ENPM673
Project 2
Perception for Autonomous Robots

Project Group -

Amrish Baskaran (116301046)
Arpit Maclay (116314992)
Bala Murali Manoghar Sai Sudhakar (116150712)

Pipeline (Histogram for Lane Pixels):

1. Input Preparation
    a. Undistort the image
    b. Denoise the image
    c. Homography to obtain orthogonal view
2. Lane Detection
    a. Image thresholding
    b. Histogram of lane pixels
    c. Required lane extraction
    d. Find the curvature
    e. Get direction of turn
3. Lane marking
    a. Inverse homograph to display the detected image lanes
    b. Overlay lane area
    c. Display curvature

Input preparation

Fig: a) Distorted Image, b) Undistorted image

Undistort the image

The first step is to correct the raw image for distortion that arises due to camera lens, sensor and manufacturing defects. In case of pinhole camera model, radial distortion and tangential distortions are much pronounced than other several distorsion. All these distortions can be corrected and a near perfect image can be obtained by calibrating the camera and finding the intrinsic matrix. This intrinsic matrix is first multiplied to the image and all subsequent operations are performed on this image

Denoise the image

A common way to reduce noise is applying gaussian filter. This is a very fast, cheap and suitable operation considering other choices. Following filters are tried and discarded.

- Box Filter - Gaussian filter is robust and performs better generally for various types of noise.
- Median filter - This is costlier than gaussian filter since this filter is not separable. Also sorting operation is slower than addition and multiplication operations.
- Bilateral filter - This filter is a potential candidate since it preserves the edge pixels and- blurs the non feature pixels. Though this filter performs much better that gaussian filter, it is extremely slow and the video stalls when this filter is applied to each frame. Also we are not using edge detection in our pipeline. Hence gaussian filter is the good candidate for out purpose.

Homography to obtain orthogonal view

The image is a perspective projection of the real world objects. Since perspective projection does not preserve parallelism, it is necessary to transform the image to real world coordinates. Lane detection can be done without applying homography, but to find the curvature of the lanes it is necessary.

Homography procedure

1. Take 4 points on the image. The points along the lanes are potential candidates for consideration. When the car travels in a straight road, the lanes appear as trapezoid which in real world is a rectangle. This behaviour is exploited to find 4 point correspondence.
2. Find the homography matrix using the points collected above.
3. Warp the image to get orthogonal view of the road

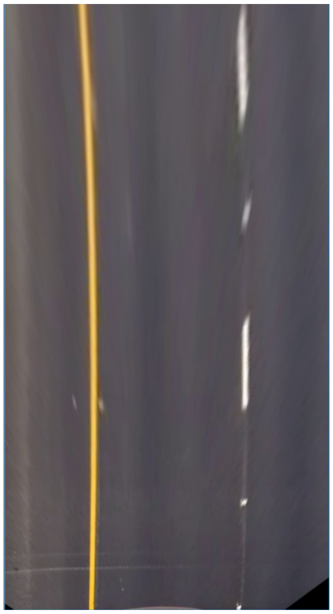


Figure after homography

Lane Detection

Once the orthogonal view of road is obtained, the next step is to do further processing on the image to extract the lanes. Below procedure is followed

Image thresholding

We know that the lane will be colored with yellow and white. Hence other features are masked and only the lanes are extracted using thresholding methods. HSL color space is giving better results that RGB color space since the range of white and yellow color are varying a lot and overlaps with other image features. We carefully tune high and low cutoff values of Hue , Saturation and Lightness parameters to extract only the lane pixels. Now we can comfortably move to binary image since all the features in the mask are important features and color differentiation is no longer needed. The image can be comfortably thresholded to for further processing.

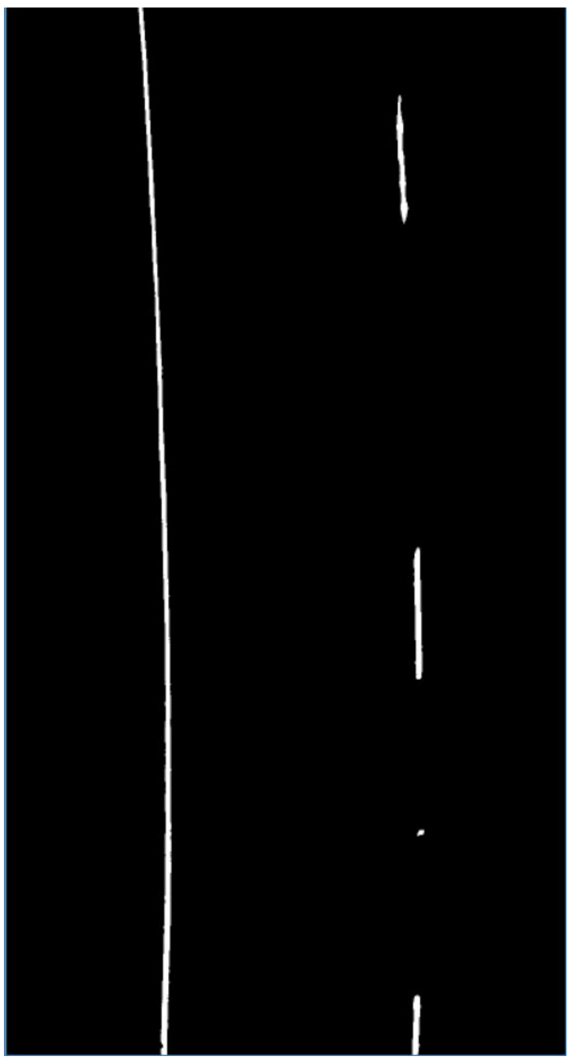
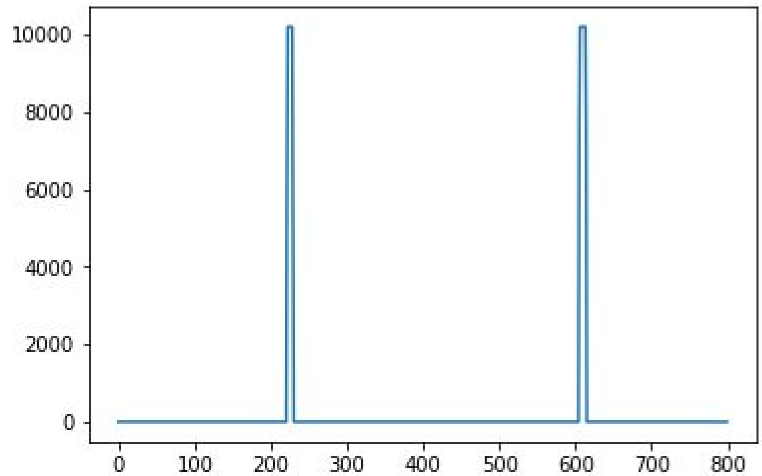


