Algorithms for Real Time Detection and Depth Calculation of Obstacles by Autonomous Robots

Vikas Singh, Anvaya Rai, Hemanth N., Rohit D.A., Asim Mukherjee
Electronics and Communication Engineering Department
Motilal Nehru National Institute of Technology
Allahabad 211004, U.P., India
vikkifundoo@gmail.com

***Abstract*—As robotics is increasing its influence over the technological developments, the need for autonomous robots capable of performing various tasks has increased. These task include the operations in which they detect the obstacles and navigate themselves in an unknown environment .Here we present two algorithms for wireless detection and depth calculation of obstacles. These algorithms are implemented using image processing techniques. The sensor assembly used for wireless sensing of the obstacle comprises of a LASER source and a camera.**

***Keywords*— Wireless obstacle detection, Sensor assembly, depth calculation, Digital image processing, LASER source and Camera.**

I. INTRODUCTION

With robots being made capable of performing various tasks in which they need to navigate themselves, the need for wireless and efficient detection of obstacles in their path is an important aspect to be developed. In this paper we have presented two different algorithms for efficient real time detection and depth calculation of obstacles. Optical sensors are designed by us to implement two algorithms. They comprise of a camera and a LASER source. This assembly is connected to a digital signal processor capable of image processing.

The proposal discusses the two algorithms namely:

- Vertical Shift Algorithm
- Horizontal Shift Algorithm

The sensor assemblies have been discussed under respective algorithms. When there is an obstacle the LASER is obstructed and a LASER point is obtained in the image. The algorithms are based on the sensing the shift of this LASER point with the change in the depth of the obstacle from the robot.

II. VERTICAL SHIFT ALGORITHM

In this algorithm the sensor assembly used is shown in Fig.1. The LASER source and camera are aligned vertically. When there is an obstacle in the path of the robot the LASER light is obstructed. The camera which takes snaps at regular intervals, obtains a Laser spot in the image. The centroid of this point is calculated by the digital signal processor and is used to calculate the distance of the obstacle. With the change in the separation between robot and obstacle there is a shift in the obtained LASER point in the vertical direction. As shown below, the depth of the obstacle varies inversely with the pixel position of the LASER spot along the Y axis.

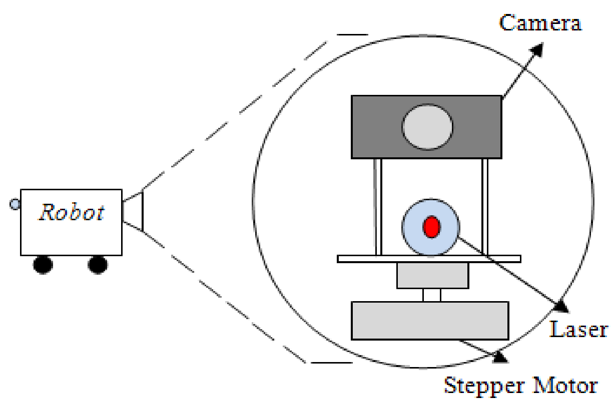

Figure 1. Sensor Assembly for Vertical Shift Algorithm

Referring to Fig.2 and Fig.3:

$$\tan\left(\frac{\theta}{2}\right) = \left(\frac{P}{D}\right)_{actual} = \left(\frac{P}{D}\right)_{virtual} \quad (1)$$

$$M = \frac{I}{O} = \frac{D_V}{D_A} = \frac{y}{h} \quad (2)$$

Now,
We know that since the resolution of the image is 640 X 480,

$$P_V = \frac{Y}{2} = \frac{480}{2} = 240 \quad (3)$$

$$\tan\frac{\theta}{2} = \frac{Y/_2}{M\, D_A} = \frac{Y/_2}{\left(y/_h\right) D_A} \quad (4)$$

$$D_A = \left[\frac{h\,Y}{2\ \tan\frac{\theta}{2}}\right]\frac{1}{y} = \frac{k}{y} \quad (5)$$

$$k = \left[\frac{h\,Y}{2\,\tan\frac{\theta}{2}}\right] \qquad (6)$$

Therefore,

$$D_A \propto \frac{1}{y}$$

where,

D_A is the actual depth of the obstacle.

k is a constant.

Y is the maximum pixel value along Y-axis, Y = 480.

h is the vertical separation between camera and Laser.

Θ is the angle of view of the camera.

y is the pixel distance of the obtained Laser Spot from the center of the image.

