An Optimal Approach for Use of Lane Detection Algorithm Using Reliable Lane Markings

Kanishk, Chandan Kumar, Kundan Singh, R.B. Sarooraj

Abstract: This paper proposes efficient and innovative way of Lane Detection Algorithm implementation for developing nations like India, Sri-Lanka, Pakistan and Urban Nation as China, USA. The paper consists of analytical approach for the use of algorithm and implementation of the same. The approach for the implementation of the algorithm is robust and effective. We would be using Active Safety System first; the collective images of the objects (nodes) are captured by the front – end camera as binary image from the Region of Interest. The Binary images are then converted to Grey – Scale Images. The Region of Interest (ROI) is determined and Lane Detection Algorithm is implemented to get the matched value with the real time object. An Active Safety System is designed and embedded with the micro controllers and Adaptive Cruise Control and further, Sensor Fusion is done. The Experimental results show that the algorithm was implemented successfully and identifies lanes in complex situations and worn out areas and curved lanes.

Index Terms: Objects, Binary Images, Grey Scale Images, Bird's eye view, Inverse Perspective Mapping, Gaussian's density function, Histogram images, Region of Interest.

I. INTRODUCTION

This paper introduces use of Reliable Lane Markings in Lane Detection Algorithm for better and safe travelling experience in the developing countries. In this paper, it covers the basic algorithms used for Lane detection technique using image processing and Nodal Object Recognition. India and other developing countries transporting vehicles use Automotive Navigation System, which uses satellite technology for GPS Navigation and Google Maps with image processing. In this type of technology, Satellite Navigation Device is used which gives position data which gives correlated position on the road. Lane Detection Technology uses Automated Lane Departure Warning System and Lane Keeping Assistant System which typically helps in alerting the driver for correct timing and positioning of the vehicle on the roads. Reliable Lane Markings (RLM) are the Standard Marking System; it gives the ability to convert the Road Markings and Symbols in Object Nodes and process it into real time images and graphics detection.

This paper develops a robust and effective vision based approach in the field of Lane detection. In the paper it is theoretically explained the approach of capturing real time image with the help of Video Processing camera and cover live stream and image processing with the help of pre-defined Lane Markings object nodes. The object nodes are armature to the lane markings and these should be considered as pre-defined objects which can later be used by the image processing sensors and identify them as objects. The first step is embedding video and image camera on the front and rear side of the vehicle which will capture the video and image of the objects. The image captured is analyzed for getting the positioning of the vehicle and it is matched with the satellite image using GPS technology. The processed image in then converted into Binary images combined and converted to Grey Scale Image for a particular Region of Interest(ROI), This region of interest is the scalable field of image sensing area that is cropped according to the Region of Interest(ROI) and the particular ROI is extracted on which we need to work on. Once, we have the extracted Region of Interest (ROI), we check the state flow and model it with algorithm tuning. The tuned algorithm checks the image and extracts object in the form of nodes which is marked and matched. These objects are exactly similar to the objects in real time that we get from Reliable Lane Markings (RLM). These Lane Markings are non-other than: ***1. White Lines:*** These are painted on the Lane to indicate the traffic travelling in our direction. ***2. Solid White Lines:*** This indicates us to stay at the single lane and changing of lane is not allowed. ***3. Broken White Lines:*** This indicates that we can change the Lane if we wish to. ***4. Yellow Lines:*** It is used to mark the centre of the lane for a two way traffic road and we may pass the traffic if we wish to. If the yellow line is accompanied with a broken white line, then we are not allowed to pass the traffic and follow in a single lane. We would be using various algorithms to check the state flow and match the object with real time objects and the processing of image.

II. LITERATURE SURVEY

We did a Literature Survey from 6 previous year published papers which includes 1 base paper and its corresponding reference papers. We analyzed that the object the whole process can be described in six phases of implementation. The six phases of implementation are described as follow as per previous authors.