Figure- After color thresholding for white and yellow colors and Binary image conversion

Histogram of Lane Pixels

To extract the lane from the thresholded binary image without cropping the area of interest. Extracting the yellow lane is easier since it is continuous. But the white lane is tricky since it is discontinuous. We follow the below procedure to detect the right and left lane lines.

1. Split the image into 20 horizontal strips.
2. In each strip, count number of white pixels along each column. This gives the histogram as shown below.



3. Find the peaks in the histogram. Then take a window around the peak and find the white pixel indexe through which plie can be fit. A better approach would be to apply ransac and discard outliers rather than fitting a window. There might be different scenarios
    a. Two peaks - one on left and one on right - a perfect situation were each peak denotes one lane.
    b. Multiple peaks on the right half of histogram and one peak on left half of histogram - In this case right peak is taken as reference. We know that the distance between lanes is constant even in curves. We take the peak that matches this criteria.
    c. No peaks on right of histogram - in this case only left lane is extracted.

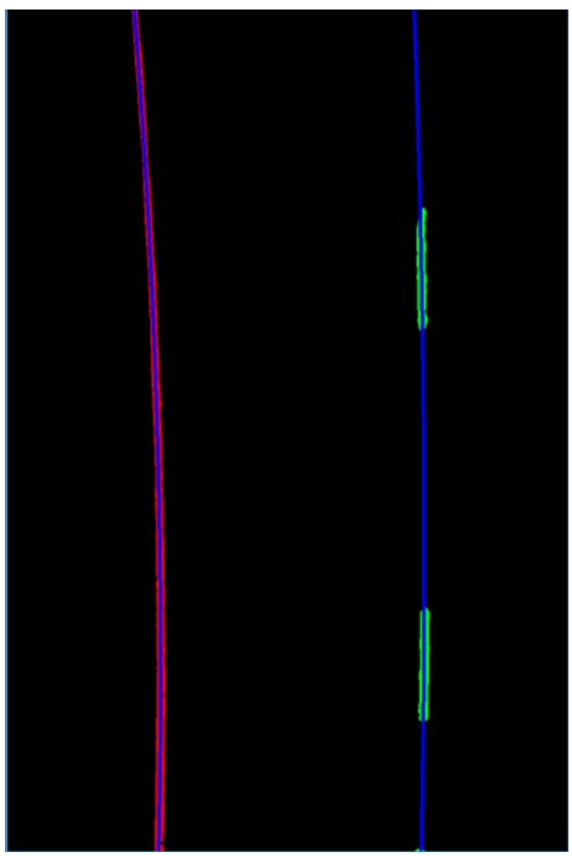
Required Lane Extraction

For the lane extraction, since now we have the peaks and the corresponding pixels corresponding lane pixels are taken. The program collects points from the left half of the plane and appends the points into a list. Similarly for the right lane it collects the points from the right half of the plane and appends the points into one more list.Finally to create the lane a polyfit function is used on the 2 lists of points such that the lane is extracted.

Find the curvature
For n number of points if a circle is generated to pass through all of them then the reciprocal of the radius of this circle is called the curvature of the curve.
The function to find the curvature of the set of points is given in
https://github.com/windowsub0406/Advanced-lane-finding/blob/master/finding_lines.py

Lane Marking

Inverse homograph to display the detected image lines
The mathematical relationship between two planes is defined as a homography matrix H. The matrix H can be expressed as H = sMR.
A calibration-free approach is proposed to iteratively attain an accurate inverse perspective transformation of the road, through which a "birds eye" view of the target road, as well as parallel lanes, can be gained.This method obtains 4 endpoints of a pair of lanes in perspective image by our lane detection algorithm first. Considering that the road is flat, we project these points to the corresponding points in the IPM view.



Overlay lane area

The program takes the initial point from the left lane and appends all the points coming in the lane in clockwise direction continuing through the horizontal connection between the lanes till the right lane right bottom most point. The lanes in the real world space are then plotted using a polygon fill on to an overlay image with a transparency factor but not before unwarping the overlay back into perspective view so that it matches the camera view.

Display Curvature

After the lanes and the lane area are determined as above they are then added to the undistorted image by using weighted average.