International Core Journal of Engineering Volume 6 Issue 5, 2020
ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

Dynamic Path Planning for Collision Avoidance of USV based on D* Algorithm

Lin Xu

School of ShangHai Maritime University, College of Transport & Communication, China.

1211454066@qq.com

Abstract

This paper proposes a dynamic path search method for collision avoidance of the unmanned surface vehicle (USV) based on D* algorithm to solve the problems of slow search speed, long route and low obstacle avoidance efficiency faced by the USV in path planning with dynamic obstacles. The method uses the information of voyage environment based on the electronic sea chart. Firstly, it carries on the reverse calculation from the target point to the starting point and stores the information of each possible position on the path. Then, the USV is allowed to move from the starting point to the target point. When dynamic obstacles appear on the next path, it only needs to change the local path behind to avoid obstacles so that it is more efficient than other methods because of less computation. The part of model simulation is using MATLAB programming to realize the running program of dynamic collision avoidance path planning of USV based on D* algorithm, and comparing with the simulation results of path planning based on the traditional Dijstra algorithm and A* algorithm. The test results show that the D* algorithm can significantly reduce the search time and the path length when USV meet dynamic obstacles.

Keywords

USV; Collision avoidance; Dynamic path planning; D* algorithm.

1. Introduction

The research and exploration on USV are in the ascendant since the realization of USV has been supported by science and technology with the rapid development of new concepts and technologies such as communication and artificial intelligence. Since it is inevitable for USV to encounter other ships in the course of collision avoidance and navigation control, dynamic collision avoidance path planning for USV has become a hot issue. The dynamic collision avoidance path planning of USV refers to the navigable path from the origin to the destination and the completion of dynamic collision avoidance based on the principles of fast search speed and short voyage distance in the sea environment with known or unknown obstacles. A difficulty arises because of the need to avoid both static obstacles and dynamic obstacles encountered during the voyage.

At present, the algorithms used for USV path planning are becoming mature, mainly including the following algorithms: Dijstra Algorithm[1], Ant Colony Algorithm[2], A* Algorithm[3]-[4], Artificial Potential Field Method[5], Particle Swarm Optimization Algorithm[6], Firefly Algorithm[7] and Genetic Algorithm[8]. Research results on collision avoidance of USV are also abundant. Restricted by international rules for collision avoidance at sea[9], relevant experts and scholars have devoted themselves to qualitative research and quantitative analysis of ship models[10]-[11], and proposed methods for collision avoidance of ships for collision avoidance scenarios of single ship and multiple ships[12]-[15].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

However, the research on dynamic collision avoidance path planning of USV is relatively slow. Yutao Kang, Daqi Zhu[16] conducted a classified review and evaluation of ship collision avoidance model and path planning methods, and they analyzed the development trend of ship collision avoidance path planning. Ming cheng Tsou[17] uses ECDIS as the information platform for navigation decision support to generate a route that can simultaneously avoid collision and geographical obstacles for multiple ships. Yanlong Wang, Xuemin Yu[18] reported the preliminary research results of an automatic obstacle avoidance method for USV based on COLREGs. Haiqing Shen, Chen Guo[19] proposed an intelligent collision avoidance navigation method of USV based on deep competition Q learning algorithm and A* algorithm, aiming at the problem of automatic collision avoidance of USV under complicated maritime navigation conditions. Xue, Y, Lee, B.S.[20] proposed an automatic trajectory planning and collision avoidance method based on artificial potential field and velocity vector. Ning Li and Fei Xue [21] proposed a hybrid path planning algorithm combining global path planning and local path planning by using the improved particle swarm optimization algorithm and the improved artificial potential field method, to deal with the unknown surrounding environment and the existence of moving obstacles during the navigation of USV.

In the algorithm research of dynamic path planning for USV above, although the goal of dynamic obstacle avoidance can be basically achieved in different degrees, the uncertainty of obstacles under dynamic environment is not taken into comprehensive consideration, and there are still some problems such as low search efficiency and long path planning when encountering dynamic obstacles. Therefore, in this paper, aiming at the difficulty of dealing with both static and dynamic obstacles in dynamic path planning of USV, local path planning is carried out when the USV completes the collision avoidance process based on D* algorithm.

D* algorithm is an algorithm based on the known dynamic environment of the information part. It has the advantages of small computation, strong real-time performance, low complexity and easy combination with other algorithms. The process of implementing D* algorithm in this paper can be divided into two parts: environment awareness and dynamic path planning. Environmental perception is the selection of appropriate methods to describe the sea conditions of USV, mainly referring to static and dynamic obstacle information, which is the most basic part of the problem of dynamic collision avoidance path planning for USV. The quality of environmental perception determines the quality of the path. Dynamic path planning is based on environmental awareness, planning from the starting point to the target point of the collision of the better path. The path planning method of USV is studied by using D* algorithm, aiming at realizing autonomous navigation function of USV in complex Marine environment.

