Lane Detection and Obstacle Avoidance Techniques in Autonomous Driving Vehicles

Phanindra Amaradi[1], Nishanth Sriramoju[2]

[#]Electrical &computer Engineering, Kettering University

1700 University Ave, Flint, MI, 48504, United States

[1]venk2168@kettering.edu

[2]srir3757@kettering.edu

***Abstract*-In this paper, we have developed lane detection and obstacle avoidance techniques using Matlab. Our system is based on single 180 degrees fish eye camera and LIDAR sensor. For lane detection a modified lane marking technique was used representing edges of the lane with offset distance in between the lanes. This paper describes implementation of the lane detection system using Hough line and Hough transform to detect lane marker. Based on the detected lane marks, the position of the vehicle is determined as to whether the vehicle stays in its lane or stays out of lane. The system will display an offset distance values from the centre of the lane. We adopted SICK LIDAR LMS-291 S-O5 Laser range sensor to detect obstacles.**

***Keywords*— Lane Detection, Obstacle Avoidance, Image Processing, Video Processing, Hough Transform and Computer Vision.**

I. INTRODUCTION

Lane detection and Obstacle Avoidance Technique plays the most important roles in Advanced Driver Assistance Systems (ADAS). ADAS acts as an assistant to the driver while driving. Many Car Manufacturers, Universities, Research Institutes and Companies have focussed on using various kinds of ADAS systems such as ACC (Autonomous cruise control), CACC (Cooperative Adaptive Cruise Control), LDW (Lane departure warning), FCW (forward collision warning) [1], BSS (Blind Spot Information System), AFS (Adaptive Front lighting System), PCW (Pedestrian Collision Warning）, V2X (Vehicle to X), Start/Stop systems have been presented. An Accident can occur within split second due to driver's inattention or drowsiness. During a long drive returning back from a holiday or after a stressful and tiring day at work, drivers tend to fall asleep for a few seconds, which results in accidents. To reduce the increasing mortality rate due to traffic accidents, a system needs to warn the driver if the vehicle is drifting out of its lane. Therefore many researchers and research teams are working to improve intelligent vehicle systems to assist the driver, avoid collisions or to control the vehicle autonomously [2]-[4]. Lane Detection System is one of the most studied areas among intelligent vehicle systems. The lane detection system is the process of identifying the lane markings and edges to estimate the position of the vehicle within the two lanes measured from the centre of the lane. Fig. 1 illustrates a sample vehicle lane detection scenario.

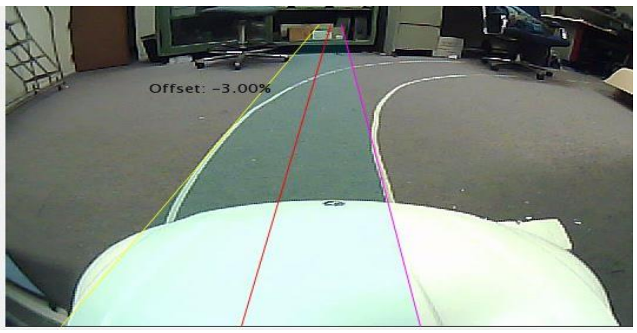


Fig. 1: Lane detection scenario

Although Lane detection is capable of detecting lanes yet obstacles detection was possible accurately, only using LIDAR sensor. This obstacle avoidance is done using the SICK LMS-291 LiDAR sensor. LiDAR is a laser measurement system that emits infrared laser beams and measures the angle of incidence and distance of objects that are obtained by the beams. Using these measured angles and distances the vehicle can scan points that are within a buffer zone around itself to detect objects.

The main objective of this paper is to implement an effective lane detection and obstacle detection system. In this paper, image processing based lane detection system is comprised of the following steps: filtering, edge detection, lane detection and offset distance display based on position of the vehicle. Warnings to apply a proportional turn respective to offset distance to the wheel is produced after applying edge detection and Hough Transform method in order to adjust the vehicle to stay in its lane. By using the LIDAR raw data distances are measured to detect obstacles around it.

The Rest of the paper is arranged as follows, Section II describes related literature. Section III provides an overview of the proposed system, Section IV provides results of various experimental conditions and Section V is conclusive and presents directions for future work.

