Path Planning of an Autonomous Robotic Vehicle for Outdoor Cleaning

Darsana A.S.
CET Center for Interdisciplinary Research
College of Engineering
Trivandrum, India
darsana.2705@gmail.com

Lal Priya P.S
Dept. of Electrical and Electronics
College of Engineering
Trivandrum, India
lalpriyaps@gmail.com

***Abstract*—Path planning of a robot is to find a path through which the robot could maneuver smoothly avoiding all kinds of obstacles. For a cleaning robot, it is mandatory to pass through all the points in a given area or environment unlike other robots which in most cases are meant to find the shortest path between two given points. The main challenges in coverage path is the reduction of repetition rate and reducing the number turns taken by the vehicle. Minimization of the repetition rate, will lead to an increase in the number of turns that are taken by the vehicle. In the proposed method, a new algorithm which reduces the number of turns as well as the repetition rate is been developed. It is compared with one of the latest algorithm and the proposed method is shown to give better results. Obstacle detection is done with the help of ultrasonic sensors. The algorithm was simulated in Python and the kinematic model of the vehicle was developed in Matlab.**

***Index Terms*—path planning, cleaning robot, coverage path planning**

I. INTRODUCTION

Path planning of a robot is the process of determination of its path that is to be taken for its smooth trajectory. Complete Coverage Path Planning Algorithms [1]are used in cases where the robot is intended to travel or cover each and every point in the work space. These kind of applications need to take the shortest path or time to complete the task with minimum number of overlapping and minimum number of turns as possible.

Coverage path algorithms can be applied in many different fields like cleaning [2], mining [3], agricultural sectors for automatic harvesting [4] and also in lawn movers [5] etc. Complete coverage of a given area can be accomplished using any of the different type of patterns like random walk, spiral, s shaped/ boustrophedon and wall following, as explained in [6]. Most commonly used one is s shaped pattern because of its simplicity and it takes comparatively less time to complete the task.

Zelinsky et al. [7] uses the method of distance transformation which divides the whole task space into square cells and uses the distance transform method to determine the shortest path from the goal point back to the start point. Zengyu Cai1 et al. [8] demonstrates the coverage path planning algorithm form one point to another using A* algorithm. This enables in handling situations of deadlock, where the robot or the vehicle will be surrounded by cells which are obstacles or already covered ones.

The latest technique existing in the area of offline coverage path planning is the cost evaluation algorithm [11], which assigns cost to each and every cell and selects a path with minimum cost. The cost determination is done at each position of the vehicle and thus it advances to its goal position. In this paper, a weightage based path planning algorithm is been developed which is compared with the cost evaluation algorithm and was seen to reduce the repetition rate and number of turns further to a minimum of 35 percentage.

For the purpose of demonstration, the task space is divided into square shaped cells considered to be equal to the size equal to that of the vehicle size. Covering each cell represents the corresponding whole area is being cleaned. The path planning section was tested on a prototype of a robot with size 25 cm by 15 cm, where the speed control as done with the help of a motor drivers. Ultrasonic sensors are used for detecting the obstacles on its path. The algorithm can be implemented in any environment of different shapes in presence of any obstacles with the prior knowledge of the environment. The proposed method is able to cover a given area with minimum number of turns and minimum number of repetition rates.

II. COMPLETE COVERAGE PATH PLANNING

A. System Overview

The robotic vehicle for outdoor cleaning is a 4 wheeled differential driven with a total of 3 brushes in the front section for collecting waste. The rectangular shaped vehicle has dimensions about 90 cm by 60 cm with a height of about 25 cm. Instead of using vacuum cleaning technology, brushes rotating in inward direction is been used which helps in collecting the waste materials. In an outdoor environment, dust and sand will also be present which is not needed to be collected. By using the proposed method only the solid waste waste like leaves papers and cans will be collected in the waste bin in the vehicle which can be manually removed after the cleaning task.

