Mehran University Research Journal of Engineering & Technology
Vol. 38, No. 1, 113-128, January 2019
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.1901.10

Sensor-Fusion Based Navigation for Mobile Robot in Outdoor Environment

MUHAMMAD YOUSAF ALI KHAN*, EHTASHAM MUSTAFA*, AAMIR NAWAZ*, NASIR SALEEM*, AND USMAN ILLAHI*

RECEIVED ON 12.10.2017 ACCEPTED ON 12.02.2018

ABSTRACT

Autonomous navigation of the vehicles or robots is very challenging and useful task used by many scientists and researchers these days. By keeping this fact in mind, an algorithm for autonomous navigation of mobile robot in outdoor environment is proposed. This navigation track consists of colored border containing obstacles and some unplanned surfaces along with some specific points for GPS (Global Positioning System) alarms. The main goal is to avoid colored border and obstacles. For the vision problem webcam is used. First the colored border is detected by using OpenCV library by following HSI (Hue Saturation Intensity)technique. The Canny edge algorithm is used to find both edges of the border, then for detecting straight lines on both sides of track, Hough transformation is used. Finally, the closest border line is detected and its center point is calculated for which the mobile robot has to steer to avoid it. Second step is to avoid the obstacles, which is done by LRF (Laser Range Finder), first the range of LRF is defined, because not all obstacles have to be avoided, only those obstacles are detected and avoided which are in specified range as defined before, and finally GPS receiver is used to make alarms at some specific points. As a result, a successful navigation of the mobile robot in the outdoor environment is implemented.

Key Words: Mobile Robot, Navigation, Vision, Laser Range Finder.

1. INTRODUCTION

Autonomous Navigation of Mobile Robot is a very hot topic among the engineers and scientists who are working in the field of mobile robots. In this paper we proposed a method to move autonomously the mobile robot for some specific trackin the outdoor environment.

Some researchers use combination of precise localization of GPS and laser in the urban and non-urban environment [1]. Some authors adopted vision system and encoder to localize the target approximately by introducing the concept of decision-making space, forward passageway [2]. Some propose a novel-based navigation method in contrast with appearance-based approaches. This algorithm is based on motion estimation by a camera to plan the next moment of a robot and robust feature matching to recognize home and destination locations [3]. Some use mobile robot by control station and control

Authors E-Mail: (myousafak@gu.edu.pk, ehtashammustafa@gu.edu.pk, aamirnawaz@gu.edu.pk, nasirsaleem@gu.edu.pk, usmanillahi@hotmail.com)
* Department of Electrical Engineering, Gomal University, Dera Ismail Khan, Pakistan.

this robot by receiving the image data from camera by sending to the control station [4]. Some use the differential GPS and odometry data by detecting the curbs by laser range finder and building the map [5]. For boundary detection, many vision techniques have been used by using image segmentation algorithm and have shown the experimental results [6]. Tracking the people for mobile robot in motion is discussed in [7]. The kinematics of 4-wheeld mobile has been derived by using two 2-wheeled mobile robots [8]. Motion in skid-steering mobile robots is achieved by varying the speed on opposite sides of wheels [10]. For vision some reference use hybrid localization approach which switchs between local, in strong illumination changes and global image features [11]. Some authors have used both the vision and laser data for tracking moving obstacles like humans on outdoor environment depending on the speed of the obstacles [12].A hybrid approach is used to decrease the computational time of type-reduction process [13]. The navigation is mainly dependent on the parameters like perception, localization, cognition and motion control [14]. The WLPS (Wireless Local Positioning System) provides local positioning with sufficient coverage, reliability and accuracy [15].

This paper has two main portions. In the first portion, hardware design of mobile robot is explained, and in second portion, navigation algorithm is proposed.

For the hardware, 4-wheeled skid steering mobile robot is used. These 4-wheels are controlled by four BLDC (Brushless DC motor) having four motor drivers [9]. It also has GPS receivers which provide one with the latitude and longitude of the mobile robot platform by using WGS-84 geographic coordinate system. A computer laptop is used as a controlling device. The Logitech 2MP autofocus webcam is used for vision propose. The real-time EtherCAT network is used. For the purpose of obstacle detection and avoidance, Sick LMS (Model Name) 200 laser Range Finder is used. The power is supplied by 24V 18Ah sealed lead acid battery.

For the navigation algorithm, the kinematics of mobile robot is derived. For vision purpose, OpenCV library is used. The common technique used to detect colors is RGB (Red, Green, Blue) but here we have used HSI, where H is Hue, S stands for saturation, and I stand for intensity because the reflection of light is reduced in this system. First, we take the binary image (0,1). To detect the both edges of colored track (border), canny edge algorithm is used. To detect the straight border lines on both sides of the track and then finding closest line by finding its center point is done by Hough transformation. For obstacle detection LRF is used.

The Navigation track was about 150m in length consisting of obstacles, unplanned surface, and speed breakers. Also it has some specific positions for GPS alarms. There was a colored border of the track. The successful navigation was to pass through the navigation track without striking the obstacles, without moving it out of the track and making GPS alarm at some specific positions. Fig. 1 shows the sample navigation track in which we can see the border, obstacles, and some slope paths and Fig. 2 shows the size of obstacle.

Finally, by using this algorithm and kinematic design, navigation of the mobile robot in the outdoor environment has been successfully implemented.

2. KINEMATICS

2.1 Kinematics of the 4-Wheeled Mobile Robot

The kinematic description of a 4-wheeled mobile robot is shown in Fig. 3. Basically it is obtained by combining two 2-wheeled mobile robots [8].

v_1, v_2, v_3, and v_4 are the velocities of the four wheels 'v' and 'w' are the linear and angular velocities of the mobile robot platform respectively, 'd' is the distance between two opposite wheels, and θ is the angle between the wheel and the perpendicular axis of each 'd'.

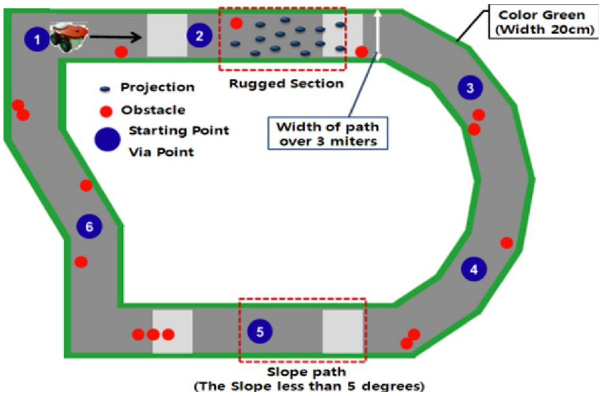


FIG. 1. NAVIGATION TRACK

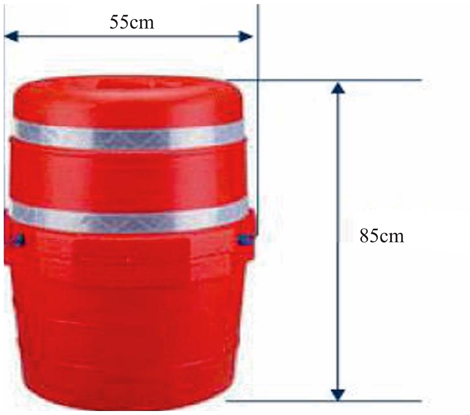


FIG. 2. OBSTACLE

The kinematic equations of the 4-wheeled mobile robot are shown below:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{-\cos\theta}{2d} & \frac{\cos\theta}{2d} \end{bmatrix} \begin{bmatrix} v_1 \\ v_4 \end{bmatrix} \qquad (1)$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{\cos\theta}{2d} & \frac{-\cos\theta}{2d} \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \end{bmatrix} \qquad (2)$$

