ISBN: 978-981-18-5852-9
WCSE 2022 Spring Event: 2022 9th International Conference on Industrial Engineering and Applications
doi: 10.18178/wcse.2022.04.068

Route Planning of Unmanned Surface Vehicle Based on Improved A-star Algorithm

Shuang Huang[1], Xu Cao[1], Lei Wang[2], Mi Yan[2], Xin Luo[2], Yang Liu[2]

[1] Wuhan second ship design and research institute Fifth design center

[2] Wuhan University of Technology School of Automation

Abstract. Path planning is one of the important topics in the USV research field. Aiming at the path planning problem of surface unmanned boats in a known environment, this paper proposes an improved A* algorithm. For the problem that the traditional A* algorithm can only search in 8 directions, this paper analyzes the path obtained by the traditional A* algorithm. The principle of the shortest line segment between two points is used for further optimization, so that the USV can search in any direction within the feasible range. Finally, compared with the traditional algorithm, the multi-direction improved A* algorithm is verified by simulation, the number of path nodes obtained by the search is less, the path distance is shorter, and it has better search performance.

Keywords: route plan; Unmanned Surface Vehicle; A* search algorithm

1. Introduction

Unmanned Surface Vehicle (USV) [1] is a new type of intelligent surface unmanned platform, which is usually used to perform some tasks with high risk factor and harsh operating environment. It can complete complex tasks in complex water environment. Once equipped with advanced control systems, sensor systems, communication systems and weapon systems, it can perform a variety of war and non-war military missions, such as reconnaissance, search, detection and demining, etc.

The environment in which USV travels is complex and easily affected by obstacles. Therefore, USV requires a high autonomous system, and path planning [2] is one of the most important systems. The USV path planning task needs to search for an optimal path (or suboptimal path) from the specified starting point to the target point within the safe navigation area according to certain optimization criteria. According to the degree of understanding of environmental information, the path planning system of USV can be divided into two types, one is a global path planning system for static obstacles to avoid [3]; the other is to realize USV to avoid dynamic obstacles The local path planning system of [4].

Many scholars have done a lot of research work in this field, and have explored a variety of effective solutions, which have continuously enhanced the planning ability of unmanned aerial vehicles. At present, the commonly used solution methods mainly include Djkstra algorithm [5], artificial potential field method [6], probability map method [7], A* algorithm [8], ant colony algorithm [9], etc. Among them, Djkstra [10] algorithm. It is a typical single-source shortest path algorithm. The advantage of this algorithm is that the operation is simple and the shortest path can be planned, but the amount of computation is large; the central idea of the artificial potential field method is to define the environment where the unmanned vehicle is located by a potential field, the obstacle avoidance driving of the UAV is carried out through the position information, and the environmental adaptability is strong, but there are problems of local optimality and inaccessibility of the target, and the design of the potential field space is the key to its application; The A* algorithm [11] uses a heuristic function to weigh the position of the optimal node of the path in the global map, and iteratively calculates the evaluation function to find a low-cost path. This algorithm improves the efficiency of the algorithm by reducing the search range, resulting in a limited search space. Therefore, the final path is not a global path. The optimal solution; the probabilistic map method is a path planning method based on heuristic node enhancement strategy. By randomly generating valid sampling points in the map to obtain a probabilistic roadmap, it can effectively avoid accurate modeling of complex environments, but this The algorithm is a probabilistic complete type. When the number of searches is too small or the sampling points are too small, the solution may not be found; the ant colony algorithm is a simulation optimization algorithm that simulates the foraging behavior of ants. The algorithm has strong robustness and inherent parallelism in solving performance. It is easy to be implemented by computer, but it is easy to fall into local optimum. Often used to solve the traveling salesman problem.

This paper improves the traditional A* algorithm, so that the algorithm is not limited to searching in 8 directions, but can search in any direction without encountering obstacles. First, a relatively optimal path is obtained by using the traditional A* algorithm, and then the obtained path is optimized according to the principle of the shortest line segment between two points, so that the overall path length is reduced, and a better path is obtained. And compare and analyze the better path with the traditional path to verify its rationality.