II. Related Work

The related literature on advanced driver assistance system (ADAS) and autonomous driving vehicles is to autonomously navigate through the Waypoints to reach the target from the start point by using global-positioning system (GPS) to obtain the data relating to latitude and longitude. The accurate location of the Vehicle is obtained from the GPS device in a standardized NMEA0183 string format. These strings are passed through a parser that will call a specified function when a complete set of sentence is received. This function updates the current position of the Vehicle. While using the GPRMC string, it is also possible to get the current heading using only the GPS [5]. To perform waypoint navigation, the desired heading value is calculated using trig functions based on the current position of the Vehicle and the current target waypoint. There are many systems which are based on using sing/dual cameras to detect the lanes. Sensors especially LIDAR, RADAR are used in detecting the obstacles. 3D modelled simulation of the road environment is generated for testing the autonomous driving vehicles. Some works are used to observe the driver's behaviour using front facing camera to detect possible drowsiness, sleeplessness, and fatigue based on the eye-movement, yawn detection and head rotation.

Other works are based on Traffic sign recognition and detection, simulating a 3D modelled road environment (e.g. [6], [7], and [8]). For detecting and avoiding collision, they focused on LIDAR sensor data classifying object as static, dynamic and tracking, using extended Kalman filter for getting a wide view of the surrounding environment. Once we have an idea of where the lane marks are at the first frame, a Kalman filter is used in order to predict where the lane marks will be in the next video frame, then evolve step-by-step, frame after frame, through Kalman predictions, until and unless this is achieved, (i.e. Kalman is able to lock-on the lane mark). Every Kalman prediction identifies a very thin region of the next frame, in which we can perform a local search in a very efficient way. For detection of forward collision Srinivasa et al. have used forward looking camera and radar data [7].

Urban and countryside locations are used by many people to demonstrate the example of lane detection and departure warning system. Most of these works have used machine learning and image processing, computer vision approach for detecting and extracting the lane features. Some of the works assume straight and plane roads and work only on highways. This happens because of assumption of a strong lane marking and low traffic. In lane tracking, Kalman filter and particle filter [9] based approach is used. Another work based on vanishing point detection [10] gives a good result on the countryside road.

III. Lane Detection and Obstacle Avoidance

Lane Detection

The Lane detection algorithm is a mechanism which detects the driving lane boundaries and estimates the geometry of the lane in 3D coordinates, respective to the vehicle. Lane detection algorithm allows in building various applications, like:

1) Lane keeping 2) Lane Centering 3) Lane departure Warning system.

Lane Keeping actively applies torque to the steering wheel, to avoid unintentional drift of the vehicle out of the lane .Based on the lane boundary geometry the steering angle decides how much torque is required to be applied in order to steer the vehicle which prevents the vehicle to veer off the lane. Lane centring is a method of continuously controlling the steering wheel in order to keep the Vehicle put stay at the centre of the lane. Based on the lane geometry the steering angle decides as to how much torque is required to be applied in order to keep the vehicle at the centre of the lane. The Lane Departure Warning System uses the information from the lane detection algorithm, to estimate whether the vehicle is about to cross the lane boundary within a few seconds. In that situation, and if the driver did not plan to signal his intent to switch the lane, a warning is displayed.

Lane detection is a most researched area in the field of computer vision with applications in autonomous driving vehicles and advanced driver assistance system. A lane detection algorithm must be capable of detecting all manners of markings from cluttered roadways and filter them to produce an estimate of position of the vehicle and trajectory respective to the lane markings. To develop a solid lane detection algorithm that can work in variety of environmental conditions, we developed the lane detecting system by integrating the lane marking detection with lane tracking algorithm.

In order for a lane detection system to be robust, [11]:

- It should be capable to detecting image features which Correspond to road markings
- It should ignore irrelevant road markings, i.e. text on road Surfaces or road markings not parallel to the vehicles Lane.

The process of lane detection system includes the following, viz.

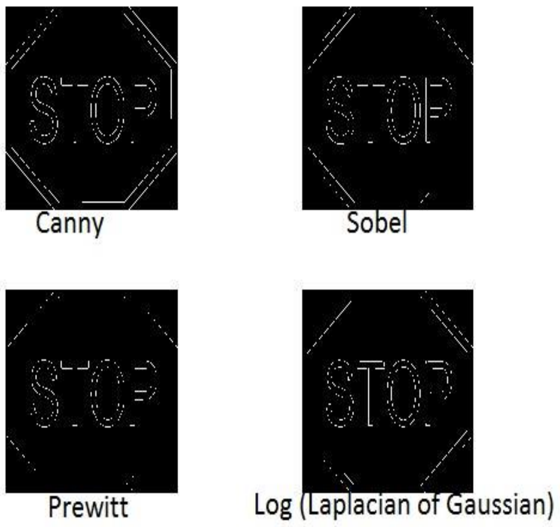
- Smoothening,
- Edge Detection,
- Line Detection and
- Lane Detection.