Inverting Equations (1-2), we have

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1 & \frac{-d}{\cos\theta} \\ 1 & \frac{-d}{\cos\theta} \\ 1 & \frac{-d}{\cos\theta} \\ 1 & \frac{-d}{\cos\theta} \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \qquad (3)$$

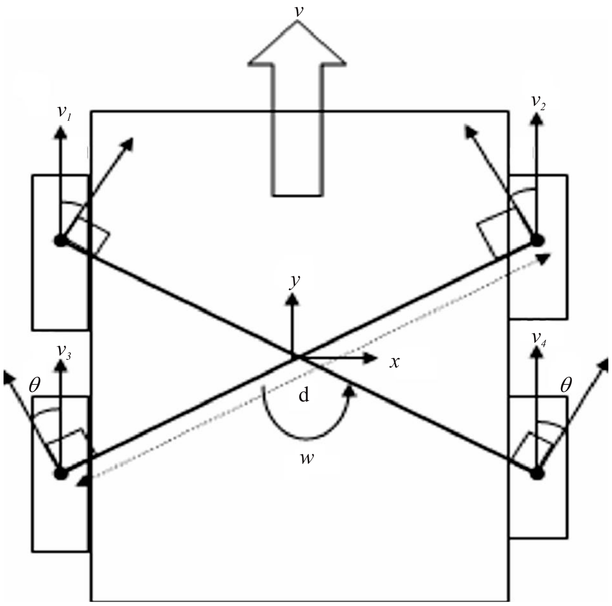


FIG. 3. KINEMATIC DESCRIPTION OF 4-WHEELED MOBILE ROBOT

3. MOBILE ROBOT HARDWARE

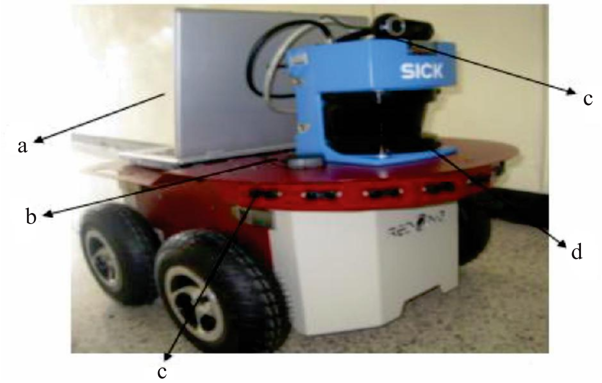
3.1 Mobile Robot Platform

Fig. 4 shows a differential type skid-steering mobile robot which is used for the experimentation and navigation. In such mobile robots, steering is achieved by differentially varying the speed of the lines of the wheels on different sides of vehicle [10]. BLDC motors have been chosen for motion because of their high performance. Skid steering is accomplished by creating a differential velocity between the inner and outer wheels. This mobile robot has been provided by Redone Technologies, Co. Ltd [6]. This mobile robot is redundantly actuated, so if one or more of its motor got problem still it will work.

This mobile robot platform consists of 4 skid-steering wheels having four BLDC motors. It has notebook PC as a control station. Sick LMS 200 Laser Range Finder and Logitech 2MP autofocus cameras are also installed on the platform. It also has a GPS receiver.

3.2 Control Network System

EtherCAT is one of the industrial EtherNET communication technique applied to internal control system of this mobile robot. In EtherCAT, each component of the mobile robot (i.e. motor, sensors, GPS, etc.) can be overall controlled. Fig. 5 shows the structure of the internal control system of the mobile robot. EtherCAT for internal control network of mobile robot platform is an advanced technology of communication with transmitting speed of 100Mbps. EtherCAT is the optimal network solution for intelligence robot, field robot, industrial robot, and motion systems by using discrete clock synchronization.



a. Computer Laptop
b. PSD-IR Sensors
c. Logitech 2MPAuto Focus Vision camera
d. GPS (Global Positioning System)
e. Sick LMS 200 Laser Range Finder

FIG. 4. MOBILE ROBOT PLATFORM

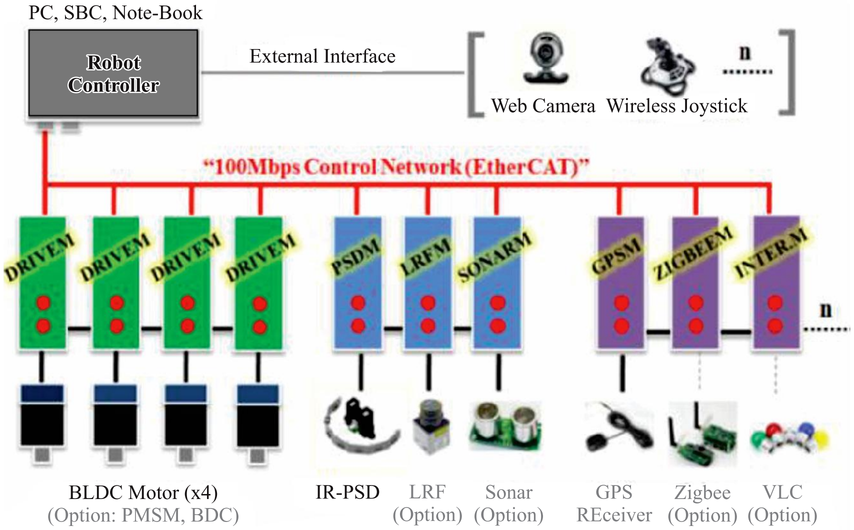


FIG. 5. INTERNAL CONTROL STRUCTURE OF THE MOBILE ROBOT

Internally installed modules of the mobile robot system are controlled through high speed EtherCAT network. It is easy for each module to be changed, replaced, and reconstructed. It can recognize the replaced modules on network without resetting addresses, baud rates, and so on.

4. VISION SYSTEM FOR DETECTING COLORED BORDER

In vision process OpenCV library is used. The steps in the vision process are shown in the flow chart in Fig. 6.

In the vision process, following steps have been done on the original image.

4.1 Median Filtering

In image processing, it is usually necessary to perform high degree of noise reduction in an image before performing higher-level processing steps, such as edge detection. The image filter is a non-linear filtering technique, often used to remove noise from images or other signals.

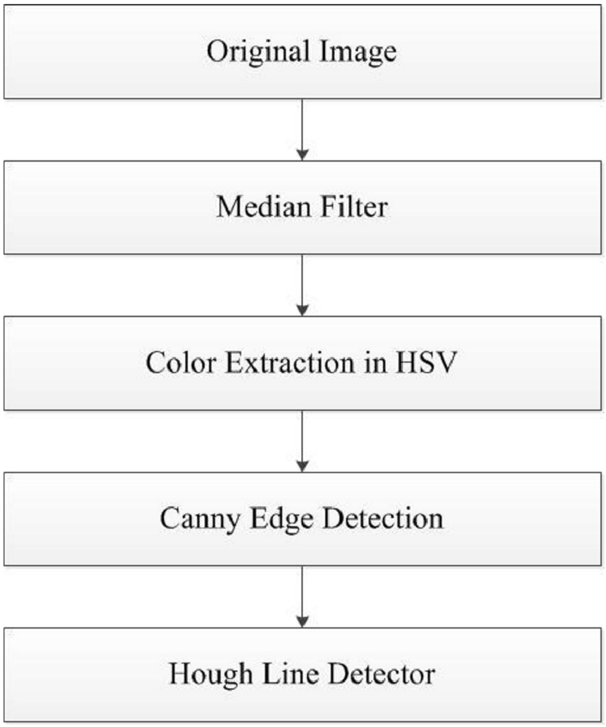


FIG. 6. FLOW CHART OF VISION PROCESS

Median filtering is a common step in image filtering. It is particularly useful to reduce noise and salt and pepper noise. Its edge-preserving nature makes it useful in cases where edge blurring is undesirable.

