Mobile robot path planning compares wavefront and A* algorithms

Siyan Zhang¹,*,†, Zehan Liao², b, †, Yichen Wu³, c, †

¹Shenyang Urban Construction University, 2-24-1, Building 2, Gelin Rose Bay, Heping District, Shenyang, Liaoning
²Oregon State University, 102, Unit 4, Building 1, Yuanbojiayuan, Changxindian Town, Fengtai District, Beijing
³University of California, Irvine, 403, Building 19, Oriental Lily Garden, Hanjiang District, Yangzhou, Jiangsu

a3138910164@qq.com
bliaoze@oregonstate.edu
cyichew15@uci.edu
†These authors contributed equally.

Abstract. In this study, the A* algorithm and wavefront algorithm are adopted to simulate the motion of the robots using Matlab. And a complex Maze with different complexity is created to compare the time and path that taken by the robot using the A* algorithm and wavefront algorithm. Before testing, we have determined the types of wavefront algorithms. The wavefront algorithms are divided into 4-Sector Connectivity and 8-Sector Connectivity. After summarizing and comparing the previous research results, we decide to use the 8-Sector Connectivity wavefront algorithm and the A* algorithm for testing. After that, the maze (environment) is established by Matlab simulations, and the corresponding algorithm, A* algorithm and wavefront (8-Sector Connectivity) algorithm are assigned to the robot. Finally, the robot moving to the specified position in the maze is tested. We use the built-in function in Matlab to record the time and distance of the robot moving to the specified position in the maze. The data are collected and analysis to compare the A* algorithm and wavefront algorithm. The results show that the A* algorithms achieve better results in a simpler environment, that is, it takes less time to get the goal in the maze. The wavefront algorithm has better performance in some complex environments. In general, we think the A* algorithm better than the wavefront algorithm.

1. Introduction

With the development of science and technology, people are becoming more and more skilled in robotics. We can find robots working in dangerous environments like chemistry, combustion, radiation, and war environments. In those environments, the robotics need to complete the assignment by themselves and overcome complex environments. Therefore, path planning is one of the key research areas in Dynamic Robotics. However, the efficiency of the robot to complete the task depends on the algorithm we use. Some commonly used algorithms include depth-first search, A* search algorithm, single query planner, rapidly exploring random Tree, algorithm (RRT), and waveform algorithm. We will choose the A* algorithm and the waveform algorithm because these two algorithms are the most...
basic, the simplest, and have fast processing speed [1]. In this article, we want to compare the A* algorithm and wavefront algorithm to find the optimal solution. A problem with wavefront algorithms is to divide this algorithm into 4-Sector Connectivity and 8-Sector Connectivity. We must first study these two aspects to decide which algorithm of wavefront we will use. A subsection

1.1 A* algorithm

A* algorithm is a searching algorithm. It is usually used to find the shortest path between the initial and the final state. We usually use the A* algorithm on map applications [2]. The A* algorithm's principle is to set the starting point and the final point in an environment full of obstacles. The goal is to get from the starting point to the endpoint in the shortest time [Figure 1]. The A* algorithm works based on a heuristic method, which helps to achieve optimality. A* is another form of the best-first algorithm. Optimality enables the algorithm to find the best solution to the problem. Such an algorithm can also provide completeness. If there is a possible solution to an existing problem, the algorithm will definitely find it [8].

1.2 Wavefront algorithm

Wavefront algorithm is suitable for grid-based maps, and it is a cell-decomposition path planning method. It will divide the environment into equal polygons like a square grid. Obstacles are represented by specific numbers [3] as shown in figure 2. Set the specific number as an obstacle, set the initial point and the final point, such as the initial point is 1 and the final point is 4. The robot will act in the order of 1, 2, 3, 4 [4].

![Figure 1 A* algorithm path](Image)
1.3 Wavefront Based on 4-Sector Connectivity and 8-Sector Connectivity

The wavefront algorithm is divided into two types, the first is 4-Sector Connectivity, and the other is 8-Sector Connectivity [7]. The principles of 4-Sector Connectivity and 8-Sector Connectivity are similar. The wavefront algorithm is essentially breadth-first search (BFS), each cell is assigned a value that corresponds to the number of moves required for the shortest path from that cell to the goal [6]. We can use Neighborhood connectivity to explain the difference between 4-Sector Connectivity and 8-Sector Connectivity. Now there is a cell C with four neighbors around it, as shown in Figure 1. If cell C wants to move and only moves in these 4 directions, this is 4-Sector Connectivity, if cell C has 8 neighbors around it, as shown in Figure 2. Cell C has 8 ways to leave the origin, this is 8-Sector Connectivity [5]. That means the path that 8-Sector Connectivity can move is shorter than that of 4-Sector Connectivity because when the robot is using 8-Sector Connectivity, it can move to 45 degrees while 4-Sector Connectivity can only turn 90 degrees. Therefore, we will use the 8-Sector Connectivity wavefront algorithm to make tests.

