

Research on collision avoidance route planning based on A-star algorithm

Wenbin BAI* and Hitoi TAMARU**

*Tokyo University of Marine Science and Technology, 2-1-6 Etchujima, Koto-ku, Tokyo, 135-8533 Japan

BAIWENBIN0529@OUTLOOK.COM

Keywords: ship collision avoidance, Route planning, OZT, A-star algorithm, Maritime safety

ABSTRACT

Under the impetus of economic globalization, maritime traffic develops rapidly. With the increase of the number of ships, the number of marine traffic accidents caused by human error has surged. Therefore, for the purpose of reducing the occurrence of such accidents, the development of unmanned ships has become an important research topic and has received a lot of attention. Route planning is one of the essential key technologies in the research of ship collision avoidance. A ship collision avoidance route planning system combining OZT method, and A-star algorithm is proposed in this paper. The effectiveness of system was validated in simulation. The result indicated that the system has applicability in multi ships encounter situation.

1. Introduction

1.1. Background

In recent years, the frequent occurrence of marine traffic accidents has caused huge economic losses. These accidents are mainly caused by human factors (1). In order to reduce the economic losses caused by maritime traffic accidents, all countries are actively conducting research on autonomous ships.

Path planning algorithm is important Sub-research in the field of autonomous ships. The purpose of the path planning algorithm is to provide the Optimal collision avoidance navigation route for the unmanned boat through the route planning algorithm. The theoretical support was provided for realizing the interaction between autonomous ships and navigation situation.

As the busiest port in Japan, Tokyo Bay is connected to the Pacific Ocean by the Uraga Traffic Route. Therefore, the Tokyo Bay ferry which traverse at the end of Uraga channel are facing complex traffic situation. The autonomous navigation research can effectively reduce the pressure of both crew and Maritime Bureau.

1.2 Related works on collision avoidance route planning

At present, research on route collision avoidance route planning has been carried out among the worlds. There are two stages during the route planning: conflict detection and decision making. For the first stage, X.W. proposed a collision avoidance support system by utilizing DCPA (distance to closest point of approach) and TCPA (time to closest point of approach) as the risk evaluation method (2). In addition, the mathematical model of ship maneuvering was combined into the system. However, DCPA and TCPA is a numerical-based risk assessment that cannot be visualized on a map. For the path planning problem of surface unmanned boats in open waters, M.C. Tsou proposed a collision avoidance model based on the Ant Colony Algorithm, which integrates maritime laws and AIS data to plan safe and economical collision avoidance routes. However, due to the large amount of calculation of the algorithm, real-time path planning cannot be realized (3); Kim, Hanguen, et al. proposed a path planning algorithm based on ThetA-star algorithm, which can create paths in real-time, and considered maneuverability of USV. In addition, the algorithm has superior path tracking time performance compared to the traditional A-star algorithm. Nevertheless, the moving obstacles wasn't considered in the research and mainly applicable to global route planning (4). C Liu proposed an improved A-star algorithm, which realizes the collision avoidance route planning in the river channel and plans the design of the berthing route. However, the accuracy of the ship trajectory algorithm is missing the forecasts of encounter ships (5).

1.3. Outline

The current research is mainly reflected in the research of global path planning algorithm and local real-time collision avoidance algorithm. Each collision route planning algorithm has its advantages and disadvantages, such as simulate but without considering the traffic situation, the poor real-time performance due to large amount of calculation, lack of trajectory prediction for other ships, etc. In order to solve some shortcomings in the research, this paper will use the OZT method to predict the ship's trajectory, and then use the A-star algorithm to plan the ship's route. These two algorithms have relatively low computational complexity and high applicability and can solve the above problems well.

The innovation and main work of this paper are as follows:

(1) A grid map of Tokyo Bay was created. Besides, the OZT method and the AIS data of ferries was utilized to

predict the ship trajectory.

