radius of curvature seems to be consistent with the lane's curvature.

The fit from the rectified image has been warped back onto the original image and plotted to identify the lane boundaries. This should demonstrate that the lane boundaries were correctly identified. An example image with lanes, curvature, and position from center should be included in the writeup (or saved to a folder) and submitted with the project.

Annotated lines and patch on the rectified image are properly "unwarped" and annotated on the original images, well done!

Pipeline (video)

The image processing pipeline that was established to find the lane lines in images successfully processes the video. The output here should be a new video where the lanes are identified in every frame, and outputs are generated regarding the radius of curvature of the lane and vehicle position within the lane. The pipeline should correctly map out curved lines and not fail when shadows or pavement color changes are present. The output video should be linked to in the writeup and/or saved and submitted with the project.

lane detection works pretty well in project video.

SUGGESTIONS:

- In addition to other filtering mechanisms. You can also use cv2.matchShapes as a means to make sure the final warp polygon is of good quality. This can be done by comparing two shapes returning 0 index for identical shapes. You can use this to make sure that the polygon of your next frame is closer to what is expected and if not then can use the old polygon instead. This way you are faking it until a new frames appear and hence will get good results.

Discussion

Discussion includes some consideration of problems/issues faced, what could be improved about their algorithm/pipeline, and what hypothetical cases would cause their pipeline to fail.

Very good job including your discussion in the README, pointing where the pipeline could fail and also provided solutions to improve lane detection.

⬇ DOWNLOAD PROJECT

RETURN TO PATH