The process of Smoothening involves, removing of noise from the image by using one of the various [12] filters. 2D-FIR filter is selected among other edge operators to generate a binary edge map. Hough Transform is applied to the edge map to detect lines. Hough Transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough *Tr*ansform is most commonly used for the detection of regular curves such as lines, circles and ellipses, whereas Hough Line is a transform used to detect straight line in lane boundaries.

Removing noise from the image is very important because the presence of noise in the system will affect in detection of the edge. Three fundamental steps performed in edge detection system are [13]:

- Smoothening : images for noise reduction
- Edge point detection: this operation extracts image from all the points that are potential members to become edge points.
- Edge localization: is a process to select a member from the edge point only, the points that are truly members of the set of points incorporated in an edge.

Figure 2 illustrates a variant of different edge detection filters that we used to find edges in a stop sign board. When we compare this image with different edge detection filters, we see the difference and can say how it works to find the edges. There are many edge detection methods [14] that can be used in matlab. Fig. 2, below shows Canny, Sobel, Prewitt, log (Laplacian of Gaussian) edge detection methods by looking for intensity in image. Whereas, canny edge detection looks for local maxima of the gradient of intensity in finding edges and edge function calculates the gradient using the derivative of Gaussian filter. Sobel edge detection function uses Sobel approximation to the derivative. It displays edges at those points where the gradient of intensity is maximum. Prewitt edge detection returns at those points, where the gradient of intensity is maximum. Log (Laplacian of Gaussian) edge detection finds edge by looking for zero-crossings after filtering the intensity value with a Laplacian of Gaussian filter. Comparing all the methods, we can observe that the canny method is less likely to be fooled by noise, and more likely to detect true weak edges when compared with the other methods.



- Fig. 1 Example of an unacceptable low-resolution image

Line Detection

We used Hough Transform [15] to detect lines. With this transform, straight line can be drawn in between the Cartesian space and a parameter space. The advantage of using Hough Transform is that the pixel available on one line need not be contiguous with that of the other. Therefore, it is required to detect lines with short breaks in them due to noise or partially occluded by objects. Following steps have been used by us in detecting lines in an image. [16].

- Finding all the edge points in the image using any Edge Detection Method.
- Quantifying the (m, c) space into a two-dimensional matrix H with appropriate quantification levels.
- Initializing the matrix H to zero value.
- Considering each element of H matrix, H (mi, ci), which is found to correspond to an edge point is incremented by one. The result is a histogram or a vote matrix showing the frequency of edge points corresponding to certain (m, c) values (i.e. points lying on a common line).
- The histogram H is a threshold limit, where only the large valued elements are considered. These elements correspond to a line in the original image.

Lane Detection

Lane detection process uses the edge of the image, the Hough Lines [17] and the horizontal lines as input. The Hough Lines object finds Cartesian coordinates of lines that are described by rho and theta values. The object inputs are the theta, rho is the values of lines and a reference image. The object outputs the zerobased row and column positions of the intersections between the lines and two of the reference image boundary lines. The boundary lines are the left and right vertical boundaries and the top and bottom horizontal boundaries of the reference image.

Lane Tracking

Lane tracking phase is processed by matching frames from image and Kalman filter. First, the distance between the lines is found in the current frame and these are stored. Then, the best matches between the current lines and those in the stored lines are found. If a line in the stored line matches with an input line, it replaces it with the input one and increase the count number by one, otherwise, reduce the count number by one. The count number is then saturated.

Kalman filtering algorithm [18] works in two-steps. In the prediction step, the filter produces estimates of the current state variables along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average method, with more weight being given to estimates with higher certainty.

Obstacle Avoidance

Autonomous vehicles in different environment conditions are required to be capable of obstacle detection and plan an obstacle free path. In our vehicle we used LIDAR to detect obstacles and plan a path free of obstacles. LIDAR is an automatic remote sensing device that measures distance by emitting a laser and calculating the distance of reflected light. LIDAR is an acronym of Light Detection and Ranging.

The advantage of using LIDAR is the high speed and accuracy it provides in distance calculation. We used a *2-D* SICK LIDAR LMS 291 S-O5 which has a very wide-angle range of 180 degrees at a range of 0.5 degree and 8011 mm range. Every time the LIDAR scans, the array of 180-degree span can be mapped onto their respective angles to the polar coordinated (r, ϕ) of the surroundings. The LIDAR can give 12-14 scans per second.

