DOI: 10.54254/2755-2721/5/20230647

Optimization of coverage path planning algorithm for robots

Yue Wu

Futian District, Shenzhen, China, 518028
Shenzhen Foreign Languages School

193191762@st.usst.edu.cn

Abstract. In the era of technology gradually into the home, sweeping robots have become a must for many families. However, due to the lack of algorithms in the robot itself, there are still many problems with sweeping robots. In order to save cost and improve efficiency, it is necessary to improve the algorithm of robot coverage path planning. In this paper, we study the robot to complete full-area coverage path planning using sensors in an unknown environment. In this paper, the wavefront algorithm is combined with the traditional zigzag full-area path planning algorithm. The authors find that the combined algorithm speeds up the sweeping and reduces the repetition rate. Combining several different algorithms together like this can have more different applications in real life.

Keywords: coverage path planning,wavefront algorithm, breadth-first search.

1. Introduction

With the development of society and technology, many people are choosing to buy floor cleaning robots for cleaning in their homes because they are busy at work. Sweeping robots are very widely used in our life. They sense the surroundings through their own sensors, avoid obstacles and need to cover all the sweepable space around them. By the same token, in the event of disasters such as earthquakes and fires, disaster relief robots can provide help to rescuers, disaster site clean-up, while taking into account the functions of detection and communication. They need to do the whole range as soon as possible to search for survivors. The act of path planning through all points of interest or an entire region within a range like this, while avoiding obstacles, is called coverage path planning [1]. The algorithm given to the robot to perform reduces the labor cost to a great extent. Robotic sweeping, robotic floor construction, wall painters, window cleaners, and automated lawn mowers are all examples of coverage path planning. However, due to the shortcomings of today's overlay path planning algorithms, robots often have repetitive actions on a given path. Take, for example, path planning for sweeping robots. Under current conditions, an effective approach for floor-sweeping robots is randomization. This is the method that some floor cleaning robots rely on: if the floor is swept randomly for long enough, it will become clean. However, this repetitive action greatly extends the cleaning time and reduces the efficiency of the robot, while also depleting the robot itself of energy and reducing its working life. In addition, such repetitive motions greatly reduce the efficiency of the search during the search and rescue process. Therefore, in order to save cost and improve efficiency, it is necessary to improve the algorithm of robot coverage path planning.

To address the problem of robot coverage path planning, Guo proposes to combine knowledge inference techniques with breadth-first search to find the optimal path for cleaning robots in dynamic environments [2]. The method is used to combine knowledge inference with traditional path planning coverage algorithms to enable the robot to control and avoid obstacles with the best coverage. Experimental results show that using the proposed method, the robot is able to avoid both fixed and moving obstacles with optimal coverage, while reducing computational cost and time. The method has high coverage and low repetitive coverage compared to other current methods and performs better than conventional robot algorithms. Similarly, S. Gajjar, J. Bhadani, P. Dutta, and N. Rastogi developed a full coverage path planning algorithm and tested it on real hardware [3]. The algorithm was tested on real hardware. The algorithm was developed for a known 2D environment with static obstacles and was optimized in terms of minimum coverage time. The authors believe that the two algorithms can be combined to form further optimizations.

In this paper, algorithm improvements for robot coverage path planning will be investigated. In this paper, a simulation environment will be built to simulate a real dynamic environment by Matlab. matlab can verify whether the algorithm is a better solution in three latitudes: coverage, repeat coverage and time. This paper is useful for practical applications in life. China has been conducting technology tracking research on intelligent sweeping robots and cleaning robots since the 1990s, but the intelligence level of domestic intelligent sweeping robots is still relatively low due to the specificity of the environment and the characteristics of the cleaning robots themselves. It is still in the primary research stage, and many key technologies such as localization, environment modeling, path planning and sensors are to be strengthened. This paper proposes some new ideas for path planning of sweeping robots, hoping to promote domestic research and exploration in intelligent robots.

2. Problem Formulation

In this paper, we study the optimization problem of the robot full area coverage path planning algorithm in a static two-dimensional environment, without considering the robot's own motion characteristics and volume, and assuming that the robot can obtain accurate information about its own position. The authors will combine the wavefront algorithm with the traditional full-area coverage path planning algorithm, and after combining the wavefront algorithm with the traditional full-area coverage path planning algorithm, the two cases will be compared to find a more optimal solution. The authors will use Matlab to build a simulation to simulate a static, obstacle-filled two-dimensional environment. The two combined algorithms will be run in Matlab to determine if the algorithm is a better solution in the three latitudes of coverage, repeat coverage, and time.