2. ENVIRONMENT MODELING

The establishment of the environment model is very important for the path planning system. First, environmental information needs to be captured, including static information and dynamic information. Then, an environment model is established for the obtained environment information, and the established model can fully represent the real environment of the USV during the driving process. Therefore, environmental modeling is very important, and the quality of the results will affect the length of the search path and the search time during path planning. This paper comprehensively considers the way of the experimental scene and the convenience of the environment modeling method, and considers the use of the grid method to model the experimental environment. The details of the environment modeling are analyzed and explained below.

2.1 Grid method modeling

At present, the application of electronic charts in underwater weapons is in the exploratory stage. Due to the complexity of electronic charts, it is usually necessary to convert electronic charts into a chart data environment model that can be directly used.

In this paper, the chart information rasterization method is used to render the digital chart of a certain sea area. First, the two-dimensional planning space is evenly decomposed into grid units, and the grid unit is used as the minimum moving unit in the path planning. The resolution of the grid can be adjusted according to actual needs. If a grid belongs to an obstacle area, it is recorded as a type 1 grid; if it does not belong to an obstacle area, it belongs to a passable area, and it is recorded as a type 0 grid, which indicates the information of obstacles and passable areas on the chart ; Secondly, the obstacle area is processed and merged, the non-navigable road sections and trapped road sections are eliminated, and the obstacle area is standardized into a polygonal graph, so that the constructed data space includes the identification starting point, target point, obstacle area and route position information. Algorithms can be easily used for path planning. The following assumptions are made on the environmental model in this paper:

1) Consider the USV as a particle, while the length of the planned path is within the USV range.

2) The planned marine environment is a two-dimensional space, and a certain size of the rule area is used to represent the obstacle area on the sea.

3) The influence of other interference factors such as tidal currents, ocean currents, and electronic interference are not considered.

In this paper, the Cartesian coordinate system is used in the global path planning, and the advantages of grid modeling are fully utilized to make the planning path simple and clear, convenient for mathematical analysis and calculation, and easy to implement.

Taking Fig. 1 as an example, this is a 10×10 grid space, and the method of rectangular coordinate system can be used to represent the obstacle area and feasible path in the figure, which are respectively expressed as the following forms.

Obstacle area:

$$O = [(2,4),(3,4),(2,5),(3,5),(5,2),(6,2),(6,6),(7,6),(6,7),(7,7)] \tag{1}$$

Possible paths:

$$R = [(1,1),(2,2),(3,3),(4,3),(5,4),(6,5),(7,5),(8,5),(8,6),(8,7),(8,8)] \tag{2}$$

The grid method can reduce the complexity of abstracting complex environments, and its algorithm principle and implementation process are relatively simple. However, it is more complicated to divide the grid, and many factors need to be considered to find a balance.

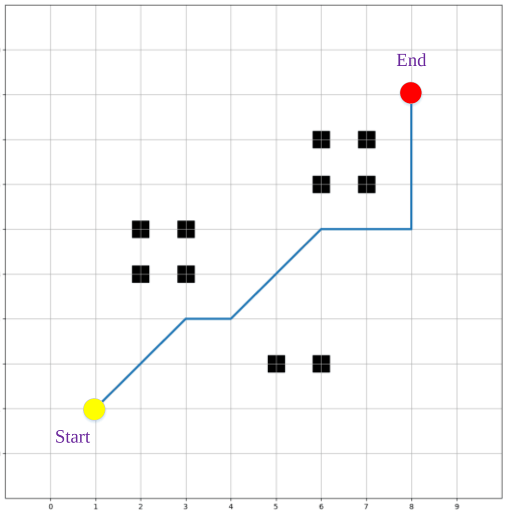

Fig.1: Schematic diagram of grid method

2.2 Grid resolution

The most important step in the raster modeling process is to determine the raster resolution of the raster map. The resolution of the raster map is different, and the precision of the raster map that is finally established will also be different. Different raster resolutions can create different raster map models according to the environment, and the time required for path search using the created raster map is also more different, and even the resulting paths are also different.

The size of the grid resolution is determined by the proportion of obstacles in the entire simulation environment. When the proportion of obstacles is small, it may have adverse effects, resulting in an inaccurate search result. Conversely, the determined grid strength can be increased, and when a better path search result is obtained, the time used by the algorithm to search is reduced.