Arrangement of camera and Lidar

Camera and Lidar are arranged on a demo vehicle which has a steering mechanism to make a turn. The arrangement of the camera is most important, as position of the camera affects a lot, to detect lanes on the road. The arrangement of LIDAR is also very important. It should be properly placed in the centre of the vehicle, only then, it can give the proper 180 degree angular view in detecting obstacles in front of the vehicle

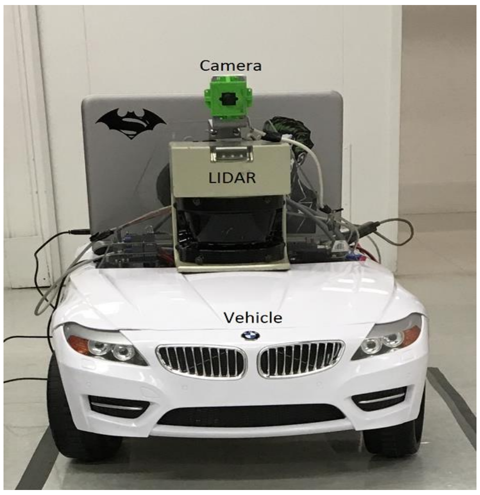
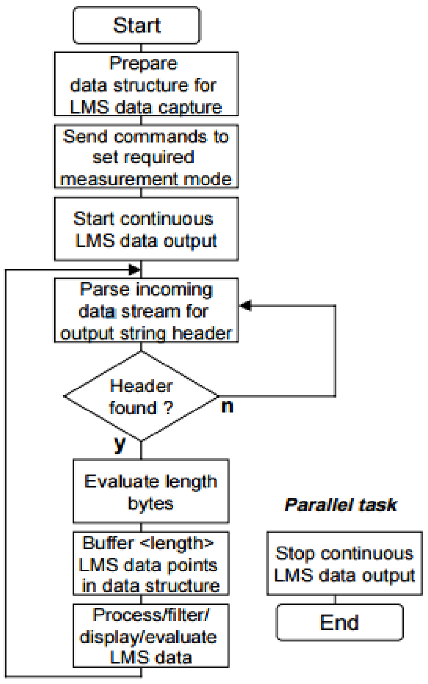


Figure 3

Figure 3 shows the experimental arrangement of camera and LIDAR mounted on the vehicle. The camera view is projecting downwards so that it is easy to detect the lanes without any noise. The camera centre and the LIDAR centre should be in line, so that we don't get confused when we look for the obstacle in the camera lane view.

Parsing LIDAR Data

The LMS LIDAR unit is supplied with 24 Volts DC +/- 15%. Depending on the length of the power supply cable, the power supply needs to provide sufficient current. For data exchange in measuring tasks, the LMS is equipped with a serial RS 232 interface. The default settings of the LIDAR are 9600 baud with angular range of 0-180 degrees at a resolution of 0.5 degrees. We change the baud rate from 9600 defaults to 38400 baud, but whenever the power is off, the LMS LIDAR goes back to its default settings and we again need to change the baud rate. So, it is better if we change the baud rate in the PC program we use. And we also change the angular resolution from 0.5 to 1 degree which increases the processing time.


Start
Prepare data structure for LMS data capture
Send commands to set required measurement mode
Start continuous LMS data output
Parse incoming data stream for output string header
Header found ?
n
y
Evaluate length bytes
Buffer <length> LMS data points in data structure
Process/filter/ display/evaluate LMS data
Parallel task
Stop continuous LMS data output
End

Interpreting/parsing the received data

The above figure actually explain the step by step procedure to parse the incoming data from the LIDAR and understand it. The only way to change the settings in the SICK LIDAR [18] is by sending set of strings to it. We need to set the required measurement mode like the SICK LIDAR has 'cm' and 'mm' mode and changing baud rate, followed by sending a continuous data output string to the device which in return sends continuous data until and unless a stop string is sent to it. The next most important step would be to look for the header in the data which is received. Notice that the header is of 7 bytes sent before actual data is sent. These 7 bytes are the packet header. It contains information about the data being sent. If you know what this header is supposed to be, you can use it to identify the beginning of the data.

In our case, this header will be:

Hexadecimal Form:	02	80	6E	01	B0	B5	41
Decimal Form:	2	128	110	1	176	181	65

The incoming data from a scan is sent sequentially as the sensor scans through 180 degree. In our case, since the sensor is set to scan 180° with resolution of 1 degree, the first data point sent will correspond to 0 degree, the next will correspond to 1 degree, the next to 2 degree, and so on. This means we should have a total of 181 data points. Each distance measurement is sent in the form of two bytes. The least significant byte is sent first, followed by the most significant byte. Since we're operating in mm mode, the unit for the measurements will be in mm.

Now after getting enough variable to calculate the 180 degrees, we can take the first 2 bytes and perform the following operation.