In this research, we will use MATLAB to simulate the robotics movement by A* and wavefront algorithm. We will use the A* algorithm and wavefront algorithm (8-Sector Connectivity) to work on the same maze and we will show the time spent on creating the proper path. At the same time, we will compare the path and consider which path is optimal.

2. Methods

To compare the advantages and disadvantages between the two, you need to design experiments. We used an algorithmic path recognition simulation using Matlab robot playground's code. At the beginning of the design experiment, we first decided on variables. In practice, the difficulty of maze becomes a prerequisite for each algorithm to accomplish the task well. Therefore, variables can be derived, that is, the difficulty of the maze. We chose 4 difficulties. 5, 10, 15, 20 compare the speed of the two algorithms to solve the maze and the rationality of solving the maze under these four difficulties. You can get a rough look at the advantages and disadvantages of the two algorithms in the case of the same maze. The best measure of the speed at which two algorithms solve the maze is time. But in addition to time, solving the rationality of solving the maze is also one of the biggest problems. Therefore, the amount that needs to be compared is the rationality of time and solving the maze, but the rationality of solving the maze cannot be compared as a standardized data, so we abstract it, and take the total path length obtained in our experiment as the basis for the comparison of the rationality of the maze.

Therefore, we started designing experiments. Four control trials were conducted in each difficulty, and two control experiments under the same maze were conducted in each group. And use the tic-toc function that come with Matlab for timing. The time recorded is the speed at which the two algorithms solve the maze. The resulting picture is then used by CAD to record the path.
Maze level 5

Figure 3: Example: A set of experiments

a)
Figure 4 Path planning results based on wavefront or A* on maze level 5
Maze level 10

a)

b)

c)
Figure 5: Path planning results based on wavefront or A* on maze level 10

Maze level 15

a)

b)
Figure 6: Path planning results based on wavefront or A* on maze level 15

Maze level 20

a)
3. Results

Table 1: Data on level 5

| number of experiment | algorithm | Riming time of algorithm | Generated route distance |
|----------------------|-----------|--------------------------|--------------------------|
| 1                    | A*        | 11.2021                  | 23.115                   |
| 1                    | wavefront | 0.6597                   | 19.862                   |
| number of experiment | algorithm  | Riming time of algorithm | Generated route distance |
|----------------------|-----------|--------------------------|-------------------------|
| 2                    | A*        | 8.3405                   | 26.425                  |
| 2                    | wavefront | 0.6827                   | 22.089                  |
| 3                    | A*        | 5.0031                   | 19.05                   |
| 3                    | wavefront | 0.4847                   | 17.5                    |
| 4                    | A*        | 0.7623                   | 17.425                  |
| 4                    | wavefront | 0.4827                   | 17.678                  |

Table 2: Data on level 10

| number of experiment | algorithm  | Riming time of algorithm | Generated route distance |
|----------------------|-----------|--------------------------|-------------------------|
| 1                    | A*        | 3.7896                   | 28.738                  |
| 1                    | wavefront | 0.4730                   | 20.489                  |
| 2                    | A*        | 20.6844                  | 51.175                  |
| 2                    | wavefront | 0.4840                   | 26.511                  |
| 3                    | A*        | 11.2803                  | 39.2                    |
| 3                    | wavefront | 1.5333                   | 19.767                  |
| 4                    | A*        | 19.2398                  | 69.475                  |
| 4                    | wavefront | 0.6519                   | 21.133                  |

Table 3: Data on level 15

| number of experiment | algorithm  | Riming time of algorithm | Generated route distance |
|----------------------|-----------|--------------------------|-------------------------|
| 1                    | A*        | 60.1030                  | 62.9                    |
| 1                    | wavefront | 0.5978                   | 35.344                  |
| 2                    | A*        | 51.0395                  | 64.813                  |
| 2                    | wavefront | 0.7845                   | 36.178                  |
| 3                    | A*        | 18.5849                  | 39.225                  |
| 3                    | wavefront | 0.6469                   | 23.911                  |
| 4                    | A*        | 55.3182                  | 53.875                  |
| 4                    | wavefront | 0.7552                   | 36.3                    |
Table 4: Data on level 20

| number of experiment | algorithm | Rimming time of algorithm | Generated route distance |
|----------------------|-----------|---------------------------|--------------------------|
| 1                    | A*        | 101.5861                  | 68.875                   |
| 2                    | wavefront | 0.8503                    | 50.178                   |
| 2                    | A*        | 9.0558                    | 38.8                     |
| 3                    | wavefront | 0.8821                    | 31.5                     |
| 3                    | A*        | 74.3703                   | 57.575                   |
| 4                    | wavefront | 0.7604                    | 41.4                     |
| 4                    | A*        | 106.3337                  | 51.963                   |

