

Advanced Lane Finding

OVERVIEW

In this project we try to find the lane in a more accurate way. In this writeup I will go through each step in the pipeline and I will show the output for each. *(All the code used to show the examples is commented).*

The following are the steps followed to get my output video:

1. Pipeline processing
2. Lane Finding Images processing

PIPELINE PROCESSING

The step is composed from the following:

1. Camera calibration
2. Distortion correction
3. Image Tranform
4. Color/gradient threshold

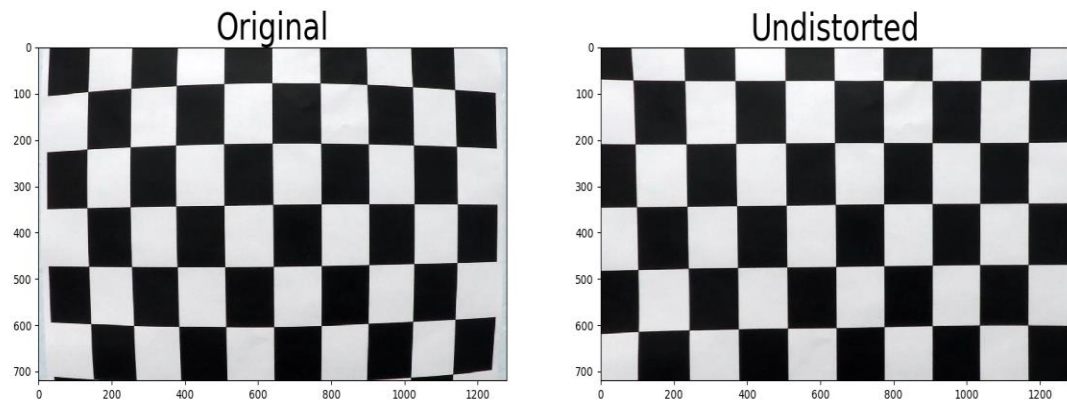
Camera Calibration

The first step we do is to calibrate the camera using the images in ./camera_cal/*.jpg. This step is done only once.

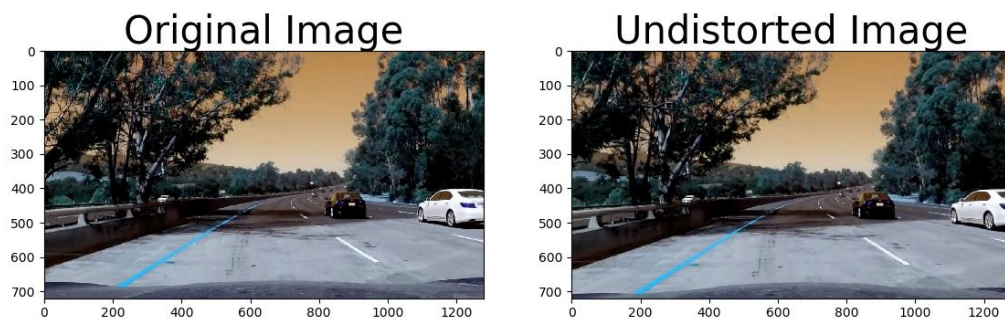
Undistorted Images

After calibrating the camera, we start to undistort the frame we receive. Below is an example of image before and after the distortion on **the chess board** , also there's an example on the road image.

You can find the images as well in the output_images/undistimages



Un-Distortion On chess Board



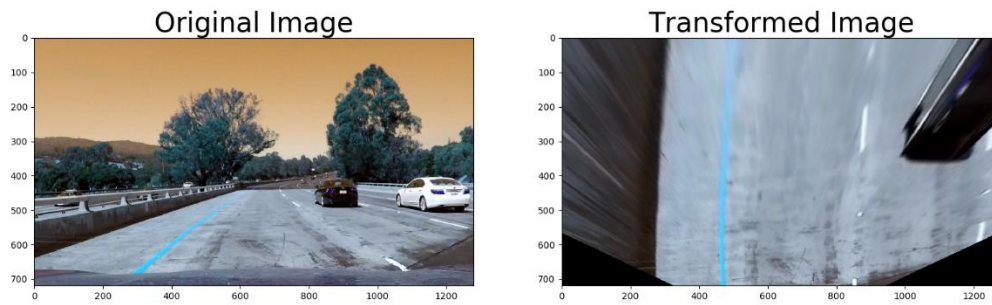
Un-Distortion On chess Board

You can find more images in the following path `output_images/undistimages`

Check also `UndistortedChessBoardTest` in `Funcs.py` that's used to undistort chess board.

Image Transform

After finishing the distortion we need to get a birdview so I warp the image. In this part we should keep Minv as it will be used later.