(A.) Vehicle Positioning Abstraction: For this very first process of Positioning Details Abstraction, Global Mapping Algorithm [1] is used. In this Algorithm, we extract the object and obstacles position using coordinate mapping system. This is achieved with the help of analysis and extraction from the Gray Scale Image.

Revised Manuscript Received on May 05, 2019

Kanishk, Department of Computer Science Engineering, SRM Institute of Science and Technology, Chennai, India

Chandan Kumar, Department of Computer Science Engineering, SRM Institute of Science and Technology, Chennai, India

Kundan Singh, Department of Computer Science Engineering, SRM Institute of Science and Technology, Chennai, India

R. B. Sarooraj, Department of Computer Science Engineering, SRM Institute of Science and Technology, Chennai, India

The Grey Scale Image is extracted from the Binary Image. It is then transferred to Global Coordinates [1]. Global Mapping Algorithm though cannot fulfill all the conditions and certain problem arises in the abstraction of vehicle, object and co-relating them at the real time aspect. The other problems that occur with the use of this algorithm are: ***1. Data Density Problem:*** This includes the problems that occur in position tracking and position extraction, Plane fitting, combining stereo cameras, and LIDAR's [1], with differential depth. The Algorithm is not able to give a precise position coordinates and it is difficult to tune with the model. ***2. Pace Error:*** This includes simultaneous localization [1] and mapping, which means the mapping of object distance and time value, does not match with the Range-speed graph of the vehicle and object distance calculation. It effects on calculation of Iterative Closest Points [1]. This gives approximate value between the time lapse of the object and the vehicle. The Scans matching is not sufficient and gives errors in the waveform length of the historical graph of the images.

(B.) ***Vehicle System Configuration:*** We need to configure the vehicle with some specific hardware and sensor peripherals that can give us the object input. To obtain this we use: ***SNUCLE:*** This is used in the Autonomous Vehicles but, a custom made device enables electronic actuation of brake with a wire and motor. Two stick LIDAR'S and one IBEO LIDAR is installed at the front bumper guard of the vehicle. There should be Three SICK LIDAR's [1] roof rack which integrate with the camera and peripherals.

(C.) Vision Detection: Vision Detection is the Third phase according to the base paper and the other reference papers, the vision system mentioned in the reference paper does not recognise the Lane and Markings by the LIDAR's easily. For achieving the correct image for processing, we would be using: ***1. Greyscale Camera:*** This camera is a pedestrian crossing camera that gives input image in Grey scale format. The image quality is typically low thus it is hard to get a précised image in the same format. ***2. Color-camera:*** It is a camera that gives input in RGB color format and it helps in managing the bumps while travelling as it helps Bump detection and Lane Detection. The Lane Detection rate is 85% (Accuracy) with no false positive.

(D.) Algorithm Classification: It is the theory, which helps in classification of algorithm and understanding the usage of algorithms for various period of task. The algorithms and usage of these algorithms are classified on the basis of Mapping of ground detection. Some of the Algorithms described in the previous works are:

(1.) Height – Difference based Algorithm: This Algorithm helps in detecting the height and difference between the object points with the help of précised object distance calculation and calculation of height distance. For getting the most accurate results we use:

$$A \rightarrow x \rightarrow B,\ where\ x \geq \alpha\ (Dangerous\ Point) and\ \alpha\ is\ Pre\ defined\ Range.$$

After we have got the correct height and distance values between the distances object points, we will have gone for further Risk Integration. ***(2.) Risk Integration:*** Risk Integration helps in getting the reliability of each algorithm with help of $R_{int} = W_{gd}.R_{gd} + W_{hd}.R_{hd} + W_{ab}.R_{ab}$. This gives us final risk value. Here, R_{int} is the integrated Risk Value. R_{gd} is he risk value from the ground detection algorithm. R_{ab} is the risk value from the height difference based algorithm. W is the height of each algorithm. It is given by the reliability of each algorithm and the detected point (x, y) value.