2. Situational Awareness

2.1 Rasterization of navigable environment

In order to express the position information of obstacles conveniently, this paper uses grid method to divide the chart environment of USV into a square unit. By judging whether each unit contains obstacles, the navigable area and unnavigable area of the whole chart environment can be accurately expressed. Firstly, the environment map of USV navigation was obtained through AIS. Secondly, according to the working principle of raster method, the navigation environment of USV is rasterized, that is, a two-dimensional grid is divided on the environment map according to the appropriate cell size, and a two-dimensional matrix is established according to the size of the grid, so that each cell of the grid can correspond to the elements in the matrix. Then, determine whether the USV can sail in each unit in turn. For the unit that can sail, set the element of its corresponding matrix as 0. For unnavigable elements (that is, containing obstacles), set the element in the corresponding matrix to 1. Finally, the 0-1 matrix is used in MATLAB to express the navigation environment of USV, and path planning is made according to this environment.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

2.2 Methods for determining the size of raster cells

However, this approach also leads to a difficulty: the size of the unit affects the complexity of the algorithm and the ability to find the path. Therefore, this paper refers to the ship domain model to determine the size of a unit. In view of the actual size and safety of the USV, the ship domain model proposed by Japanese scholar Fujii Ra is adopted. According to the principle of the model, the ship field of USV is an ellipse centered on the ship and with the direction of the terminal connection of USV as the direction of the long half axis. The size of the ellipse will be adjusted according to different navigation conditions. Assuming that the captain of the USV is L, then in general, the ship domain dimensions of the USV are an ellipse composed of 8L and 3.2l. If the USV is navigating in a port or narrow waters with speed limit requirements, the dimensions of the ship field drop to an ellipse consisting of 6L and 1.6l, as shown in figure 1.

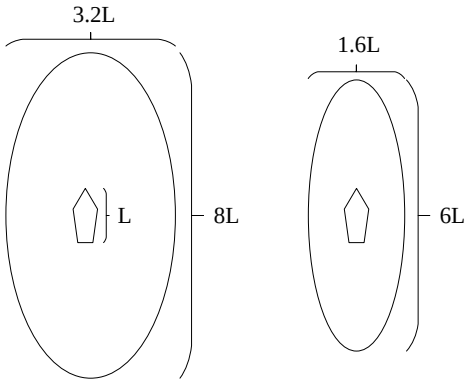

Figure 1. Fujii model

Considering the USV usually sail in open waters, the size of the ship domain adopted in this paper is an ellipse composed of 8L and 3.2L. Based on the ship domain of USV, grid cells processed by "expansion" are generated so that a unit can just contain a ship domain. Therefore, the side length of a grid cell is 8L, as shown in figure 2.

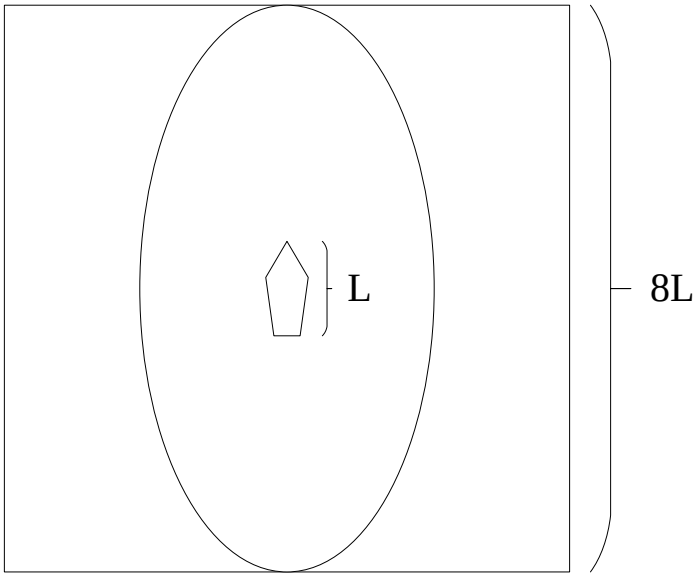

Figure 2. Determine the size of the grid cell

2.3 Establishment of Grid Environment Model

The electronic chart provides detailed and accurate environmental information about the Marine environment, so electronic charts are used to obtain information about the Marine environment, as shown in figure 4. When modeling the environment, we use matrix to store the coordinate information of each grid, and use cartesian coordinate method and ordinal method to represent the coordinates of each grid. The sequence number of the grid is consistent with the element number in the MATLAB

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

index matrix, and the sequence number of each grid corresponds to the coordinates of the matrix elements, as shown in figure 3.

