Steering Assist for Drivers based on Road lane Articulation

VIT University, Vellore, Tamil Nadu, India

Abstract. Today Road Accidents are a major cause of deaths .The main reason for Road accidents is over speeding and getting lost of lanes thus leading accidents ,forms one of the main causes of impermanence, also have an economic impact of the owners of vehicles.Most of the automotive industries are developing technology to reduce risk in vehicles. In this way, lane detection systems is important, because from this data is possible to determine risk situations hence reducing accidents. Here we propose a technique for lane detection, based on image processing, which allows identifying the position of lateral or partial lanes. The idea is based on converting the given image to gray scale image and then tracing the white lines of lanes and getting to a point where maximum of intersections occur and therefore tracing graphic lines hence assisting the driver based on the graphics thus implemented.

Keywords: Hough Transform · Gaussian Blur · Canny Edge Detection.

1 Introduction

Steering Assist system is one of the most important feature of the modern vehicles to ensure driver safety and decrease vehicle accident on roads. Apparently, the road lane detection on road boundaries detection is the complex and most challenging tasks. A vision system using on-board camera looking outwards from the windshield is presented in this paper. The system acquires the front view using a camera mounted on the vehicle and detects the lanes by applying few processes. The lanes are extracted using Hough transform through a pair of hyperbolas which are fitted to the edges of the lanes. The proposed lane detection system can be applied on painted lanes The proposed system does not require any extra information such as lane width and time to lane crossing The system can be applied under various situations of changing illumination, and shadows effects in various road types without speed limits. The system has demonstrated a robust performance for detecting the road lanes under different conditions.The main motive is Region of Interest(ROI) as compared to previous projects.

2 Related Work

Several different image processing algorithms have been developed such as Likelihood of Image Shape (LOIS) algorithm [2], B-Snake algorithm [3] and others

[1] using a feature-based method to extract features in an image (e.g. edges), with different techniques.

A feature-based method has the advantage to offer to researchers a large choice of the kind of techniques to be used in each step of the algorithm and then contribute on its optimization. However, it has two major problems. The first issue of lane detection is the amount of calculation required. Thus, for real-time application, the performances and the processing time of the algorithm proposed above can be improved by using other operators for edge detection on the one hand and for line tracking on the other hand. The second issue is the difficulty of identifying lane boundary lines, because a real road image can contain other candidate edges (e.g. noise edges). Thats why; it should be a judicious choice of an edge detection technique to reduce the number of edge points in the output image of this step. Indeed, the edges must be detected well for Hough transform to be efficient.

3 Proposed Method

We aim to develop a robust algorithm and ensure a real-time detection, because we are interested on a real-timeimplementation. For this purpose, our optimizations will concern image preprocessing, edge detection and line tracking steps. The purposes for the construction of the ROI (Region of Interest)are the reduction of processing time and the reduction of memory space required for each video frame. Moreover, we limit the number of memory access. Then, the RGB color image is converted to an intensity image and the rest of operations are performed on intensity image.. Finally,we threshold the intensity image to obtain road lines

Our Model Flow:

1. Grayscale image
2. Gaussian Blur
3. Canny Edge Detection
4. Region of interest
5. Bitwise AND
6. Hough Transform

3.1 GrayScale Image

Basically we convert the given image to grayscale image because it consists of black and white intensisty value no coloured values therefore reducing memory utilization because in real world memory plays a major role .Another major reason for conversion to gray scale being for faster processiong and execution because it is easy to process gray scale image as compared to coloured image.So for above reasons the given image is converted to gray scale image.

3.2 Gaussian Blur

The given gray scale image is passed through the gaussian blur, the main reason being i.e to reduce noise present in the image.In our case the reasons for noise being fog,sand,rain, water and etc other natural factors can result in making the image noisy i.e reducing the amount of fine deatails required for further processing, i.e it is important to pass the given image through gaussian blur i.e smoothening the image.