The information about the environment can be obtained from the sensors that are used like camera and ultrasonic sensors for obstacle detection and the distance calculation from those obstacles. Rotary encoders are used for localization and tracking of the vehicle.

The proposed algorithm was tested on a small prototype of the above mentioned design. This prototype of the cleaner robot was made with size 15 cm by 25 cm as shown in Fig.1. The vehicle has a castor wheel in the front and two rear wheels which are controlled by using two DC motors of 60 RPM. These motors are controlled by using the motor driver L2938. An ultrasonic sensor is attached in the front section of the prototype for sensing any kind of obstacles. The waste collection mechanism is obtained by using two rotating brushes connected on the sides of the vehicle. These brushes are controlled by using a single channel relay module which enables the brushes to rotate in the inward direction.

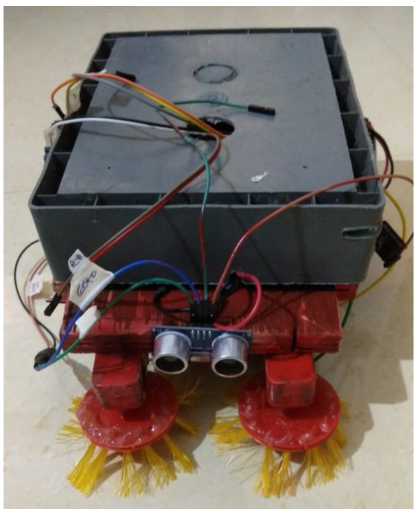

Fig. 1. Prototype of the cleaner vehicle

B. Map Representation

The whole area to be covered is divided into square shaped grids each of size assumed to be equal to that of the robotic vehicle. Each cells is initially assigned the value as zero which represents the area to be cleaned and the visited positions as one. The obstacles are given value two which need not be covered. All the values are given with specific colours for identification purpose. Empty cells with white colours, obstacles with black,visited locations of the robot with blue and the overlapped areas with green colour as shown in Fig. 2. The start point is always set at the left most corner and is assigned red colour. When the robot reaches an already cleaned are the brush rotation stops and restart when it reaches the uncleaned path.

C. Coverage Algorithm

The proposed algorithm is developed under the assumption that the vehicle has basically three movements - Straight line movement(S), Left movement(L) and Right movement(R). Straight mostly refers to the forward and backward movement, but the vehicle rarely moves in the reverse direction like in case of deadlock situations.So the vehicle is assumed to have

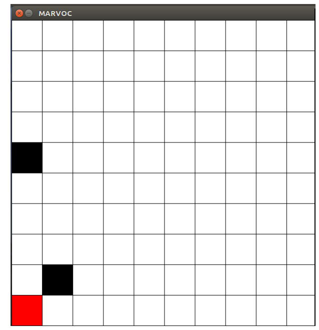

Fig. 2. Task space divided into cells (different colour for representation)

four neighboring cells as shown in Fig. 3. Also here the initial position is assumed at the left most bottom corner which can be changed manually by the user.

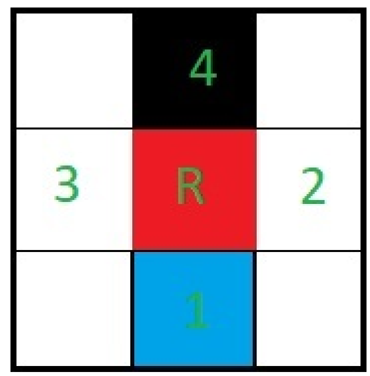

Fig. 3. Robot position with neighboring cells

Fig. 3 shows the present cell which is marked as R and its four neighboring cells. The state of each cells is identified by checking the assigned values to it. The black cell representing the obstacle that is assigned with a value of two. So the robot checks for the value two to confirm whether there is any obstacle in it rather than using the colour for identification. The blue cells corresponding to the visited cell that has a value one and all the white cells representing the vacant or free cells has a value of zero.