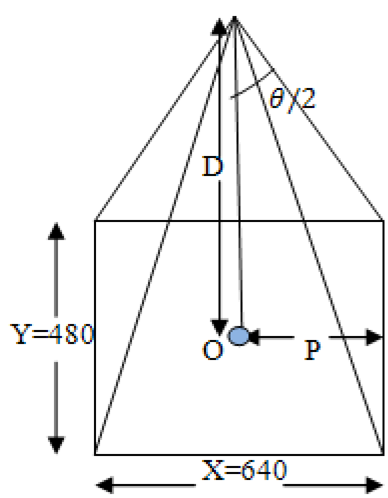

Figure 2. Top view of Image plane for analysis

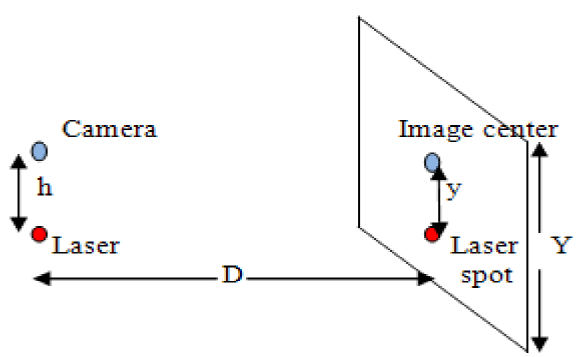

Figure 3. Front view of Sensor assembly and Image Plane for Analysis

III. Horizontal Shift Algorithm

In this algorithm the sensor assembly used is shown in Fig.4 (a). The LASER source and camera are aligned horizontally. When there is an obstacle in the path of the robot the LASER light is obstructed. The camera which takes snaps at regular intervals, obtains a Laser spot in the image. The centroid of this point is calculated by the digital signal processor and is used to calculate the distance of the obstacle. In this assembly both camera and Laser assembly are capable of rotating.

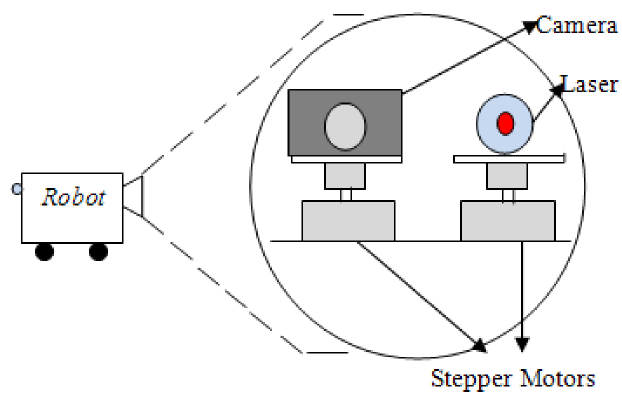

Figure 4(a). Sensor Assembly for Horizontal Shift Algorithm

It can be shown from Fig.4 (b) that the point of intersection the axis of the camera and LASER follows the following trajectory:

$$x^2 + y^2 + yd\cot(\propto_L + \propto_C) - xd = 0 \qquad (7)$$

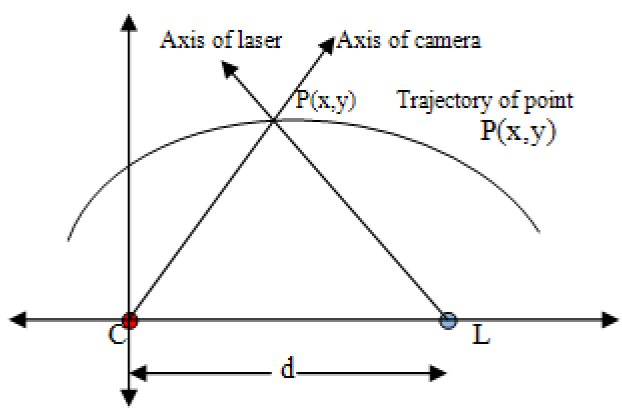

Figure 4(b). Trajectory of the point of intersection of the axis of Laser and Camera

Now depending upon the position of the obstacle three cases arise:

- Obstacle inside the trajectory.
- Obstacle outside the trajectory.

- Obstacle on the trajectory.