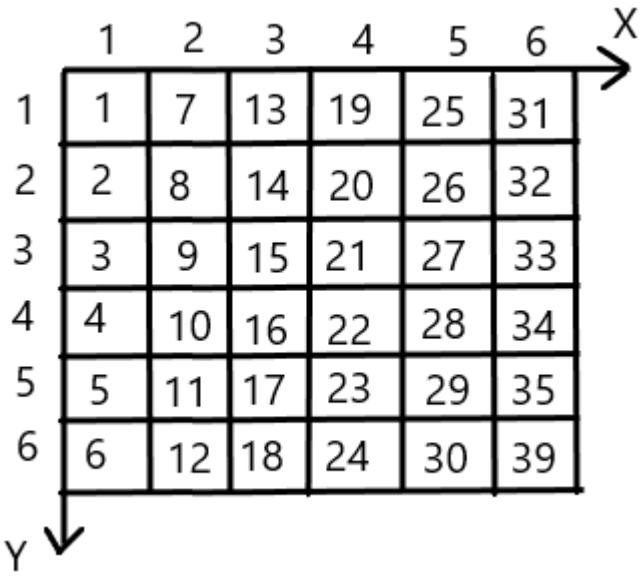

Figure 3. Grid and coordinate diagram

The above two grid representations can be converted to each other, and the cartesian coordinates (x, y) can be converted to the ordinal number I using the following formula:

$$i=(x-1)*nrow + y \tag{1}$$

In this formula, nrow represents the number of rows of the matrix.

Accordingly, the serial number I can also be converted into cartesian coordinates by the following formula:

$$\begin{cases} x = fix(i/nrow) + 1 \\ y = mod(i/10) \end{cases} \tag{2}$$

In this formula, mod refers to the operation of seeking mod, and fix refers to the operation of rounding mod.

Depending on whether the USV can navigate in a certain grid, the environment matrix determines the values of each element by the following formula:

$$a_{ij} = \begin{cases} 1 & \text{Impassable grid} \\ 0 & \text{Passable grid} \end{cases} \tag{3}$$

When making a grid chart, 0 is represented by white and 1 by black, as shown in figure 5.

