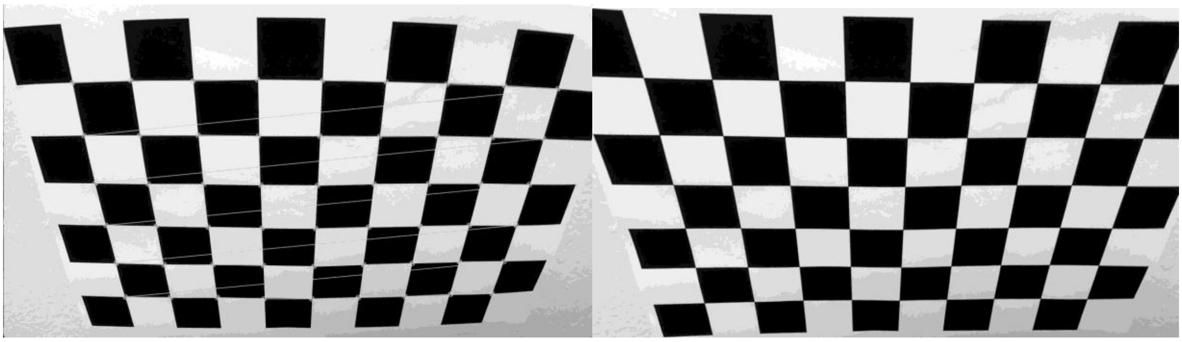
Advanced Lane Lines Detection

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 and Distortion Correction Test

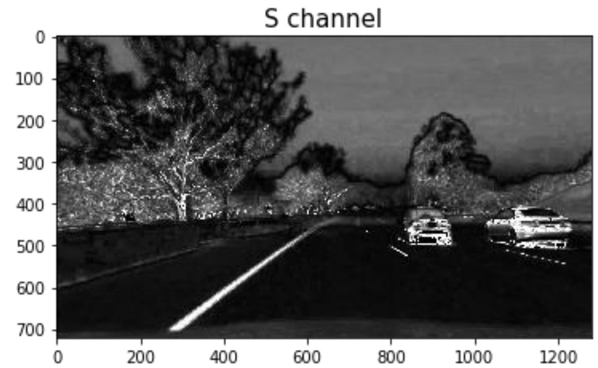
The very first step of detecting curved lane lines is camera calibration. Raw images taken from non-calibrated camera are distorted. Non image-degrading abberations such as pincussion/barrel distortion can easily be corrected using test targets. Chessboard images taken from different angles, helps to map all corner cordinates. Those cordinates are used to compute the camera calibration and distortion coefficients.



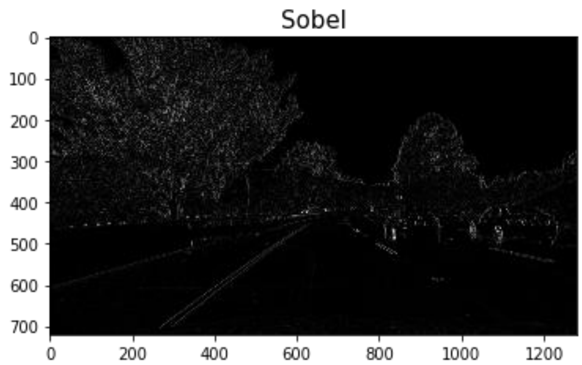
Detected corners on distorted image(on the left side), undistorted image(on the right side)

Color and Gradient Threshold

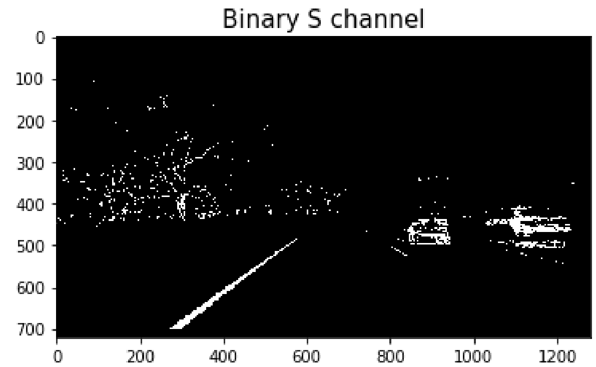
Simple edge detector like Canny algorithm or Sobel filter is insufficient to properly extract lane lines. If the surface of a road is very bright than the gradients will be too small and white lane lines will be not detected. Another way to extract lane lines is to use color threshold. This kind of approach works quite well for yellow lines. Combined color and gradient-base methods gives best results. For gradient threshold I used absolute value of Sobel "X" derivative and for color threshold I used S-channel from HLS color space.