(E.) Hough Transform Algorithm: This algorithm is used twice in the previous works, it defines the usage of Fixed Region of Interest (ROI). It gives the usage of Candidate Lane Pixels and is detected by 1- bit transform. The feature points/ nodes are determined by SOBEL filter. Lines are detected and extracted by Hough Transform Algorithm. It creates a difference between the object as nodes and real time.

(F.) Image converse Transform: This algorithm helps to extract the information from Grey Scale Images. Grey scale images carry information to detect through Lanes and can be handled with small computational times. The RGB image received as input from Color Camera is converted to Grey scale image.

III. SYSTEM DESCRIPTION

The System description can be understood as it has five important components. The components can be classified as ***(A.) Input State – Active Safety System (B.) Processing state - Adaptive Cruise Control (C.) Extraction state - Radar and Camera Sensor Fusion (D.) Output State.***

(A.) Input State – Active Safety System: The input stage consists of all the important hardware and processing connectors which helps us to get the input. The input consists of video graphic camera which can be divided into frames and these frames can used as image input and these RGB image can be then further processed into Grey scale image.

(1.) RGB Camera: The RGB camera will give us input as video enabled graphics and algorithms will help us to extract the information from the RGB image and help us to convert these images into Grey scale image. The input taken from the RGB camera are converted to Object nodes point which helps us mark the nodes and these nodes gives information of the threshold value of the image. Then the gray scale image is matched with the object nodes. ***(2.) Image filter:*** These filters are used to get the best result from the frames that are derived from the RGB image group. These filtered images are then used by the grey scale image. ***(3.) Fetcher:*** This feature will help us to fetch the best image in grey scale format and this will be used for further process. ***(4.) Adruino:*** Adruino is a micro controller that is to be used for processing the hardware and software configuration and tuning of the process. Adruino contains input and output peripherals. This will help us to connect our software configuration part.***(5.) Sensors and Radars:*** The hardware sensors are to be used for the receiving the signals.

(B.) Processing state - Adaptive Cruise control: The processing state consists of the software configuration with the system and hardware. We will use following software's to achieve our desired result: ***(1.) OpenCV:*** Open CV is package software integrated with Python. This is used to write the mathematical portion of the calculation. These calculations will be connected to the analysis part further.

(2.) MATLAB: MATLAB is to be used for writing the functional part of the system to integrate it with the hardware part of the system. ***(3.) Simulink:*** Simulink helps us to get the simulation part of the system. It helps us to understand the functioning and working of the software. ***(4.) Radar system simulation in MATLAB:*** The radar that is to be used should be configured. Therefore, we will be using ***(4.1) End to End Mono-static Radar*** and ***(4.2) FMCW Radar Range Estimation*** for configuring the radar and sensors with Adruino.

We would be using various tool boxes for the desire result: ***(1.) Array system tool box (2.) RF toolbox,*** these tool boxes configures the RADAR and sensors.