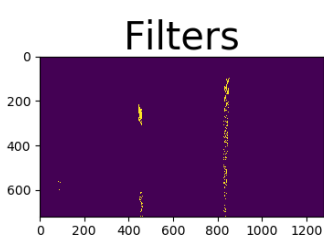
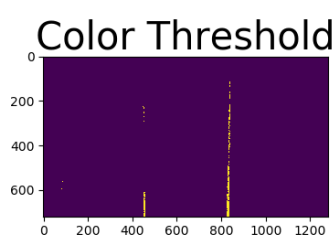
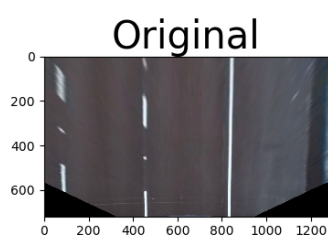
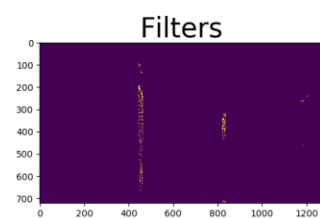
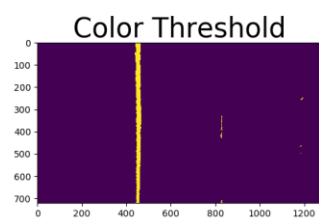
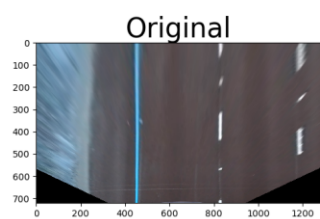
Color/gradient threshold

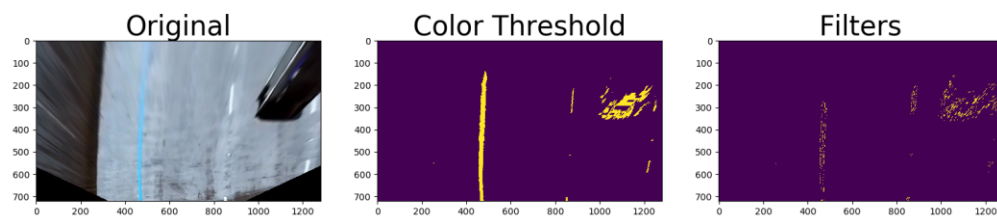
After Calibrating and undistort the frame (image) I start to make some filtering to the image using both gradient and color thresholds. I am oring the two results to get a well combined image.

1. I used first Sobel filter for x and y and I and their result
2. I used magnitude of the gradient and I and it with the direction of the gradient
3. I used S_Channel color space filter
4. I used as well B_RGB to remove any low R, G, B like black objects

Then all the four filters I have OR them together to get my final filtered image.

Below is an example of using Color Space and gradient filters, that's in an other word an OR for step 1 and step2 and the other image is an OR between step 3 and step 4

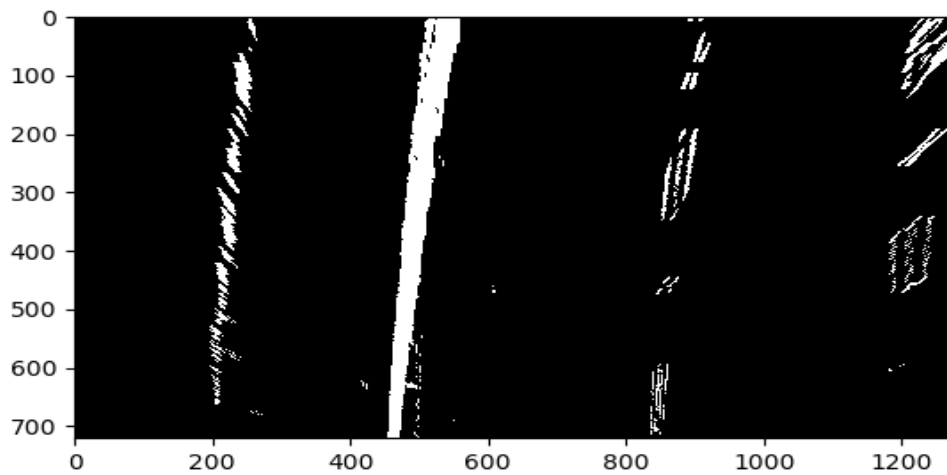




You can find more images in ColorGradientThreshold in the outputimages folder . Below an image for oring between the two thresholds



