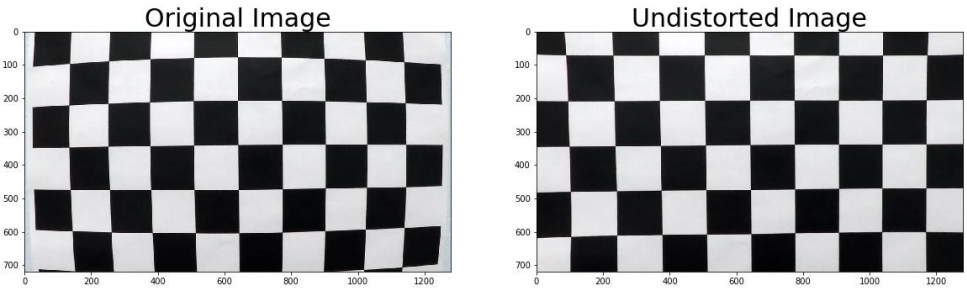
**Advanced Lane Finding Project**

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

## Camera Calibration:

A camera projects 3D world-points onto the 2D image plane. I used 2D Camera Calibration as we learned in chapter 15 by extracting calibration matrixes and parameter from 20 Checkerboard Images. Using this technique set the world coordinate system to the corner of the checkerboard, so that all points on the checkerboard lie in one plane. Below is an example from Checkboard images. The Code for this step is provided in Notebook Cell 2-3.




## Pipeline (Single Images)

### 1. Distortion Correction in Test Images

As described in section before are Test images provided by Udacity generated again. Left is original and right corrected image. At first look, it can be little hard to see the difference between the images.





## 2. Binary Images – Lane Lines Extraction

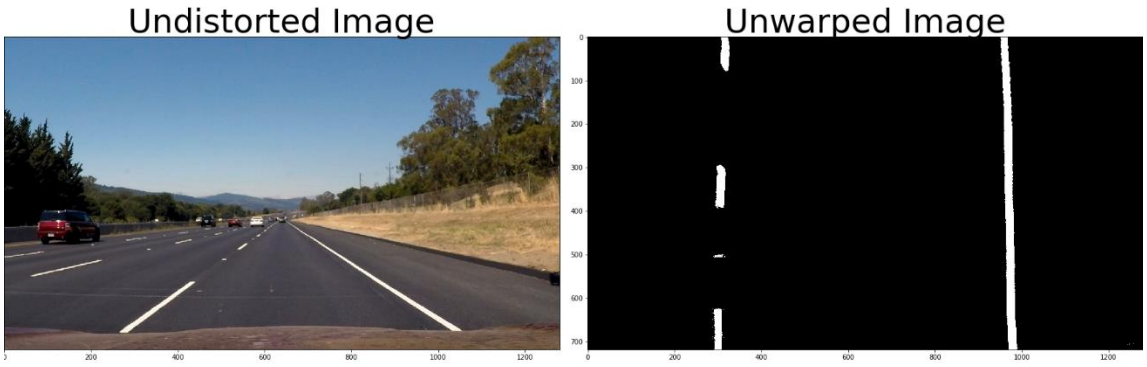
0. Perspective Transformation:

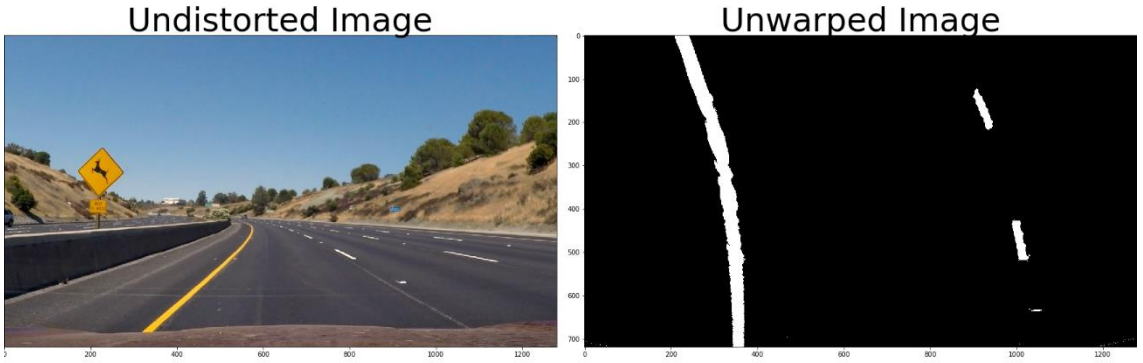
| Source | Destination |
|---|---|
| 273,680 | 300,720 |
| 1046,680 | 980,720 |
| 585,455 | 300,0 |
| 700,455 | 980,0 |