4.2 Color Extraction in HSV

There are two main kinds of color models. RGB, and the other one is HSI. Here, we have used HSI model because it is more tolerant to the intensity of light and also has more accurate color relationship as compared to RGB. Fig. 7 shows RGB and Fig. 8 shows HSI color models.

4.3 Canny Edge Detection

The Canny edge detection characterizes boundaries and is therefore a fundamental step in image processing. Edges in the images are areas with strong intensity contrast. Edge detection of an image significantly reduces the amount of data and filter out useless information, while preserving the important structural properties in an image.

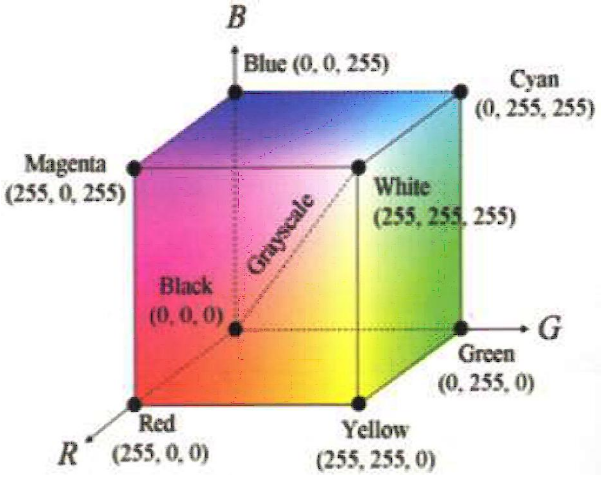


FIG. 7. RGB COLOR MODEL

4.4 Hough Transformation

The Hough Transformation is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of this technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.

4.5 Vision Results

The original picture taken by vision camera is shown in Fig. 9(a) which shows the track in green color as well as the obstacle in red color. The picture taken after the HSI (Hue Saturation Intensity), color extraction is shown in Fig. 9(b), where we can see the border very clearly. The picture after canny edge detection is shown in Fig. 9(c), where we can see both edges of the track. The picture taken after Hough Transformation is shown in Fig. 9(d), here we can see the edges of the track in terms of straight lines which is done by Hough Transformation.

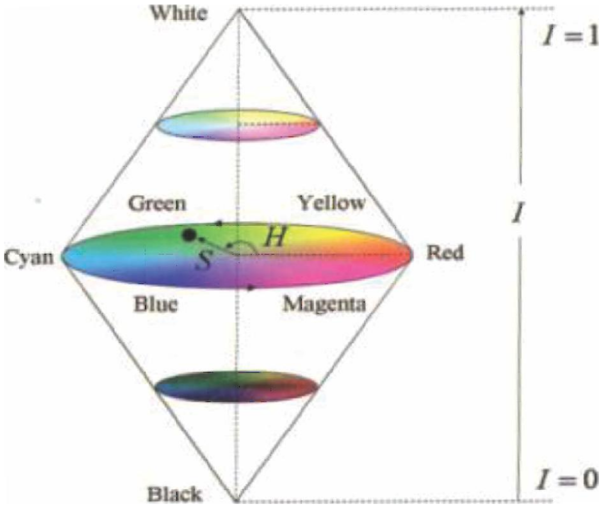
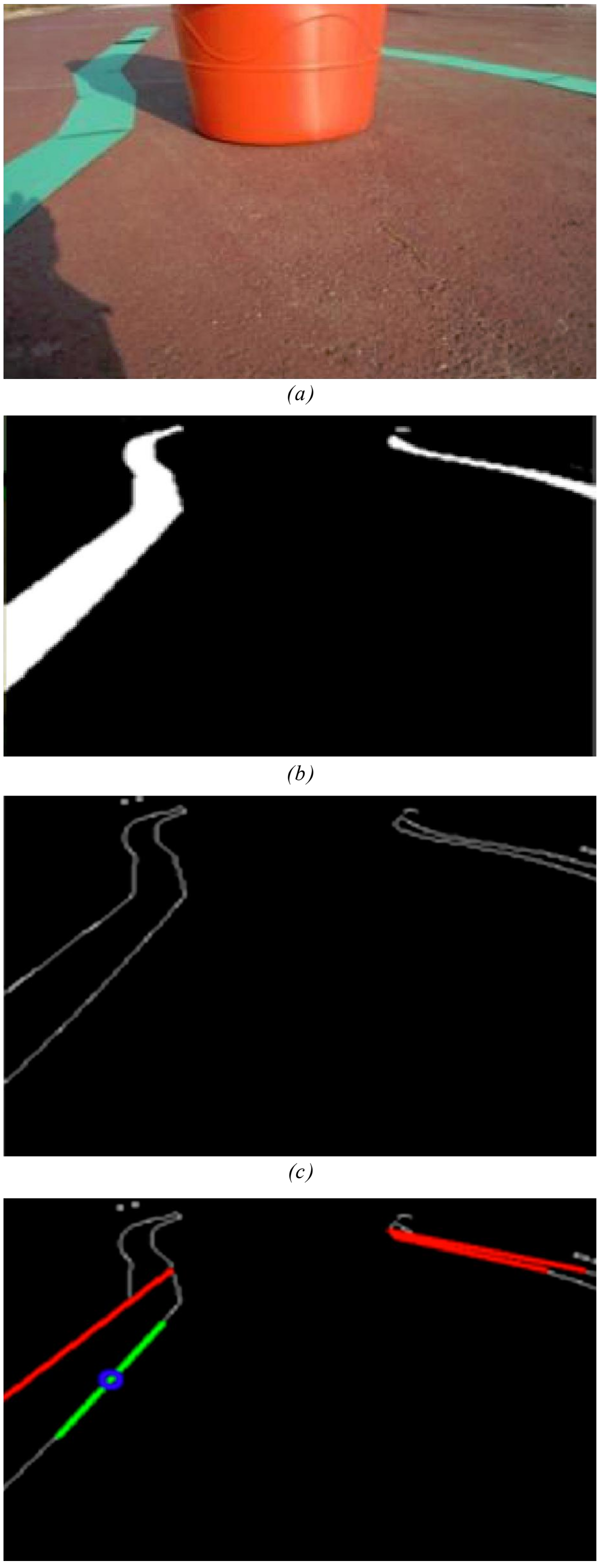


FIG. 8. HSV COLOR MODEL


(a)

(b)

(c)

(d)

FIG. 9. IMAGES OF DIFFERENT STEPS IN VISION PROCESS

Now after detecting the lines of the borders on both sides, we need to find out the closest line because generally the border lines are on left side, right side or in the front side of mobile robot. After finding the closest line we have to find out centre of that closest line so that we can steer accordingly. The distance of the closest line is expressed in terms of pixels or picture element. The picture elements are represented in terms of rows and columns. The first element represents the row and the second element represents the column. The value can be presented in the form (x,y) which are discrete quantities.

The way to find out the distance of closest line is discussed with the help of Fig. 10.

After detecting the closest line and then finding its centre point we need to know the direction of the line that whether this line is on the left, right or in the front. For that we have to find out the slope of the line and then by comparing the slope of that line with some specific value. Now if the slope of the line is in between -0.25 and +0.25 the closest line is located in the front side. If the slope of the line is greater than 0.25, then the closest line is located on the left side, and if the slope of the line is less than -0.25 then the closest line is on the right side.