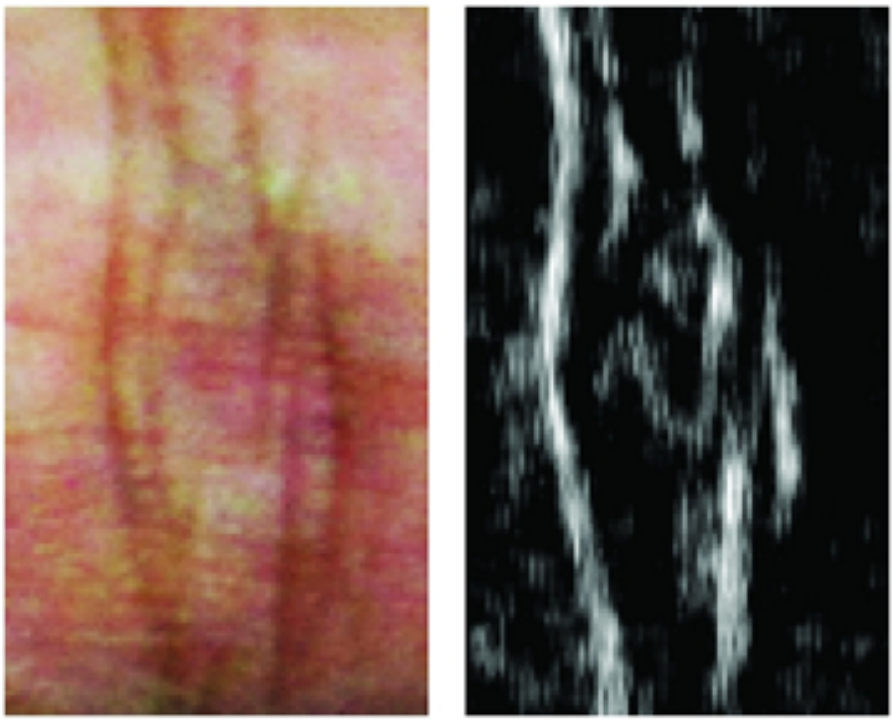

Fig. 1. Gaussian Blur

3.3 Canny Edge Detection

Basically an Edge is formed when we go through higher to lower intensities or vice versa.This edge detection technique is basically used to distinguish road lanes from other non usefull stuff hence reducing the memory utlilization as well as faster processing. In the edge detection system we are using Canny inbuilt function of cv2 library.The function automatically identifies edge when an image is passed through it.

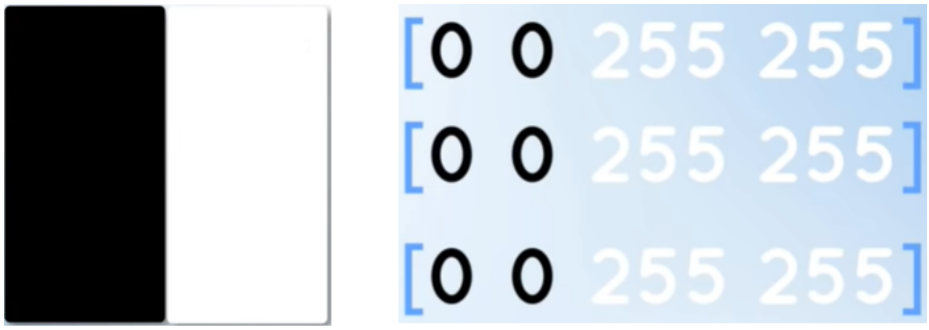

Fig. 2. Edge Detection

3.4 Region Of Interest

In the road lane detection system we are basically interested in road lanes. So we design a specific area to be considered while detecting raod lanes.This helps in faster execution and less memory utilization. Coming to the area to be considered can be designed according to the vehicle size as well as to the gap between the right and left lane and many other factors.

Fig. 3. Edge Detection

Hence we have pointed out the region of interest in white colour rest of the part can be ignored.

3.5 Hough Transform

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure.

.

We see that when the point (2,12) is passed through different lines we obtain a straight line on the hough space.

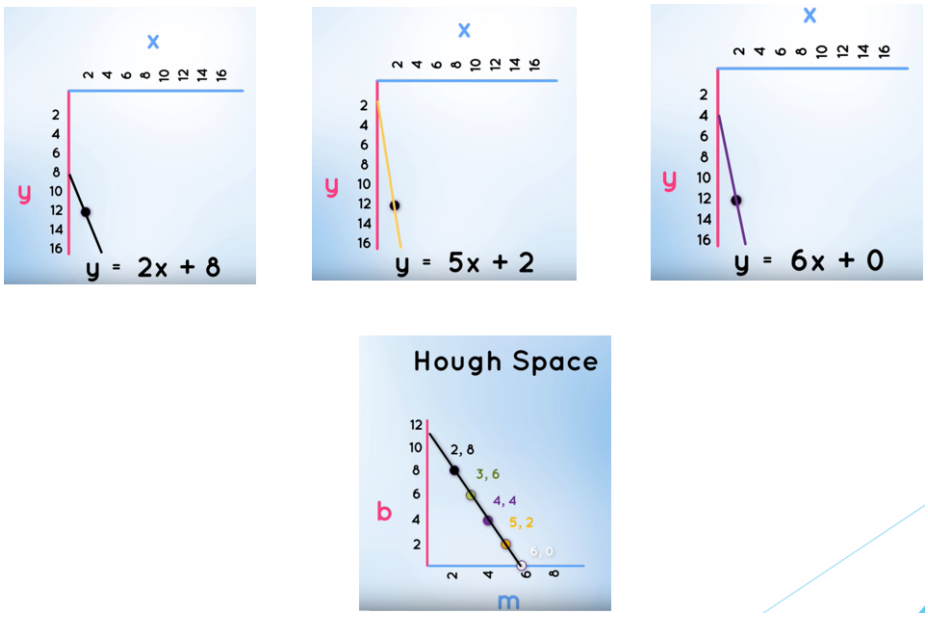

Fig. 4. Hough Transform 1.1

Now considering different point on the same line we obtain different lines on the hough space therefore on intersection point is the point or more specific line of best fit, for example in the above case (4,4) is the intersection point and hence the line of best fit is y=4x+4.

