Project 2: Advanced Finding Lane Lines on the Road

The final video can be found at the same directory, under the name of result_video.mp4.

Reflection

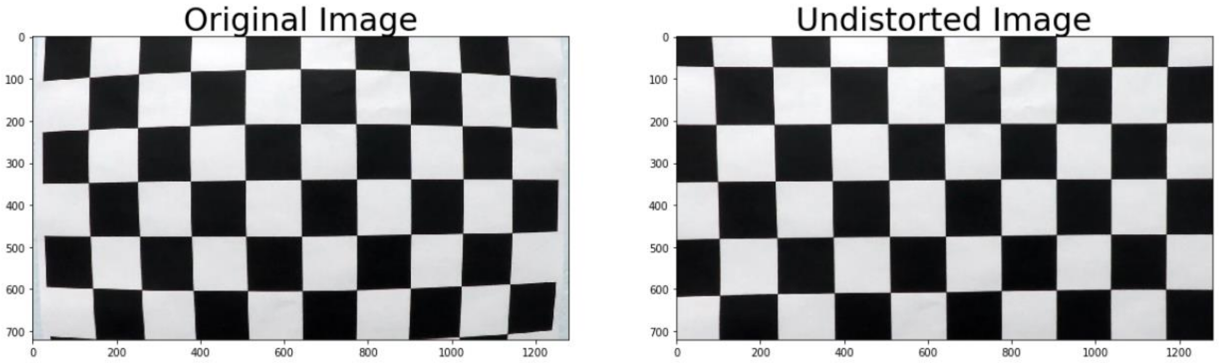
1. Describe your pipeline.

Below is the approach I took for this project

- Do a proof of concept for the software pipeline for this project by coding and applying to one image.
- Once completed, see if the software pipeline is robust enough with other images
- If unsatisfactory, try tweaking the parameters
- If satisfactory, enable the code to allow video (From the 1st project)

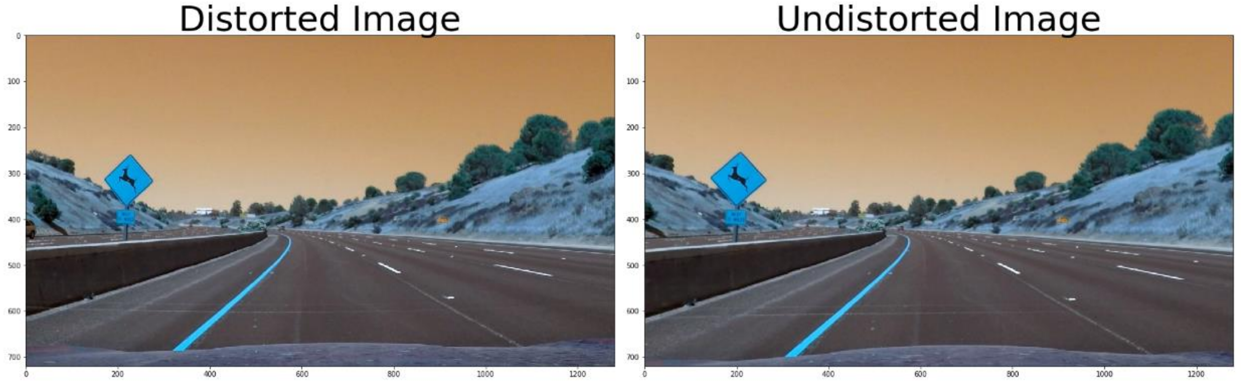
My pipeline consisted of the following steps in sequential order in the code.

Firstly, the camera calibration matrix and distortion coefficients were computed based on given a set of chessboard images (need around 20 chessboard images from different orientations but on a flat surface). By applying the openCV chessboard library (find chessboard corners) to obtain the calibration matrix and distortion coefficients. The parameter regarding the corners should be reflective of the size and number of boxes in the height and width of the chess board.


