## ADVANCED LANE FINDING

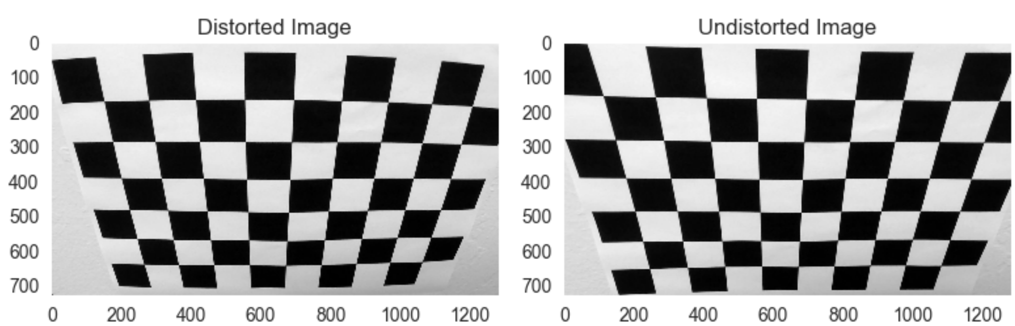
The following is a description of each step in a software pipeline designed to identify lane boundaries from a front-facing camera. The processing steps include:

- **Distortion Correction:** Correct image distortion produced by camera.
- **Image Thresholding:** Create a binary image that only includes lane pixels by processing color and gradient information within a region of interest.
- **Perspective Transformation:** Rectify the binary image to a "bird's eye-view".
- **Lane Segmentation and Curve Fitting:** Detect right/left lanes and fit boundary curves using information from previous frames when appropriate.
- **Curvature and Position Estimation:** Determining lane curvatures and vehicle position.
- **Visual Display:** Warping detected lane boundaries onto the original image.

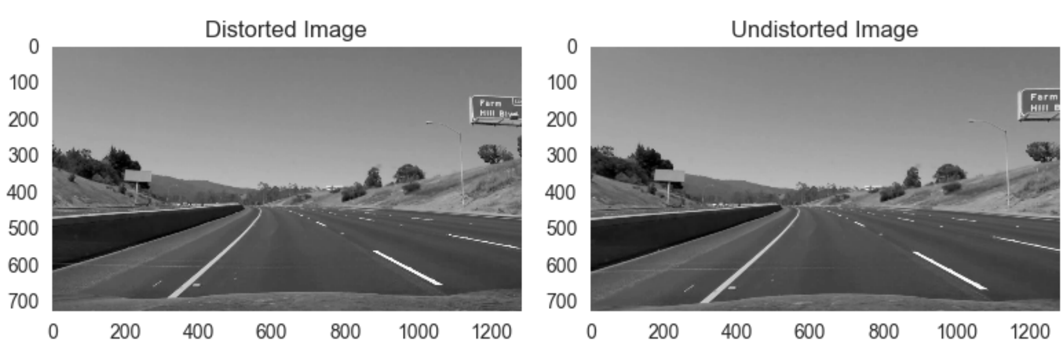
### DISTORTION CORRECTION

Correcting camera image distortion involves using calibration images to derive the camera matrix and distortion coefficients, and then using those parameters to undistort newly acquired images.

The images below illustrate a calibration image before and after correction. In the original image (left) one can see the curvature introduced by radial distortion at the top of the image. In the corrected image (right) radial distortion is drastically reduced.

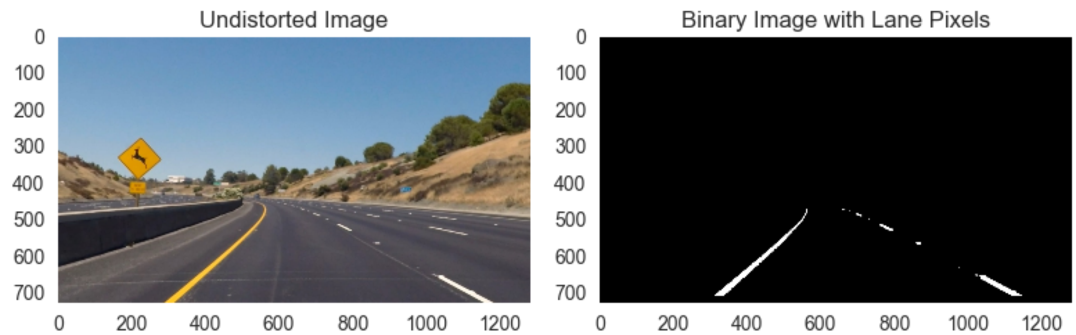



The images below illustrate correcting distortion in a frame from the project video. In the original image (left), the highway exit sign (upper right corner) appears slanted due to radial distortion. In the corrected image (right) the exit sign appears to be squarely facing the camera