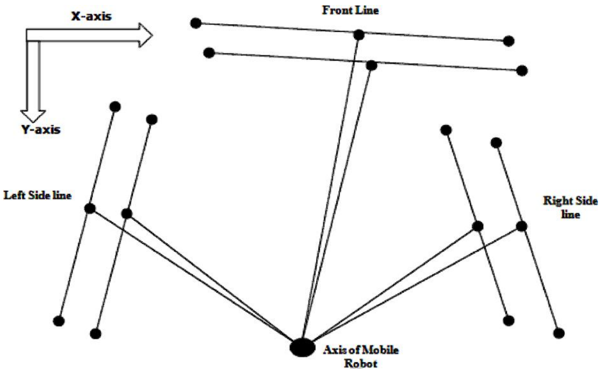


FIG. 10. DIAGRAM TO FIND THE CLOSEST BORDER LINE IN VISION PROCESS

If the center of closest line lies within some specific range, then the mobile robot needs to avoid it otherwise it has to go straight. In Fig. 11 we can see the range of the line.

We can see in Fig. 11 that if the line lies within the range of 200 Pixels then the mobile robot has to avoid it otherwise it will go straight.

Now we have to calculate how much it has to steer after finding the closest line. Basically, we have two types of velocities for mobile robot, the linear velocity ‘v’ and the angular velocity ‘w’. The maximum linear velocity of this mobile robot is v = 30cm/sec and the maximum angular velocity is w = 0.4rad/sec. If it finds the obstacle within the range of 200 pixels it has to calculate its own linear and angular velocity on the basis of the distance of obstacle from the mobile robot in order to steer it.

The formula in case of this mobile robot for both velocities is given by:

$$\text{Angular Velocity } w = w_{max} \text{ x } (200\text{-}d)/200 \quad (4)$$

where w_{max} = 0.4 rad/sec and ‘d’ is the distance between mobile robot and obstacle at that specific point.

$$\text{Linear Velocity } v = v_{normal} \text{ x } d/200 \quad (5)$$

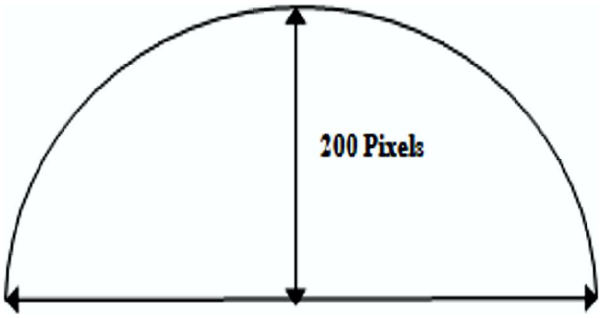


FIG. 11. DIAGRAM TO SHOW THE RANGE OF THE BORDER LINE

where v_{normal} =30cm/sec and 'd' is the distance between the mobile robot and obstacle at that specific point.

5. LASER RANGE FINDER FOR OBSTACLE DETECTION

So far, the colored border has been detected by vision data using a vision camera. However, the performance of the vision-based technique is very sensitive to the conditions of camera setting such as view point and angle of pixels, and so on. Moreover, the common change of illumination and weather conditions are another major obstacle to the reliability and robustness of vision system so to detect obstacles LRF is used [7]. Fig. 12 shows how the obstacle is detected within the range of LRF.

Fig. 12 shows how the obstacle is detected when it is entered inside the range of LRF. First the obstacle is appeared inside the range of LRF and then it starts moving towards the centre of the LRF range then the robot has to avoid it and finally the obstacle is disappeared from the range of LRF. The data obtained by LRF is in cylindrical coordinate system having the value of (r,θ).

In Fig. 13 we can see how we can find the center point of detected obstacle.

The small circles on the big circle denote the points scanned by LRF. And p_1 and p_2 are the start and end points of the obstacle. The main thing here is how the LRF should know which is the start point p_1 of the obstacle and which is end point p_2 of the obstacle. The start point p_1 is detected when the LRF data is suddenly reduced by a specific amount if the obstacle comes within the specific LRF range. That is, in this case if the horizontal data is decreased below 1m and vertical data is decreased below 4m then it means it has detected an obstacle. Now after detecting the start point of the obstacle, the diameter of the obstacle is defined and every time the array data is compared with the diameter of the obstacle. If the array data is increased layout the diameter of obstacle, then the point before that point is the end point p_2 of the obstacle.

The centre point p_m of the obstacle is computed by considering the geometry by the following equation.

$$\begin{pmatrix} x_{obs} \\ y_{obs} \end{pmatrix} = \begin{pmatrix} x_{p1} \\ y_{p1} \end{pmatrix} + \begin{pmatrix} r_c \cos(\alpha+\beta) \\ r_c \sin(\alpha+\beta) \end{pmatrix} \tag{6}$$

Where r_c is the radius of the obstacle, and x_{p1}, y_{p1}, α, and β can be calculated by the formulas shown below:

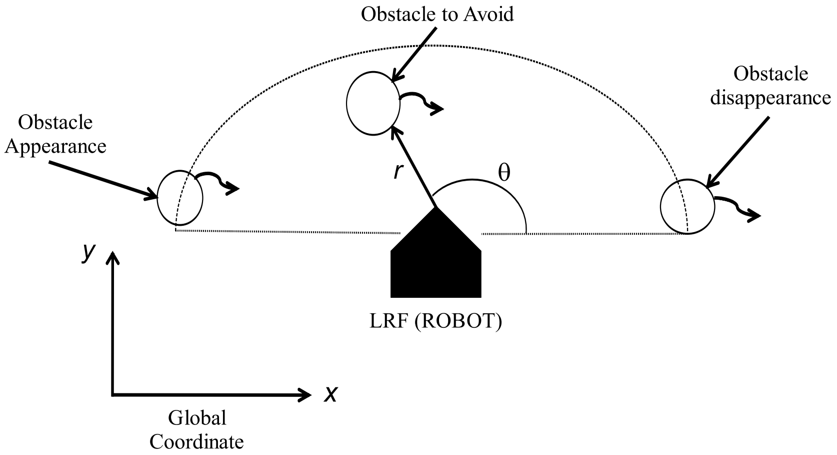


FIG. 12. OBSTACLE DETECTION WITHIN THE RANGE OF LRF

$$x_{p1} = r_1 \cos q_1 \tag{7}$$

$$y_{p1} = r_1 \sin q_1 \tag{8}$$

In the same way, the coordinates of p_2 can be calculated by:

$$x_{p2} = r_2 \cos q_2 \tag{9}$$

$$y_{p2} = r_2 \sin q_2 \tag{10}$$

The coordinates of the centre point p_m can be calculated by:

$$x_{pm} = \frac{x_{p1} + x_{p2}}{2} \tag{11}$$

$$y_{pm} = \frac{y_{p1} + y_{p2}}{2} \tag{12}$$

$$p_m = (x_{pm}, y_{pm}) \tag{13}$$

Now the angles α and β can be calculated as:

$$a = a \tan 2(y_{pm} - y_{p2}, x_{pm} - x_{p2}) \tag{14}$$

$$d = \sqrt{\left(x_{p1} - x_{pm}\right)^2 + \left(y_{p1} - y_{pm}\right)^2} \tag{15}$$

Radius of the obstacle and the angle β are given as: $r_c = 27.5$ cm

$$\beta = \cos^{-1}\left(\frac{d}{r_c}\right) \tag{16}$$

After finding the center point of the obstacle through Equation (6), the mobile robot is steered by that specific angle.

