ENPM673: PERCEPTION FOR AUTONOMOUS ROBOTS

Project 2

BHARADWAJ CHUKKALA
118341705

PROBLEM 2: Straight Lane Detection

ANSWER:

Procedure followed to tackle this problem:

- Convert the video frame from BGR (blue, green, red) color space to gray. This helps to eliminate handling the complex 3 channel image and dial it down to a single channel image.

Original frame of road

- To eliminate any noise that may hinder the edge detection process, the image is smoothed out using Gaussian blur function. Depending on the size of the kernel, the balance between elimination of noise vs loss of details is determined.

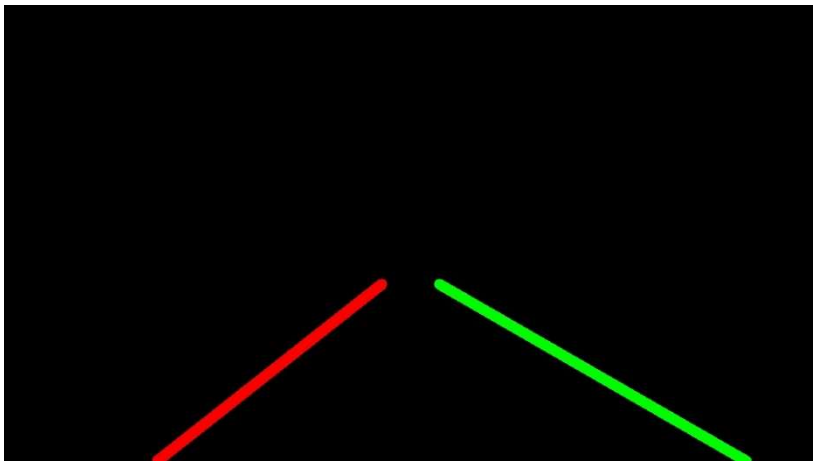
- Edge detection (we can use canny): Perform Canny edge detection on the blurred image to detect sharp discontinuities in the pixel intensities along the video frames. Sharp changes in intensity from one pixel to a neighboring pixel means that an edge is likely present. We want to detect the strongest edges in the image so that we can isolate potential lane line edges.

Canny edge detection

- We select a Region of interest that is the lane lines, we can do this by understanding the camera and the car position. Here we chose a trapezoidal shape as the lanes tend to meet at the vanishing point in the frame of the image, trapezoidal region of interest is chosen.

Region of interest has been chosen

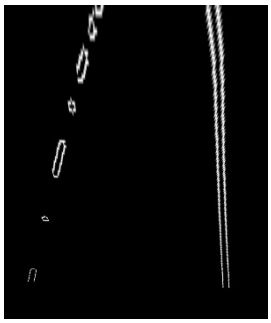
- Perform the bitwise AND operation to reduce noise in the image caused by shadows and variations in the road color. Lane lines should be pure in color and have high s channel values. The bitwise AND operation reduces noise and blacks-out any pixels that don't appear to be solid colors (like white or yellow lane lines.) Thus by performing bitwise_and, we perform AND operation for every element in both the provided images, thus extracting only the desired region of interest.
- After getting the region of interest, we perform hough transformation on the edge detected image. Hough Transform: The Hough transform is a popular feature extraction technique that converts an image from Cartesian to polar coordinates. Any point within the image space is represented by a sinusoidal curve in the Hough space. In general, hough transform gives a line which is a set of points representing the slope (m) and y intercept (b) which satisfy a point in cartesian coordinates. The intersection of these lines in hough space satisfies all the points in cartesian space. But, to avoid the case where a point with slope being zero, the cartesian coordinates are converted into polar coordinates and then the system of lines are drawn in hough space. Since we are dealing with polar coordinates, Hough space has sinusoidal waves in it. Therefore, Using Hough Transform gives the line coordinates of the white pixels from the edge detected image with most intersections in hough space, yielding straight lines mapping the solid and dashed lines.



Hough lines plotted in the Region of interest

- Even though the lines are plotted, the distinction between left lane and right plane has not been made. To do so, Homography is performed on the region of interest to get it in a bird's eye view perspective.

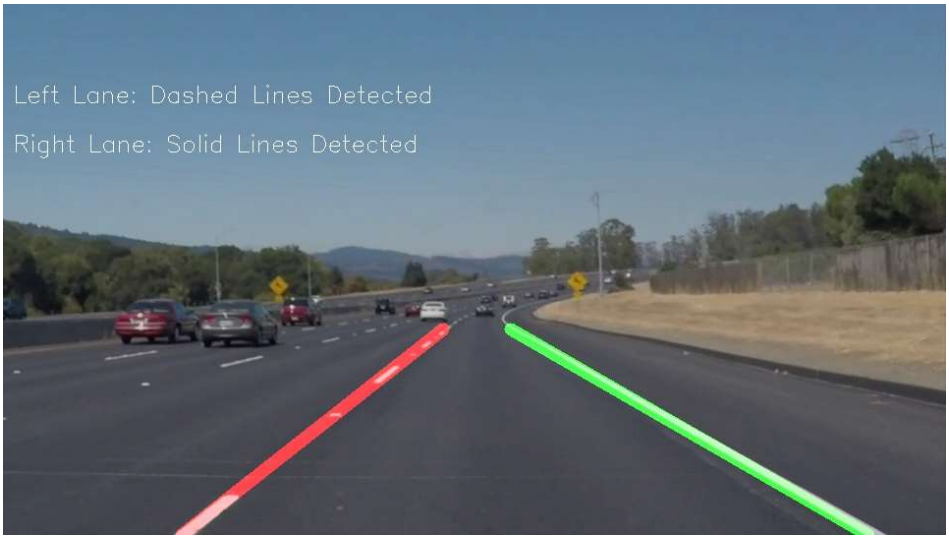
- Now, histogram is analyzed from the warped image whose graph contains two spikes with the highest spike representing the solid white line and the second highest spike representing the dashed line on the road.