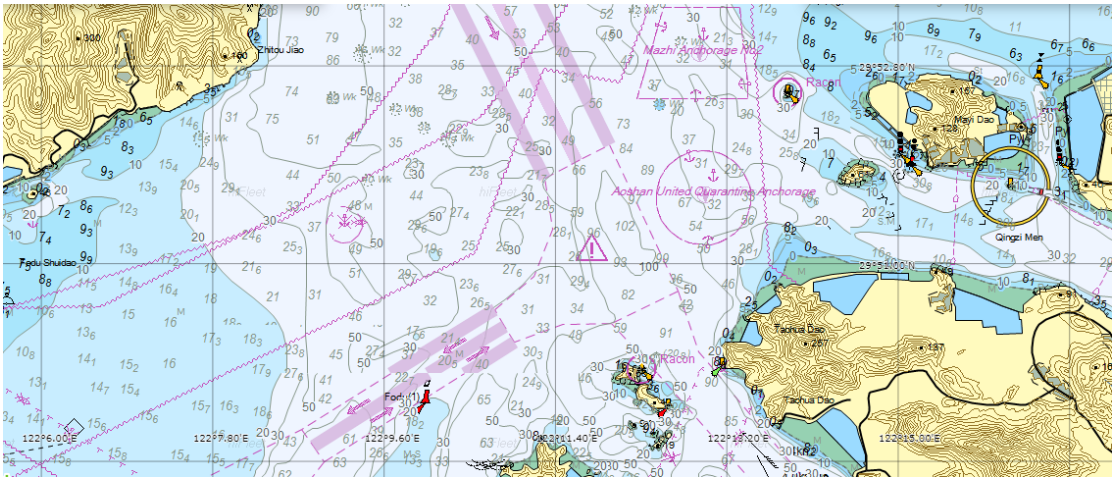

Figure 4. Original electronic sea chart

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

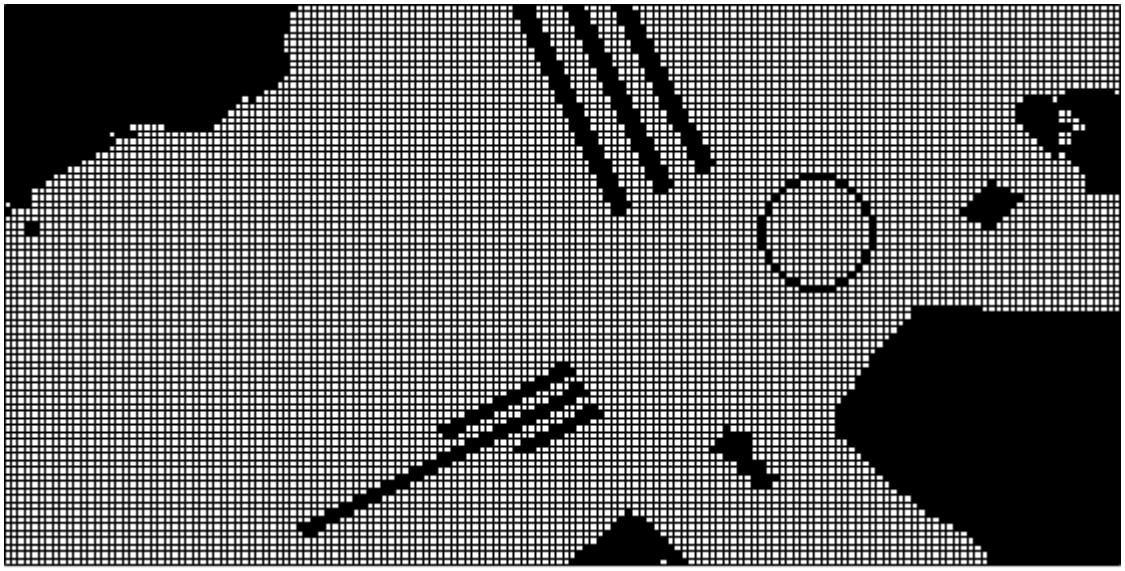

Figure 5. Grid environment model

3. Dynamic Path Planning

After the environment modeling is completed, the dynamic path of the USV is planned by D* algorithm, and the path points in the environment are represented by State. The principle of this algorithm is to reverse search the path points in the scene by maintaining a priority queue (OPEN list), and start the search by placing the target point in the OPEN list until the node of the current position of USV is out of the queue. Of course, if there is a dynamic change in the state of a node in the middle, we need to find the way again, so this is a dynamic pathfinding algorithm. The algorithm opens up three state lists, OPEN, CLOSED and NEW, to store different node information: OPEN list is used to store the path representation value of nodes to be updated; CLOSED lists are used to store the path cost of visited nodes, that is, node information that was in the OPEN table but has now been removed; The NEW list is used to store the path cost of an unvisited node, that is, one that has not joined an OPEN list.

The information to be used by each node is represented by the following parameters:

- X: location information of the current node of USV.
- Y: location information of the next node of USV.
- Backpointer: a pointer to the previous state, which is the parent of the current state. The current state is called the descendant of the state, and the target state has no Backpointer.(after the path search is completed, the USV can move step by step to the by backpointer, the Goal state is expressed in terms of G),b(Y)=X: represents that the parent node of node Y is X, that is, node Y is extended from node X.
- c(X,Y): path cost from X to Y.
- t(X): labels with state X (such as OPEN, CLOSED, and NEW).
- h(X): shortest path cost from state X to G.
- k(X): It is the Key Function and this value is the ordering basis for each node in the priority queue OPEN List. The State with the minimum k value is located in the queue header. For State X in the OPEN list, k(X) represents the minimum cost h(X) from X to G after X is placed in the OPEN List. In other words, when the state X changes, for example, the original free grid becomes an obstacle grid during the traveling process, then the cost h(X) will change. The value of k is the minimum value of h(k(X)=min(hold(X), hnew(X)) before and after the change.

There are 8 Y's corresponding to each node X, that is, 8 fields of X, as shown in figure 6. The gray square represents the square containing obstacles.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

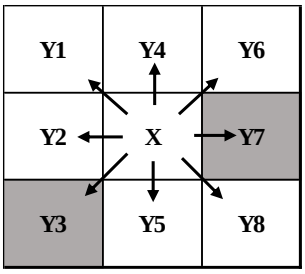

Figure 6. 8 fields of node X

Depending on the search direction from X to Y and whether Y is a free grid, the calculation methods of c(X,Y) are shown in table 1.

Table 1. Calculation formula of c(X,Y)

	Horizontal/Vertical Traversal	Diagonal Traversal
Y is a free cell	1 (e.g.(X,Y4))	1.4 (e.g.(X,Y1))
Y is an obstacle cell	Inf (e.g.(X,Y7))	Inf (e.g.(X,Y3))

The practical significance and calculation method of each parameter of D* algorithm are determined. The path planning of USV is composed of the following 6 steps:

Step1: convert the actual chart into the grid environment model, t for all states is set to NEW, h(G) is set to 0, and G is placed in the OPEN List;

Step2: accessibility discrimination: if the Open list is empty, the target point cannot be reached; otherwise, move to the next step;

Step3: put all the 8 child nodes of the current node into the Open list, make the b=G of each node, and calculate its h value and k value. Where, h=h+c, select the node X with the smallest k value in the Open list as the next node.