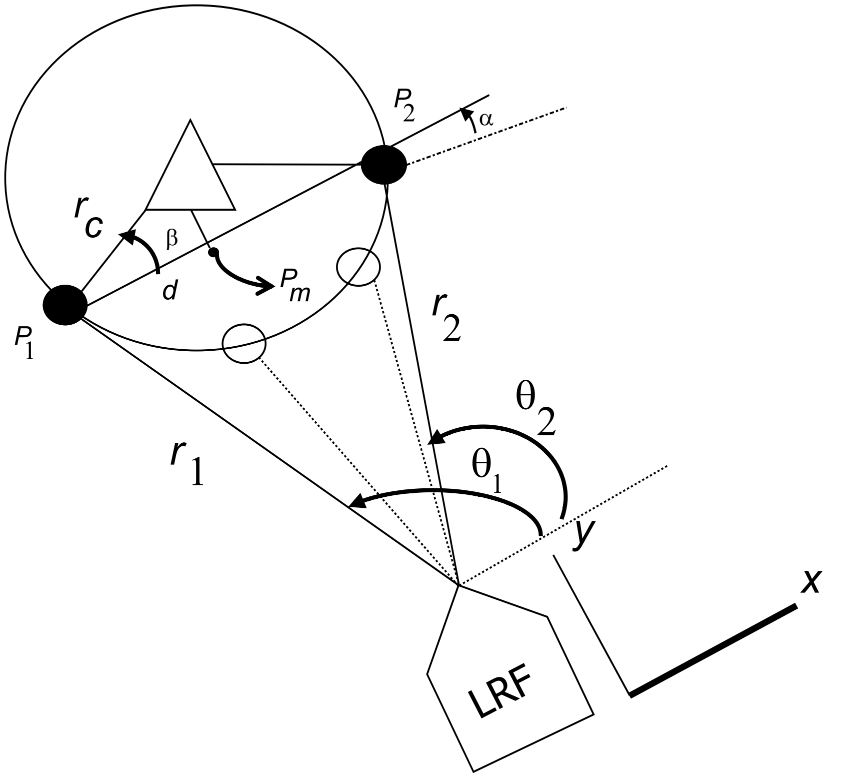


FIG. 13. OBSTACLE LENGTH & CENTRAL POINT DETECTION WITHIN LRF RANGE

As there are many obstacles within the range of LRF, we have to consider only that obstacle which is in specific range by ignoring other obstacles. This is explained in Fig. 14.

In Fig. 14, the small circles show all the obstacles within the range of LRF, but only the obstacle which is within the specific range will be considered and avoided by ignoring the other two obstacles because they are not making any hurdle for the motion of mobile robot.

The same phenomenon in the computer simulation window is shown in Fig. 15.

The red spot showing the obstacle within the specified range of LRF which will be detected and avoided and by ignoring other obstacles which are outside of that range.

6. EXPERIMANTAL RESULTS AND DISCUSSION

Now, we discuss different cases by considering different positions of the mobile robot and its behavior of avoiding the colored border. Both on the left side and on the right side of the border, four cases are considered and in each case we can see the view of the border in the vision camera and the way to avoid it.

Fig. 16 shows the mobile robot on the navigation track and Fig. 17 shows different cases of the mobile robot on the right and left side of the border. First, the cases on the left side are discussed.

6.1 Left Side

There are four cases if the mobile robot is moving on the left side of the track.

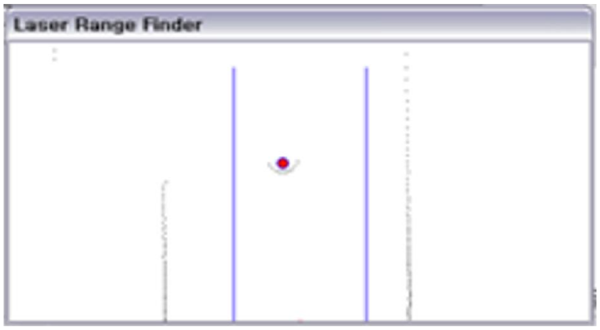


FIG. 15. RANGE OF OBSTACLES IN LRF COMPUTER SIMULATION

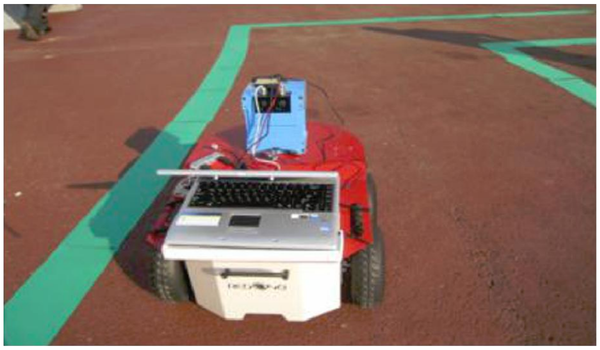


FIG. 16. MOBILE ROBOT ON NAVIGATION TRACK

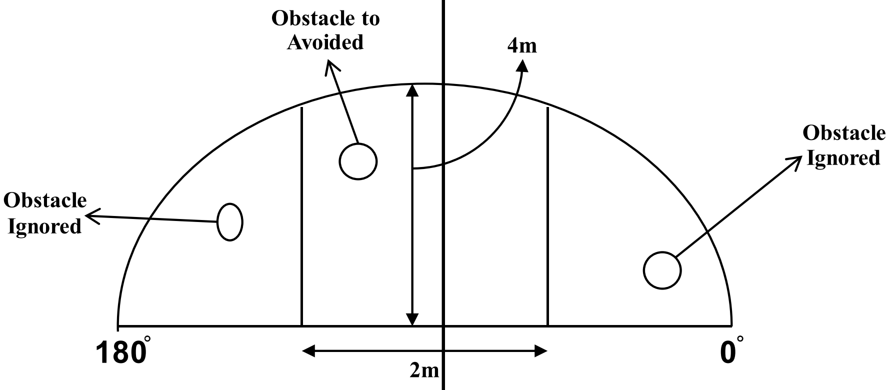


FIG. 14. RANGE OF OBSTACLE IN LRF

Case-1: We can see the vision of the camera showing the border at position 1 on the track in Fig. 18(a).

Here in this case the mobile robot will steer towards right because of detecting the border on the left side by the vision camera.

Case-2: We can see the vision of the camera showing the border at position 2 on the track in Fig. 18(b).

Here we can see that because of steering towards right side, the length of border is getting reduced within the range of vision camera, but it will keep on steering little towards right side because of the border.

Case-3: We can see the vision of camera showing the border at position 3 on the track in Fig. 18(c).

Here we can see that because of constantly steering towards right side, the length of border within the range of camera is getting reduced, but still it is inside the range of vision camera and thus it will keep on steering towards right side.

Case-4: We can see the vision of camera showing the border at position 4 on the track in Fig. 18(d).

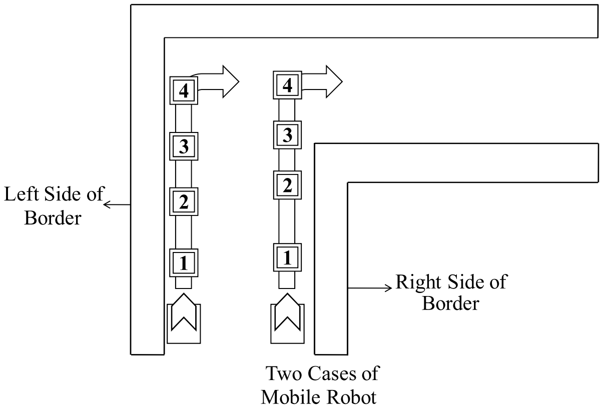


FIG. 17. DIFFERENT CASES OF MOBILE ROBOT IN VISION PROCESS

Here the vision camera is showing both borders i.e. the left side border and the front side border at the same time. Now at this point the robot has to decide whether it has to move towards left side or right side. Here the algorithm is written in such a way that if it detects both sides left side or right side and the front side at the same time, it must steer opposite to that side where is has detected the side border i.e. in the case of detecting both left side and front side border it has to steer on right side and vice versa.