1. Absolute Sobel thresholding (cell 5): Function that applies Sobel x and y to undistorted image, then takes an absolute value and applies a threshold.
2. Magnitude of the gradient (cell 6): not only x gradient, but also y gradient takes into calculation to undistorted image
3. Direction of the gradient (cell 7):
4. HLS S-Threshold (cell 8): As suggested or shown in examples in Lesson 15 Section 28 I picked up S channel and applied thresholding to undistorted image
5. LAB B – Threshold (cell 8): Try to extract yellow lines with.

6. Color Threshold (cell 8): An image is converted to HLS Value and then applied 2 mask (white and yellow) to undistorted image to find lane lines. This is not a binary image!

Below is shown the applied thresholding to undistorted image with 3 examples. In Notebook all Test Images are shown. (cell 13)



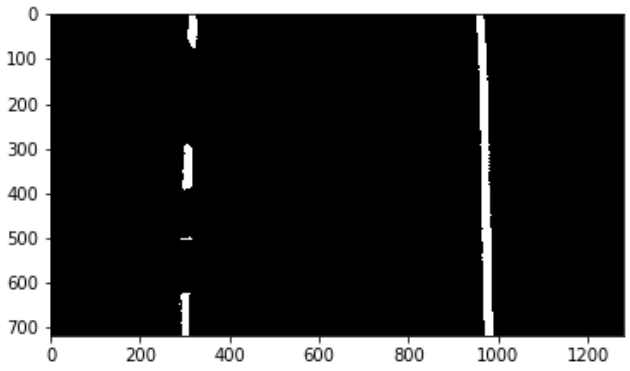


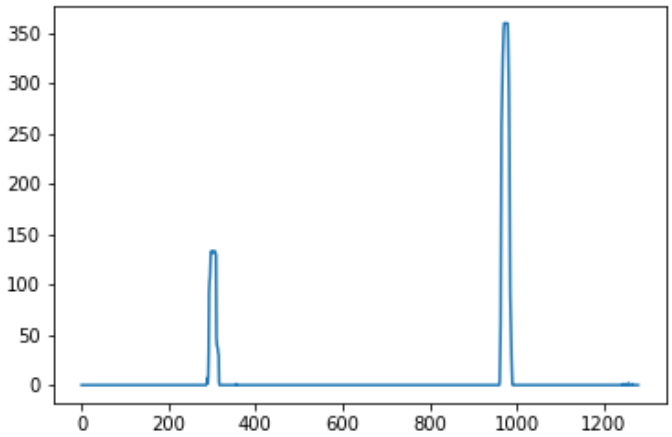
## 3. Applying perspective transformation.

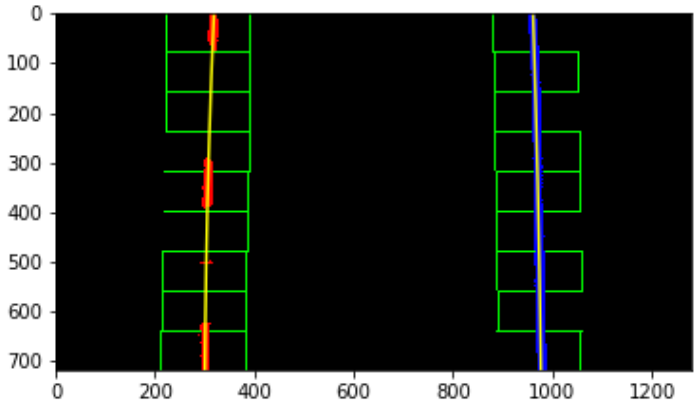
It is shown in the section above. You can find related code in cell 10.