Original Image

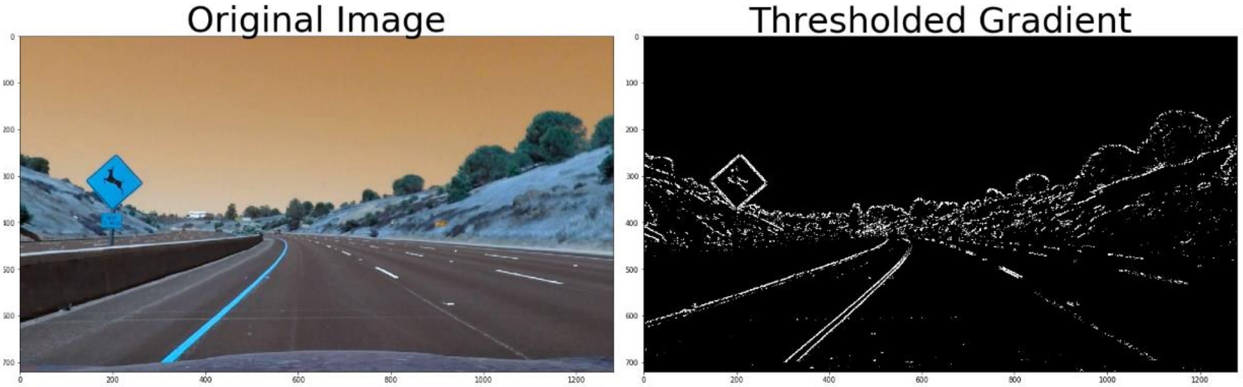
Undistorted Image

Secondly, apply the distortion correction to raw images. Print out the original and undistorted image to see the difference.


Distorted Image

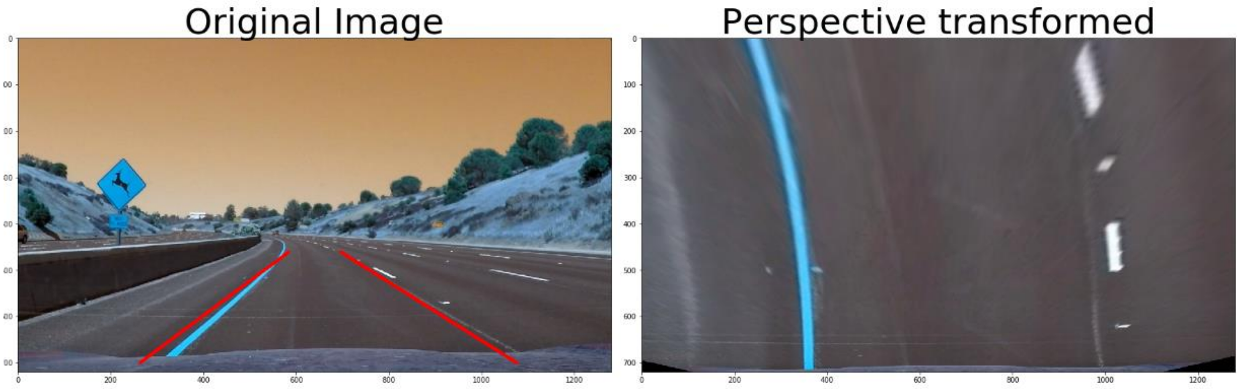
Undistorted Image

Thirdly, create the threshold binary image, to prepare the data for better analysis of this application. The threshold binary image is a combination of color transforms, Sobel x and y and gradient directional thresholds.

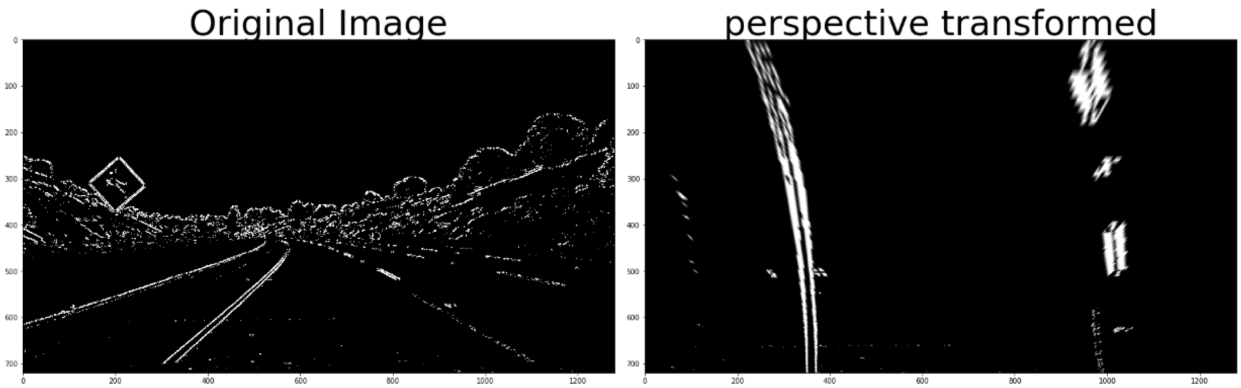

Original Image

Thresholded Gradient

Followed by, applying perspective transform to take into curved roads. This is different from project one which was handling straight roads. Identify the region of interest and apply the perspective transform to have a tops down view. This will ensure any curved lanes will be identified and used for further analysis in the following steps.