6.2 Right Side

There are four cases if the mobile robot is moving on the right side of the track.

Case-1: We can see the vision of camera showing the border at position 1 on the track in Fig. 19(a).

Here the mobile robot will steer towards left because of detecting the border on the right side by vision camera.

Case-2: We can see the vision of camera showing the border at position 2 on the track in Fig. 19(b).

Here, the vision camera has detected the right side corner of the border, thus still it will keep on steering towards the left side.

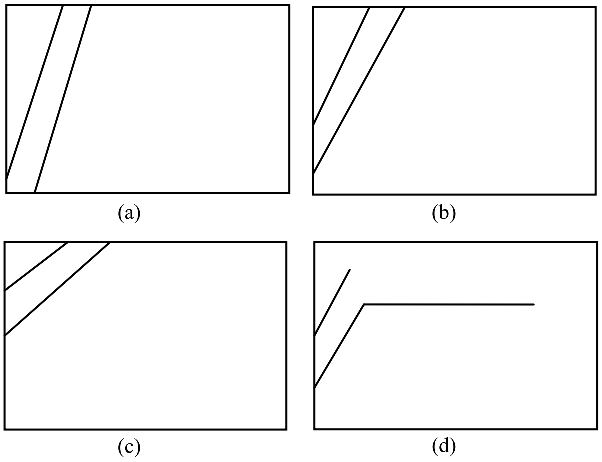


FIG. 18. DIFFERENT CASES OF VISION CAMERA WHEN MOBILE ROBOT IS ON LEFT SIDE

Case-3: We can see the vision of camera showing the border at position 3 on the track in Fig. 19(c).

Here the vision camera has not detected any border line as it has come out of the range of the right side border and has not entered in the range of any side border yet, So it will keep on moving straight until it came across any border.

Case-4: We can see the vision of camera showing the border at position 4 on the track in Fig. 19(d).

Here the mobile robot has detected the front line border. Here it has to decide whether it has to move towards left side or right side. but here the correct side is the right side, so the algorithm is written in such a way that if it comes across the situation that before detecting front line border, it has no border line within the camera range as mentioned in the case 3, it must have to steer on the same side where it has detected the last side border before getting no border data within the camera range, as it has detected the border on the right side as in case 2, it will move towards right side and vice versa.

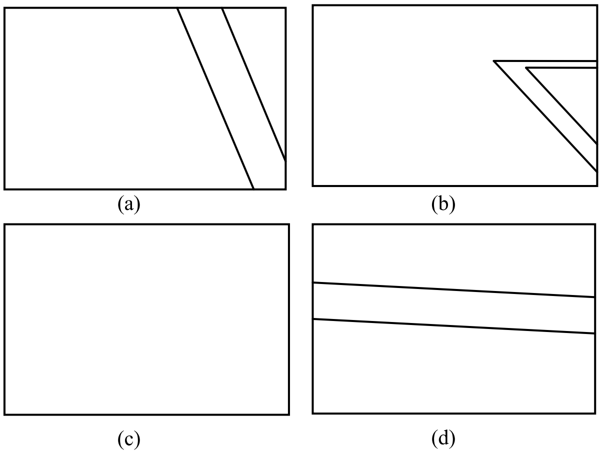


FIG. 19. DIFFERENT CASES OF VISION CAMERA WHEN MOBILE ROBOT IS ON THE RIGHT SIDE

7. VISION AND LASER RANGE FINDER

So far we have only discussed the vision process to detect the border and LRF to detect obstacles individually, but now we will combine the data of vision and LRF to avoid border and obstacles simultaneously. Different cases having both vision camera and laser range finder have been discussed.

In Fig. 20(a), we can see the mobile Robot on navigation track with obstacle and in Fig. 20(b), we have two cases whether the robot will be on the right side of the track or on the left side of the track.

Case-1: If the mobile robot is moving on the left side as shown in Fig. 20(b) and comes across obstacle, then it will behave in the same way as front side border and as before detecting front line border or obstacle it has left side border in the record and by following the same algorithm it will steer towards right side.

Case-2: If the mobile robot is moving on the right side, then it will go straight by avoiding the right side border.

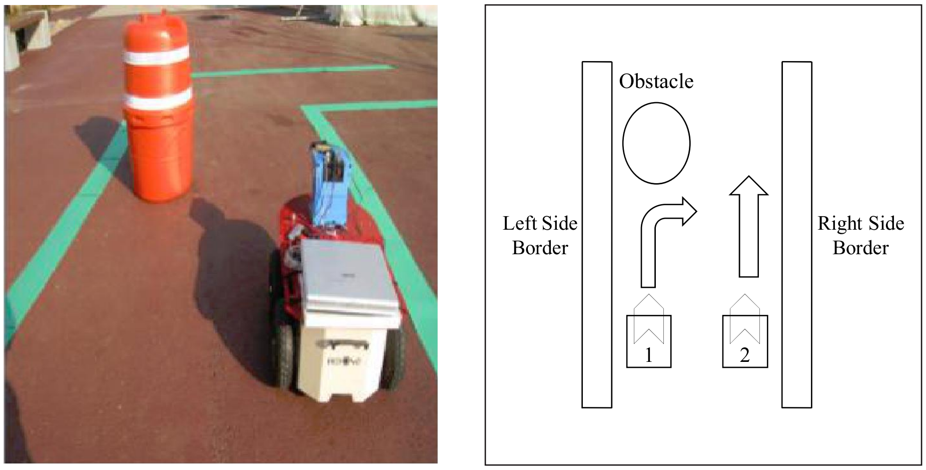
Now we can consider another case by taking the obstacles in different positions as shown in Fig. 21(a-b).

Case-1: If the mobile robot is on the left side, it will go straight by avoiding the border.

Case-2: If the mobile robot is moving on the right side of border and it comes across an obstacle, then it will consider it as a front line border and by following the algorithm it will steer towards left to avoid the obstacle.

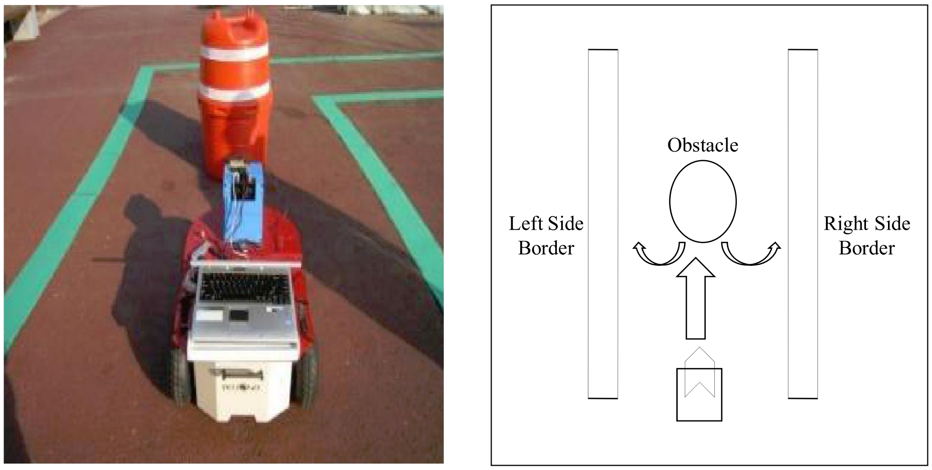
Now we can see another case of mobile robot with obstacle in Fig. 22(a-b).

