ISSN No:-2456-2165

Lane Tracking Methods: A Review

Aditya Kalra
Department of Information Technology,
Indian Institute of Information
Technology Sonepat, Sonipat, India

Dr. Sourabh Jain
Department of Computer Science & Information
Technology, Indian Institute of Information
Technology Sonepat, Sonipat, India

Abstract:- Every Year, many individuals die of road mishaps caused because of drivers' inattention. Lane Detection Systems are beneficial in avoiding these mishaps as wellbeing is the principal motivation behind these Driver assistance systems. Such frameworks are intended to distinguish the lane markings and help the driver on the off chance that the vehicle will in general leave the path. These Lane Detection Systems are significant components of intelligent transport systems. Lane Detection is a troublesome task because of varying road and climate conditions. In the past couple of years, numerous researchers have proposed and shown different lane marking detectors. In this paper, a thorough survey of writing in Lane Detection Techniques is introduced.

Keywords:- *Lane Detection, Driver assistance system, intelligent transport systems, Lane Detection Techniques.*

I. INTRODUCTION

Driver Assistant Systems are fundamentally intended to help the drivers in the view of any risky circumstance, to stay away from mishaps in the wake of detecting it. To date, there has been various research carried out by numerous specialists on the acknowledgment. These days, Traffic mishaps have become perhaps the most difficult issue. The explanation is that most mishaps occur because of drivers' carelessness. Rash and careless driving could push different drivers' and travelers' lives in peril out and about. Numerous mishaps can be kept away from on the off chance that such risky driving circumstances are distinguished early. Many systems like adaptive cruise control, collision avoidance system, night vision, blind spot detection, and traffic sign detection are a part of ADAS[10]. Path takeoff framework is likewise a piece of this class. This framework has an objective to distinguish the path marks and to exhort the driver if the vehicle tends to leave the path.

A. LANE

A lane is a part of a street that is held to be utilized by a solitary line of vehicles.[2] It is utilized for controlling and directing drivers and lessening traffic clashes. For traffic toward every path, there are something like two paths on the greater part of the streets and isolated utilizing Lane markings. Lanes are indicated by street surface markings on multi-path streets and more occupied two-lane streets on multi-lane roadways and busier two-lane roads.

B. LANE DETECTION

Lane detection is one significant technique in the visualization-based driver support structure and is fit for being utilized for vehicle steering, cross power, crash evasion, or path takeoff cautioning system[12]. Different street conditions that make this trouble more complicated incorporate a disparate assortment of paths, impediments brought about by hindrances, haze, haziness, brightening change (like evening, etc. Hence it is the technique to find paths in the image and is a huge empowering or appealing expertise in various car applications, including path takeoff acknowledgment and cautioning, travel control, cross-control, and independent driving. A Lane Departure Warning System(LDWS) is an innovation intended for advance notice to a driver when the vehicle starts to leave its path. straight & curved, yellow &white, single & double, solid and broken & pavement or highway boundaries. The system should be able to detect lanes even under noisy conditions such as fog, shadow, and stains. The framework should have the option to significantly distinguish lanes under noisy circumstances like mist, shadow, and stain.

II. LITERATURE REVIEW

Mr. Mustafa Surti [1], started with an image captured with a high-resolution camera and passed the image to an embedded demonstration board like raspberryPi. The captured image has been preprocessed for noise reduction, quality enhancement, color management, etc. Marked lane lines are detected by applying Canny edge detection on an ROI followed by Hough transform for lane detection. The lane detection system was demonstrated on an embedded development board like RaspberryPi.

Using Gaussian Based road model [2] lanes could be detected using a multi-stage algorithm. The processes involved in these stages are histogram generation, fitting, and normalization. Different ordered polynomial functions are used for lane fitting.

In this approach, [3], the curvature is detected using image processing techniques. The captured image is converted into binary space further it is segmented into three sections i.e. center, left, and right. To increase the efficiency and to make sure that the road is considered for curvature detection, some amount of the upper portion of the image is excluded for processing. The curvature direction is determined based on the section having the most non-zero pixels. The resultant image is passed through several processes i.e. quality enhancement, noise reduction, canny edge detection, hough transformation, etc. for lane detection.