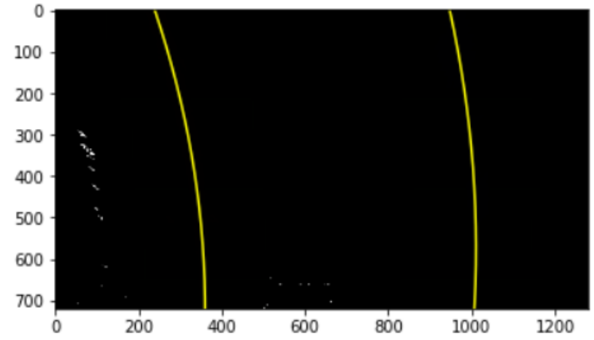

Original Image

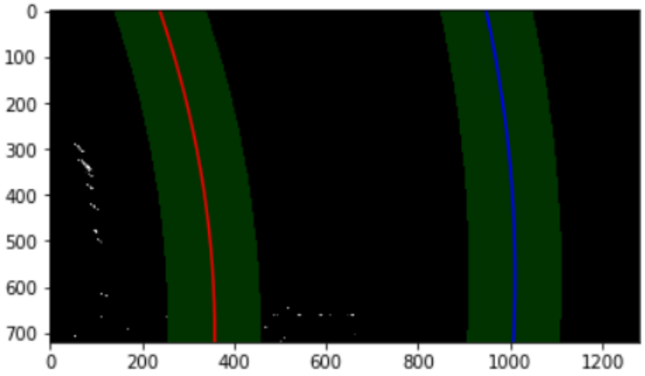
Perspective transformed



After that, the lane pixels needs to be detected for the left and right lane, and the lane boundary.

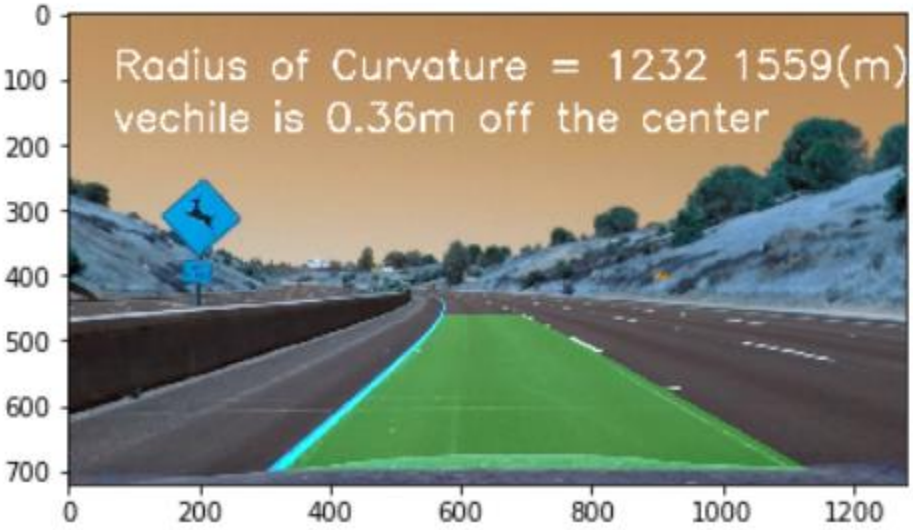
```
True
```





Followed by, detecting the curvature of the lane (around 1 km) and location of the vehicle based on the offset from middle, from left to right lane.

Once identified, warp back the detected lane boundaries into the original image, with display of numerical estimation of lane curvature and vehicle positon.



Radius of Curvature = 1232 1559(m)
vechile is 0.36m off the center

2. Identify potential shortcomings with your current pipeline

Potential Shortcoming 1: The colour consistency of the road. The roads can be asphalt black or concrete grey. Due to wear and tear or poor construction works, there are chances of different shades of black in the road. Since we are using thresholds for the binary image, we can face some problem.

Potential shortcoming 2: Light conditions affect the vision based system. At a bright setting, some of the lane lines might not be visible. Shadows too causes issues.

Potential shortcoming 3: Wet weather. When the road is wet(with puddles or just wet), It can cause refection which messes with the lane detection

3. Suggest possible improvements to your pipeline

A possible improvement would be to tweak the parameters and identify the optimal set of parameter configuration to further enhance the final result.

Another potential improvement could be to ensure under all environment conditions the pipe line works and use more color transform and thresholds to make the pipeline more robust.