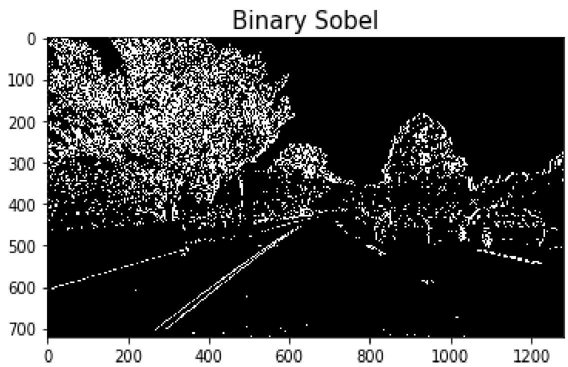


S channel of HLS image space

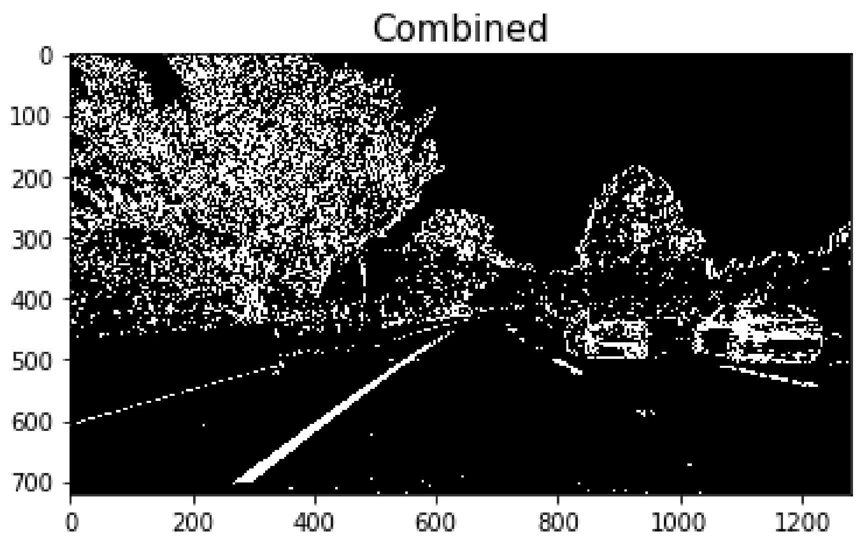


Egdes detected by Sobel operator





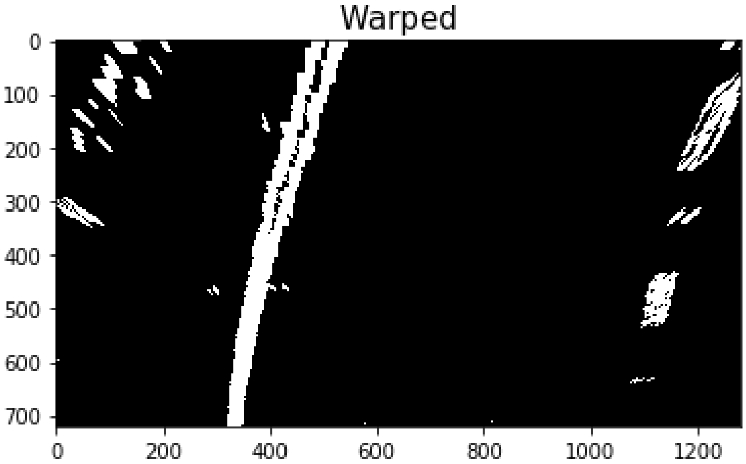
Binary images of s-channel and Sobel



Combined color and gradient threshold

Bird's-eye Transform

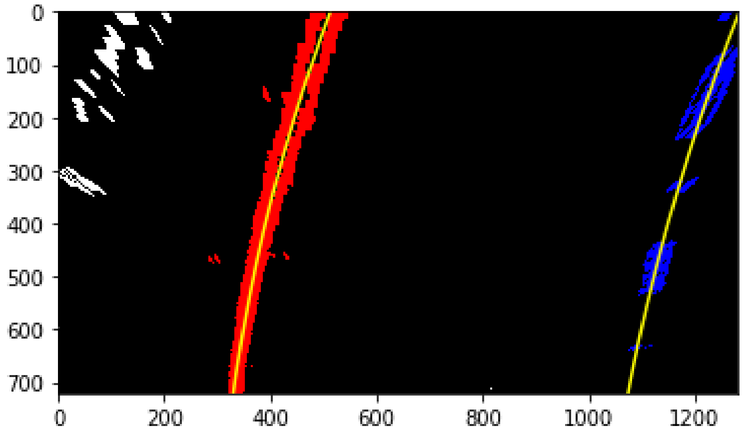
Detecting curved lane lines is much easier if image is transformed to "bird's-eye view". This kind of image represetation gives in most cases explicit information about lines shape(are they parallel, curved or straight). To avoid noise such as edges of a trees or sky, the empirically-picked points of ROI was chosen. cv2.getPerspectiveTransform and cv2.warpPerspective functions provide inverse perspective matrix and warped image.



Warped image of chosen ROI

Get Lane Lines and Determine Curvature

Next step is to locate left and right lane line. Possible method is to compute histogram and locate two highest peaks(left from center and right from center). The sliding fixed-sized windows are placed around the line centers(cordinates of peaks), to find and follow the lines up to the top of the frame. Positions of all pixels which belong to left or right line are used to fit second order polynomial. It is necessary to compute radius of lines curvature. We can get the radius of curvature in an arbitrary point using the following equation R = [(1+(dx/dy)^2)^3/2]/|d^2x/(dy)^2|. Radius of curvature was scaled from pixel space to real world space by using two coefficients: one for "x" and one for "y". The coordinates of points at the bottom of the frame which make up the curve have been used to calculate vehicle position within the lane.



Detected both right, left lane line have been used to fit second order polynomial,approximate curvature of a lines and vehicle position with respect to center

Draw region between lane lines

Last step is to transform image from warped space. Measurements of the line positions in warped space and inverse transform matrix help to project lines onto original image.

Visual display of the lane boundaries and numerical estimation of lane curvature and vehicle position.

Conclusion

Color and gradient threshold methods are insufficient for noisy images, especially images where the color of road surface and lane lines are almost the same. In such cases gradients and changes in color are too small. Also the model fails when the image contains shades, and it will fail under many circumstances such as harsh weather conditions (snow, rain, fog, etc.). One approach to solve that problem could be GPS system. Precise measurements could provide the necessary information, for steering system, about car position on the road. Hybrid system which uses both GPS and color-gradient threshold could be more robust and stable.