The approach [4] presents an efficient version of the sliding window algorithm for lane curve fitting. One of the advantages of this improved version is that it allows sliding windows to be applied even on uneven lane markings. This method is applied to the road edges from the binary image. The directions of the street edges are then handled to get the curve value and the vehicle offset.

ISSN No:-2456-2165

The RALPH framework [10], is used to control the lateral position of the vehicle. Path's curve and horizontal counterbalances are dissuaded by a layout to find the middle value of the output line power profile which is adaptively changed and adjusted utilizing a matching procedure. The same college developed another framework called AURORA which tracks the path markers present on structured streets utilizing a variety of cameras mounted on a vehicle pointed downwards toward the street [11].

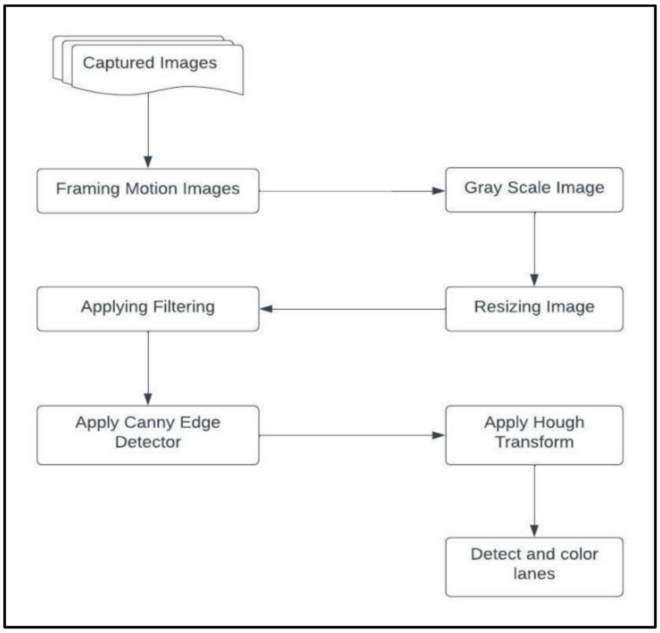


Fig. 1: Generalized Lane Detection Algorithm

III. LANE DETECTION TECHNIQUE

A. HOUGH TRANSFORM

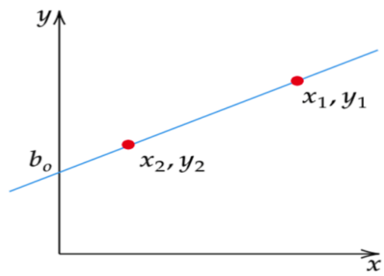
Hough Transform [7] is a method that is used for feature extraction that can be utilized for image analysis and digital image processing. Conventional Hough Transform is used for distinguishing lines in the image. There was trouble in identifying straight lines, circles, and so forth in the automated analysis of digital images. The edge detector has been utilized in the preprocessing stage for acquiring focus on pictures that lie on the ideal bend; however, because of some issue in the picture, a portion of the pixels was absent on the ideal bend. So for taking care of this issue Hough Transform is utilized. Hough Transform is an efficient [8] method for the detection of straight lines in an image, even within the sight of commotion and impediment. By counting novel conditions for each possible line through the point of the image, finding predominant lines in an image is capable. By choosing pixels from the picture object set, the edge pixels can be assembled into an object class. For the detection of lines in an image, it is first changed over into a binary image using some threshold. Then the dataset is added with appropriate instances. Hough space is the principal part of Hough Transform. For the Hough Transform algorithm, it is crucial to perform edge detection first to produce an edge image which will then be used as input into the algorithm[12].