Warped image

- Once we have identified the pixels that correspond to the left and right lane lines, we draw a best-fit line through the pixels using the slope values of the pixels in each spike with different color to identify each of them.

- This algorithm works for both the normal video and the flipped video.



Final output on normal video

Final output on the flipped video

PROBLEM 3: Lane Detection and Turn Prediction

ANSWER:

Brief overview before getting into the problem

Procedure Followed to tackle this Problem:

- Convert the video frame from BGR (blue, green, red) color space to HLS (hue, saturation, lightness). The HLS color space is better than the BGR color space for detecting image issues due to lighting, such as shadows, glare from the sun, headlights, etc. We want to eliminate all these things to make it easier to detect lane lines. For this reason, we use the HLS color space, which divides all colors into hue, saturation, and lightness values.

Original Frame

- Perform binary thresholding on the S (saturation) channel of the video frame. Doing this helps to eliminate dull road colors. A high saturation value means the hue color is pure. We expect lane lines to be nice, pure colors, such as solid white and solid yellow. Both solid white and solid yellow, have high saturation channel values.

Saturated image

- Edge detection (we can use canny): Perform Canny edge detection on the S (saturation) channel of the image to detect sharp discontinuities in the pixel intensities along the video frames. Sharp changes in intensity from one pixel to a neighboring pixel means that an edge is likely present. We want to detect the strongest edges in the image so that we can isolate potential lane line edges.

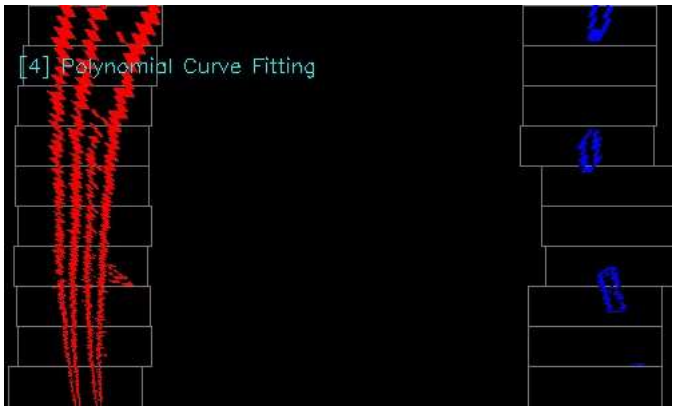
- We select a Region of interest that is the lane lines, we can do this understanding the camera and the car position. Here we chose a trapezoidal shape as the lanes tend to meet at the vanishing point in the frame of the image, trapezoidal region of interest is chosen.

ROI

- Perform the bitwise AND operation to reduce noise in the image caused by shadows and variations in the road color. Lane lines should be pure in color and have high s channel values. The bitwise AND operation reduces noise and blacks-out any pixels that don't appear to be solid colors (like white or yellow lane lines.)

Warped image

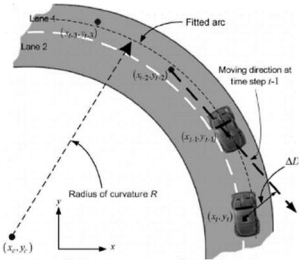
- The next step is to use a sliding window technique where we start at the bottom of the image and scan all the way to the top of the image. Each time we search within a sliding window, we add potential lane line pixels to a list. If we have enough lane line pixels in a window, the mean position of these pixels becomes the center of the next sliding window.



Sliding window and curve fitting

- Once we have identified the pixels that correspond to the left and right lane lines, we draw a polynomial best-fit line through the pixels. This line represents our best estimate of the lane lines.
- We calculate the radius of curvature for the lane lines that have been detected. Using the polynomial curve fit equation. The formula for the calculation of radius of curvature of lanes is given by:

$$\text{Radius of Curvature, } R = \frac{(1 + (\frac{dy}{dx})^2)^{3/2}}{|\frac{d^2y}{dx^2}|}$$



- Then we calculate the center offset for the sliding window, doing this will give us the necessary data required for prediction, along with the radius of curvature. We can predict the turn direction using the offset, if the offset is in the positive x direction, the lane is curved right and if the offset is in the opposite direction then the lane is curved in the left direction.

Final output

Drive link:

https://drive.google.com/drive/u/0/folders/1rlzJxrkSt9MmnpSy5gOn-WSxS7RBETWY