3. Method

In this paper, we study the optimization of a robot full-area coverage path planning algorithm in a static 2D environment without considering the robot's own kinematic properties and volume, assuming that the robot can obtain accurate information about its own position. The authors will use Matlab to build a simulation environment to simulate a static, obstacle-filled 2D environment. The two combined algorithms will be run in Matlab to determine whether the algorithm is a better solution in the three latitudes of coverage, repeated coverage and time.

Coverage path planning algorithms can be classified into two types, offline and online. Offline planning algorithms can only rely on stationary information, i.e., the assumed environment is known [4]. However, offline planning algorithms are not implementable under most conditions. Therefore, the problem discussed in this paper is the improvement of the overlay path planning algorithm in a dynamic environment, i.e., online. The robot needs to sense its surroundings in real time through its own sensors, avoid obstacles and achieve coverage of a given environmental range.

3.1. Environment

We define the whole space as S. Each lattice has two states, E-empty and O-Occupied. Because the machine is overwriting in an unknown environment, the machine does not get any prior knowledge about the environment. For example, it does not know the boundaries of the space it is in, which positions

DOI: 10.54254/2755-2721/5/20230647

are free and which are occupied, the unknown it is in at the time and the enclosed space formed by the occupied positions. Therefore, the robot must decide which direction it can or should move along, using only its sensors and its mental map of the environment, as described below [5].

Upon reaching an unknown environment, the first step should be to determine the external shape of the environment, as well as the direction and shape of the internal obstacles. Usually, this depends on the robot's own sensors (i.e., sonar system). The machine should have sensors in all four directions, facing directly in front of it, directly behind it, to the left and to the right. These sensors provide the machine with enough information to make a mental map of its surroundings and the direction in which it will move.

In behavioural geography, a mental map is a person's point-of-view perception of their area of interaction [6]. In this paper, a mental map is the result of the machine's perception of its surroundings. It consists of at least four pieces of information: I is the machine's initial snapshot, which is the empty space that the machine first estimates about the environment. v represents the space that has already been visited. It contains the space that is initially empty and covered, F is the space that is initially free, but not yet accessed, and U is the space that is still unknown. All four variables will change as the machine moves.

The grid-based approach uses a representation of the environment that is decomposed into a uniform set of grid cells. This grid representation was first proposed by Moravec and Elfes for mapping indoor environments using a sonar ring mounted on a mobile robot [7]. The method divides the environment into small grids as a way to record the presence of obstacles at a location and whether they are passable. The robot marks the grids with obstacles as dark, i.e., not passable, and the grids without obstacles as blank, i.e., passable. In this paper, we discuss three methods of dividing the environment into separate cells. They are the large square grid, the small square grid and the triangular grid.

After arriving at an unknown environment, the first step should be to determine the exterior shape of the environment, as well as the orientation and shape of internal obstacles. Typically, this depends on the robot's own sensor (i.e., sonar system). The grid-based approach uses a representation of the environment decomposed into a uniform collection of grid cells. This grid representation was first proposed by Moravec and Elfes for mapping an indoor environment using a sonar ring mounted on a mobile robot. The method divides the environment into small grids as a way to record whether there are obstacles on the position and whether they are passable. The robot marks the grids with obstacles as dark, i.e., impassable, and marks those without obstacles as blank, passable. In this paper, we discuss three ways to partition the environment into independent cells. They are large square grid, small square grid and triangle.

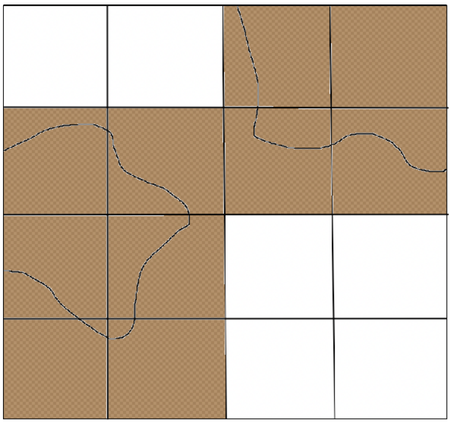

Figure 1. This image shows how the markers behave when the environment is divided into separate cells of a large grid. The irregular shapes are the shapes of the assumed obstacles, and the darker parts are the grids that are marked as having obstacles.