Original Image



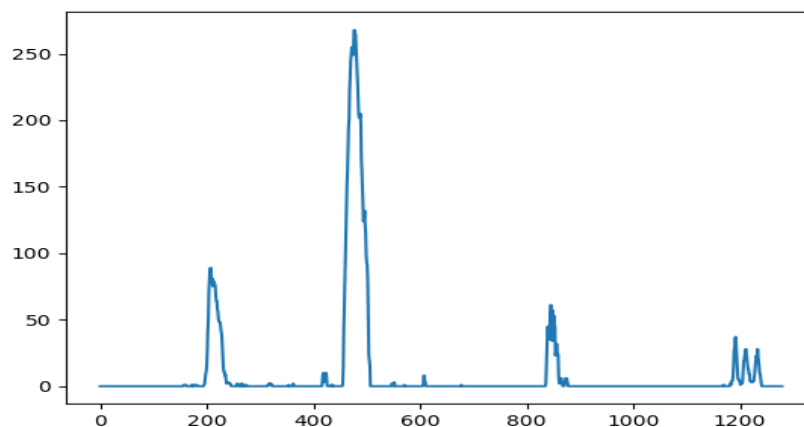
Birds View for the lane

LANE FINDING

Now and after we get a birdview for the image so now we need to find the left the right and left lane line.

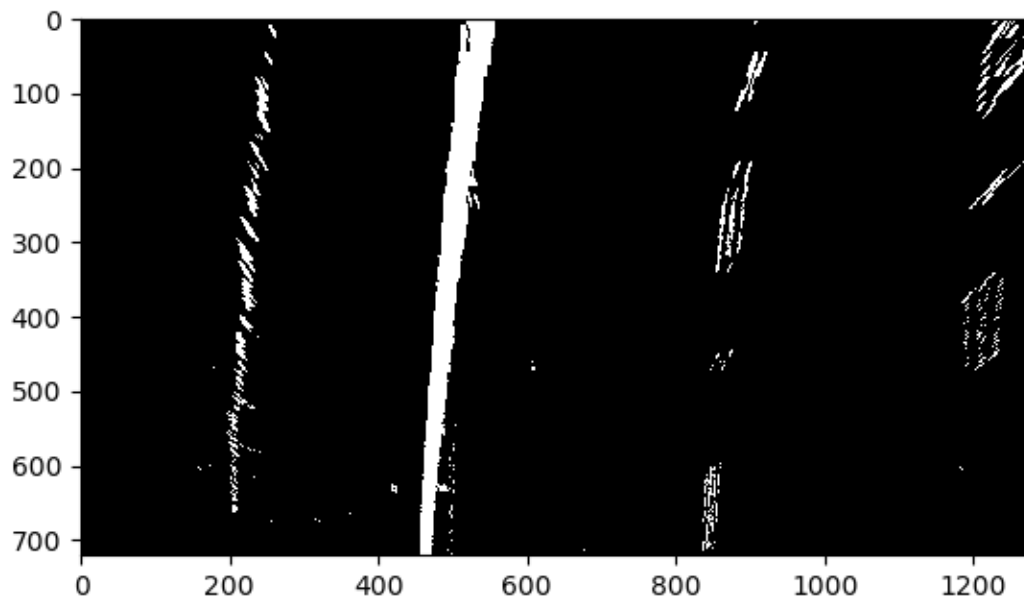
Image Histogram

As a first step we get the histogram of the warped image. This step is very important as it shows us where we most probably have the image. In this step we check where is the peak in the left side and right side to get an intuition about where is the line

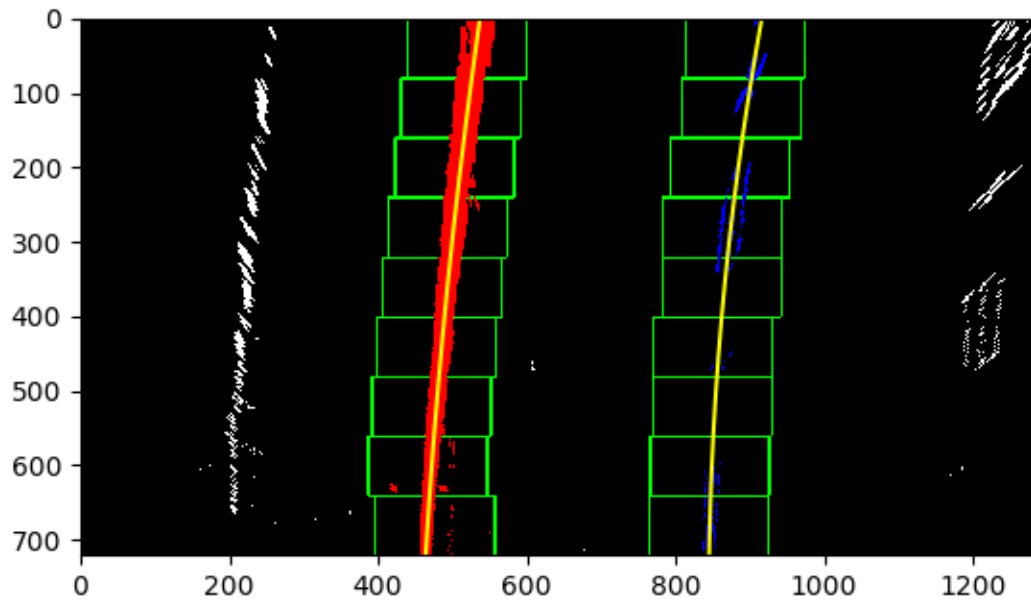


Sliding window search

To find the lane line we use sliding windows, I have used nine windows the minimum number of pixels found is 40 I have pass them on the image to detect the center of the line