1st Byte + (2nd Byte *256) = Distance –mm

Since each measurement is represented by 2 bytes, and there a total of 181 data points, the incoming data stream should have a total of 362 bytes. This is important for choosing the buffer size and for parsing the data. The final step would be to send the Stop string to the device, as only then, it stops sending commands to the PC.

Obstacle Detection

In practice, we base our work on two assumptions:

1) All objects are rigid.

2) The instantaneous curvature of the object trajectories (i.e., the ratio between their angular and translational velocities) is small enough to assume that its motion is purely translational over short time intervals. Hence, the translational velocities of all points on an object are identical and equal to that of its centroid. Both assumptions are plausible for the projection on the ground of walls, most vehicles and even pedestrians.

The obstacle points could be present anywhere in the environment, but it is important to ignore those points that would not disturb the path of the vehicle [19]. Anyways, if these obstacles are directly in the path of the vehicle, they can be called as an obstacle. The method of detecting the obstacle involves studying a certain degree-span of readings directly in front of the vehicle which are calculated based on a safe-distance and the minimum clearance required for the vehicle to pass safely

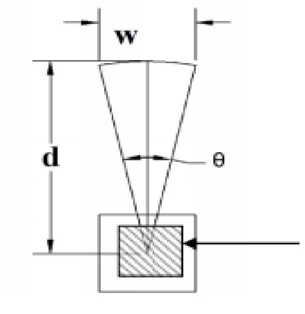


Figure 3: Obstacle Scanning Range

If θ is the range to be studied, w is the clearance width and d is the safe-distance, then:

$$\text{Cos}\ (\theta/2) = d / g \qquad \text{where } g = d2 + w2 /4$$

Path Planning

In a scan we get 180 degrees although the optimum path is to be selected within 120 degrees which we further divide into 6 parts with 20 degrees in each part. As shown in figure 4.

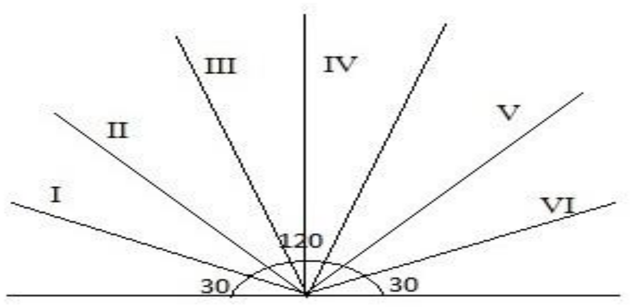


Figure 4

When there is an obstacle in any part we look for the tail in the adjacent or the next part, through which we can estimate the size of the obstacle and can predict the turn need to be made by the vehicle to avoid the obstacle. If no obstacle is detected we ask the matlab problem to select the longest free line on which the robot can traverse.

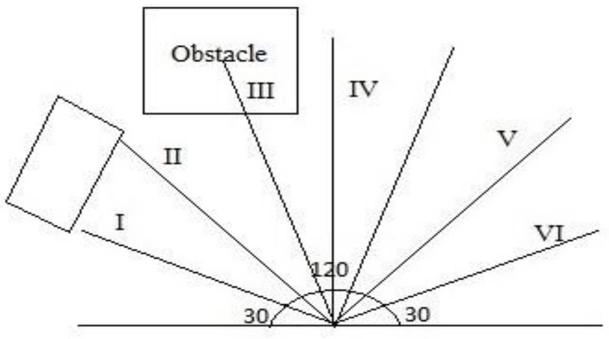


Figure 5

As shown in figure 5 if there is an obstacle in part I, II or III, the robot to look for obstacle free path in part IV,V and VI, and if it is an obstacle free path as shown in figure 5, we can make enough turn towards part IV,V and VI to avoid the obstacle. The minimum distance to detect the obstacle is 0.3 m that is based on the turning radius of the vehicle as it needs to avoid the obstacle from hitting before it makes a good turn.

Lane and Lidar combination.

Now we have the proper program to detect the lanes and obstacles in front of the vehicle. If both the lanes and obstacles are combined, vehicle should make a decision to select the best between lane and obstacle. As shown in the figure 6, if there is an obstacle in left lane and part I, II, III or IV. Then the vehicle should look for the obstacle in remaining parts, and if it is an obstacle free place it should move until it clears the obstacle. The next step would be to find the offset of the lane and make the vehicle straight to the lane by moving forward. In this way the combination of lane and obstacle detection is done in this report.