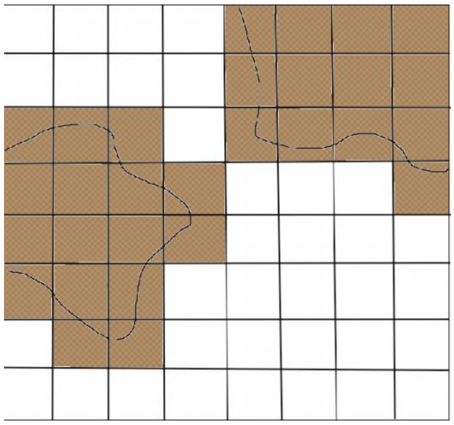

Figure 2. This image shows how the markers behave when the environment is divided into separate cells of a large grid. The irregular shapes are the shapes of the assumed obstacles, and the darker parts are the grids that are marked as having obstacles.

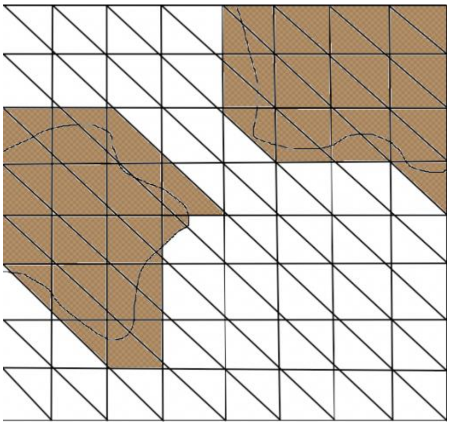

Figure 3. This image shows how the markers behave when the environment is divided into separate cells of triangles. The irregular shapes are the shapes of the assumed obstacles, and the darker parts are the meshes that are marked as having obstacles.

Based on the comparison of the three images above, it can be observed that the area marked by the separate cells of the large grid is the largest and the area marked by the separate cells of the small grid is the smallest. This is because the independent cells of the small grid have a higher resolution. During motion, the machine avoids the cells that are marked darker, so the larger the marked area, the smaller the area covered by this path planning. The algorithm requires a higher resolution in the coverage path planning because of the need for the robot to be able to achieve the goal and the robot needs to make very fine motion adjustments.

3.2. Algorithm

Two algorithms are frequently used in the current coverage path planning algorithms. They are randomization and zigzag, respectively [8].

Randomization, as the name suggests, is to let the machine move randomly and continue moving as long as it encounters an obstacle to change direction. Given enough time, the machine can cover the required range. Such an algorithm is often found in home sweeping robots. By simply giving the sweeping robot enough time to clean up, the robot can achieve a clean floor. Efficiency is usually not required, and the path repetition rate is not cared for.

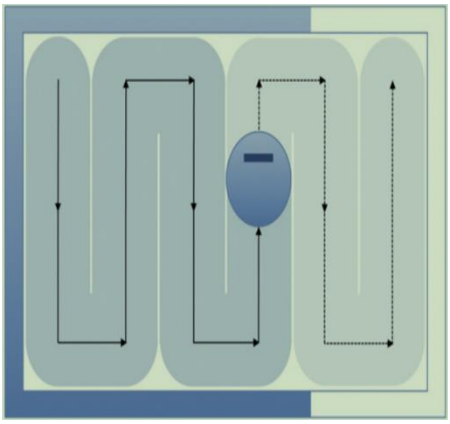

Figure 4. The zigzag path planning, on the other hand, largely avoids the problem of path repetition. Each individual unit can be covered. But again, such path planning has the problem of low efficiency [1].

DOI: 10.54254/2755-2721/5/20230647

When the machine encounters an obstacle, there is still the problem of path duplication. The wavefront algorithm is a breadth-first search algorithm. It first analyses the nearest few grids as the center, and then extends the range to more distant regions [9]. The wavefront algorithm consists of a breadth-first search of the graph starting from the target point until the starting point is reached. First, all grids covered with obstacles need to be marked as 1, which is used to tell the robot that these grids are occupied and the grids marked as 1 need to be avoided. Also, in order to avoid dead zone positioning errors, you can choose to surround the grid occupied by the original obstacle with "1" to expand the coverage of the obstacle. Then, the grid where the target point is located is marked as 2, all the grids adjacent to the grid where the target point is located are marked as 3. After marking all the grids of 3, the grids adjacent to 3 and not yet covered are marked as 4. The same method is repeated until the starting point is marked. The result of the wavefront search is a wavefront map [10]. Once the wavefront map is generated, the wavefront algorithm traverses the path from the grid marked with the highest value to the grid marked with the lowest value. It will not repeatedly return to the grid that has already been covered. This will result in a map whose coverage ends at the grid of the target point, and there will be no duplication.