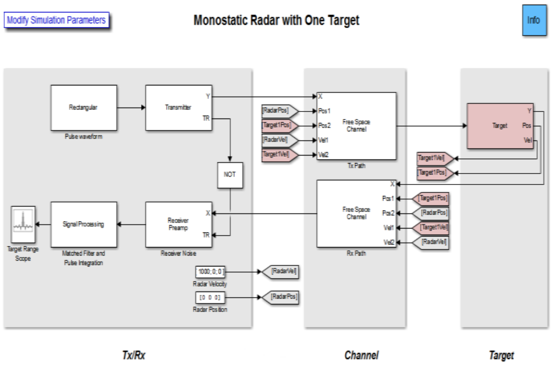

Fig1: Mono-Static Radar with one target

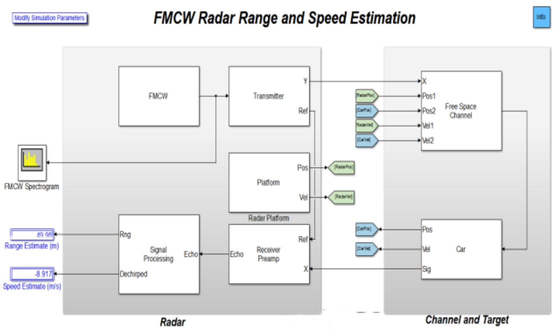

Fig 2: FMCW Radar Range and speed Estimation

(C.) Extraction State - Radar and Camera Sensor Fusion: The prepared result should be integrated with the output gadget. For structure complete Active Safety System, we would utilize RADARS and Sensors. We would integrate the RADAR and sensor with the assistance of Automotive Adaptive Cruise control Technology. An Automotive Cruise Control System utilizes Frequency Modulated Continuous Waveform Technology. This precedent tells the best way to show an automotive adaptive voyage control system utilizing the frequency modulated continuous wave (FMCW) technique. This model performs range and Doppler estimation of a moving vehicle. Not at all like beat radar systems that are ordinarily found in the resistance industry, automotive radar systems often adopt FMCW technology. Contrasted with beat radars, FMCW radars are littler, utilize less power, and are less expensive to manufacture. As a result, FMCW radars can just monitor a lot littler distance

.

Fig 3: FMCW radar working; Detecting close object

FMCW Waveform: Think about automotive long range radar (LRR) utilized for adaptive cruise control (ACC). This sort of radar for the most part possesses the band around 77 GHz, as demonstrated in [1]. The radar framework always appraises the separation between the vehicle it is mounted on and the vehicle before it, and cautions the driver when the two become excessively close. The figure beneath demonstrates a sketch of ACC. A well known waveform utilized in ACC framework is FMCW. The guideline of range estimation utilizing the FMCW procedure can be outlined utilizing the accompanying figure

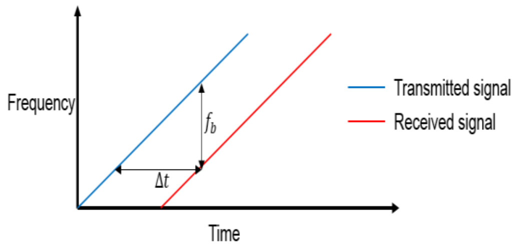

Fig 4: Frequency- Time Range graph

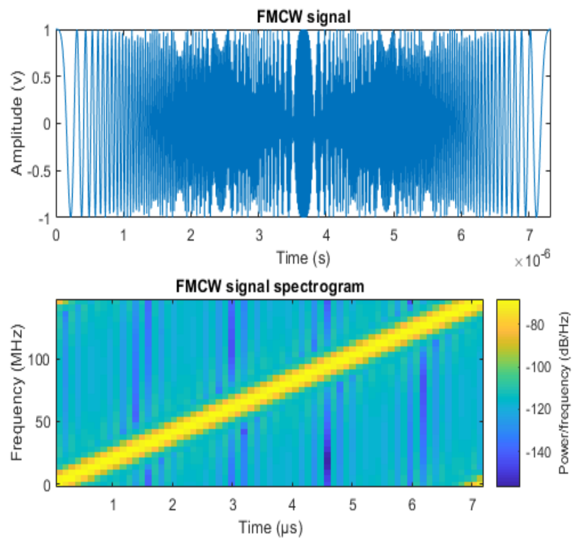

Fig 5: Frequency Wave Transform

(E.) Output State:

The output state consists of various. Here, it gives us the description about the type of output from beginning of the process and it helps us to understand and match the optimum result. The formats are:

(1.) Capturing:

It includes output that we get the RGB video feed camera. The video input is fragmented into Frames and the best frame is filtered out from the group to process. The frame that we obtain is in RGB format which we need to convert into Grey-Scale Format. The Gray-scale format image is used to filter the Region of Interest (ROI).