Step4: put node X into Closed list;

Step5: if node X is the target point, it indicates that the path has been found and the algorithm is over; otherwise, it will enter the next step.

Step6: expand the 8 child nodes Y around node X, calculate the h value and k value of each child node, and update the Open list with the state transfer function process_state, and turn to Step2.

The running process of the state transfer function is shown in figure 7.

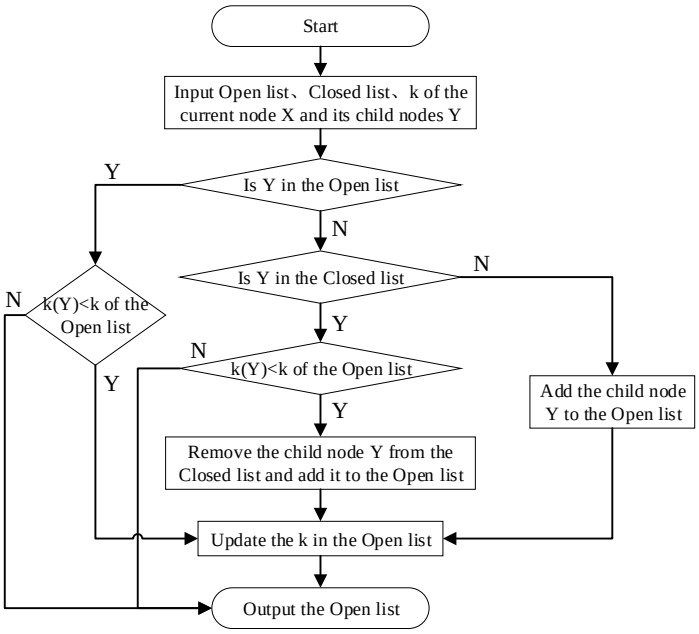

Figure 7. Running process of the Process state function

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

According to the above steps, a simple grid cost calculation is shown in figure 8, where the gray grid is the obstacle grid, the arrow indicates the current grid's father node, the blue grid is the path of the plan.

6	h=0 k=0 b= GOAL	h=1 k=1 b=(1,6)	h=2 k=2 b=(2,6)	h=Inf k=Inf b=(3,6)	h= k= b=	h= k= b=
5	h=1 k=1 b=(1,6)	h=1.4 k=1.4 b=(1,6)	h=2.4 k=2.4 b=(2,6)	h=Inf k=Inf b=(3,6)	h= k= b=	h= k= b=
4	h=2 k=2 b=(1,5)	h=2.4 k=2.4 b=(1,5)	h=2.8 k=2.8 b=(2,5)	h=Inf k=Inf b=(3,5)	h=Inf k=Inf b=(4,3)	h=6.6 k=6.6 b=(5,3)
3	h=3 k=3 b=(1,4)	h=3.4 k=3.4 b=(1,4)	h=3.8 k=3.8 b=(2,4)	h=4.2 k=4.2 b=(3,4)	h=5.2 k=5.2 b=(4,3)	h=6.2 k=6.2 b=(5,3)
2	h=4 k=4 b=(1,3)	h=4.4 k=4.4 b=(1,3)	h=Inf k=Inf b=(2,3)	h=5.2 k=5.2 b=(3,3)	h=5.6 k=5.6 b=(4,3)	h=6.6 k=6.6 b=(5,3)
1	h=5 k=5 b=(1,2)	h=5.4 k=5.4 b=(1,2)	h=5.8 k=5.8 b=(2,2)	h=6.2 k=6.2 b(4,2)	h=6.6 k=6.6 b=(4,2)	h=7 k=7 b=(5,2) START
	1	2	3	4	5	6

Figure 8. Grid cost calculation of D* algorithm

The USV is running along the shortest path of calculation, and constantly determines whether there is a new obstacle in the next node. If there is no new obstacle in the next node, it is no need to calculate, and the shortest path information that is calculated before using the shortest path is going to continue to navigate from the starting point. When the other ship was found, the USV adds the information to the map, adjusting the three lists of OPEN,CLOSED and NEW, adjusting the h value in the affected grid to Inf, and then moving to the second step of the path plan, using the stored information, and replanning the new shortest path from current coordinates to the target point, as shown in figure 9. The USV repeats the process in the voyage until it reaches the target point.