For the first case, the Laser Spot is obtained on the left half of the image plane. This is shown in the Fig.7(c) and it can be shown by triangulation in Fig.5 that the depth of the obstacle is given as:

$$D_A = d\left[\frac{\sin\propto_L}{\sin(\propto_L+\propto_C-\beta)}\right]\sin(\propto_C-\beta) \qquad (8)$$

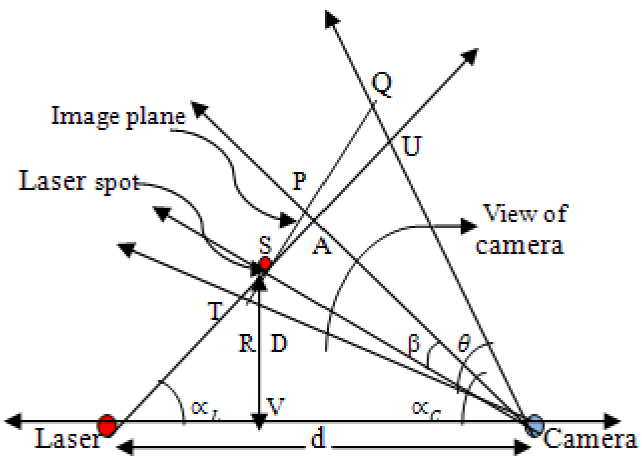

Figure 5. Analysis for obstacle inside the Trajectory

For the second case, the LASER spot is obtained on the right half of the image plane. This is shown in the Fig.7 (b) and it can be shown by triangulation in Fig.6 that the depth of the obstacle is given as:

$$D_A = d\left[\frac{\sin\propto_L}{\sin(\propto_L+\propto_C+\beta)}\right]\sin(\propto_C+\beta) \qquad (9)$$

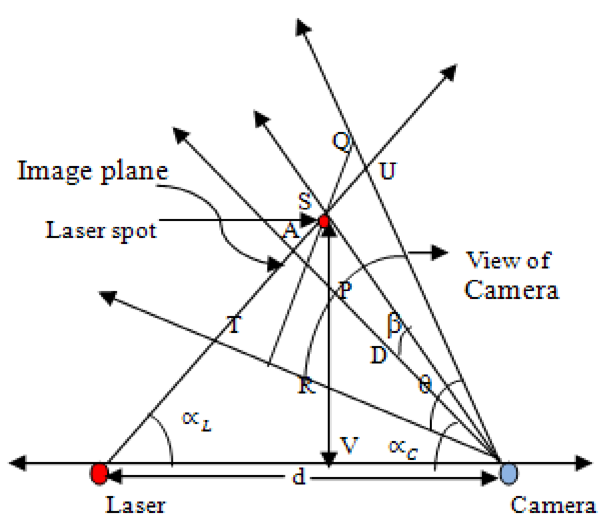

Figure 6. Analysis for obstacle outside the Trajectory

For the third case, the Laser Spot is obtained at the center of the image plane. This is shown in the Fig.7 (a) and it can be shown by triangulation that the depth of the obstacle is given as:

$$D_A = d\left[\frac{\sin\propto_L}{\sin(\propto_L+\propto_C)}\right]\sin(\propto_C) \qquad (10)$$

Where,

d is the horizontal separation between the camera and LASER.

$\propto_L$ is the angle which the axis of the LASER makes with the horizontal at any instant.

$\propto_C$ is the angle which the axis of the Camera makes with the horizontal at any instant.

β is the angle which the obtained LASER spot makes with the axis of the Camera.

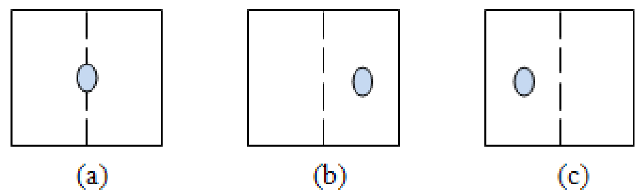

Figure 7. Position of LASER spot in the image depending upon the depth of the obstacle.

IV. RESULTS

The data from the sensor assembly was fed into a computer and were processed using the MATLAB software. The following results were obtained.

Results of vertical shift algorithm:

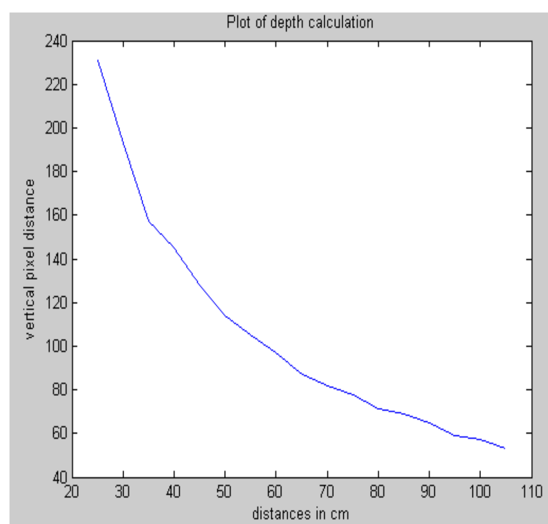

Figure 8. Plot of Pixel distance of LASER Spot Vs Actual depth of object.

