ENPM 673

Perception of Robotics

Project 2

By

Mothish Raj Venkatesan Kumararaj

Logo

Description automatically generated

# **Question 1:**

**Graphical user interface, text, application

Description automatically generated**

## **Process:**

**Histogram:**

* Get all the image from the adaptive\_hist\_data folder.
* Read the given image using OpenCV.
* Flatten the image in 1D array.
* Calculate the histogram graph for each color gradient from 0 to 255.
* Get the cumulative sum for each intensity in histogram graph and normalize it.
* Replace the position in the flattened array of the image with the corresponding cumulative sum.
* Reshape the image

**Adaptive Histogram:**

* Read the given image using OpenCV.
* Flatten the image in 1D array.
* Separate the image into small grid.
* Apply the histogram for each grid separately
* Replace the position in the flattened array of the image with the corresponding cumulative sum for each grid.
* Reshape the image

## **Results:**

**Input Image:**

A picture containing text, tree, outdoor, street

Description automatically generated

**Histogram Image:**

A picture containing text, outdoor

Description automatically generated

**Adaptive Histogram Image:**

A picture containing text, mammal

Description automatically generated

# **Question 2:**

**Text

Description automatically generated**

## **Process:**

* Read the video and get separate frame.
* Convert the image to gray scale and mask the image to specific coordinate to get only the road.
* Convert the masked image to binary format.
* From the binary converted image, do Hough line transform to get the line coordinates.
* The lines received is ordered from longest to short so get the slope of the longest line.
* Check the signs of the slope of the longest line with other lines.
* If the sign is same, then they belong to solid line else it belongs to dashed line.
* Fill green for solid line and red for dashed line.

## **Results:**

**Input Image:**

A picture containing text, road, way, scene

Description automatically generated

**Masked Image:**

A picture containing chart

Description automatically generated

**Binary Masked Image:**

Chart

Description automatically generated with medium confidence

**Output Image**

A road with cars on it next to a body of water

Description automatically generated with medium confidence

## **Link for output video:**

https://drive.google.com/file/d/1OHV8\_M0kzU-1JKFEya9jfW6A0WmcJLFG/view?usp=sharing

# **Question 3:**

Graphical user interface, website

Description automatically generated

## **Process:**

* Read the given image using OpenCV.
* For each frame do binary and canny edge detection and combine the two outputs for better results.
* Specify the coordinated points to perspective warp the threshold image to get only the lanes.
* Get histogram for ones in each pixel of the first half and second half of the image in x direction separately.
* The maximum intensity of the left and right half gives us the lanes location along x
* Add margin of 100 to get the start and end of the left and right lanes.
* Form a non-zero matrix for x and y direction of the image
* Separate the image into 9 grids for the y direction and obtain the non-zero points on the left and the right.
* Concatenate the left and right land index points and put the index in the non-zero matrix in x and y direction to get the coordinate position.
* Use Polyfit to get the constants of curve equation and calculate the variable.
* Convert the pixels to meters and multiply the conversion to the variables and constants to get left and right lanes fit.
* Get the warped image with marking of the lanes for the output.
* The radius of curvature can be calculated using the formula.

Graphical user interface, text

Description automatically generated

* Use fill poly to draw the marking on the lane and warp it according to the original image and use added weightage
* Add the radius of curvature to the image and get the final image
* Store the final image for each frame in a video and display it in the console.

## **Results:**

**Input Image:**

A screenshot of a video game

Description automatically generated with medium confidence

**Threshold Combined Image:**

Diagram

Description automatically generated with low confidence

**Perspective Warped Image:**

Chart

Description automatically generated

**Warped Image with lane marking:**

A picture containing chart

Description automatically generated

**Warped image with fill poly:**

Chart

Description automatically generated

**Warped image to input size with fill poly:**

Chart

Description automatically generated

**Final Image**

A screenshot of a video game

Description automatically generated with medium confidence

## **Link for output video:**

https://drive.google.com/file/d/19ir8WMhl9TR8-uBay-YFlS10FwGCHrZG/view?usp=sharing

# **Problems Faced:**

**Question 1:**

In question 1 when looping through the grid, I was reusing the code used in AR tag and since it was a square the loop did not have any issue. But for the given problem, it was a rectangular image so the divisor of the x and y length was used instead.

**Question 2:**

Initially, the left and right lines were separated by using positive and negative slope which was inefficient. Then the longest line was detected according to the sign when longest line and any line in the image is multiplied, the left and right lane were separated considering that the longest line lies in the solid lane.

**Question 3:**

For detecting the lanes, the logic used in question 2 was tried initially which did not render the required output. So, histogram of the threshold image was taken in the left half and right half of the image along x axis and the highest intensity of 1 on either half gave the position of left and right lanes and margin was added to it.

To get a better threshold image, I combined canny and threshold method which gave an efficient output.