The Hough Space is a 2D plane that has a horizontal axis representing the slope and a vertical axis representing the intercept of a line on the edge image. A line-on-edge image is of the form:

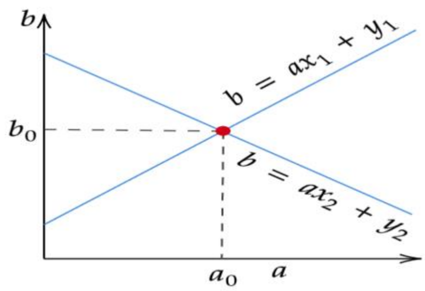
$$y = ax + b \qquad (1)$$

One line on the edge image produces a point on the Hough Space since a line is characterized by its slope a and intercept b [12]. Then again, an edge point (x_i, y_i) on the edge picture can have a boundless number of lines go through it. Subsequently, an edge point delivers a line in the Hough Space as:

$$b = ax_i - y_i \qquad (2)$$



(2a)



(2b)

Fig. 2(a & b): Mapping from the edge points to the Hough Space.

$$m = \frac{rise}{run} = \frac{y_2 - y_1}{x_2 - x_1}$$

m = slope

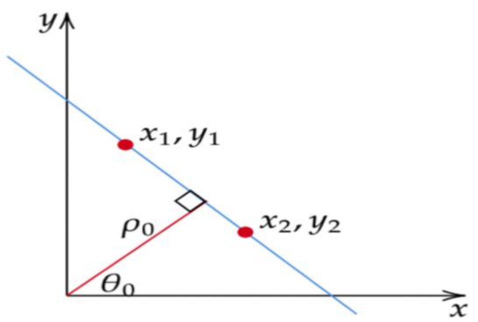
(x_1, y_1) = first point

(x_2, y_2) = second point

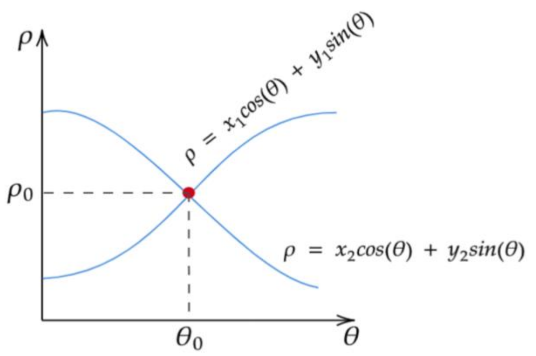
Fig. 3: The equation to calculate a slope of a line

There is one imperfection with representing lines as $y = ax + b$and the Hough Space with the slope and intercept. Here, the calculation will not have the option to distinguish the vertical lines as the slope 'a' is undefined (infinite) for vertical lines. Programmatically, this implies that a computer would require a boundless measure of memory to address all possible values of a. To keep away from this issue, a straight line is rather represented by a line called the normal line that passes through the origin and is perpendicular to that straight line. The type of the normal line is $\rho = x \cos(\theta) + y \sin(\theta)$ where ρ is the length of the normal line and θ is the point between the normal line and the x-axis[12].

As referenced, an edge point delivers a cosine curve in the Hough Space. From this, if we somehow happened to map all the edge points from an image onto the Hough Space, it would produce a ton of cosine curves. If two edge focuses lay on a similar line, their related cosine curves will intersect each other on a particular (ρ, θ) pair. Consequently, the Hough Transform algorithm detects lines by seeing the (ρ, θ) pair that have various convergences larger than a specific threshold. It is important that this technique for thresholding could not necessarily in all cases yield the best outcome without doing some preprocessing like neighborhood suppression on the Hough Space to eliminate comparable lines in the edge image. The Hough Transform is essentially utilized for the detection of straight paths. Yet, it tends to be improved to detect the curved path effectively and efficiently.



(4a)



(4b)

Fig. 4(a & b): An alternative representation of a straight line and its corresponding Hough Space.