3. A* ALGORITHM AND ITS IMPROVEMENT

3.1 Principle of A* algorithm

The A* algorithm is actually a heuristic algorithm and one of the most commonly used algorithms in path planning. The A* algorithm is not only used for path planning, and at the same time, there is not only A* heuristic method in path planning. Compared with other path planning algorithms, such as genetic algorithm and ant colony algorithm, the A* algorithm has a simpler algorithm process, is easier to understand, and runs faster. Moreover, the path planning results of applying A* are not bad. Therefore, in general, the A* algorithm should be a cost-effective path planning algorithm.

The basic idea of the A* algorithm is that, for the current search point $CNode$, first find all the neighbor nodes of the node $CNode$, such as moving in four directions, there are 4 neighbor nodes, and moving in eight directions, there are 8 neighbor nodes; then calculate all neighbors The objective function value of the node is $f(x)$, the objective function in the path planning is the distance from the starting point to the end point through the current node; finally, select the node with the smallest $f(x)$ value from all neighboring nodes, and repeat The above process until the search reaches the end point. The cost function $f(x)$ is calculated as follows:

$$f(x) = g(x) + h(x) \tag{3}$$

Among them, $g(x)$ refers to the actual minimum distance from the starting point to the current node $CNode$, and $h(x)$ refers to the estimated minimum distance from $CNode$ to the target node. The quality of the search results of the A* algorithm largely depends on $h(x)$ choice.

There are two main methods for calculating distance. One is the Euclidean distance, that is, the object can move in eight directions when moving, and its calculation formula is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4}$$

The other is the Manhattan distance, that is, the object can only move in four directions when moving, and cannot move diagonally. The distance calculation formula is:

$$d = (x_2 - x_1) + (y_2 - y_1) \tag{5}$$

This paper adopts the Euclidean distance method, so that the USV can move in eight directions, and the obtained path is closer to the shortest path, as shown in Fig. 2.

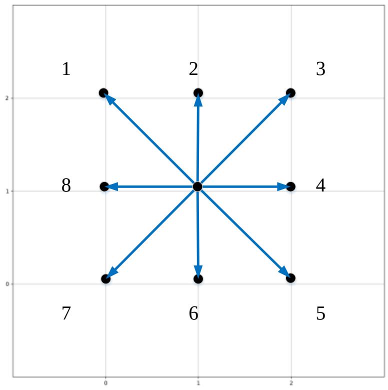

Fig. 2 :USV 8-direction search diagram

The A* algorithm search path needs to create two lists, the $OPEN$ table saves all the nodes that have been generated but not expanded, and the $CLOSED$ table saves the obstacles and the nodes that have been expanded. First put the starting point S into the $OPEN$ table and use it as the starting node, and save all obstacle grid nodes in the $CLOSED$ table. Secondly, the starting node is expanded to generate child nodes, the child nodes point to the parent node, and the child nodes that are not in the $CLOSED$ table are added to the $OPEN$ table, and the node with the smallest cost f(x) in the $OPEN$ table is searched and added to the $CLOSED$ table. Then take this node as a new starting node, and perform the above search operation again. The search stops when the target point is reached or the $OPEN$ table is empty. The specific search steps of the A* algorithm are as follows:

1) Create two empty lists, $OPEN$ and $CLOSED$.

2) Store the starting node S information in the $OPEN$ table and use it as the starting node, and save the obstacle grid node in the $CLOSED$ table.

3) Determine whether the $OPEN$ table is empty. If the $OPEN$ table is empty, the search fails and the loop ends; otherwise, the search continues.

4) Put the node x with the smallest cost value in the $OPEN$ table, which is the first node in the $OPEN$ table, into the $CLOSED$ table.

5) Determine whether the node x is the target node, if it is the target node, trace the parent node from the node to the start node, the line is the best track path, and the search is successful, and the loop ends; otherwise, go to step 6).

6) Expand the node x as the parent node of the next group of nodes, generate child nodes $x_i (i = 1,2\cdots)$, and calculate the cost value $f(x_i)$ of all child nodes.

