

Advanced Lane line detection

The goal is to write a software pipeline to identify the lane boundaries in a video, utilizing the following steps:

- 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.

In my pipeline I created different classes that implement different functions as below:

Note: I used a flag to switch between whether I am using separate images or video stream.

- `calibrate_distort.py`: This class is for calibrating the camera and correct distortion
- `imgvidprocessing.py`: This class is for generating binary image after applying HLS and `abs_sobel_threshold`, then apply perspective transform.
- `Findlanes.py` : This class is for detecting lane lines and it has `search_around_poly` function that is used in case of videos where we have successive frames and the probability of have same lane lines and curvature is high. It also included the smoothing steps and averaging the line pixels over n frames.
- `Linetrack.py`: This is to keep track with every line detected.

I attached a folder with test Images output and saved the output image in three stages, first after correct distortion, second after generating the binary image and finally after detecting the lane lines and draw them.

I attached also the processed output video.

All the code files are attached.

Reflection:

Smoothing could be enhanced by incorporating the lanes curvature as well. In the code I checked the distance between the left and right lines only.