(2) The A-star algorithm and OZT method is combined in this paper, which contains ship safe distance setting and route node selection optimization. Even in a real complex encounter situation, the algorithm can return a proper route for collision avoidance.

(3) Two different real sea conditions were selected. To get the optimal route returned from the algorithm and compare with the real ship route to verify the effectiveness of the algorithm.

The remainder of this paper is organized as follows. The improved A-star algorithm and are first introduced in section II. Then Section III is devoted to simulating and verification. Some conclusions of this paper are in Section IV.

2. Method

2.1.1. Traditional A-star algorithm

The A-star algorithm was first proposed by Hart, P. E, et al (6). A-star is a path search algorithm, which is aims at finding an optimal path to the given goal node. As shown in Fig. 1(a), In via of extending a tree of node originated from a starting node, iterates the optimal path continuously to reach the goal. Comparing with the conventional Dijkstra algorithm and the Breadth-First Search algorithm, this algorithm has higher search efficiency. The traditional A-star algorithm is donated in Fig. 1 (a).

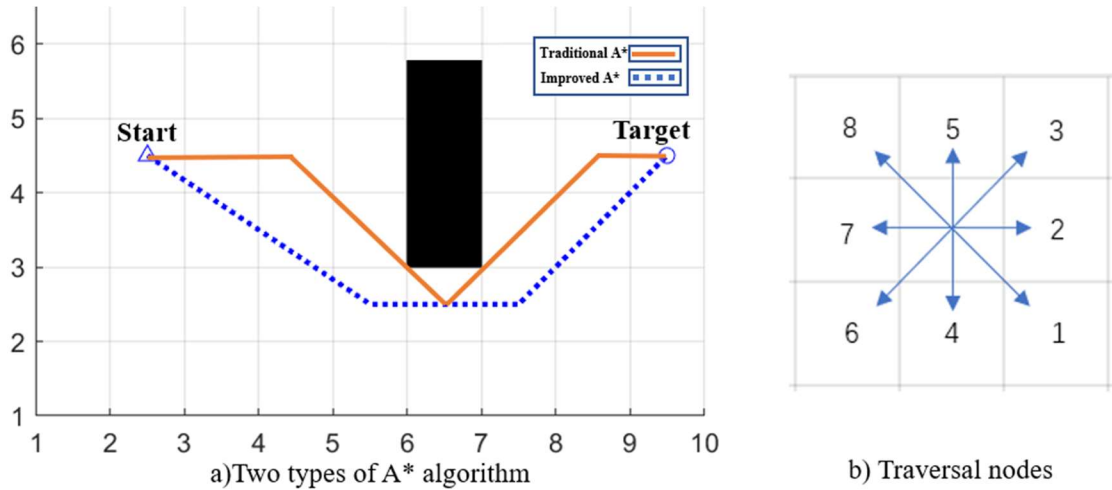


Fig. 1. A-star algorithm

In A-star algorithm, Eq. (1) is employed to ensure the optimal path from the start node to the target node. The priorities of the nodes in the OpenList are arranged according to the value of $f(n)$. The smaller the value of $f(n)$, the higher the searched priority.

$$f(n) = g(n) + h(n) \quad (1)$$

Where, n represents current node, $f(n)$ is the estimated value of the minimum cost path from the current node to the destination; $g(n)$ evaluates the cost from the starting node to the n -th node; $h(n)$ was called heuristic function, which evaluate the cost from n -th node to target node.

The Manhattan Distance written in Eq. (2) is used in this paper as the cost function.

$$h(n) = |x_t - x_n| + |y_t - y_n| \quad (2)$$

With regard to all the traversal nodes must satisfy the Eq. (3) for all path nodes n to ensure the optimality of the calculation results of the A-star algorithm.

$$f(n) \leq f(n)_{min} \quad (3)$$

Where, $F(n)_{min}$ is the minimum cost from n -th node to the target node.