7) Remove the nodes in the $CLOSED$ table and put the remaining child nodes in the $OPEN$ table. Compare the cost value of the child nodes that originally existed in the $OPEN$ table. If the cost value is smaller, update the parent node and cost value of the node. Then sort all nodes in the $OPEN$ table according to the cost value from small to large, and go to step 3).

3.2 Improved A* algorithm

The traditional A* algorithm uses eight-direction search, which limits the path angle to a certain extent, resulting in the search path being longer than the ideal optimal path, and the path with any navigation angle is the shortest path in the true sense. The path obtained by the A* algorithm is optimized. As shown in Fig. 3, due to the need for obstacle avoidance, we assume that the corner of the obstacle cannot be directly reached, that is, the feasible path of the USV cannot pass through the area surrounded by the purple dotted line in the figure. The optimal path obtained by the traditional eight-direction search A* algorithm is $P_1 \to P_2 \to P_3 \to P_4 \to P_5 \to P_6 \to P_7 \to P_8$, obviously connecting P_1 with P_4 and P_4 with P_8, the resulting path $P_1 \to P_4 \to P_8$, with shorter distances and fewer intermediate node.

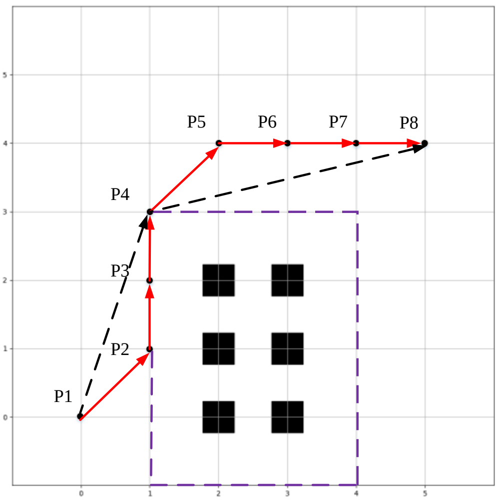

Fig. 3: Eight-direction A* algorithm search results

Based on the path obtained by the traditional A* algorithm, the paper uses the local search method to optimize the path.The optimal path node is extracted to obtain the optimal path of any angle. The schematic diagram of path optimization is shown in Fig. 4. Initialize the optimization path $Path\_new = [P_1]$, take the node P_1 as the optimization point and connect the node P_3 with a straight line. If there is no obstacle grid between the lines, continue to connect with P_4, and there is no obstacle grid between P_1 and P_4, then P_1 continues to connect with P_5, and there is an obstacle grid between the lines, then save the path $P_1 \to P_4$,update $Path\_new = [P_1; P_4]$, the current optimization node is updated to P_4, P_4 is connected with P_6 and P_7 in a straight line, and there is no obstacle grid. Continue to connect with P_8 in a straight line, and there is no obstacle grid, and P8 is The target node, ends the loop, and adds P_8 to $Path\_new$, that is, $Path\_new = [P_1; P_4; P_8]$. Output $Path\_new$ at this time, which is the path optimized by the multi-direction improved A*algorithm.

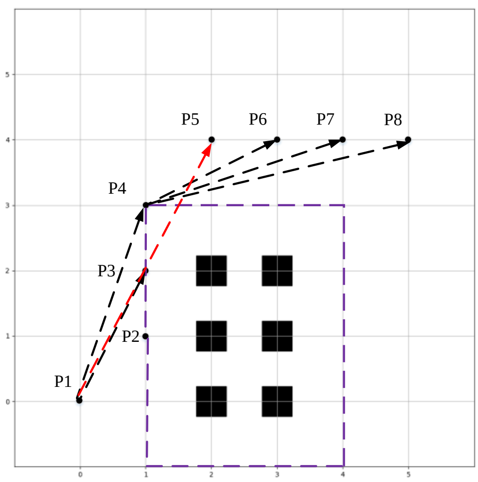

Fig. 4: Schematic diagram of improved A* algorithm path optimization

The specific optimization steps of the multi-directional improved A* algorithm are as follows:

1) Enter the path obtained by the eight-direction search of the A* algorithm $Path = [P_1; P_2; \cdots; P_i; \cdots; P_s]$, where P_i represents the coordinates (x_i, y_i) of the current grid, and s represents the total number of path nodes. Initialize $i = 1$, add P_i to the optimized path $Path\_new$, that is, $Path\_new = [P_i]$, and use it as an optimized node, that is, $P_{op} = P_i$.