We convert the given coordinates to polar coordinates for more convenience although the concept is the same,Here also the intersecton occours hence helping us to get the line of best fit.

The Results are clearly visible in fig7.

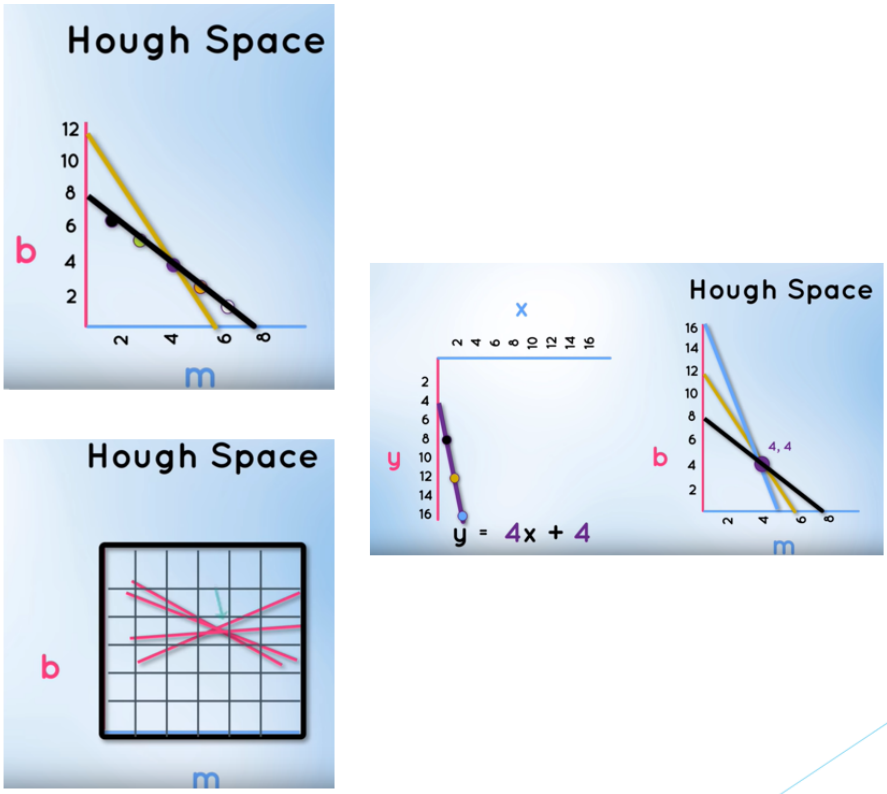

Fig. 5. Hough Transform 1.2

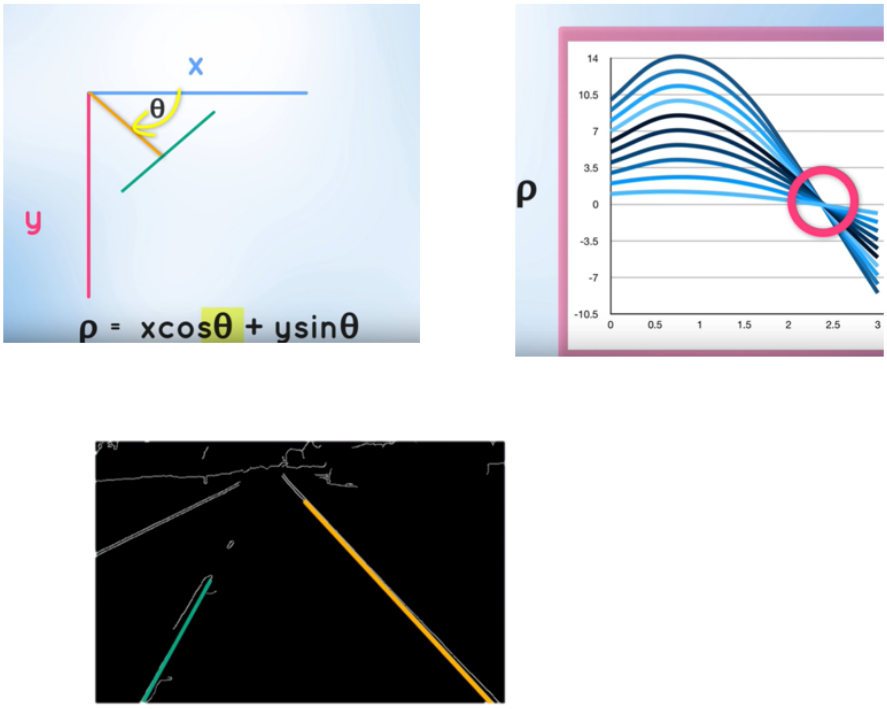

Fig. 6. Hough Transform 1.3

3.6 Implementaion:

The project is implemented in python. The libraries used are cv2,numply and matplotlib.