6	h=0 k=0 b= GOAL	h=1 k=1 b=(1,6)	h=2 k=2 b=(2,6)	h=Inf k=Inf b=(3,6)	h= k= b=	h= k= b=
5	h=1 k=1 b=(1,6)	h=1.4 k=1.4 b=(1,6)	h=2.4 k=2.4 b=(2,6)	h=Inf k=Inf b=(3,6)	h= k= b=	h= k= b=
4	h=2 k=2 b=(1,5)	h=2.4 k=2.4 b=(1,5)	h=2.8 k=2.8 b=(2,5)	h=Inf k=Inf b=(3,5)	h=Inf k=Inf b=(4,3)	h=Inf k=6.6 b=(5,3)
3	h=3 k=3 b=(1,4)	h=3.4 k=3.4 b=(1,4)	h=3.8 k=3.8 b=(2,4)	h=Inf k=Inf b=(3,4)	h=9.2 k=5.2 b=(4,2)	h=9.6 k=6.2 b=(5,2)
2	h=4 k=4 b=(1,3)	h=4.4 k=4.4 b=(1,3)	h=Inf k=Inf b=(2,3)	h=Inf k=Inf b=(3,3)	h=8.2 k=5.6 b=(4,1)	h=9.2 k=6.6 b=(5,1)
1	h=5 k=5 b=(1,2)	h=5.4 k=5.4 b=(1,2)	h=5.8 k=5.8 b=(2,2)	h=6.8 k=6.2 b(3,1)	h=7.8 k=6.6 b=(4,1)	h=8.8 k=7 b=(5,1) START
	1	2	3	4	5	6

Figure 9. Grid cost calculation diagram for the appearance of new obstacles

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

4. Simulation

4.1 Simulation experiment

Based on the above analysis, the simulation experiment was carried out on the MATLAB platform, and the navigation environment of USV was modeled by means of grid chart in this paper. The length of the USV provided for the experiment is about 12.5 meters, and the side length of a grid cell is determined to be 100 meters according to the ship domain model. Based on the AIS electronic chart information display system, a section of complicated intersection chart with a side length of 4.4 nautical miles (about 8 kilometers) is intercepted, and the ocean region is divided into a square grid with a side length of 100 meters by grid method, which is mapped to a spatial coordinate system, thus producing 80*80 grids, that is, 80*80 coordinates. Simulation planning sets the starting point coordinates as (13,80) and the ending point coordinates as (80,11).

In the simulation experiment, the traditional A* algorithm, Dijstra algorithm and D* algorithm were respectively used for experimental comparison under the same environment.

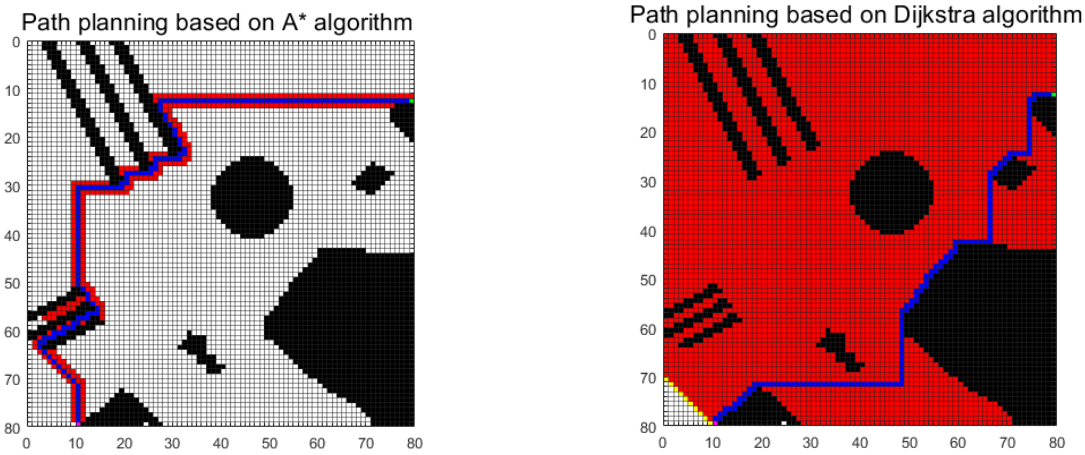

Figure 10. A* algorithm and Dijstra algorithm search path

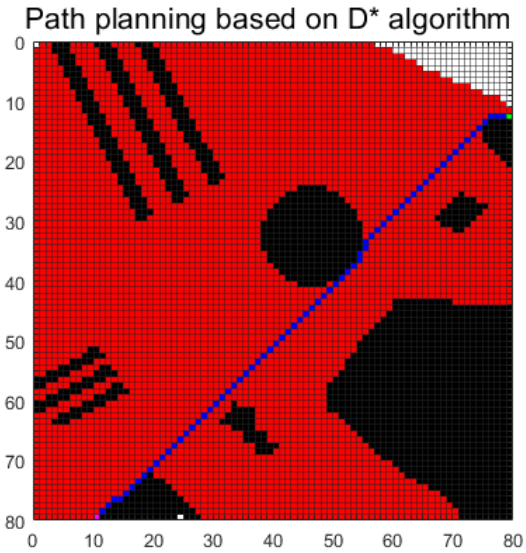

Figure 11. D* algorithm search path

If the USV encounters obstacles in the course of sailing, the obstacle information can be added to the algorithm in time (the new obstacle grid in the figure below is the other ship), and the D* algorithm can adjust the route in time.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