## 4. Lane line pixels have been identified in the rectified image and fit with a polynomial.

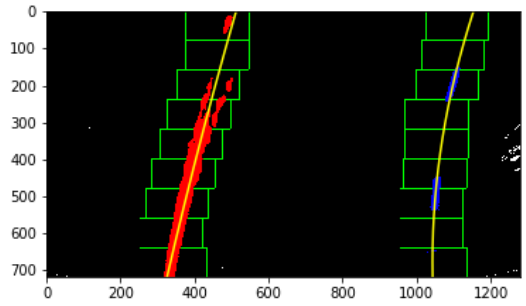
The functions *window_search-polyfit using windows* and *polyfit_using_prev_fit*, which identify lane lines and fit a second order polynomial to both right and left lane lines (cell 15-16), uses a histogram extracted from binary image. The most prominent peaks in this histogram will be a good indicator of the x-position of the base of the lane lines. Width of windows and minimum pixel are chosen relatively 85 and 50. Below are shown the binary image, its histogram and result from window search

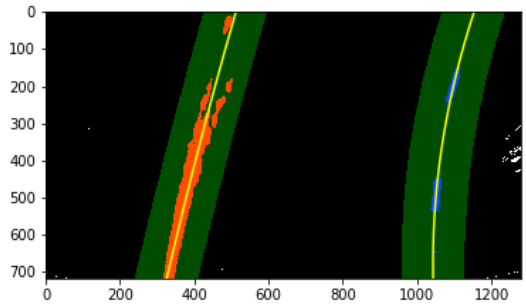


The same thing is doing by using the function polyfit_using_prev_fit from the fitted point/polyfitted x interceptes of lines of the previous frame. Basically, I am not doing the blind search again. Left found by using sliding windows, right found by without windows search again.

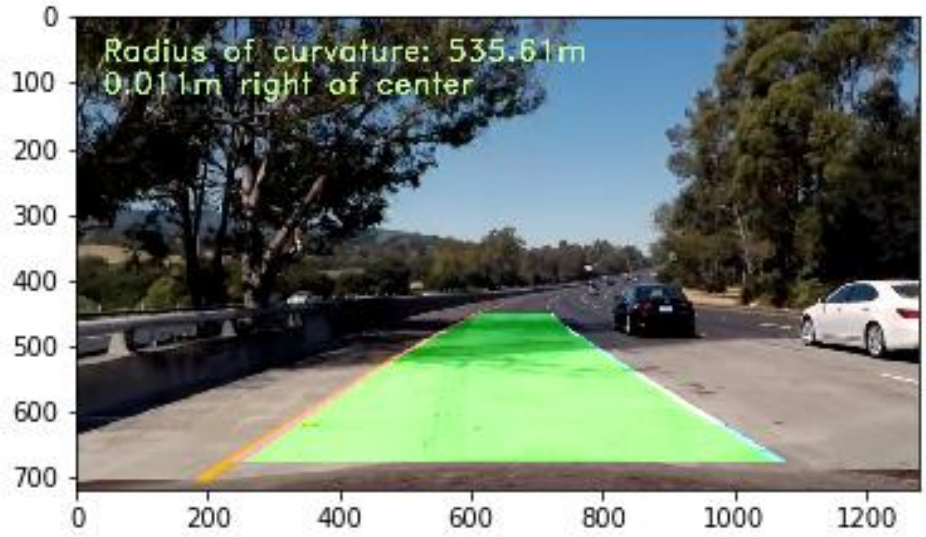



## 5.Radius of Curvature and lines projection

The radius of curvature of the lane lines was calculated from the polynomial fit parameters using the formula provided in the course material. The transformation from pixels to real-world metric values was included. The position of the vehicle with respect to center of the road was calculated by taking first intercepts of the two lanes and calculating its distance from the center of the image, assuming that the camera should be at the center of the car.

2 example images warped back to the front view with texted curvature and center line position information is provided below.




## Discussion:

My pipeline can not be used at his moment for challenge video or harder challenge video.

I need to work on more accurate implementation of thresholded binary images like shadowed, light reflection, heavy brightness changing on images in harder challenge video or light white lines in challenge video. In real world example like snowy day my pipeline will not work. I hope to revisit to develop more robust and independent pipeline.