As shown in Fig. 1 (b), in the traditional A-star algorithm, the current node is used as the parent node to traversal all the neighbor node that are not obstacles and are not in the CloseList among the 8 nodes around the current point. These nodes are then added to the OpenList and stored as candidate nodes. The traditional A-star algorithm is

Traditional A-star algorithm

Input: Start node, Target node, Obstacle nodes
Output: Path node list, or “No path”

```

1:  Choosing a starting point and adding to OpenList.
2:  While (The target node isn't in the OpenList) {
3:      Select the node with the minimum  $f(n)$  from the OpenList as the parent node, and Traverse all the
        reachable neighbor nodes, and for each of the 8 reachable nodes, executes the loop from 4-18.
4:      if (it is an obstacle or it is already in CloseList) {
5:          do nothing.
6:      }
7:      if (it's not in the OpenList) {
8:          Add current child node into the OpenList, and calculate the  $f(n)$ ,  $g(n)$ ,  $h(n)$  of the current point
9:      }
10:     if (it is already in the open list) {
        // Calculate the  $g(n)$  of the n-th parent node and all the  $g(n)$  of the adjacent node
11:         if ( $g(n) < g(n+1)$ ) {                                //in via of cost function, optimize route.
12:             Set the current child node as the parent node and update the  $g(n)$ ,  $h(n)$  of the adjacent node.
13:         }
14:         Else {
15:             Calculate and set the  $f(n)$ ,  $g(n)$ ,  $h(n)$  of the current node and the adjacent node.
                Set the current node as the new parent node
16:         }
17:         Switch it to the Closelist.
18:     }
19:     If (Fail to find the target square, and the open list is empty.){
20:         Return: “No path”
21:     }
22:     Else Save the path node list.

```

2.1.2. Improved A-star algorithm

Improved A-star algorithm		
Input: path node list, called AList[0,n-1] (Alogrithm1 node List) here		
Output: optimized path node list, called OptList[0,n-1] (Optimized node List) here		
1:	Loading path node list from Algorithm	
2:	Int n=0, count =0;	//count turning node
3:	While (n+2 < Alist (.). size){	//Traverse all the node of AList
4:	Vector 1=Alist(n+1)- Alist (n);	
5:	Vector 2= Alist(n+2)- Alist (n+1);	
6:	If (Vector 1= Vector 2) {	// Determine if they are on the same line
7:	OptList(count)= Alist (n+2);	
8:	n++;	
9:	}	
10:	Else {	//if there is no obstacle, optimize node number
11:	If (there is no obstacle between AList(n) and AList(n+1)) {	
12:	OptList(count)= Alist (n+2);	
13:	n++;	
14:	}	
15:	Else {	
16:	OptList(count)= Alist (n+2);	
17:	count ++;	

```

18:         n++;
19:     }
20: }
21: }
22: Return OptList;

```

Algorithm 2

2.2. OZT method

The OZT method (Obstacle Zone by Target) proposed by Imazu (8). The OZT method has the property of indicating a danger zone. In via of OZT method, the risk area of target ship can be predicted. Even in complex waters, a feasible collision avoidance route can be suggested. Therefore, the risk areas predicted by the OZT method will be imported into the grid map as obstacle areas in this paper. Eq. (4) is employed to give the courses of OZT method.

$$c_o = \begin{cases} Az \pm \alpha - \arcsin \left\{ \frac{V_T}{V_o} \sin (Az \pm \alpha - C_T) \right\} (V_T \leq V_o) \\ Az \pm \alpha - \pi + \arcsin \left\{ \frac{V_T}{V_o} \sin (Az \pm \alpha - C_T) \right\} (V_T > V_o) \end{cases} \quad (4)$$

The results are restricted by Eq. (5):

$$\left(\left| \frac{V_T}{V_o} \sin (Az \pm \alpha - C_T) \right| \leq 1 \text{ and } 0 \leq C_o \leq 2\pi \right) \quad (5)$$