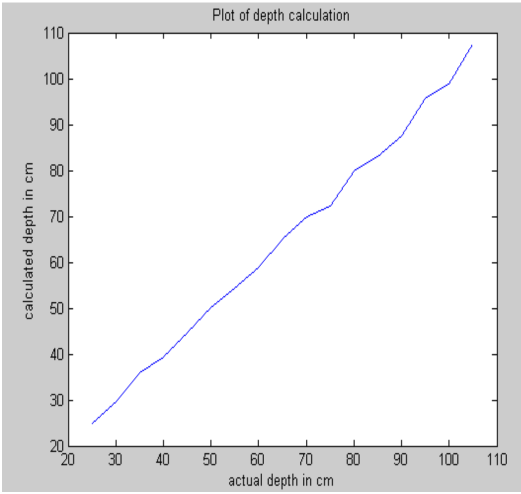

Figure 9. Plot of calculated distance of object Vs Actual distance of object.

TABLE I. DATA SHOWING THE RANGE OF VALUES OF THE CONSTANT 'K' AND SHOWING THE ERROR IN THE CALCULATED DISTANCE BY USING 'K=9055':

	A	B	C	D	E	F
1	d(cm)	y	y-240	k=(y-240)d	d(k=9055)	error(k=9055)
2	30	518.788	288.7877	8663.63	31.35518096	1.355285354
3	35	488.9023	248.9023	8711.58	36.37973823	1.379742885
4	40	459.6754	219.6754	8787.016	41.21990901	1.219909011
5	45	436.0538	196.0538	8822.421	46.18630192	1.186301923
6	50	418.1923	178.1923	8909.615	50.81588823	0.815888229
7	55	404.0283	164.0283	9021.5565	55.2038886	0.203888597
8	60	391.375	151.375	9082.5	59.81833196	-0.181668043
9	65	379.75	139.75	9083.75	64.79427549	-0.205724508
10	70	370.3387	130.3387	9123.709	69.47284268	-0.527157322
11	75	362.2778	122.2778	9170.835	74.05268986	-0.947310141
12	80	355	115	9200	78.73913043	-1.260869565
13	85	347.5	107.5	9137.5	84.23255814	-0.76744186
14	90	343.34	103.34	9300.6	87.62337914	-2.376620863
15	95	336.5	96.5	9167.5	93.83419689	-1.165803109
16	100	332.3571	92.3571	9235.71	98.04335563	-1.956644373
17	105	328.5	88.5	9292.5	102.3163842	-2.683615819

Results of Horizontal shift algorithm:

After locating the centroid of the LASER spot correctly in the image the next task was to accurately calculate the distance of the object. For this the value of the constant 'β' is calculated first. The table showing the obtained results is shown for $\propto_L = 75$ deg $\propto_C = 75$ deg $d = 30cm$.

TABLE II DATA SHOWING THE RANGE OF VALUES OF THE CALCULATED DISTANCE "D" USING HORIZONTAL SHIFT ALGORITHM

D(actual)cm	x	E	β	D(cal)cm
15	268.4142	51.5858	46.1527373	14.4321
20	273.9987	46.0013	37.8383859	18.9321
25	280.687	39.313	29.0399626	24.3215
30	287.2912	32.7088	21.6879739	29.64321
35	293.6093	26.3907	15.8123401	34.7343
40	299.9473	20.0527	10.8968609	39.8415
45	306.4527	13.5473	6.69384783	45.08351
50	312.6201	7.3799	3.34907827	50.05321
55	318.9097	1.0903	0.45595976	55.1214
60	324.426	4.426	1.93173198	60.0361
65	328.573	8.573	3.95355219	64.8215
70	333.0592	13.0592	6.40821055	71.6512
75	335.3035177	15.30351772	7.75409285	75.9951
80	337.1602241	17.16022407	8.93292557	80.2315
85	339.2663365	19.26633648	10.3468895	85.9511
90	341.144376	21.14437602	11.6810951	92.1234
95	341.6539907	21.6539907	12.0556536	94.0151
100	343.605472	23.60547202	13.5416583	102.34
105	343.9776817	23.97768165	13.8346709	104.1567
110	345.4183398	25.41833984	14.9988402	112.0567
115	345.1471359	25.14713593	14.775985	110.4532

