Example Writeup

You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to submit a pdf if you prefer.

Advanced Lane Finding Project

The goals / steps of this project are the following:

Compute the camera calibration matrix and distortion coefficients given a set of chessboard images. Apply a distortion correction to raw images.
Use color transforms, gradients, etc., to create a thresholded binary image.
Apply a perspective transform to rectify binary image ("birds-eye view").

Detect lane pixels and fit to find the lane boundary.
Determine the curvature of the lane and vehicle position with respect to center.
Warp the detected lane boundaries back onto the original image.
Output visual display of the lane boundaries and numerical estimation of lane curvature and vehicle position.

Rubric Points Camera Calibration

1. Have the camera matrix and distortion coefficients been computed correctly and checked on one of the calibration images as a test?

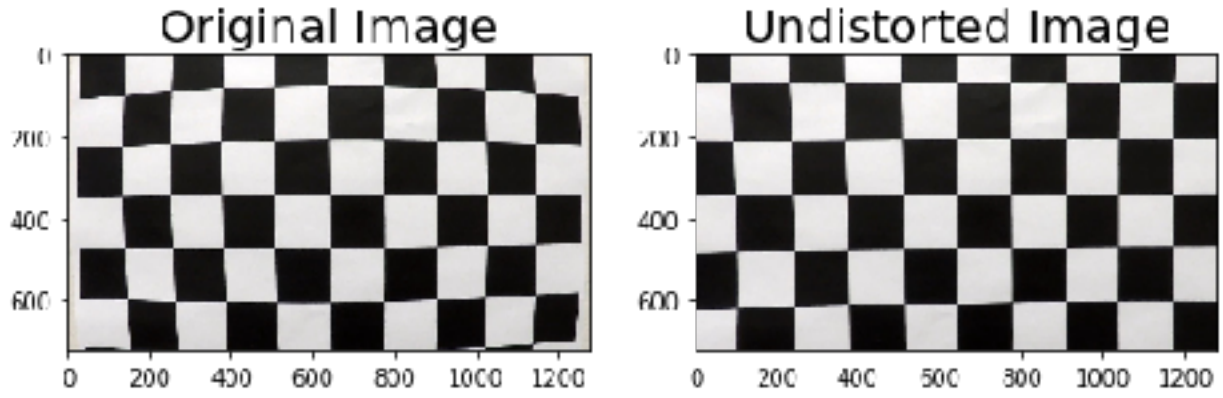
The code for this step is contained in the first code cell of the IPython notebook code cell 2, 3 and 4

I start by preparing "object points", which will be the (x, y, z) coordinates of the chessboard corners in the world. Here I am assuming the chessboard is fixed on the (x, y) plane at z=0, such that the object points are the same for each calibration image. Thus, `objp` is just a replicated array of coordinates, and `objpoints` will be appended with

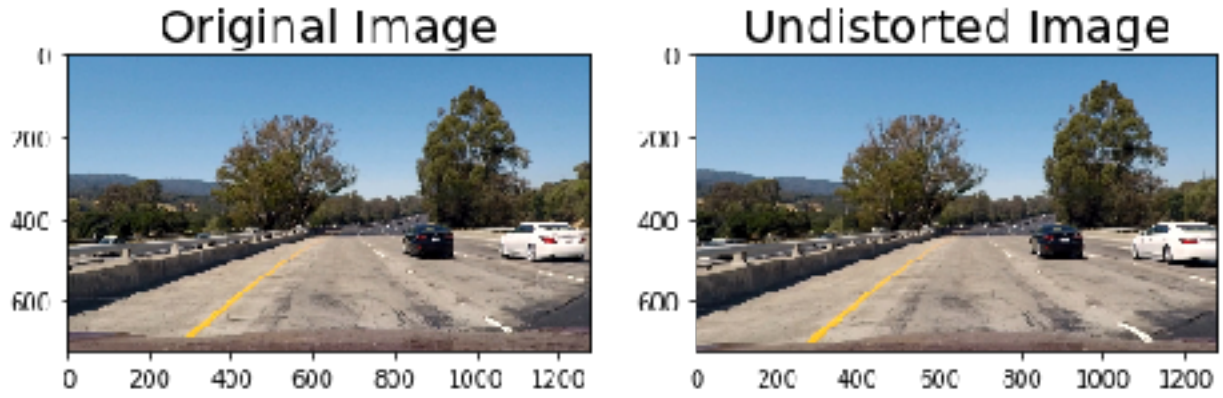
a copy of it every time I successfully detect all chessboard corners in a test image.

