Perception for Autonomous Robots
ENPM673

Project 2

Name: mano srijan battula

UID : 118546490

Date: 04/04/2022

Problem 1 – Histogram Equalization:

In this project we aim to improve the quality and lighting conditions of the image sequence provided . We will be using histogram equalization which will distribute the intensities of the image and make them evenly spread. I have stitched the all the images into a single video file and used this video. Then I have converted the video into LAB color-space. After that equalized and adaptive equalized ‘L’. then merged them with ‘A’ and ‘B’.

Fig1: original stitched image

In Fig 2, it is easy to notice the upper region of the image can show more details than original image. The darker region is also brighter than original image.

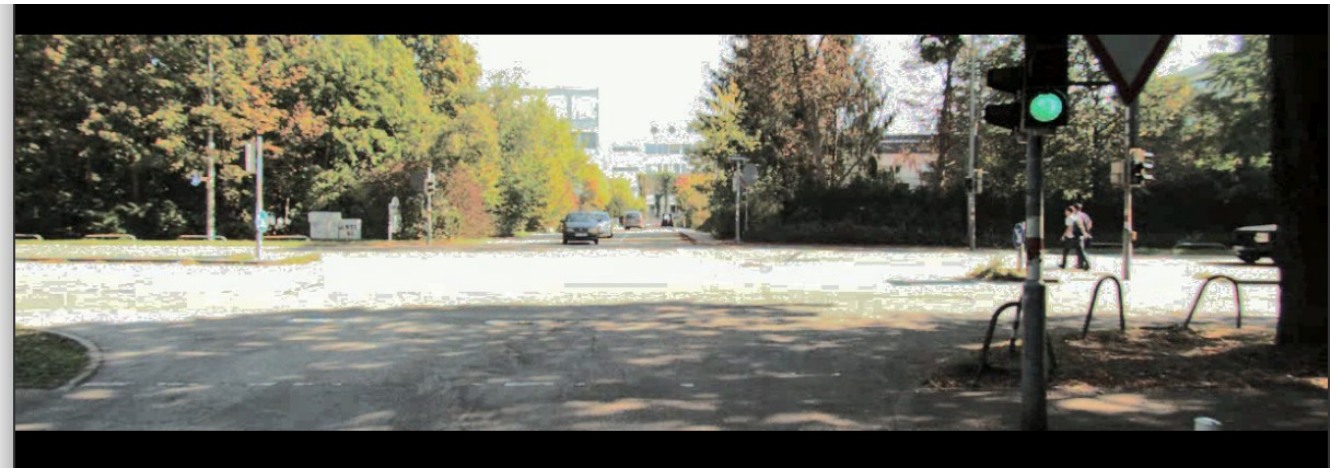

fig 2: histogram equalization

For adaptive histogram equalization, it is based on histogram equalization. This method divides image into several pieces and utilize histogram equalization on each piece.

fig 3: adaptive histogram equalization

Compared to the Normal Histogram equalization, the Adaptive Histogram Equalization works a lot better .We can view more details on the trees, people, and the car in the background.

Problem 2 – Straight Lane Detection:

we are required to classify the dashed and solid lines on the road. Green is for solid line and red is for dashed line. In order to do this following steps were taken,

1) The video is converted into gray scale

2)Then Gaussian blur is applied to smoothen the image, canny edges is performed on it to find edges,and a copy of canny is created.

3) a black image which acts as mask is created which will be imposed on the video thus giving only the required lanes which needs to tracked.

4) later, lines are detected using inbuilt Houghlines function is algorithm which detects lines then we draw the lines.

5) To distinguish the lines by deriving the gradient (i.e slope) of any left or right lane line: * left lane: as x value (i.e. width) increases, y value (i.e. height) decreases: slope must thus be negative * right lane: as x value (i.e. width) increases, y value (i.e. height) increases: slope must thus be positive thus we can separate the lines into right and left one.

6) The using the defined draw lines function we create two lines with red for dashed line and green for solid line.