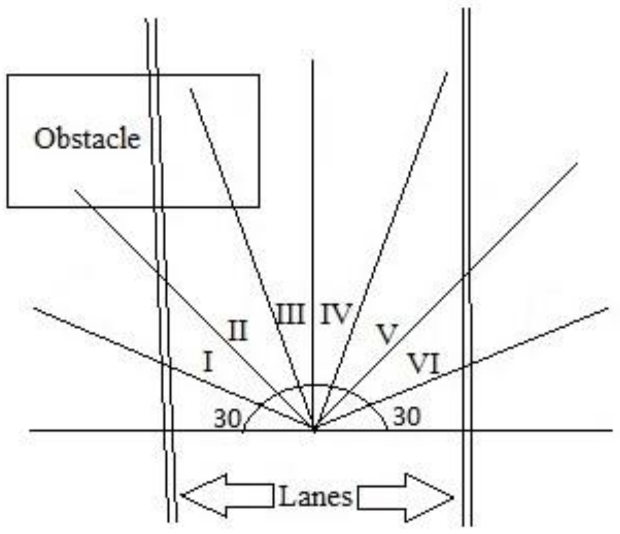


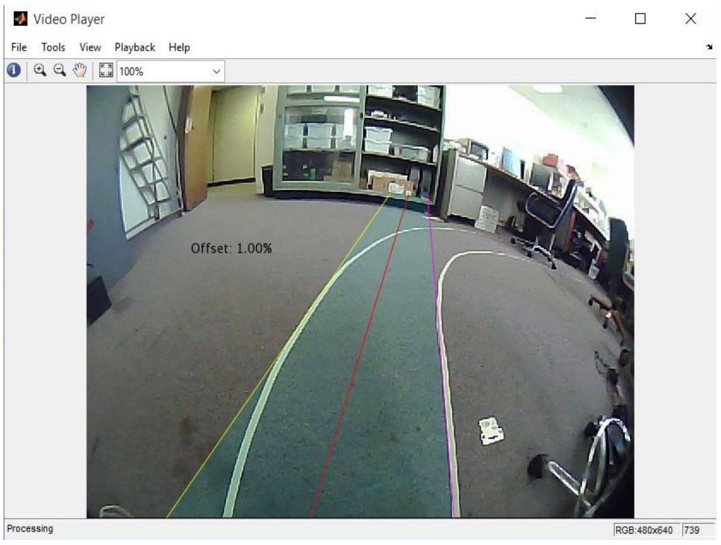
Figure 6: Lane and Lidar

Once the best path is chosen, the vehicle **<u>yaws</u>** in the said direction and begins traversing the intended path. This entire procedure is run every 0.12 seconds and hence leads to a smooth trajectory assumable around the obstacles.

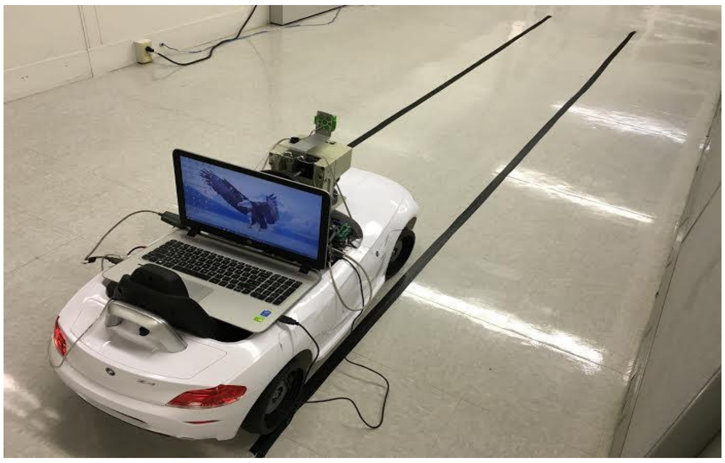
IV. EXPERIMENTAL RESULTS

This paper explains a lane detection with obstacle avoidance system based on image processing and computer vision methods. Input to this system is streams of video captured by the single camera mounted on the vehicle. The input image was divided horizontally into two sub-frames. The main intention behind dividing the frame, was to reduce the image processing time and to set a field of view to the lower part of the frame, which is needed to detect various types of lane marks such as solid line, dashed line, etc. There was difficulty detecting lane marks on various types of lighting conditions due to shadows. This can be equalized by applying filtering techniques. 2D FIR filter is applied at the input image data to remove noise and to detect edge of the lane. Filtered image is then applied with automatic thresholding to convert the input image from grayscale to binary image that detects lane markers under various lighting conditions. Then by finding the left and right lane markers is done by using Hough Lines and Hough Transform. This system detects lane markers by matching the current lane markers with the markers from previous video frame and by using Kalman filter. Based on the classified lane marks and the offset value, the position of the vehicle is determined whether to go straight or to take a turn.