4. Discussion

According to this experiment, it is clearly to show that two kinds of algorithm (A* algorithm and Wavefront algorithm) in which comparing the time solving the same level of maze and the cost of each route used. It is lucid to see from the table above that A* takes much longer time than Wavefront. This finding also corrects under the conditions of some simple maze. Moreover, it obviously proves that Wavefront has higher executive power than A* has, which means Wavefront has more advanced ability to find the way out of the maze. For example, when the maze level is 10, A* spends 3.7896, 20.6844, 11.2803, 19.2398, and Wavefront spends 0.4730, 0.4840, 1.5333, 0.6519. From these data, it is clearly that A* takes approximately 13.7 second to find the path and Wavefront spends around 0.786 second to solve the same maze. These information show that A* takes more than 20 times longer than Wavefront to solve the same maze. Furthermore, it does not have any recurring relationship with the difficulty of the maze, so, in any easy case, A* takes more time than Wavefront. Therefore, it can be concluded in cases which the maze level is below 20. Wavefront’s route planning ability is better than A* algorithm in the easier maze, Wavefront has more higher working efficiency to find the correct path than A*. From the above figure, it is obvious that A* algorithm has much more nodes and repeating path during the route planning process and Wavefront is following the edges of obstacle to find the shortest path, so A* will take much longer time to find the path compared with Wavefront algorithm.

For the length of the route, Wavefront algorithm also shows better efficiency than that of A*algorithm, but it has a little difference comparing to the cost of each algorithm. It has an increasing relationship with the difficulty of the maze. To be specific, the more difficult it is, the more cost A-algorithm and the wavefront algorithm need to take. For example, in maze level 5, A* needs to take average 21.5 m and Wavefront costs average 19.3 m to the goal. In the maze level 10, A* takes average 47.1 m and Wavefront spends average 22.0 m to the destination. In the maze level 15, A* takes average 55.2 m and Wavefront costs average 32.9 m to the end. In the maze level 20, A* costs average 54.3 m and Wavefront spends average 39.3 m to the goal point. From the above data, Wavefront has less cost than A* and it slightly shows that with the difficulty of maze goes up, the cost of Wavefront will exceed the cost of A* since A* has less difference compared with Wavefront in level 20 than that in level 15 and Wavefront will continue to increase. The reason is that Wavefront will only find the path which is very close to the wall of obstacle, which means it has to overcome the first obstacle and then to solve the next obstacle, so the cost will not reduce with the number of obstacle goes up. A*, on the other hand, is one algorithm which use nodes to detect the correct path in order to avoid the most expensive choices. In this way, A* algorithm will be the more efficient method in the complex maze.
To sum up, in the simple maze (difficulty of the maze below 20), Wavefront is more efficient than A*, and in the complex maze (difficulty of the maze above 20), A* is more convenient than Wavefront.

5. Conclusion
This project produces one situation that using A* and Wavefront algorithm to help robot to find the optimal path from the starting point to the goal without colluding obstacles. MATLAB software helps us to build this circumstance and to make simulation for each test. By comparing time and cost each algorithm spend for solving the same maze, it is easy to find out that Wavefront algorithm is more efficient than A* algorithm in handling these kinds of problems with the maze difficulty below 20. And in the higher maze level, A* will be more satisfied than Wavefront. Therefore, there is no best algorithm, only suitable for the environment. Due to defects of equipment and software, the maximum difficulty of maze simulation in this study is 20, so it does not have capability to test comprehensively higher level maze.

Exploring robot path planning is very wide studying field. Thus, many algorithm and technology need to be developed in order to provide more optimal plan for solving the problems like maze. In the future, this domain can be extended by increasing more trial in order to find one costless path.

References
[1] Simon Bøgh, "Little Helper" - An Autonomous Industrial Mobile Manipulator Concept, 2011
[2] Edpresso Team, What is the A* algorithm?
[3] Adel Al-Jumailly, Cindy Leung, Wavefront Propagation and Fuzzy Based Autonomous Navigation, 2005
[4] Intelligent Robotics
[5] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, “Principles of robot motion: theory, algorithms, and implementations. 2005,” MITPress, Boston
[6] Aboul Ella Hassanien, Khaled Shaalan, Tarek Gaber, Ahmad Taher Azar, M. F. Tolba, Proceedings of the International Conference on Advanced Intelligent Systems and informatics, 2016
[7] Campbell, J., Sukthankar, R., Nourbakhsh, I., Pahwa, A.: A robust visual odometry and precipice detection system using consumer-grade monocular vision. In: Proceedings of ICRA 2005, Barcelona, Spain (2005)
[8] Marina Chatterjee, A* Search Algorithm in Artificial Intelligence (AI), 2020