Now the plot of above obtained values as obtained on MATLAB is as shown:

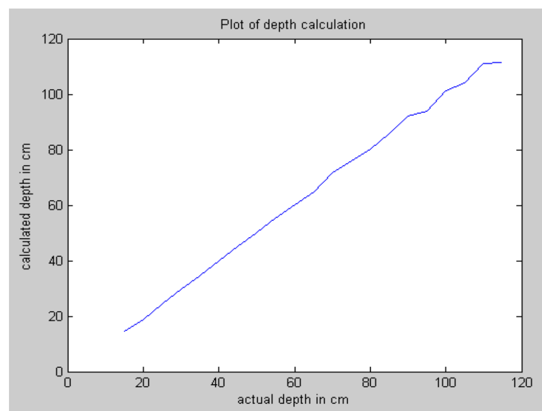

Figure 10. Plot of calculated distance of object Vs Actual distance of object.

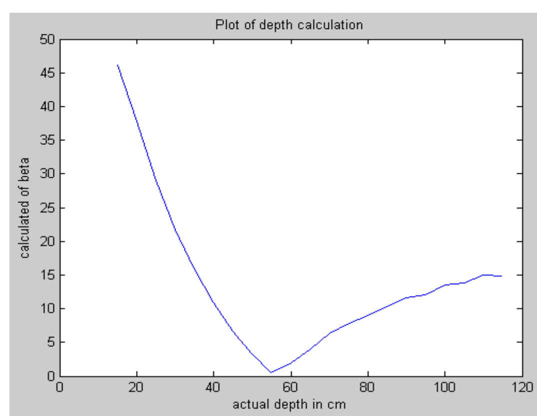

Figure 11. Plot of value of β Vs Actual distance of object.

V. CONCLUSION

Following comparison can be made between two algorithms:

- In this paper we have calculated the real time depth of an obstacle from the Robot up to an accuracy of 98.93% over the range of 20cm to 105cm using vertical shift algorithm. However range of depth calculation can be increased or decreased by varying the distance between camera and LASER in the vertical shift sensor assembly that is “h” in our case. But in case of horizontal shift algorithm there is no constrain on range of depth calculation, without varying any parameters of its sensor assembly.

- Sensor assembly of horizontal shift algorithm is more complex since it requires two stepper motors rotating simultaneously with same revolutions per minute. But it is not the case with vertical shift algorithm since only one stepper motor is used for calculating the same depth.

Applications of these algorithms other than depth calculation are:

- Autonomous navigation of vehicles.
- Calculation of the speed of an approaching object.
- Detection and counting of objects.

REFERENCES

[1] D. R. Pugh an E. A. Ribble, V. J. Vohnout, Th. E. Bihari, Th. M. Walliser, M. R.Patterson and K. J. Waldron. “Technical description of the adaptive suspension vehicle”. *The International Journal of Robotics Research*, pages 24–42, 1989.

[2] K. Berns, R. Dillmann, and S. Piekenbrock. Neural networks for the control of a six-legged walking machine. *Robotics and Autonomous Systems*, 14:233–244, 1995.

[3] St. Cordes, K.Berns, M. Eberl, W. Ilg, F. Schoenung, and A. Baeker. On the design of a four-legged walking machine. In *International Conference on Advanced Robotics ICAR’97*, 1997.

[4] W. Ilg, Th. Mhlfriedel, and K.Berns. A Hybrid Learning Architecture based on Neural Networks for Adaptive Control of a Walking Machine. In *IEEE International Conference on Robotics and Automation (ICRA‘97)*, Albuquerque, New Mexico, US, april 1997.

[5] E.P. Krotkov, R.G. Simmons, and W.L. Whittaker. Ambler: Performance of a six-legged planetary rover. *Acta Aeronautica*, 35(1):75–78, 1995.

[6] D. Wettergreen, C. Thorpe, and R. Whittaker. Exploring mount erebus by walking robot. *Robotics and Autonomous Systems*, 11:171–185, 1993.

[7] P. Buehrle and St. Cordes. Modeling, simulation and realization of an autonomous six legged walking machine. In *ASME*, 1996.