Fig 4: original

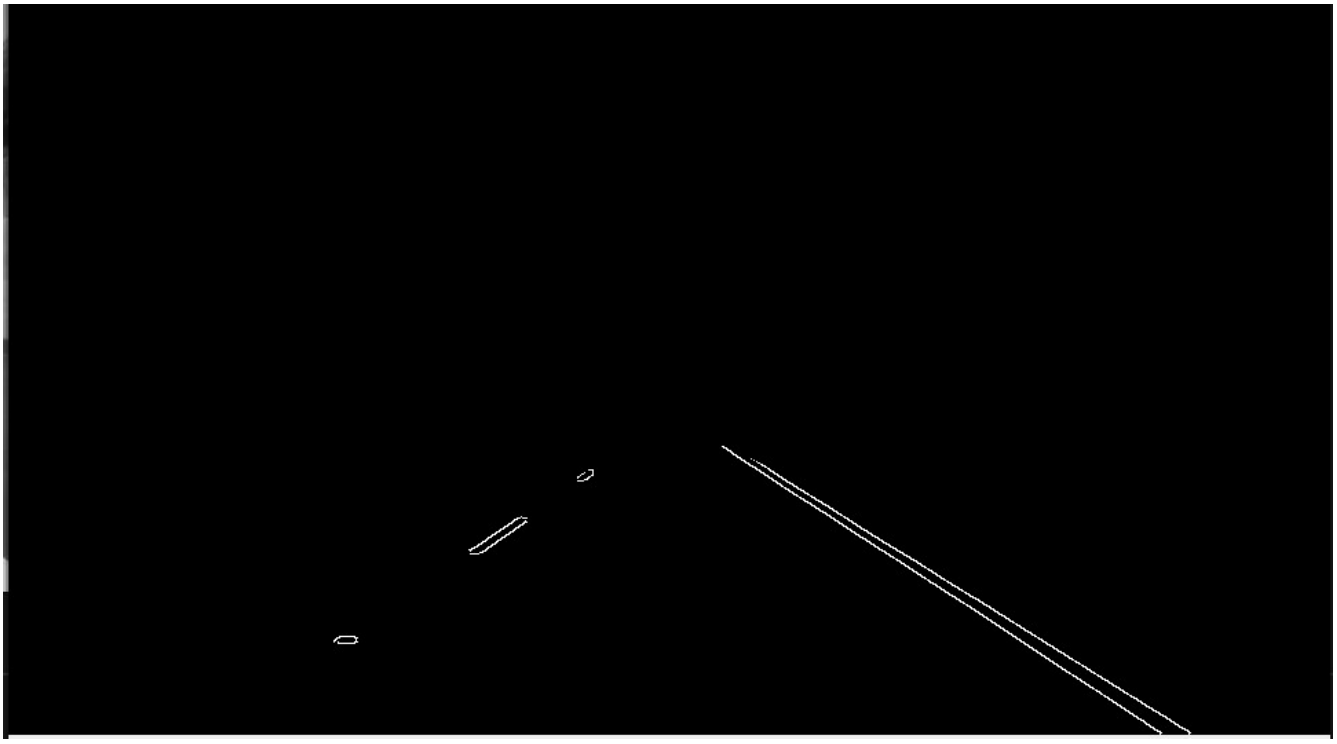

fig 5: after threshold

fig 6: lane detection

Problem 3 –Predict Turn:

we aim to detect the curved lanes and predict the turn. There are yellow and white colored lanes in video for predicting turn.

Steps performed are:

1. Apply filter to show yellow and white colors only and ignore unnecessary edges

2. we convert the image to grayscale then we mask the video with a region in which the lanes are in.

3.Used Gaussian Blur to smooth the image then thresholed the image from 150 to 255 and select the interested region.

4.Then perceptive transform is used this function calculates a perceptive transform from the four pairs of corresponding points.

5. later, the filtered image is warped, with perceptive transform this gives the birds eye perspective and then converted into then blur and threshold it.

6. A series of filters are applied to improvise the lane quality(Gaussian blur followed by thresholding).

7. Line fitting is done using polyfit function and the lane projection is superimposed over the original image.

8. Find the radius of curvature to estimate the turn

Fig 7: Input lane frame

Fig 8 : Filtered input

Fig 9: Masked(ROI)+ Threshed Frame

Fig 10 : Birds Eye perspective)

Fig 11: Turn Super Imposition