In Fig. 22(b), the mobile robot can follow any direction depending on the situation and distance between the obstacle and the border.



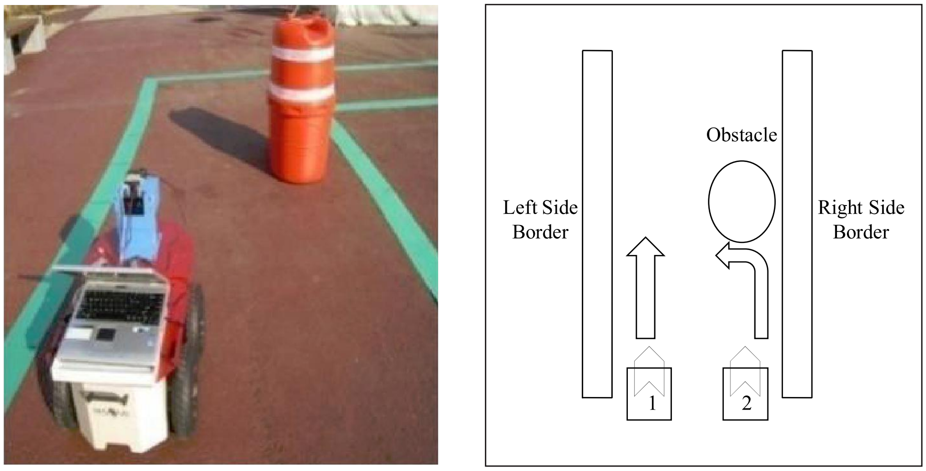
(a) (b)

FIG. 20. MOBILE ROBOT ON NAVIGATION TRACK WITH OBSTACLES



(a) (b)

FIG. 21. MOBILE ROBOT ON NAVIGATION TRACK WITH OBSTACLES



(a) (b)

FIG. 22. MOBILE ROBOT ON NAVIGATION TRACK WITH OBSTACLES

8. COMPUTER SIMULATION WINDOW OF NAVIGATION ALGORITHM IN VISUAL C++

Microsoft Visual C++ tool is used for making the algorithm of navigation. Fig. 23 shows all the control program of computer simulation window.

8.1 Computer Simulation Window of Vision Camera

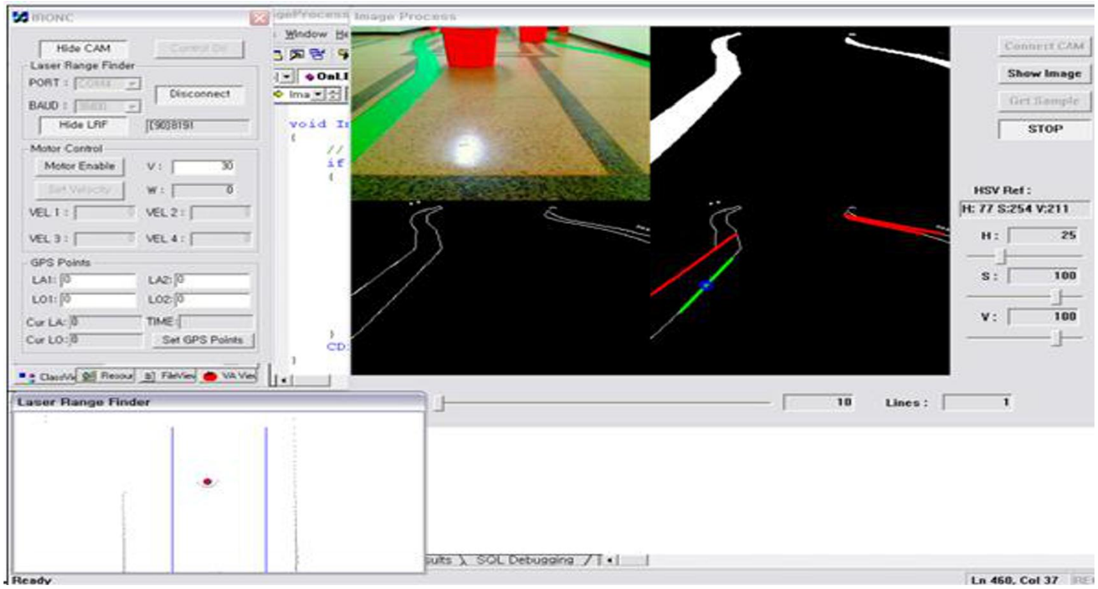
First the simulation window of vision camera is shown in Fig. 24. Here we can see that on the right side of simulation window is the screen of vision camera which is showing four images at a time. The image on the top left corner is showing the original image taken by the camera. The next on the top right corner is showing the image which we get after HSI color detection. The left bottom image is showing the image which is obtained after Canny edge detection and the last one on the right bottom side is showing the image we get after Hough transformation.

There are four control buttons on the right side of the screen which include Connect CAM, Show Image, Get Sample and Stop. First the Connect CAM button is clicked to connect the camera and then Show image is clicked to show the image of the track and then Get Sample Button is clicked to take the samples of color of the border in terms of HSI. After getting the samples of the images of the colored border, we get some specific color values of the border in the boxes shown below individually in H, S, and I edit boxes. The STOP button will finish the working of simulation window of the vision camera.

8.2 Computer Simulation Window of Laser Range Finder

The simulation window of the laser range finder is shown in Fig. 25.

In Fig. 25,On the left side, we can see two drop down lists which include PORT and BAUD and below is the Hide LRF control button. On the right side there is Disconnect button and edit box which shows LRF data for 180 degrees. The PORT drop down list box is used to select the specific communication port and the other BAUD drop down list box is used to select specific baud rate for the LRF. The



[90]8191

V : 30

W : 0

LA1: 0 LA2: 0

LO1: 0 LO2: 0

Cur LA: 0

Cur LO: 0

HSV Ref :

H: 77 S:254 V:211

H : 25

S : 100

V : 100

10 Lines : 1

Ln 460, Col 37

FIG. 23. COMPUTER SIMULATION WINDOW OF NAVIGATION SYSTEM IN VISUAL C++

Disconnect button is used to connect or disconnect the LRF and the edit box is used to show the data of LRF for the entire 180 degrees.

8.3 Computer Simulation Window of Motor Control

The computer simulation window for the motor control is shown in Fig. 26.

Here we have two control buttons. One is Motor Enable button which is used to enable the motors and the other is Set Velocity button which is used to set the angular and linear velocity. Two edit boxes 'V' and 'W' are used to input the required value of linear and angular velocity of the mobile robot. There are four edit boxes, VEL 1, VEL 2, VEL 3, and VEL 4, which show the current value of velocities of four motors at any instant of time during motion.

8.4 Computer Simulation Window of GPS

The computer simulation window for the GPS is shown in Fig. 27.

In Fig. 27 we have four edit boxes, LA1 shows the latitude of starting point, LO1 shows the longitude at the starting point, LA2 shows the latitude at the ending point and LO2 shows the longitude at the ending point. Cur LA shows the current latitude and Cur LO shows the current longitude. There is a TIME edit box which is used to show time at the current moment and finally the Set GPS Points button which is used to show the GPS data.