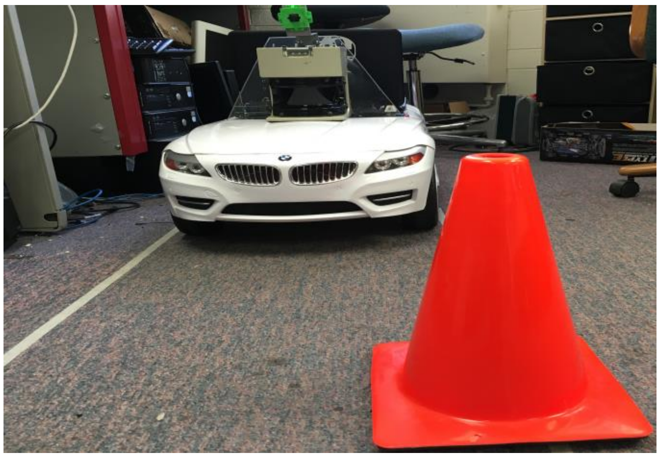
In obstacle detection, an incoming stream of LIDAR data is sent in a scan, where we get 180 degrees, although the optimum path is to be selected within 120 degrees which we further divide into 6 parts with 20 degrees in each part to define the field of obstacle. If an obstacle is detected in any part, we look for the tail in the adjacent or the next parts. Through which we can determine the distance and angle of the obstacle that can predict enough room to avoid obstacle .After determining the path, the vehicle **yaws** in the defined direction and begins traversing the intended path. This entire procedure is repeated every 0.12 seconds and hence leads to a smooth trajectory is being assumed around the obstacles.



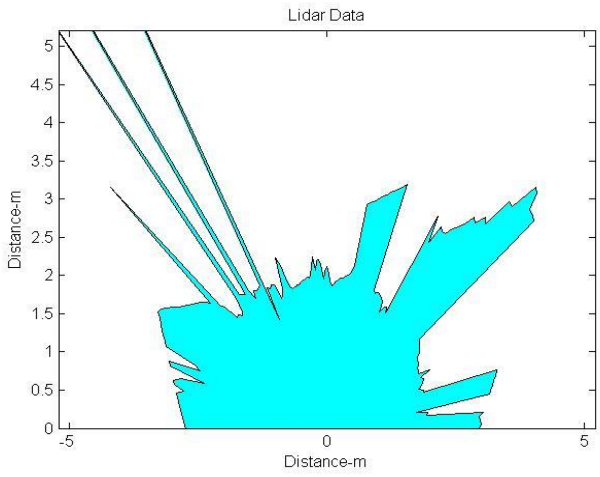
(a)



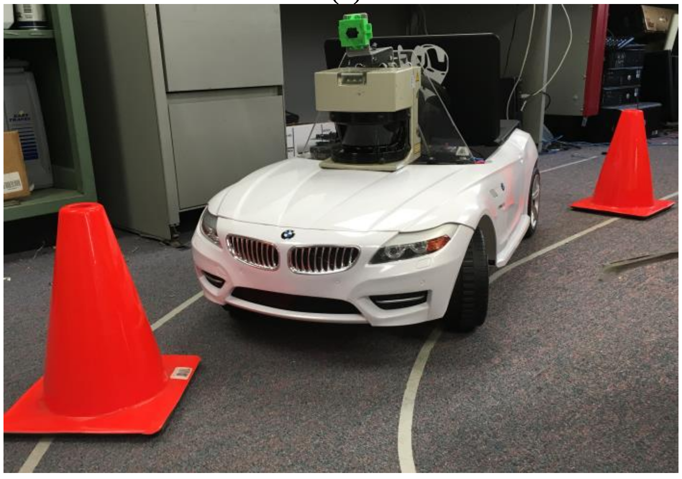
(b)



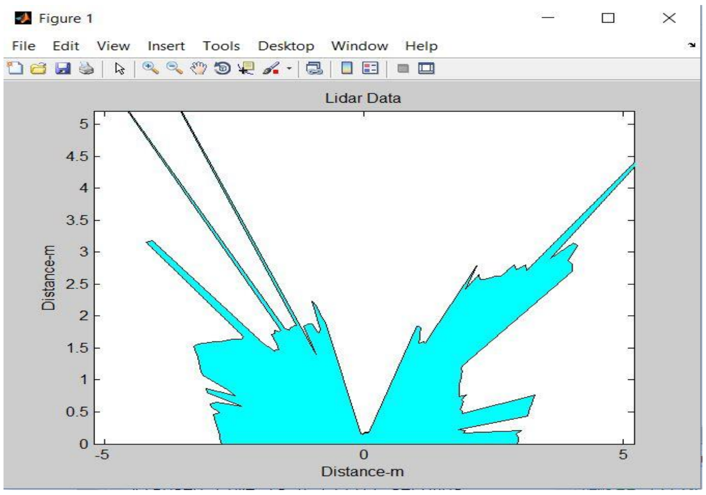
(c)



(d)



(e)



(f)

Fig. 4 illustrates the lane and obstacle detection algorithm results (a) shows lane detection on straight lane displaying offset value (b) shows the experimental test scenario conducted for lane tracking (c) shows the experimental Vehicle mounted with LIDAR and cameras sensor In front of an obstacle. (d) Shows laser scan graph plotted from LIDAR data. (e) Shows the obstacle avoidance by the vehicle in a course. (f) Shows laser scan data when an obstacle is detected in front of it.