Step 1 : Initializing the workspace

The task space is divided into grids of cells each representing the vehicle. Fig. 4 shows the map representation of the task space. It can be divided into square or rectangular shaped cells which is chose conveniently. Here it is shown as square shaped cells. The red cell shows the starting position, white cells are the empty cells or not visited cells and black cells shows the obstacles.The initial position can be set manually and is selected at left most bottom corner. The orientation of the robot also places a major role in the path planning.

Step 2 : Calculation of the weightage function

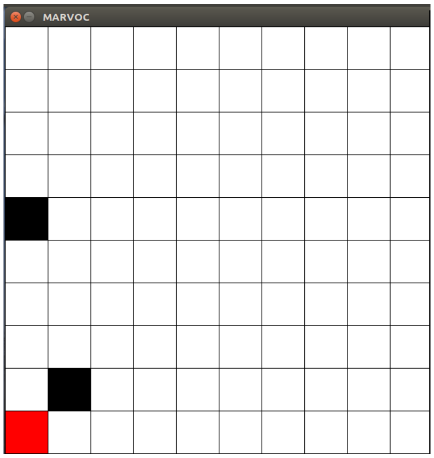

Fig. 4. Map representation

General equation for calculating the weightage value for each of the neighboring cells is as in (1).

$$W = ((A \times F) + (B \times V) + (C \times O)) \tag{1}$$

In (1), F represents the number of free cell in that direction. V represents the already visited cell and O represents the number of obstacles. Whereas A,B and C are constants values for determining the weightage function in the order A $>$ B $>$ C. C is chosen as zero because the cell with obstacle should not be chosen as the robot's path under any circumstances. B is given a value greater than zero and A will be given a value much greater than B since if there is two boundary cells, one with a free cell and the other with a cell that is been already covered, it should chose the vacant cell.

Weightage function for each of the four the neighboring cells will be calculated as shown in 2.

$$\begin{aligned} Ws &= ((A \times F) + (B \times V) + (C \times O)) \\ Wl &= ((A \times F) + (B \times V) + (C \times O)) \\ Wr &= ((A \times F) + (B \times V) + (C \times O)) \\ Wb &= ((A \times F) + (B \times V) + (C \times O)) \end{aligned} \tag{2}$$

Ws is the weightage function for straight motion in the forward direction which calculates the weightage for the immediate front cell. Wb for the straight function in the backward direction (which is not usually used) calculates weightage for the immediate back cell. Wl and Wr represents the weightage function in the left and right direction.

Step 3: Applying the preference value

Now these function values will be multiplied with a preference values. The preference value is given in case if two or more neighboring cells gets the same weightage value, the robot may get confused in selecting the next cell to move on. So preference value is selected for each of the front, left, right and back cell. If the robot's starting position is from the left most end, heading in the upward direction, the preference value of left cell will be given more, under the assumption that the left cells of the current robot position should already covered and so is given more preference than the front cell. If the robot's starting position is front right most end, then highest preference value will be given for right cell and then to the front cell. Then the preference value will be to the other side after front cell and least preference to the back cell.

Step 4: Selection of next cell

After multiplying with the preference value, maximum of the four values will be taken as shown in (3). M in (3) represents the max of all the weightage functions and Ps, Pr, Pl and Pb are preference value for the front cell, right cell, left cell and back cell.

$$M = Max[(Ps \times Ws), (Pl \times Wl), (Pr \times Wr), (Pb \times Wb)] \tag{3}$$

The path is selected according to the M value whichever value equals the value M, corresponding path to that cell will be taken. Thus weightage function for each cell is calculated as the vehicle moves and correspondingly a smooth path is obtained. By the proper selection of the preference values the path opted by the robot will be more optimized. Preference value selected is in the order of Pl $>$ Ps $>$ Pr $>$ Pb. Left is always assumed with respect to the left side of the map irrespective of the orientation of the robot. So the left side is given more preference in the concept that if there is any unvisited path in the left section it should be given more preference than in the front section.