18	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
17	17	16	15	14	13	12	11	10	9	8	8	8	8	8	8
17	16	16	15	14	13	12	11	10	9	8	7	7	7	7	7
17	16	15	15	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	6	6	6	6
17	16	15	14	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	5	5	5	5
17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	**2**

Figure 5. The wavefront algorithm.

4. Results

In this paper, we simulate the simulation environment in Matlab by creating irregular shapes in the environment as obstacles. The environment is meshed and each mesh is considered as an individual cell. A wavefront algorithm is used until the machine iterates from the starting position to the target position and the entire environment is covered. The grid-based approach, the wavefront algorithm, provides complete coverage over a discrete representation of the target environment. Moreover, it is easy to create and manipulate the grid map.

Comparing the traditional coverage path planning algorithm with the algorithm using the wavefront algorithm combined with the traditional coverage path planning algorithm, it is clear that the improved algorithm reduces the repetition rate by about 9.4% when traversing the entire environment. The time used is similarly reduced by 10s, or about 3.6%.

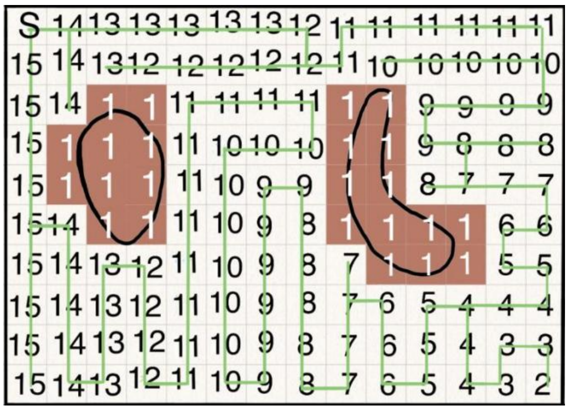

Figure 6. The wavefront algorithm executes like this.

DOI: 10.54254/2755-2721/5/20230647

5. Conclusion

This paper focuses on the combination of wavefront algorithm and traditional coverage path planning algorithm to achieve full coverage path planning in an unknown space environment. According to the Matlab simulation results, the improved algorithm can achieve a lower total cost of motion. It reduces the redundancy rate by 9.4% compared with the traditional covered path algorithm, and it also reduces the time by about 3.6%. In addition, it achieves full coverage of the area with 97.7% coverage. In the near future, the authors will investigate algorithms that combine other algorithms with traditional covered path planning. For example, the A* algorithm in the heuristic algorithm. Again, the authors will judge the success of the algorithm in three dimensions, namely coverage, repetition rate and usage time.

The article still has flaws. The article does not put the dynamic environment into the discussion. The machine cannot avoid obstacles after the initial snapshot. The authors will further investigate the application of wavefront algorithms in dynamic environments in the future. The uncertainty is amplified to take into account the uncertainty of the environment and even the uncertainty of the machine itself.

References

[1] Enric G and Marc C 2013 Robotics and Autonomous Systems Volume 61 Issue 12.

[2] Guo X 2009 WRI World Congress on Computer Science and Information Engineering 403 261-5.

[3] Gajjar S, Bhadani J, Dutta P and Rastogi N 2017 "Complete coverage path planning algorithm for known 2d environment" 2nd IEEE International Conference on Recent Trends in RTEICT 8256741 963-7

[4] Darsana A and Lal P 2018 "Path Planning of an Autonomous Robotic Vehicle for Outdoor Cleaning" ICCSDET 1-6

[5] Vikas S, Rajiv J, Ugur K and Dana N, "Real-Time Planning for Covering an Initially-Unknown Spatial Environment", Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference 63-68

[6] Qin I, Li C, Liu L, Hao J, Zhu Y and Yan J 2021 International Conference on Control, Automation and Information Sciences (ICCAIS) 52680 84-87

[7] Xue Y and Sun J 2018 Appl. Sci Volume 8 Issue9

[8] Zhou Y, Sun R, Yu S, Sun Y and Sun L 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER) 8446258 1335-40

[9] Wang X and Syrmos V 2009 17th Mediterranean Conference on Control and Automation 5164725 1295-1300

[10] Santos, Santos F, Mendes J, Costa P, Lima J, Reis R and Shinde P 2020 Path Planning Aware of Robot’s Center of Mass for Steep Slope Vineyards Robotica 684-698.