3. Case Study

3.1. Creation of grid map

In this paper, the sea area near the Urawa Channel in Tokyo Bay is taken as the research target. The map is gridded based on ENC data. The specific method is to extract the latitude and longitude data of the land edge part of the map of Tokyo Bay. The map was partition into 100*100 grids. The grid map is shown as Fig. 2(b):

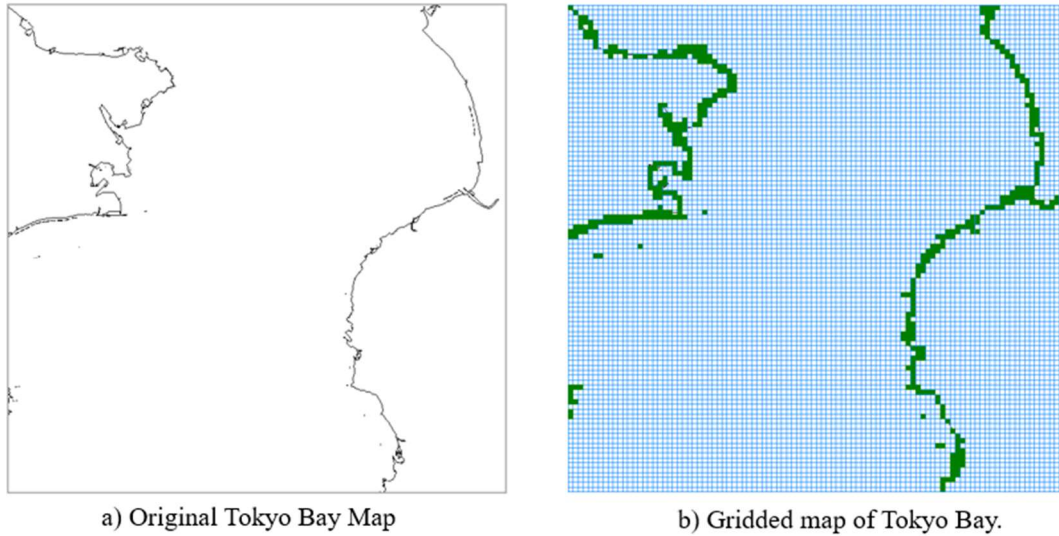


Fig. 2. Creation of grid map

In this figure, the coordinates of the upper left corner and the lower right corner are (139.68, 35.28) and (139.88, 35.12) respectively. The lateral length of each square represents approximately 200 meters.

3.2. Input of obstacle information

The OZT method is used to predict the risk area of target ships. To obtain the OZT, the most crucial argument is safe distance (SD). When target ships pass own ship within the range of minimum of SD is recognized as danger situation. However, if the safety distance is set extensive, it will contribute to unnecessary detours. Therefore, An appropriate safety distance balances safety and distance. Considering the complicated traffic in Tokyo Bay, safe distance is donated in Eq. (6):

$$SD = \begin{cases} \frac{L_{own} + L_{target}}{2} & SD > 0.07NM \\ 0.07NM & SD < 0.07NM \end{cases} \quad (6)$$

Where L_{own} denotes the length of the own ship, L_{target} denotes the length of the target ship. The risk area of the target ship will be imported into the grid graphics. The imported image is shown as Fig. 3.