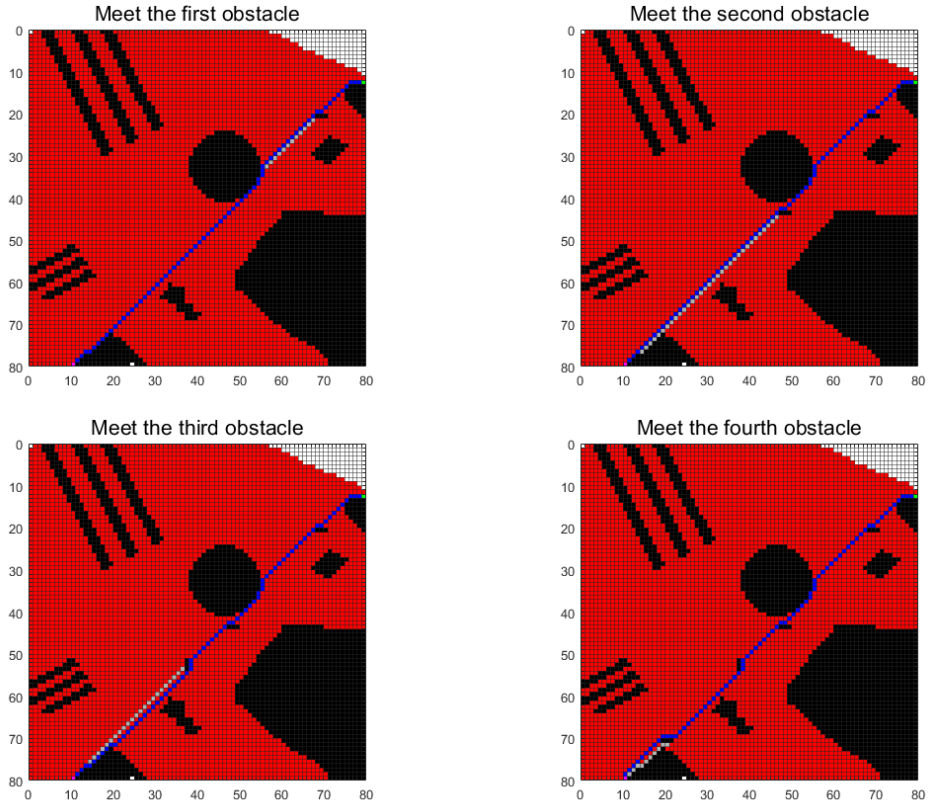

Figure 12. The path of D* algorithm changes when it encounters obstacles

4.2 Simulation and result analysis

According to the experimental results, it can be seen from figure 10-12 that the traditional A* algorithm or Dijstra algorithm USES 4 domains to search for nodes, which not only makes the node expansion too tedious, but also causes the sawtooth effect, resulting in long path polylines and many inflection points. However, D* algorithm USES 8 domain search nodes, which not only shortens the length of the optimized path, but also reduces the inflection point. Table 2 lists the data generated by three different algorithms under the same experimental conditions. From the analysis of simulation results, it can be seen that all three algorithms can calculate the shortest path from the start to the end. D* algorithm can replan the subsequent path in time when faced with dynamic obstacles, which has A good real-time performance, while the traditional A* algorithm or Dijstra algorithm cannot deal with dynamic obstacles. It can be seen from this that D* algorithm is more effective in solving the dynamic collision avoidance path planning problem of moving USV.

Table 2. Comparison of A* algorithm, Dijstra algorithm and D* algorithm

Algorithm	Search time/s	First attempt to avoid collision/s	Second attempt to avoid collision/s	Third attempt to avoid collision/s	Fourth attempt to avoid collision/s
D* algorithm	183.3	3.6	3.6	3.7	3.8
A* algorithm	51.2	-	-	-	-
Dijstra algorithm	715.7	-	-	-	-

5. Conclusion

Aiming at the problem of dynamic collision avoidance path planning for USV in complex sea conditions, A D* algorithm more suitable for USV in complex sea conditions is adopted in this paper,

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

which solves the problem that the traditional A* algorithm and Dijstra algorithm cannot handle both static and dynamic obstacles at the same time. In the case of unknown environment or dynamic obstacles, path planning using A* algorithm requires discarding the open and close tables completed in the initial planning and re-planning, resulting in an increase in planning time. In view of the shortcomings of the traditional A* algorithm and Dijstra algorithm in the dynamic path, A stronger real-time D* algorithm is adopted. In this paper, the traditional A* algorithm, Dijstra algorithm and D* algorithm are compared in the simulation experiment through MATLAB. The experimental results show that D* algorithm shortens the search length and search time, and has fast convergence speed and small calculation amount.