`imgpoints` will be appended with the (x, y) pixel position of each of the corners in the image plane with each successful chessboard detection.

I then used the output `objpoints` and `imgpoints` to compute the camera calibration and distortion coefficients using the `cv2.calibrateCamera()` function. I applied this distortion correction to the one image used for calibration and to the test image using the `cv2.undistort()` function and obtained this result:



Test images



Pipeline (single images)

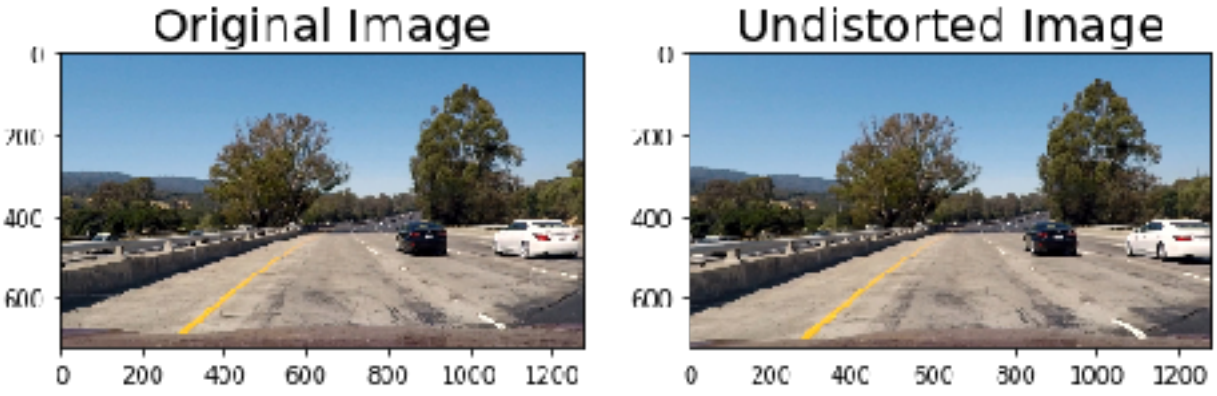
1. Has the distortion correction been correctly applied to each image?

The distortion correction is applied as follows.

The objpoints and imgpoints created while finding the corners on chessboard are used to calibrate using cv2.calibrateCamera(objpoints, imgpoints, img_size, None, None)

mtx and dist returned from above cv2.calibrate was used in cv2.undistort function to create undistorted images

The result on test image is as follows



2. Has a binary image been created using color transforms, gradients or other methods?

Binary image was created by applying following.

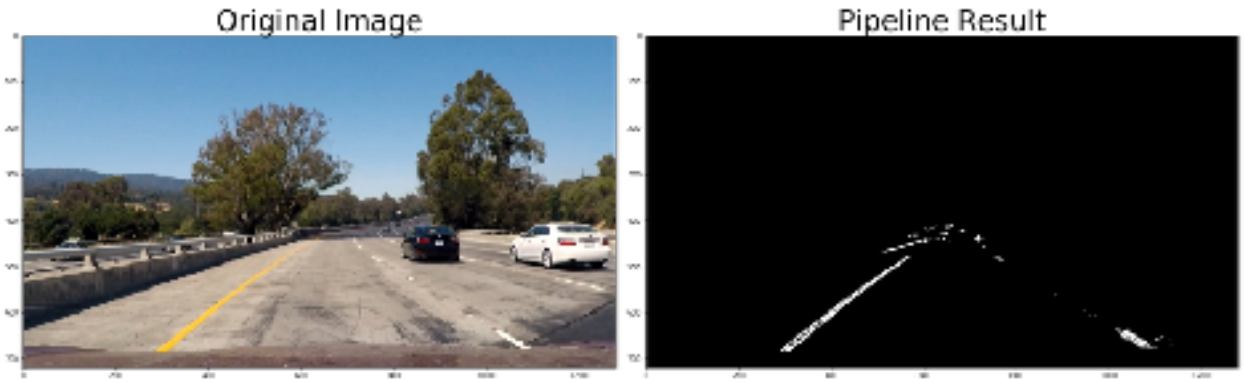
#1 Gaussian Blur was applied to smoothen the image and remove noise.

#2 S channel is extracted from HLS colorspace

#3 gradx, grady, magnitude of gradient and direction of gradient was applied to gray image

#4 The thresholded image was combined with s channel

#5 Then it was masked with region of interest to remove all unwanted area.