V. Conclusions

This paper discussed above reported Lane Detection and Obstacle Avoidance System. In this system we go frame by frame to calculate the lanes and distance of the obstacle from the vehicle. The detection of lane marks and lane boundaries are proposed using Hough Transform and Hough Lines. This system detects lane markers by matching the current lane markers with the markers from previous video frame and by using Kalman filter. Based on the classified lane marks and the offset value, the position of the vehicle is determined. as to whether to go straight or to take a turn. The Obstacle Detection is achieved by using a 2-D Lidar scanner. The Lidar sends us the raw information in ASCII Hex variables which we convert into distances and plot them on a Cartesian plot using Mat lab software.

The proposed system can detect road lanes and obstacles in front of it. This system can detect lanes and objects for 10-13 frames per second. The developed system can be used at real-time video processing and also from a recorded video camera.

The limitations of this system is its inefficient detection at low lighting as we use a fish-eye wide angle (180) camera view. Future awaits the development of a system that can eliminate the error of low lighting and detect lanes in any environmental conditions.

REFERENCES

[1] Wally Chen, Leon Jian, Sy-Yen Kuo, “Video-based on-road driving safety system with lane detection and vehicle detection”, 2012 12th International Conference on ITS Telecommunications, Taiwan, 05-08 November 2012, pp. 537-541.

[2] Seonyoung L., Haengseon S., Kyungwon M. Implementation of lane detection system using optimized hough transform circuit, 2010 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 406-409.

[3] Jia H., Hui R., Jinfeng G., Wei H. A Lane Detection Method for Lane Departure Warning System, International Conference on Optoelectronics and Image Processing (ICOIP), 2010,

[4] Vijay G., Shashikant L., An Improved Lane Departure Method for Advanced Driver Assistance System, International Conference on Computing, Communication and Applications (ICCCA), 2012,

[5] Interfacing GPS tutorial https://learn.sparkfun.com/tutorials/gps-basics

[6] M. Haloi, ”A novel pLSA based Traffic Signs Classification System”, eprint arXiv:1503.06643, 2015.

[7] N. Srinivasa et al, A fusion system for real-time forward collision warning in automobiles, IEEE Intell. Transp. Syst., vol. 1, pp. 457462, 2003.

[8] M. Aly, Real time detection of lane markers in urban streets, IEEE Intelligent Vehicles Symposium, 2008.

[9] Burak Benligiray, Cihan Topal, Cuneyt Akinlar, Video-based lane detection using a fast vanishing point estimation method, 2012 IEEE.

[10] Nasim Arshad, Kwang-Seok Moon, Seung-Seob Park, and Jong-Nam Kim. Lane detection with moving vehicle using color information. WCES Vol I, Ocotober 2011.

[11] David Hanwell and Majid Mirmehdi, “Detection of lane departure on high-speed roads”. ICPRAM 2012.

[12] http://www.mathworks.com/help/releases/R2012b/vision/ref/2dfirfilter.htm

[13] Rafael C. Gonzalez and Richard E. Woods, Digital image processing, 3rd edition, Pearson, 2008.

[14] Image and video processing edge detection in intensity image references http://www.mathworks.com/help/images/ref/edge.html?refresh=true

[15] B. M. John, N. Donald, “Application of the hough transform to lane detection and following on high speed roads”, signal &system Group, Department of Computer Science, National University of Ireland, 1999,pp.

[16] Ghassan Hamarneh , Karin Althoff, Rafeef Abu-Gharbieh, “Automatic line detection”, Project Report for the Computer Vision Course, September 1999.

[17] http://www.mathworks.com/help/releases/R2012b/vision/ref/vision.houghlinesclass.htm

[18] Interfacing with sick LIDAR LMS-291 http://www.pages.drexel.edu/~kws23/tutorials/sick/sick.html

[19] Iwan Ulrich and Johann Borenstein, VFH*: Local Obstacle Avoidance with Look-Ahead Verification, IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000, pp. 2505-2511.