B. EDGE DETECTION

This algorithm [4] depends on identifying points in an image at which there is a sharp change in image brightness. Edge is characterized as an organized set of curved line segments. This set comprises of the points where the brightness of the image sharply changes. Edge Detection is a tool utilized in image handling for feature detection and extraction. This algorithm fundamentally diminishes data to be handled and may eliminate less relevant data while saving significant properties of an image. In the algorithm that this calculation is fruitful, the task of interpreting the information in the image may be improved. Anyway, it isn't generally imaginable that ideal edges can be gotten from real-life images of present-day intricacy. An edge detection system called Canny edge detector is utilized to distinguish edges in an image. This technique involves numerous stages of calculations and points in finding the optimal edge detection. Canny edge detector is an edge disclosure computation that utilizes various stage processes to recognize edges in an image.

Fig. 5: Canny Edge Detection Algorithm. Source: AI Shack

IV. GAPS IN EXISTING WORK

Nowadays various lane detection algorithms have been used for assisting the driver in the Advanced Driver Assistance System(ADAS). Most of the techniques proposed have focused on the detection of straight lanes markings and curved lanes have been ignored or the algorithms are inefficient to detect curved lanes accurately. The improvement in Hough Transform has also been ignored for improved Lane Detection systems. The effect of fog on Lane Detection has also been ignored. The effect of over-Amplification of contrast in the images, when converted to binary images, is also ignored while developing lane detection systems which leads to inefficiency of the system to give accurate output to the user.

V. CONCLUSIONS

In this paper, different lane detection methods are audited and examined. All the reviewed methods identify the straight lanes in a very effective manner. Yet, this large number of strategies have disregarded the detection of curved lanes. The current algorithms have disregarded the idea of CLAHE for proficient picture contrast and utilized Improved Hough Transform for better-bended lane detection.

REFERENCES

[1.] M. Surti & Prof. B. Chourasia, "Real-time lane detection system using Python and OpenCV on Raspberry Pi ", International Journal for Research in Engineering Application Management (IJREAM), Vol-05, issue-06, 2019.

[2.] B.C. Yeo, W.S. Lim & H.S. Lim, "Lane Detection In The Absence Of Lane Markings For Roadway Surveillance With Thermal Vision", *ICIC International*, Vol-12, Number-03, 2016.

[3.] A. Ihare, "Image Processing to Detect Curvature of Road", Article on medium.com, 2020.

[4.] L. Bentley, J. MacInnes & R.B.T. Bose "A Pseudo-Derivative Method for Sliding Window Path Mapping in Robotics-Based Image Processing", 2019.

[5.] G. Kaur & G. Singh, "A Review Of Lane Detection Techniques", *International Research Journal of Engineering and Technology (IRJET)*, Vol-02, Issue-03, 2015.

[6.] B.C. Yeo, W.S. Lim & H.S. Lim, "Lane Detection In The Absence Of Lane Markings For Roadway Surveillance With Thermal Vision", ICIC International, Vol-12, Number-03, 2016.

[7.] Ohio University, "The Future of Driving", Article on ohio.edu, 2021.

[8.] M. Hassan, "Self-Driving Car using Raspberry Pi", Article on computer- vision.zone, 2020.

[9.] S. Kamc, D. Aksu, M.A. Aydin" Lane Detection For Prototype Autonomous Vehicle", AIRCC Publishing Corporation, Vol 09, Number 13, 2019.

[10.] D. Pomerleau & Jochem," Rapidly Adapting Machine Vision for Automated Vehicle Steering", *IEEE*, 1996.

[11.] M. Chen., T. Jochem D. T. Pomerleau, "AURORA: A Vision-Based Roadway Departure Warning System", *In Proceedings of IEEE Conferenceon Intelligent Robots and Systems*, 2004.

[12.] S. Lee, "Lines Detection with Hough Transform", Article on towardsdatascience.com, 2020.

[13.] Open Source Computer Vision," Smoothing Images", Article on docs.opencv.org, 2021.

[14.] A. Rosebrock, "4 Point OpenCV getPerspectiveTransform Example", Article on pyimagesearch.com, 2014.

[15.] Aryan," Implement Canny Edge Detector in Python using OpenCV", Article on geeksforgeeks.org, 2020.

[16.] S. Sahir, "Canny Edge Detection Step by Step in Python — Computer Vision", Article on towardsdatascience.com.