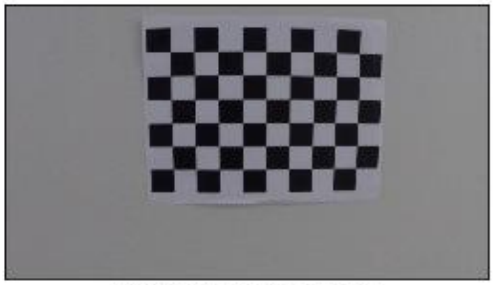
TERM 1, PROJECT 4

Udacity Self Driving Car Nano Degree

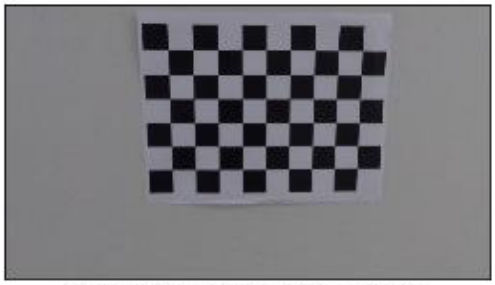
Advanced Lane Finding

Sri Harsha Kunda

1. **Writeup / README**
    a. This document mentions all the rubric points and how they were implemented in the Jupyter Notebook Code.
2. **Camera Calibration**
    a. **Briefly state how you computed the camera matrix and distortion coefficients. Provide an example of a distortion corrected calibration image.**
        i. Initially all the calibration images were loaded using glob function and passed through calibrate_camera function from cell 2. I used findChessboardCorners function to generate the camera mtx and dst values. These values were later saved in a pickle file to import for later use.
        ii. Parameters used for camera calibration function are 9,6(mentioned in Cell 3). Undistort function from OpenCV was used to correct the distortion effect of the image. The below image shows the comparison between Original Calibration Image and Distortion Corrected Calibration Image.



Original Calibration image



Distortion corrected Calibration image

3. **Pipeline (test images)**
    a. **Provide an example of a distortion-corrected image.**
        i. Below is the distortion corrected image via camera calibration matrix and distortion coefficients obtained earlier from calibration images. Cell 5 has code where all the test images are corrected for distortion.

TERM 1, PROJECT 4

Original image

Distortion corrected image

b. Describe how (and identify where in your code) you used color transforms, gradients or other methods to create a thresholded binary image. Provide an example of a binary image result.

i. Cell 9 has a function for color thresholding for white and yellow colors. The idea here is to threshold white in RGB space and Yellow in HLS space and add a combined image later. Instead of thresholding and warping I found its easier and more accurate when the image is warped and thresholded later to detect the lane line. Below is the image that shows a warped image and thresholded binary image.

Warped Image

Thresholded Image

c. Describe how (and identify where in your code) you performed a perspective transform and provide an example of a transformed image.

i. Cell 7 and 8 has the code related to perspective transformation of the image. The undistorted image was passed through perspective_transform (used cv2.getperspectivetransform from OpenCV) function with source (src) and destination (dst) points. Below is the image which shows Undistorted image and Warped form of the same image.

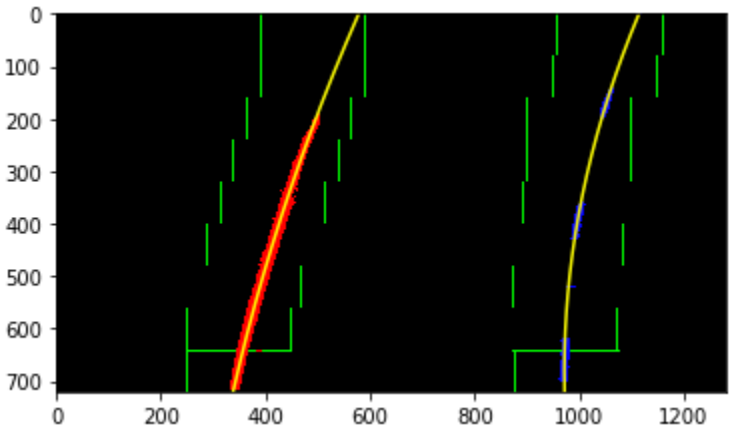
TERM 1, PROJECT 4


Original image


Warped Image

d. Describe how (and identify where in your code) you identified lane-line pixels and fit their positions with a polynomial?

i. After thresholding the warped image, I took the histogram of binary thresholded image to find the peak values in a selected area. This was done in Cell 11. In cell 12, I created a function just like in lecture videos which is called by hist_window. This function will scan the binary image for non-zero pixels and append them to a list. Cell 13 has the formula for curve fitting by taking the left and right pixel list values from Cell 12. Below image shows the curve fitting of the lane lines from Binary Image.



e. Describe how (and identify where in your code) you calculated the radius of curvature of the lane and the position of the vehicle with respect to center.

i. Cell 14 has function curvature which takes left and right curve fit values as input and outputs the curvature values. As mentioned in the lectures, a conversion of 30/720 (meters/pixel) was performed in the function to obtain right curvature values.

f. Provide an example image of your result plotted back down onto the road such that the lane area is identified clearly.

i. Below image shows a test image with lane area plotted and curvature values displayed on the top left corner.

TERM 1, PROJECT 4

4. **Pipeline (video)**
    a. **Provide a link to your final video output. Your pipeline should perform reasonably well on the entire project video (wobbly lines are ok but no catastrophic failures that would cause the car to drive off the road!)**
        i. Link to Video: https://www.youtube.com/watch?v=u_BKCNNUBfk
5. **Discussion**
    a. **Briefly discuss any problems / issues you faced in your implementation of this project. Where will your pipeline likely fail? What could you do to make it more robust?**
        i. I did not find any problems in my project video or test images. However, I had problem with the challenge video. My current thresholding function works when there is only the lane but not any road signs like a diamond sign between the lanes and the yellow lanes are barely visible in the challenge video under HLS or HSV color spaces. I will keep trying to improve my code to see if I can find the lane lines in both challenge and harder challenge videos.