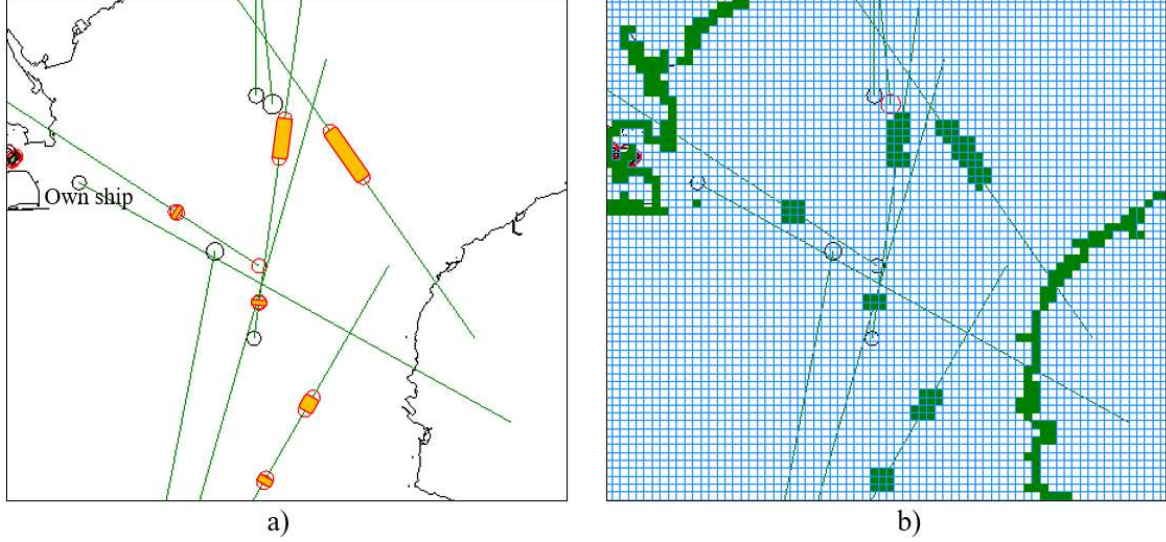


Fig. 3. Schematic of OZT

3.3. Case study

Simulation was conducted based on AIS data in Tokyo Bay. The study target is the Tokyo Bay ferry, which names KANAYA MARU and travels from KURIHAMA port to KANAYA port. The voyage on this route is about 6 nautical miles. The data studied in this subsection are from Toyo Communications Corporation. To closer inspection of verifying the effectiveness of algorithm, the simulation result and real AIS trajectory was compared.