This algorithm has reduced the repetition rate and in most cases the repetition value is zero except in case of deadlock situations. In such situations, the path may enter a cell which is surrounded by already visited cells or obstacles. In such a situation, it has to find the nearest vacant cell and moves to the corresponding cell through the already visited cells.

III. RESULTS

The algorithm was tested in two cases. First case was the scenario without any obstacles in the work space and second one was in the presence of obstacles. Fig. 5 shows the case without any obstacle. The robot follows a zig zag path and the starting position is shown as red cell and the path followed by the green lines. The boundary is shown in black rectangles. This was successfully tested on a prototype of the original vehicle of size 25 cm by 15cm on an area of 150 by 75 cm.

Fig.6 shows the case of the path taken by the robot in presence of obstacles. The end position is represented by the circular robot. Here only one obstacle of square shape is taken into consideration. Whereas Fig. 7 shows the path taken by the robot in a more complex environment where there are two rectangular shaped obstacles at two different positions.

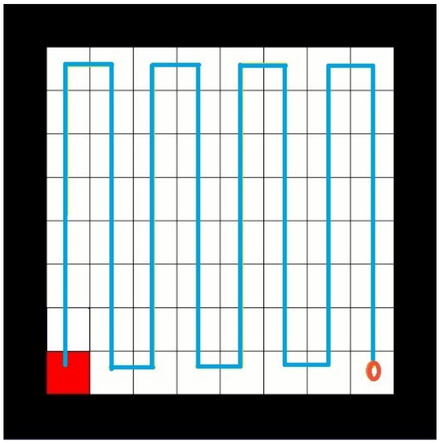

Fig. 5. Path when no obstacle is present

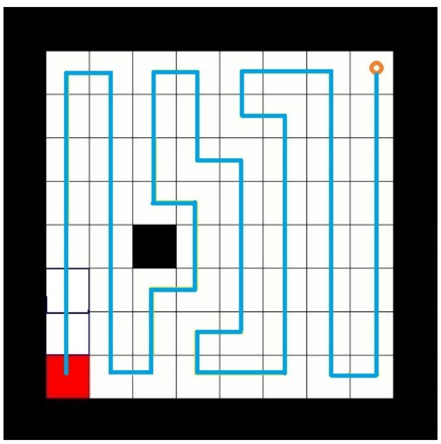

Fig. 6. Path with one obstacle

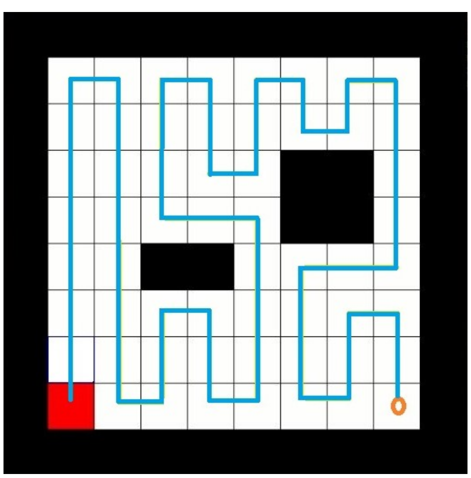

Fig. 7. Path with more number of obstacles

Overlapping or repeating the same cell occur under the condition where complicated shaped obstacles are present. These kind of obstacles leads the robot to get into an area where the robot will be surrounded by already covered area or obstacles. Such a situation is known as deadlock and the robot in such condition finds the next free cell and finds a path which will contain already covered area which is the only solution. An example of deadlock condition is shown in Fig. 8

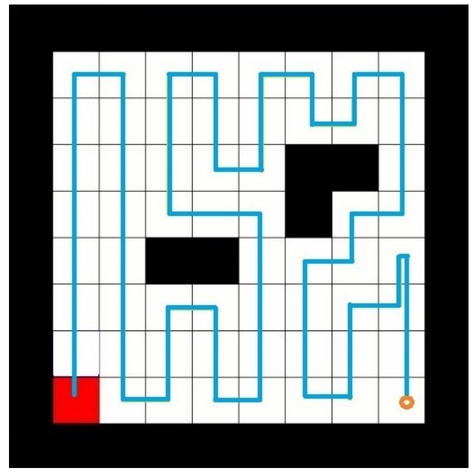

Fig. 8. Deadlock situation leading to overlapping

IV. CONCLUSIONS

In this paper, complete coverage path planning is been implemented using ultrasonic sensors. Obstacle detection and localization using the available sensors are been successfully carried out. The algorithm was tested on a prototype of the mobile vehicle with size 25 cm by 15 cm having two round brushes attached in the front.