(2.) Processing:

The Processing part of the system consists of use of Lane detection algorithm with Reliable lane markings. The Grey-Scale is used to get the object nodes and these nodes are used to match with the real time

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

image and give output as video feed, in-built display system or vibration under the seat.

(3.) Output Extraction:

For Final Output we need a few devices, such as In-built screen connected to the Adruino to get the live feed of the Lane. The In-built display icon, to alert the driver, if he is going out of range. Driver A vibrating device, which alerts the driver to check his lane.

IV. DATA ACQUISITION

The Data Acquisition consists of various steps. Here, it gives us the description about the type of data from beginning of the process and it helps us to understand and match the optimum result. The formats are ***(1.) Capturing:*** It includes output that we get the RGB video feed camera. The video input is fragmented into Frames and the best frame is filtered out from the group to process. The frame that we obtain is in RGB format which we need to convert into Grey-Scale Format. The Gray-scale format image is used to filter the Region of Interest (ROI). ***(2.) Processing:*** The Processing part of the system consists of use of Lane detection algorithm with Reliable lane markings. The Grey-Scale is used to get the object nodes and these nodes are used to match with the real time image and give output as video feed, in-built display system or vibration under the seat. ***(3.) Output Extraction:*** For Final Output we need a few devices, such as In-built screen connected to the Adruino to get the live feed of the Lane. The In-built display icon, to alert the driver, if he is going out of range. A vibrating device, which alerts the driver to check his lane.

V. ARCHITECTURAL DESCRIPTION

The system architecture can be understood by the following block diagram:

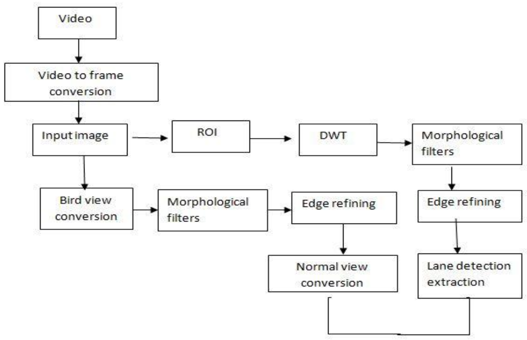

Fig 6: Block Diagram for system Implementation

Here, in the block diagram, it shows a step by step configuration on the System. The block diagram can be used to build a prototype model of the system implementation.

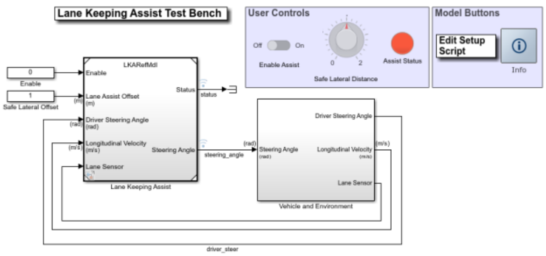

Fig 7: This shows the system flow diagram for configuration of the software modules with the Adruino

The Adruino should be set up such that, it does not get dismantle when the vehicle is facing bumps while traveling. The Camera should be placed in the front as well as the rear end of the vehicle. The RGB color camera is connected to the Adruino where, software modules for Lane Detection using Reliable Lane Markings are stored. The Software module helps us getting the desired output. Finally, The Result is displayed over the output screen.

Fig 8: Real Time test image of Lane

Fig 9: Lane detection in real time.

VI. IMPLEMENTATION

The implementation part can be divided into Three Parts: (***A.) Input Stage (B.) Processing Stage (C.) Output Stage.***

(A.) Input Stage: The input stage consist of two parts i;e ***(1.) Image and video capturing:*** The video feed is taken from the color camera in RGB format and then converted into Grey-Scale Format which is used later for the information extraction. ***(2.) Image Filtering:*** The image in Grey scale format is taken and used for filtering according to the Region of Interest (ROI), this filtered image of ROI is used for Algorithm Tuning.