2) Determine whether s is less than 3. When $s < 3$, it means that there is a straight-line path between the starting point and the target point, that is, the shortest path, and no further optimization is needed. Set $Path\_new = Path$ to end the loop; when $s \geq 3$, it means that the starting point and the target point are between the starting point and the target point. There is a path node in the middle, go to the next step.

3) Determine whether there is an obstacle grid between the optimization node P_{op} and the path node P_{i+2}. If there is no obstacle grid between the lines, go to the next step; if there is an obstacle grid between the lines, add the path node P_{i+1} to $Path\_new$, that is, $Path\_new = [Path\_new; P_{i+1}]$, and take P_{i+1} as the new optimization node, namely $P_{op} = P_{i+1}$.

4) $i = i + 1$.

5) Determine whether i is less than s. When $i < s$, go back to step 3); when $i \geq s$, add P_s to $Path\_new$, that is, $Path\_new = [Path\_new; P_s]$, and end the loop.

6) Output optimized path $Path\_new$.

4. SIMULATION AND ANALYSIS

On the basis of grid modeling, this paper uses the traditional A* algorithm and the improved A* algorithm to test two custom sea areas, and compares the number of nodes and path lengths of the two methods.

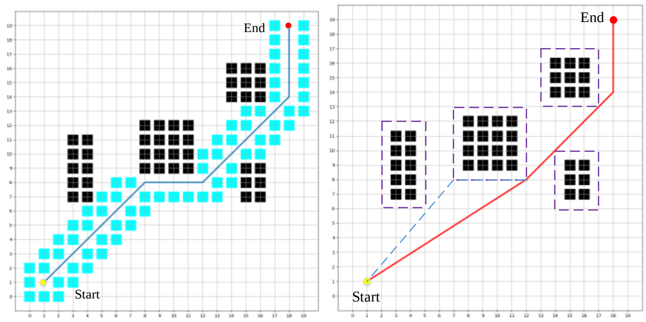

Fig. 5: Traditional and Improved A* algorithm in custom sea area 1

The traditional A* algorithm and the improved A* algorithm are used to test and search two custom sea areas, respectively. The paths of different sea areas are shown in Fig. 5, and Fig. 6. The blue solid line represents the path obtained by using the traditional A* algorithm, and the red solid line represents the path obtained by using the improved A* algorithm. It can be seen that the length of the path obtained by the improved A* algorithm is shorter, and the node Fewer numbers. Cyan squares represent nodes that have been added to the *OPEN* table, and purple areas represent inaccessible obstacle areas. In order to measure the effect of improving the A* algorithm, the comparison table of the number of nodes and distances of different search algorithms is shown in TABLE Ⅰ.

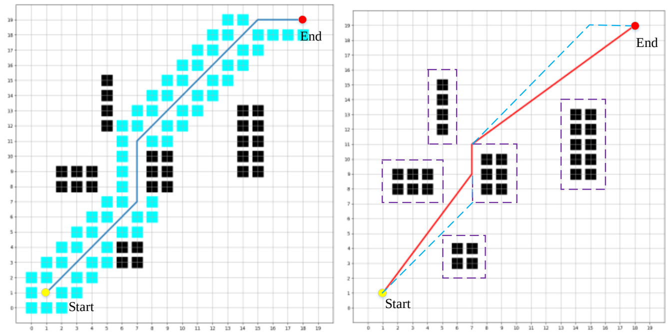

Fig. 6: Traditional and Improved A* algorithm in custom sea area 2

Table 1:Comparison of different algorithms

Custom sea area	Traditional A* algorithm		Improved A* algorithm		Node reduction rate (%)	Distance reduction rate (%)
	number	distance	number	distance		
area 1	23	27.38	4	26.52	82.61	3.14
area 2	22	26.80	4	25.60	81.82	4.48

From the reduction rate of the number of nodes in TABLE Ⅰ, it can be clearly seen that the improved A* algorithm compared with the traditional A* algorithm, in different sea areas, the number of searched path nodes is greatly reduced. This is because the patrol path network constructed by the traditional A* algorithm, the path storage is based on the grid, the number of nodes is large and the turning angle is limited. In the improved A* algorithm, the path storage uses the starting point and the end point of the straight path as nodes for storage, which greatly reduces the number of nodes. In addition, from the distance reduction rate in Table 1, it can also be seen that the path distance obtained by the improved A* algorithm is shorter than that of the traditional A* algorithm. This is because the improved A* algorithm replaces the polyline path with a collision-free straight path, with the shortest line segment between two points and therefore a shorter path distance.

