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.

Camera Calibration

1. Briefly state how you computed the camera matrix and distortion coefficients. Provide an example of a distortion corrected calibration image.

Camera Calibration is much required for a self-driving car for lane detection as it will be working on the images captured from cameras and identifying the object in our case lane marking. Here for camera calibration we are using cv2 package method `calibrateCamera().This method takes image point that’s the representation of points of the image on a flat 2D surface and object point that’s the point on the real object.`

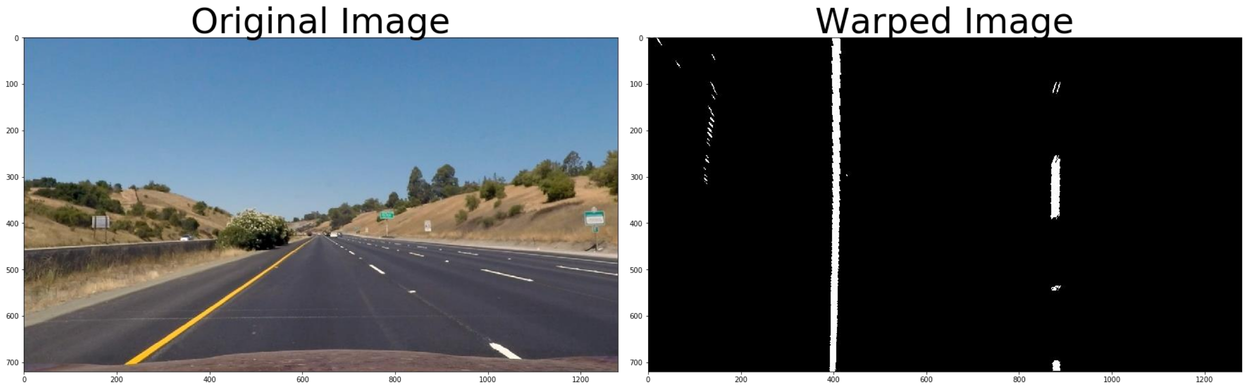
2. Apply a distortion correction to raw images.

Most of the image that’s captured from the camera’s get distorted. It becomes very difficult to predict the correct orientation of the image in that case. So we need to undistort the distorted image. For this we will use another method present in cv2 package that’s undistort().

Use color transforms, gradients, etc., to create a thresholded binary image.

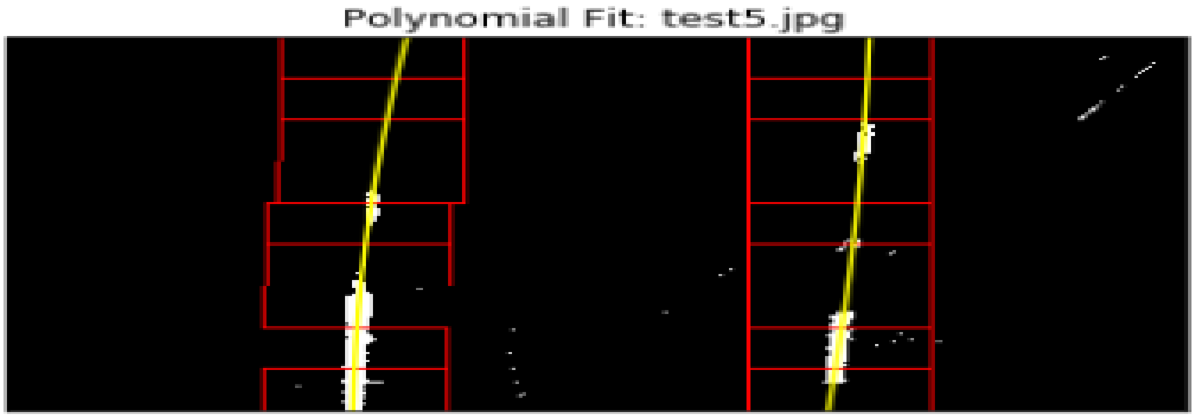
3. Describe how (and identify where in your code) you performed a perspective transform and provide an example of a transformed image.

Changing the perspective of the image orientation provide us a lot of information and allows to efficiently see the object. We can select points on the image and the destination points and we can transform the perspective using another method present in cv2 package i.e getPerspectiveTransform().



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

First of all we are using Histogram plotting to find the 2 peak values these can be taking as two base point then we can use sliding window algorithm to plot the line. This need to be done for one frame after that we will be using margin and relative position of line pixel in for previous frame to find values in the region. This makes the calculation less and improves the performance significantly. Implementation is in code



Polynomial Fit: test5.jpg

5. 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.

We have a mathematically equation that allows us to calculate the radius of curvature code is present in cell 8.

$$R_{curve} = \frac{[1+(\frac{dx}{dy})^2]^{3/2}}{|\frac{d^2x}{dy^2}|}$$

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

Pipeline (video)

1. 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!).

https://youtu.be/hPvqmAD_XxM

Discussion

1. 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?

My pipeline works fine in challenge video with some shake but suddenly when car goes inside the tunnel my marking disappears. I think we are not considering external factors like tunnels and other where due to shadow intensity of light less becomes and driver less car cause trouble.