(B.) Processing Stage: The Processing stage can be divided into three parts: ***(1.) Lane tracking (2.) Adaptive Cruise Control (3.) Software Configuration***

(1.) Lane Tracking: This is the first module code written in MATLAB environment, which enables us to the desired output and this Module is divided further into Three parts: ***(1.1) Edge Detection (1.2) Positioning of the Vehicle (1.3) Sensor Fusion***

(1.1) Edge detection: For every desired output we need to re consider the ROI and optimize it as Object nodes, these object nodes will help us to filter the image and detect Edges of the Lanes. We will be using Edge Detection function of MATLAB to get this output.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

(1.2) positioning of vehicle: After getting the perfect edges and object nodes we will use the GPS to get the positioning of the vehicle and mach it in the real time.

(1.3) Sensor Fusion: The RADAR's and Sensors are to be configured for getting the alert system and output for detection of close object. We will use Accurate CIPV(Critical In-Path Vehicles) selection for integration the range of the detection and the sensors together. It helps in getting the output relevant and object identification. The Camera will help us to detect the lane. This will help us to detect the object distance as well as the angles of alignment.

(2.) Adaptive Cruise Control System: This system helps us to configure and response with the hardware sensors simultaneously. This uses the OpenCV and Matlab environment for the same.

Fig 10: Configuration of Adruino according to FMCW radar.

This helps us to configure the steering of the vehicle, Radars and the Adruino at the same time.

(3.) Software Configuration: We tried different Algorithms for detecting the Lanes but according to the test results the best and optimum results were given by Two Lane Detecting Algorithms: ***(1.) Sobel (2.) Roberts.*** These algorithms are in-built edge detection functions that can be used for Lane Detection in the real time module.

VII. CONCLUSION

We have successfully designed and implemented the concepts of Lane Detection in a prototype image format. We implemented it as taking Image as input and configuring it with the Sensors and Radars. We were successful in configuring the Camera with the Adruino. Firstly, we took lane images as input and converted it into Grey Scale image format and derive Region of Interest. After this, we filtered the image and extracted information of the image and converted it into Object Node. This object node is used for GPS positioning of the vehicle. The position of the image is matched with real time and verified. Then, We use Lane Detection Algorithm to derive the edge and Reliable Lane Markings as an reference over the image input and extract the output of the Lanes as real time feed. We sincerely incorporated all the steps and achieved it over time. The Project scope is being used for Self-Driven Cars and regularly tested over different situations.

REFERENCES

1. Jaewoong Chaoi, Junyoung Lee, Dongwook Kin, Giacomo Sporani, Pierro Cerri, Member, IEEE, Alberto Broggi, Senior Member,IEEE and Kyongsu Yi "Environment-Detection-and-Mapping Algorithm for Autonomous Driving in Rural or Off-road Environment, IEEE transactions on Intelligent Transportation Systems (2012).
2. Florida Janda, Sebastian Pangerl, Andreas Schindler Institute of Software Systems in Technical Application (FORWISS), University of Passau, Germany.
3. Chuanxiang Li, Bin Dai and Tao Wu, College of Mechatronic Engineering and Automation, National University of Defence Technology, Changsha, Human Province, China, "Multi-Lane Detection in Urban Driving Environment Omni-Directional Camera", Proceeding of the IEEE International Conference of Information and Automation, Hailar, China, July 2014.
4. Suvarna Shirke, Phd CSE Reserch Scholar. Bharath Institite of Reseach and Educaion, Chennai,(Asst. Prof. Atharva COE, Mumbai, "A Study of Lane Detection Techniques And Lane Departure System".
5. https://in.mathworks.com/help/releases/R2016a/phased/examples/modeling-rf-front-end-in-radar-system-simulation.html?s_eid=PEP_12669
6. https://in.mathworks.com/help/phased/examples/automotive-adaptive-cruise-control-using-fmcw-technology.html?s_eid=PEP_12669

AUTHORS PROFILE

Kanishk. Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India

Chandan Kumar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India

Kundan Singh, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India

R. B. Sarooraj, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India