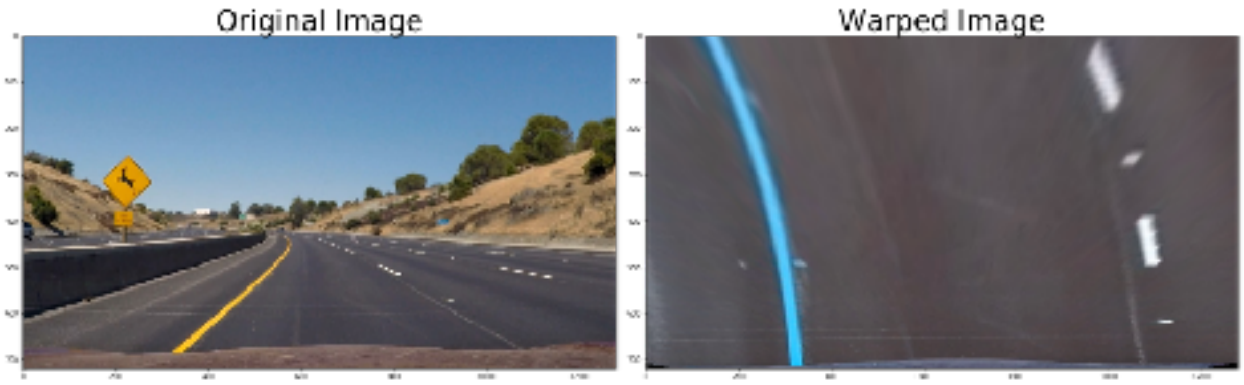


3. Has a perspective transform been applied to rectify the image?

The perspective transform is applied in code cell #11

It uses cv2.getPerspectiveTransform and cv2.warpPerspective functions to accomplish this

The result of these function is as follows



This resulted in the following source and destination points:

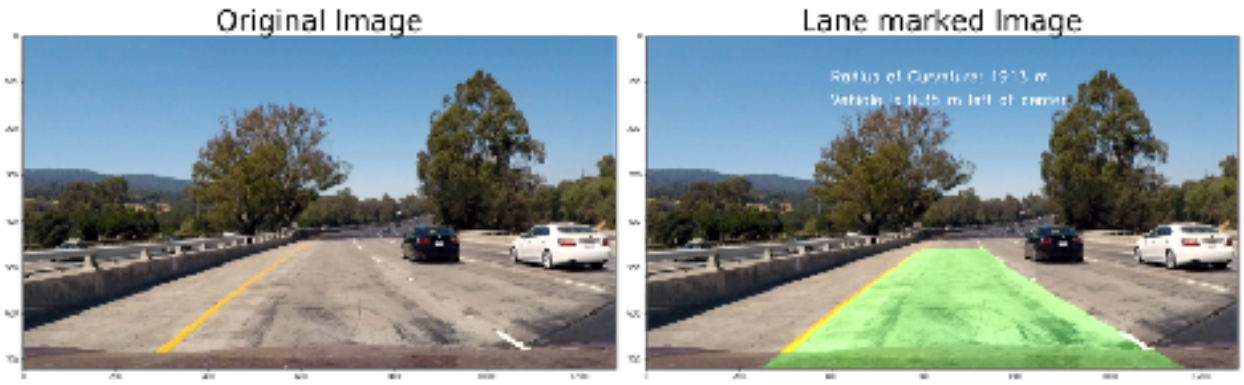
4. Have lane line pixels been identified in the rectified image and fit with a polynomial?

The lane lines were identified and function fit_lanes in cell code #13

5. Having identified the lane lines, has the radius of curvature of the road been estimated? And the position of the vehicle with respect to center in the lane?

cell code #15 is the final function which pulls all functions together and add green polygon in the lane area.

The result image is as follows.



Pipeline (video)

1. Does the pipeline established with the test images work to process the video?

Yes it works are expected the video is include with this submission under the name

project_video_lanes

README

1. Has a README file been included that describes in detail the steps taken to construct the pipeline, techniques used, areas where improvements could be made?

Yes this document is README

Discussion

#1 Area of interest is hardcoded and need some hit and trial to get it right.
#2 Lot of code was used from class material. Need to spend more time understanding it more and improve it.
#3 Warped image gets out of focus for the points far from camera. Need to see to see how to keep image in focus.

#4 Creating project video takes about 8 mins. Earlier it was taking 15 mins. Could reduce to 8 mins by doing camera calibration outside of main function so it doesn't get called again.

#5 Need to see how to speed up generation of video with marked lanes by removing code out of repetitve loops which is needed only once.