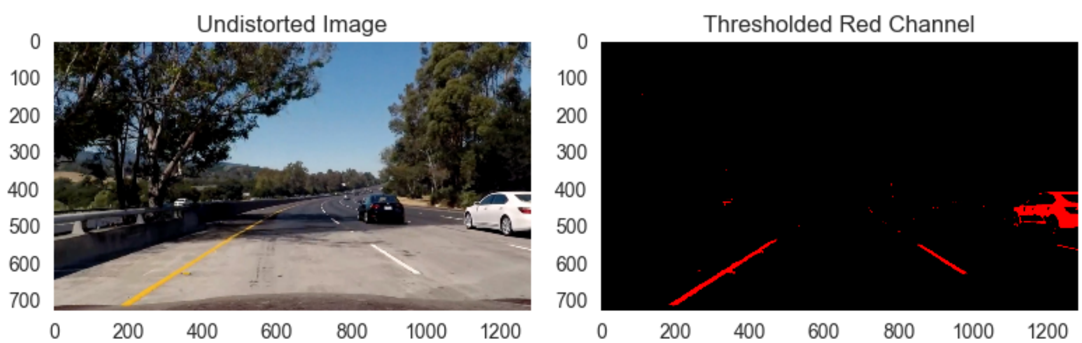


## IMAGE THRESHOLDING

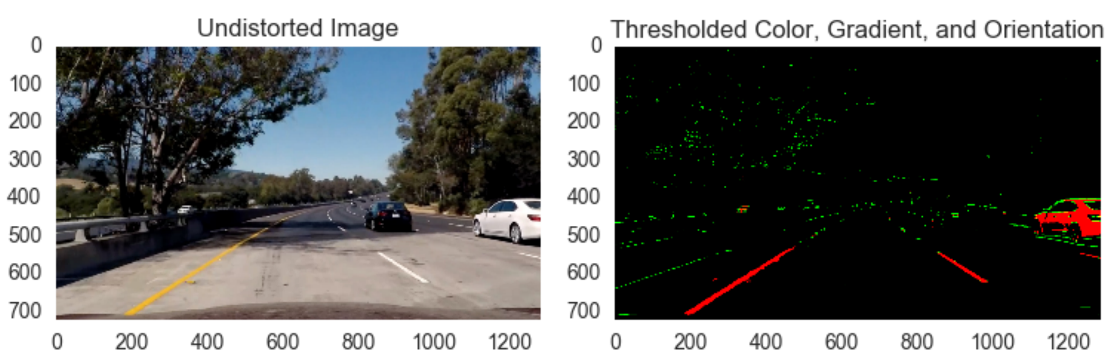
Once distortion is corrected, the image is transformed into a binary image where all pixels other than those belonging to lanes are zero. For example, the images below show a frame after correcting for distortion (left) and the resulting binary image (right). The binary image is obtained by thresholding the ***color*** and ***gradient magnitude*** and ***orientation*** within a ***region of interest***. Let's see how this works with an example.




The red channel contributes to the white and yellow lane pixels. So, retaining pixels with high red values should highlight the lanes. As an example, below is the input frame (left) and the result of only retaining pixels with red values greater than 210 (right). As can be seen, a significant fraction of both lanes is identified.




Thresholding the red channel fails to capture lane segments darkened by shadows. These segments can still be detected since they form distinct edges. Below one can see that retaining pixels with gradient magnitude greater than 85 and orientation greater than $\pi/4$ (green pixels) in addition to the pixels with red values greater than 210 (red pixels) allows us to recover more of the lane.



The final step is to only retain

## PERSPECTIVE TRANSFORMATION

## LANE SEGMENTATION AND CURVE FITTING

## CURVATURE AND POSITION ESTIMATION

## VISUAL DISPLAY