CODE:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def make_coordinates(image,line_paramters):
    slope,intercept=line_paramters
    y1=image.shape[0]
    y2=int(y1*(3/5))
    x1=int((y1-intercept)/slope)
    x2=int((y2-intercept)/slope)
    return np.array([x1,y1,x2,y2])

def average_slope_intercept(image,lines):

    left_fit=[]
    right_fit=[]
    for line in lines:
        x1,y1,x2,y2=line.reshape(4)
        parameters=np.polyfit((x1,x2),(y1,y2),1)
        slope=parameters[0]
        intercept=parameters[1]
        if slope < 0:
            left_fit.append((slope,intercept))
        else:
            right_fit.append((slope,intercept))


    left_fit_average=np.average(left_fit,axis=0)
    right_fit_average=np.average(right_fit,axis=0)
    print(left_fit_average,'left')
    print(right_fit_average,'right')
    left_line=make_coordinates(image,left_fit_average)
    right_line=make_coordinates(image,right_fit_average)
    return np.array([left_line,right_line])
```

```
def canny(image):
    gray=cv2.cvtColor(image,cv2.COLOR_RGB2GRAY)
    blur=cv2.GaussianBlur(gray, (5,5), 0)
    canny=cv2.Canny(blur,50,150)
    return canny


def region_of_interest(image):
    height=image.shape[0]
    polygons=np.array([[(200,height),(1100,height),(550,250)]])
    mask=np.zeros_like(image)
    cv2.fillPoly(mask,polygons,255)
    masked_image=cv2.bitwise_and(image,mask)
    return masked_image

def display_lines(image,lines):
    line_image=np.zeros_like(image)
    if lines is not None:
        for line in lines:
            x1,y1,x2,y2=line.reshape(4)
            cv2.line(line_image,(x1,y1),(x2,y2),(255,0,0),10)

    return line_image

image=cv2.imread('test_image.jpg')
lane_image=np.copy(image)

#canny_image=canny(lane_image)
#cropped_image=region_of_interest(canny_image)
#lines=cv2.HoughLinesP(cropped_image,2,np.pi/180,100,np.array([]),minLineLength=
#averaged_lines=average_slope_intercept(lane_image,lines)
# lane_image=image if we use like this then changees we make in lane_image ,ther
#line_image=display_lines(lane_image,averaged_lines)
#combo_image=cv2.addWeighted(lane_image,0.8,line_image,1,1 )
#cv2.imshow("result",combo_image)

#cv2.waitKey(0)

cap=cv2.VideoCapture("test2.mp4")
while(cap.isOpened()):
    _,frame=cap.read()
    canny_image=canny(frame)
    cropped_image=region_of_interest(canny_image)
    lines=cv2.HoughLinesP(cropped_image,2,np.pi/180,100,np.array([]),minLineLeng
    averaged_lines=average_slope_intercept(frame,lines)
```

```
# lane_image=image if we use like this then changees we make in lane_image ,
line_image=display_lines(frame, averaged_lines)
combo_image=cv2.addWeighted(frame,0.8, line_image,1,1 )
cv2.imshow("result",combo_image)

cv2.waitKey(1)
```

4 Results

Therefore we have used all the six concepts of our model flow and finally instead of image we can consider a video with different frames and the algorithm works well for video as well.The main thing that our project is based is on reducing high memory and time utilization.In our project we have reduced the two.

5 Conclusion

On conclusion we conclude that our projects aims at improving the current lane detection system which required high memory utilization as well as high execution time. Also the system is suitable to work in noisy image capture situations.The main motive of this project being ROI which has reduced memory utilization and also gaussian blur and other importaint things helping recover a good image for good results. So our system solves the problems as in the projects or articles previously mentioned.

References

1. A. Takahashi, Y. Ninomiya, M.O.M.N., Yoshikawa, N.: Image processing technology for rear view camera: Development of lane detection system (2003)
2. Ajit Danti, Jyoti .Y. Kulkarni, P.: A realtime road boundary detection and vehicle detection for indian roads, international journal of applied information systems, issn : 2249-0868, volume 5 no.4 (2013)
3. Y. Wang, E.K.T., Shen, D.: Lane detection and tracking using b-snake, image and vision computing, 22, p269-280 (2004)