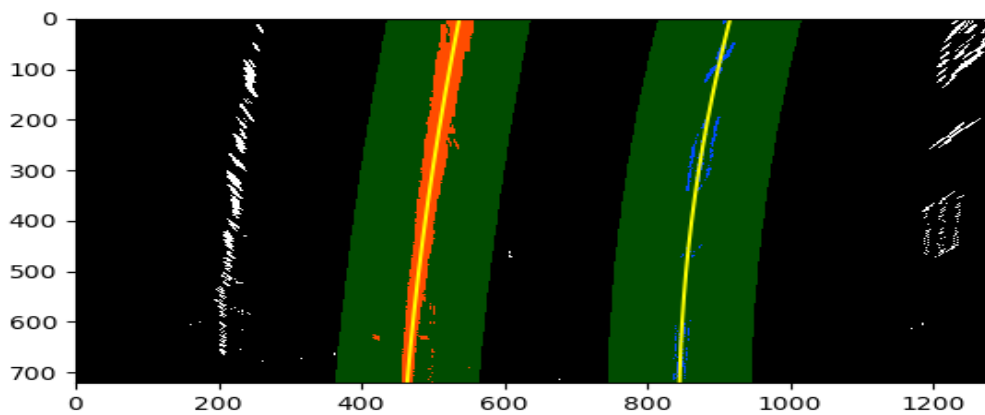


Original Image



Sliding window to search the line

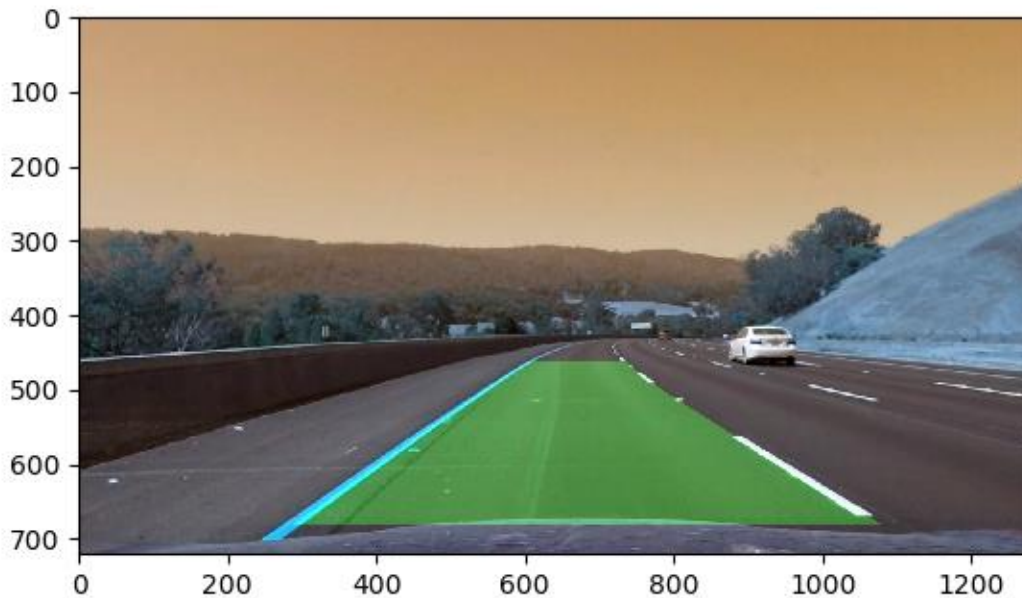
After that I filled the lane line



Visualize Lane Line

PROCESS ON ORIGINAL IMAGE

After I am done with all the above steps I have used I un-warpped the image and I highlighted where lane is



Sanity Check

The following is checked:

1. The distance between the lane lines to make sure that it is the right line. The camera is supposedly in the center of the car so we should have around same distance between the line and the center.[Update:There was an issue here and now it is fixed]
2. Curve is between 0.45 to 1.75 and if not we skip the frame.

PROBLEMS AND SOLUTIONS

One of the problem I had during processing video image is that when we had a black item like the car in the image the curve start to deviate to fix that I followed:

1. Gaussian Blurring to the image
2. Sanity Check

Also, there is some cases which may cause the pipeline to fail if for example the image captured night as we depend on RGB filter this can cause a problem so may be using another color spatial be an idea to overcome this.

One more problem our calculation may fail if we have the road with a certain slope