5. CONCLUSION

Aiming at the USV track planning problem, this paper designs a USV track planning method based on the traditional A* algorithm, which improves the A* algorithm in multiple directions. Further optimization breaks the limitation that the traditional A* algorithm can only search in 8 directions, so that the USV can sail in any direction.

It is verified by simulation that the path planned by the algorithm can effectively complete the task of USV track planning. In addition, compared with the traditional algorithm, the multi-direction improved A* algorithm can obtain fewer path nodes, shorter path distance, and better performance. search performance.

The simulation in this paper is only carried out in the known environment, and the actual navigation environment of the USV is dynamically changing, and the trajectory planning for the dynamically changing environment is the focus of the next work.

6. References

[1] Junxiang Hu;Yuan Ge;Xu Zhou;Shuo Liu;Jincenzi Wu.Research on the course control of USV based on improved ADRC[J].Systems Science & Control Engineering,2021,Vol.9(1): 44-51.

[2] Qingwu Wang 1 ;Ming Zhong 2 ;Guoyou Shi 1 ;Jian Zhao 1 ;Chunjiang Bai 1 .Route Planning and Tracking for Ships Based on the ECDIS Platform[J].IEEE Access,2021,Vol.9: 71754-71762.

[3] Nurlita Gamayanti;Rusdhianto Effendi Abdul Kadir;Abdullah Alkaff.Global Path Planning for USV Waypoint Guidance System Using Dynamic Programming[A].2020 International Seminar on Intelligent Technology and Its Applications (ISITIA)[C],2020.

[4] Xiaotian Zhang;Xiyuan Chen.Path Planning Method for Unmanned Surface Vehicle Based on RRT* and DWA[J].Multimedia Technology and Enhanced Learning,2021,Vol.387: 518-527.

[5] Application of electronic map-based modified Djkstra's algorithm in railway planning[A].2011 International Conference on Energy Systems and Electrical Power, ESEP 2011[C],2011.

[6] Yu Tian 1 ;Xiaojun Zhu 2 ;Deshan Meng 3 ;Xueqian Wang 4 ;Bin Liang 5 .An Overall Configuration Planning Method of Continuum Hyper-Redundant Manipulators based on Improved Artificial Potential Field Method[J].IEEE Robotics and Automation Letters,2021,: 1.

[7] Xiaole Yue[a];Yong Xu[a];Wei Xu[a];Jian-Qiao Sun[b].Probabilistic response of dynamical systems based on the global attractor with the compatible cell mapping method[J].Physica A: Statistical Mechanics and its Applications,2019,Vol.516: 509-519.

[8] Chen, Ge 1 ;Wu, Tao 1,2 ;Zhou, Zheng 3 .Research on Ship Meteorological Route Based on A-Star Algorithm.[J].Mathematical Problems in Engineering,2021,Vol.2021(1): 1-8.

[9] Rui Zhang;Weibo Sun;Sang-Bing Tsai.Simulation of Sports Venue Based on Ant Colony Algorithm and Artificial Intelligence[J].Wireless Communications and Mobile Computing,2021,Vol.2021.

[10] Ming Zhang, Chenglong Gong, Yuan Feng;Chao Liu.Energy-Predicted Shortest Routing Tree Algorithm in Wireless Sensor Networks[J].Lecture Notes in Computer Science,2008,Vol.5370(1): 117-124.

[11] Gang Tang 1 ;Congqiang Tang 1 ;Christophe Claramunt 2 ;Xiong Hu 1 ;Peipei Zhou 3 .Geometric A-star Algorithm: An improved A-star Algorithm for AGV Path Planning in a Port Environment[J].IEEE Access,2021,Vol.9: 1.