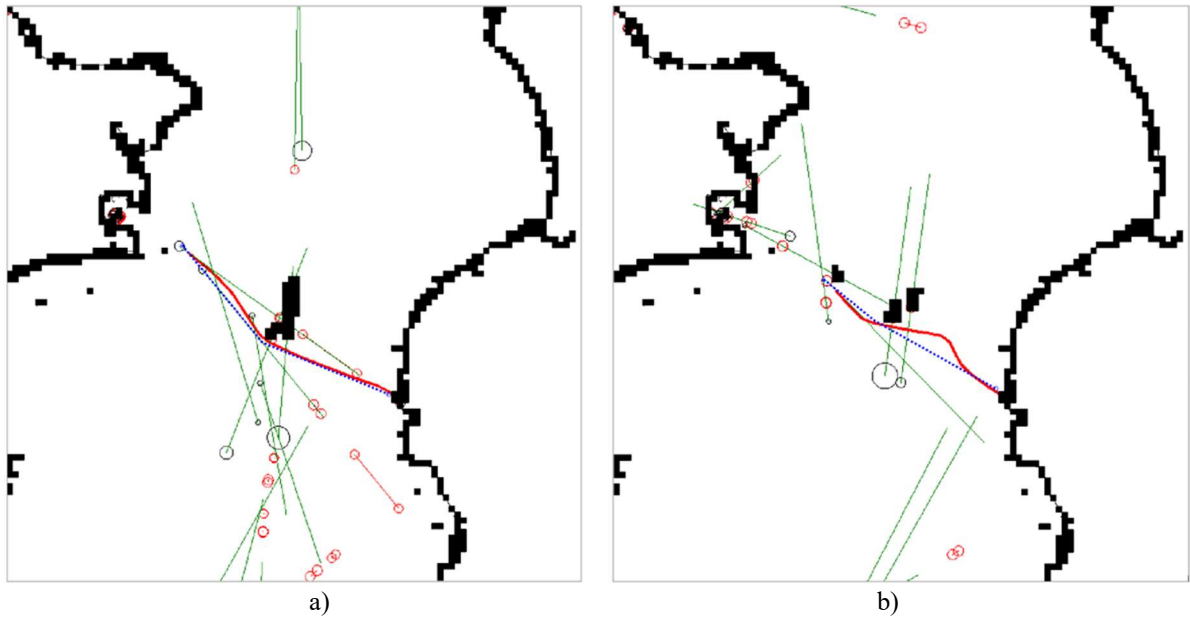


Fig. 4. Simulation result

In this image, AIS data in Tokyo Bay on January 1, 2014 is used. The black blocks represent obstacles and danger areas. The red line represents the ship's historical path, and the blue line represents the algorithmically derived path. Only the danger areas of OZT which will affect the route are shown.

4. Conclusion

As a means to solve problem of USV collision avoidance, the ability of target ships trajectory prediction and autonomous route planning is required. This paper combined the OZT method and A-star algorithm for autonomous route planning. Besides, the A-star algorithm was improved for optimizing the planning route, compare with the traditional A-star algorithm, the route is smoother and is suitable for the current marine situation. In the end of this paper, the simulation of the Tokyo Bay ferry voyage from the Kurihama to the Kanaya is carried out. The results show that the system proposed in this paper is effective.

However, two issues exist in this paper. First, the CORLEGS has not been considered. The improved A-star algorithm proposed in this paper is only suitable for finding the shortest path. Second, the OZT algorithm cannot accurately predict the path of a turning ship. In the future work, the study of improved A-star algorithm suitable for CORLEGS and high precision track prediction will be focused on.

References

- (1) Agency, EMS (2019): Annual overview of marine casualties and incidents.
- (2) Wang, Xin, Zhengjiang Liu, and Yao Cai. "The ship maneuverability based collision avoidance dynamic support system in close-quarters situation." *Ocean Engineering* 146 (2017): 486-497.
- (3) Kim H, Kim D, Shin J U, et al. Angular rate-constrained path planning algorithm for unmanned surface vehicles[J]. *Ocean Engineering*, 2014, 84: 37-44. Inaishi M, & Matsumura H (1991): Basic research on a collision avoidance system using neural networks. International Navigation Congress IAIN'91.
- (4) Kim H, Kim D, Shin J U, et al. Angular rate-constrained path planning algorithm for unmanned surface vehicles[J]. *Ocean Engineering*, 2014, 84: 37-44.
- (5) Liu C, Mao Q, Chu X, et al. An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning[J]. *Applied Sciences*, 2019, 9(6): 1057. Perera, L. P., J. P. Carvalho, and C. Guedes Soares (2009): Decision making system for the collision avoidance of marine vessel navigation based on COLREGs rules and regulations, Proceedings of 13th congress of international maritime association of Mediterranean, Istanbul, Turkey.
- (6) Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. *IEEE transactions on Systems Science and Cybernetics*, 1968, 4(2): 100-107.
- (7) <https://www.gamedev.net/articles/programming/artificial-intelligence/a-pathfinding-for-beginners-r2003/>
- (8) Imazu, H (2017): Evaluation method of collision risk by using true motion, *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation* 11.1.

Author's Biography

Wenbin Bai: he graduated from Dalian Maritime University and received his M.S. from Tokyo University of Marine Science and Technology in 2021, and currently studying for a Doctor's degree at Tokyo University of Marine Science and Technology.

Hitoi Tamaru: he is received Ph.D. degree (Mercantile Marine) from Tokyo University of Mercantile Marine (the university combined Tokyo University of Marine Science and Technology at 2003), and is presently Professor of Tokyo University of Marine Science and Technology. Institute of Navigation and The Japan Society of Naval Architects and Ocean Engineers member.