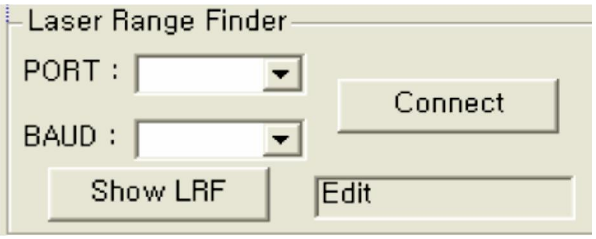


FIG. 25. COMPUTER SIMULATION WINDOW OF LASER RANGE FINDER

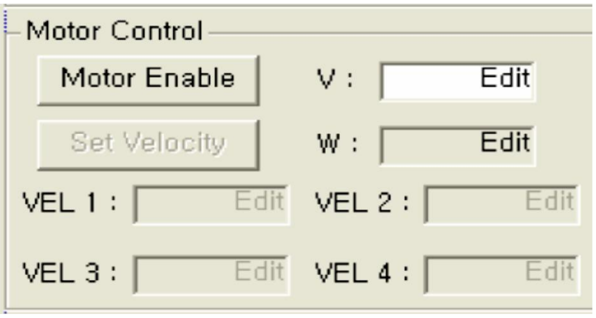
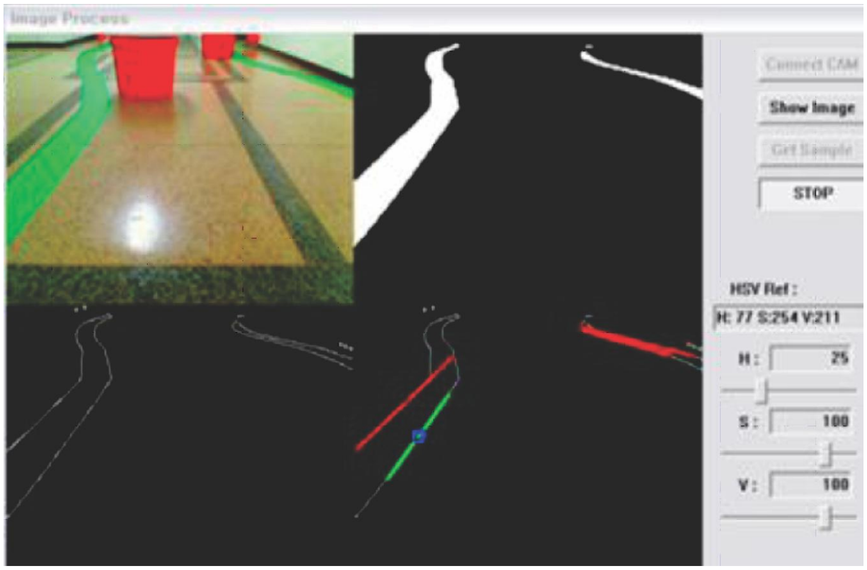
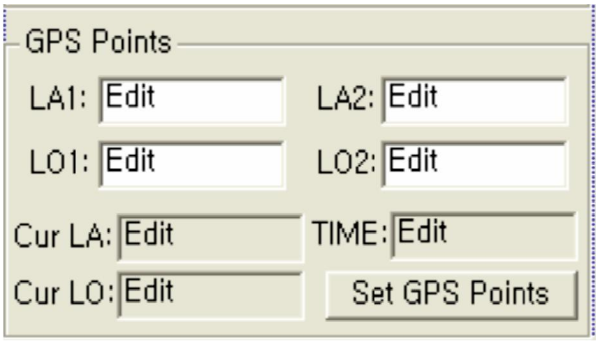


FIG. 26. COMPUTER SIMULATION OF MOTOR CONTROL



HSV Ref :
H: 77 S:254 V:211
H : 25
S : 100
V : 100

FIG. 24. COMPUTER SIMULATION WINDOW OF VISION CAMERA


GPS Points

LA1: Edit LA2: Edit

LO1: Edit LO2: Edit

Cur LA: Edit TIME: Edit

Cur LO: Edit

FIG. 27 COMPUTER SIMULATION WINDOW OF GPS

9. CONCLUSION

As it was a long navigation track thus it has made some motor problem but as it was a redundantly actuated system so it kept on moving and finally by applying all these techniques, successful navigation of mobile robot has been completed and successfully implemented.

10. FUTURE WORK

In future, this work can be implemented to more complicated navigation track by changing algorithm and hardware parameters as this is the first step and can be extended in many ways.

ACKNOWLEDGEMENT

The research work is sponsored by the Higher Education Commission, Govt. of Pakistan, under the scholarship titled: “HRD Initiative MS Leading to Ph.D. Program of Faculty Development for UESTP (Batch-II) Phase-I”.

REFERENCES

[1] Hentschel, M., Wulf, O., and Wagner, B., “A GPS and Laser-Based Localization for Urban and Non-Urban Outdoor Environment”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 149-154, 2008.

[2] Yuan, Y., Cao, Z., Hou, Z., and Tan, M., “Forward Passageway Based Collision-Free Target Tracking for Mobile Robot with Local Sensing”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 255-259, 2008.

[3] Kim, J., Bok, Y., and Kwon, I.S., “Robust Autonomous Navigation against Environment Changes”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 696-701, 2008.

[4] Kim, S.H., Roh, C.H., Kang, S.C., and Park, M.Y., “A Hybrid Autonomous/Teleported Strategy for Reliable Mobile Robot Outdoor Navigation”, SICE-ICASE International Joint Conference, pp. 3120-3125, 2006.

[5] Wolf, D.F., and Netto, C.N.S., “Vision based Navigation using Mobile Robot”, IEEE Latin American Robotic Symposium, pp. 35-39,2008.

[6] Kim, S.H., Roh, C.W., Kang, S.C., and Park, M.Y., “Outdoor Navigation of a Mobile Robot using Differential GPS and Curb detection”, IEEE International Conference on Robotics and Automation, pp. 3414-3419, 2007.

[7] Lee, J.H., Tsubouchi, T., Yamamoto, K., and Egawa, S., “People Tracking Using a Robot in Motion with Laser Range Finder” International Conference on Intelligent Robots and Systems, pp. 2936-2942, 2006.

[8] Khan M. Y. A, Jung E-J, Yi B.J., “A Simulation Tool for Outdoor Navigation of Mobile Robots”, 6th International Conference on Ubiquitous Robots and Ambient Intelligence, pp. 732-739, 2009.

[9] Redone Technologies Co., Ltd, http:// www.urc.kr

[10] Shuang, G., Cheung, N.C., Cheng, K.W.E., Lei, D., and Xiaozhong, L., “Skid Steering in 4-Wheel-Drive Electric Vehicle”, IEEE Conference on Power Electronics and Drive Systems, pp. 1548-1553, 2007.

[11] Weiss, C., Tamimi, H., Massselli, A., and Zell, A., “A Hybrid Approach for Vision-Based Outdoor Robot Localization Using Global and Local Features”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1047-1052, 2007.

[12] Kobilarov, M., Sukhatme, G., Hyams, J., and Batavia, P., “People Tracking and Following with Mobile Robot Using Omnidirectional Camera and a Laser”, IEEE International Conference on Robotics and Automation, pp. 557-562, 2006.

[13] Melingui, A., Merzouki, R., and Mbede, J.B., “Neuro-Fuzzy Controller for Autonomous Navigation of Mobile Robots”, IEEE Conference on Control Applications, pp.1052-1057, 2014.

[14] Ayub, U., Sheikh, M.J., and Ayaz, Y., “A Comparison of Various Robotic Control Architectures for Autonomous Navigation of Mobile Robots”, International Conference on Robotics and Emerging Allied Technologies in Engineering, pp. 239-243, 2014.

[15] Dobrev, Y., Vossiek, M., Christmann, M., Bilous, I., and Gulden, P., “Steady Delivery: Wireless Local Positioning System for Tracking and Autonomous Navigation of Transport Vehicle and Mobile Robots”, IEEE Microwave Magazine, pp. 26-37, 2017.