However, the research in this paper also leaves something to be improved. Whether USV in transit needs to avoid other vessels at the next node shall be determined in accordance with the international rules for preventing collisions at sea. Therefore, it will become a problem worth studying to consider the behavior decision in route planning of USV.

References

[1] Y. Singh, S. Sharma, R. Sutton, D. Hatton, and A. Khan, "Feasibility study of a constrained Dijkstra approach for optimal path planning of an unmanned surface vehicle in a dynamic maritime environment," in 18th IEEE International Conference on Autonomous Robot Systems and Competitions(ICARSC), 2018, pp. 117-122.

[2] Xinyu Liu, Liming Tan, Chunxi Yang, Chi Zhai. Adaptive dynamic path planning of ant colony and cluster in unknown environment [J]. Computer science and exploration,2019,13(05):846-857.

[3] R. Song, Y. Liu, and R. Bucknall, "Smoothed A* algorithm for practical unmanned surface vehicle path planning," Appl. Ocean Res., vol. 83, pp.9-20, 2019.

[4] Xue shuangfei, xie lei, wang shuwu, xia wentao, bao zhu. A~* collision avoidance and path finding algorithm for ships in offshore wind farm area [J]. China navigation, 2008,41(02):21-25.

[5] Liu kun. Research on USV path planning based on artificial potential field and ant colony algorithm [D]. Hainan university,2016.

[6] Xue fei. Research on path planning and obstacle avoidance based on USV [D]. Harbin engineering university,2017.

[7] Hanguen Kim,Donghoon Kim,Jae-Uk Shin, Hyong jin Kim,Hyun Myung. Angularrate constrained path planning algorithm for unmanned surface vehicles [J].Ocean Engineering2014, 84:37-44

[8] Zhang yukui. Research on path planning technology of surface unmanned craft [D]. Harbin engineering university,2008.

[9] Wasif Naeem, George W. Irwin, Aolei Yang. COLREGs-based collision avoidance strategies for unmanned surface vehicles [J]. Mechatronics.2012, 22:669-678

[10] Fujii Mihei. Survey of small ships in the area of occlusion. Report of the institute of ship technology,1996,48:147-154.

[11] GOODWIN E M. A statistical study of ship domains [J]. Journal of Navigation, 1975,28:329-341

[12] Niu liangliang, Chen guoquan, li lina. Research and application of automatic generation algorithm for multi-target ship encounter experiment [J]. Journal of jimei university (natural science edition),2016,21(04):269-274.

[13] Li lina, xiong zhennan, gao yansong, ren qisheng. Generation and optimization of intelligent decision making for single-ship collision avoidance [J]. China navigation,2002(02):51-54.

[14] Pingpeng Tang, RuboZhang, DeliLiu, LihuaHuang, GuanqunLiu, TingquanDeng. Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle.

[15] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. AUV Navigation and Localization: A Review

[16] Kang hetao, zhu daqi, Chen weijiong. Research review on collision avoidance path planning for ships [J]. Shian-hai engineering,2013,42(05):141-145.

[17] Ming Cheng Tsou. Multi-target collision avoidance route planning under an ECDIS framework.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202005_6(5).0013

[18] Yoshiaki Kuwata, Michael T. Wolf,Dimitri Zarzhitsky,Terrance L.Huntsberger. Safe Maritime Autonomous Navigation with COLREGS, Using Velocity Obstacles [J]. IEEE Journal of Oceanic Engineering.2014, 39(1):110-119.

[19] Shen haiqing, guo Chen, li tieshan, yu yalei. An intelligent collision avoidance navigation method for USV considering the rules of navigation experience [J]. Journal of Harbin engineering university.2018, 39(6):998-1005.

[20] Hossein Mousazadeh, Hamid Jafarbiglu, Hamid Abdolmaleki, el ta. Developing a navigation, guidance and obstacle avoidance algorithm for an Unmanned Surface Vehicle (USV) by algorithms fusion [J]. Ocean Engineering .2018, 159:56-65.

[21] Sun yaodong. Research on intelligent planning of a ship's voyage path [D]. Dalian maritime university,2018.