A comparative study of the proposed method with the existing algorithm in the coverage path planing filed is shown in the table I. The comparison between both the algorithm is got by applying the two algorithms on to a same environment of size 600 by 600 size work space in presence of obstacles of different shapes.

TABLE I
COMPARATIVE STUDY

	Coverage Cost algorithm		**Proposed algorithm**	
	No. of turns	Revisited cells	No. of turns	Revisited cells
Case 1	50	2	28	0
Case 2	51	1	33	1

Fig. 9 and 10 shows the path taken by the robot while using the two algorithms in presence of a square obstacle. As shown in Fig.9 and 10, the first algorithm takes more number of turnings and 2 cells are covered repeatedly. The proposed algorithm reduced the number turning to around half of the other algorithm.

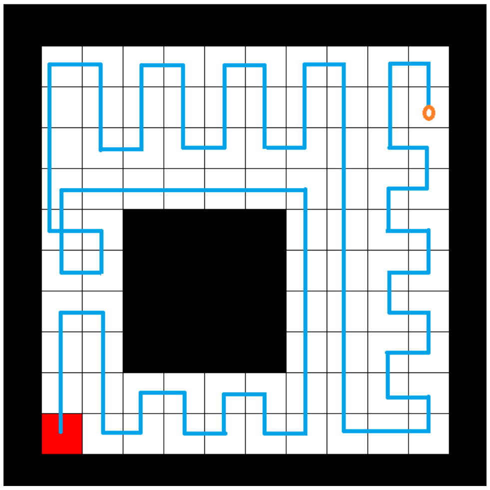

Fig. 9. Path taken using coverage cost algorithm in case 1 [11]

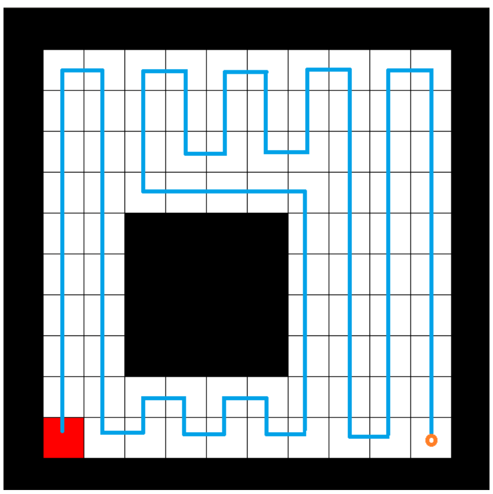

Fig. 10. Path taken using proposed algorithm in case 1

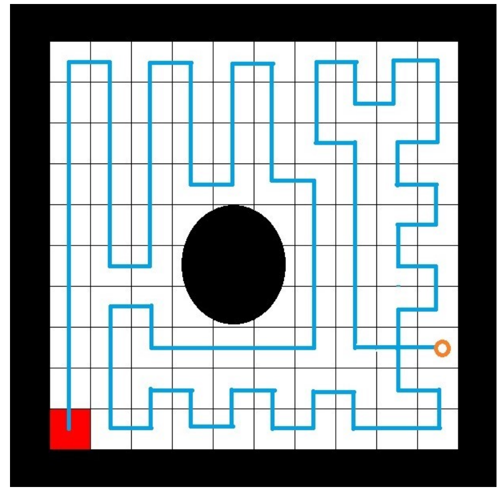

Fig. 11. Path taken using coverage cost algorithm in case 2 [11]

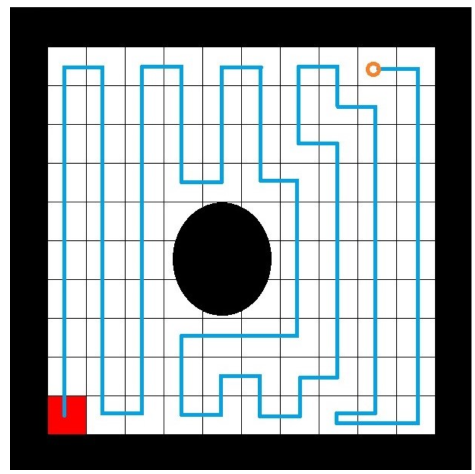

Fig. 12. Path taken using proposed algorithm in case 2

Fig. 11 and Fig. 12 shows the path taken by the robot while using the two algorithms in presence of a circular shaped obstacle. Even if the repetition rate is same for both the algorithms, the number of turnings taken by the proposed method is comparatively lower.

The comparative study shows that eventhough the cost coverage algorithm is able to minimize the repetition rate, it has more number of turnings. Whereas by applying the proposed method to the same environment it not only reduced the repetition rate but also reduced the number of turnings to considerably low value.

This algorithm can be used in both indoor and outdoor conditions where complete coverage is required. The localization method can be further improved by using camera or image processing techniques. Also by incorporation more number of ultrasonic sensors in the sides of the vehicle, it can be further used in unknown environments.

References

[1] R. N. De Carvalho, H. Vidal, P. Vieira, and M. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," vol. 2, pp. 677-682, 1997.

[2] J. Chi, G. Yang and J. Yang,"The on-line coverage algorithm and localization technique of the intelligent cleaning robot," *IEEE International Conference on Sensing and Control,* pp. 943-948, April 2008.

[3] I. A. Hameed,"Motion planning for autonomous landmine detection and clearance robots," *International Workshop on Recent Advances in Robotics and Sensor Technology for Humanitarian Demining and Counter-IEDs (RST)*, Cairo, pp. 1-5, 2016.

[4] L., Ming, K. Imou, K. Wakabayashi, and S. Yokoyama, "Review of research on agricultural vehicle autonomous guidance," *International Journal of Agricultural and Biological Engineering*, vol. 2, no. 3, pp. 1-16, 2009.

[5] Ibrahim A. Hameed,"Coverage Path Planning Software for Autonomous Robotic Lawn Mower using Dubins Curve," *Proc. of the 2017 IEEE Intel Conf. on Real time Computing on Robotics,Okinawa, Japan,* July 2017.

[6] K. M. Hasan, A. Nahid, and K. J. Reza,"Path planning algorithm development for autonomous vacuum cleaner robots," *Proc. International Conf. on Informatics, Electronics and Vision,Dhaka,Bangladesh,* pp. 1-6, May 2014 .

[7] A. Zelinsky et al., "Planning paths of complete coverage of an unstructured environment by a mobile robots," *in Proc. Int. Conf. Adv. Robot. (ICAR)*, pp.533-538, 1993.

[8] Z. Cai, S. Li, Y. Gan, and R. Zhang, "Research on Complete Coverage Path Planning Algorithms based on A* Algorithms," *The Open Cybernetics Systemics Journal,* pp. 418-426, 2014.

[9] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 34, no. 1, pp. 718-725, Feb 2004.

[10] B. Dugarjav, S.-G. Lee, D. Kim, J.H. Kim, N.Y. Chong, "Scan matching online cell decomposition for coverage path planning in an unknown environment," *International Journal of Precis. Eng. Manuf. 14* pp.1551-1558, 2013.

[11] Gajjar, Sumit, Jaydeep Bhadani, Pramit Dutta, and Naveen Rastogi, "Complete coverage path planning algorithm for known 2d environment," *2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*, pp. 963-967. IEEE, 2017.