Journal of Marine Science and Application (2025) 24: 224-237
https://doi.org/10.1007/s11804-024-00434-1

RESEARCH ARTICLE

Fusion Algorithm Based on Improved A* and DWA for USV Path Planning

Changyi Li[1], Lei Yao[2] and Chao Mi[1,2]

Received: 01 January 2023 / Accepted: 25 November 2023

Abstract
The traditional A* algorithm exhibits a low efficiency in the path planning of unmanned surface vehicles (USVs). In addition, the path planned presents numerous redundant inflection waypoints, and the security is low, which is not conducive to the control of USV and also affects navigation safety. In this paper, these problems were addressed through the following improvements. First, the path search angle and security were comprehensively considered, and a security expansion strategy of nodes based on the 5×5 neighborhood was proposed. The A* algorithm search neighborhood was expanded from 3×3 to 5×5, and safe nodes were screened out for extension via the node security expansion strategy. This algorithm can also optimize path search angles while improving path security. Second, the distance from the current node to the target node was introduced into the heuristic function. The efficiency of the A* algorithm was improved, and the path was smoothed using the Floyd algorithm. For the dynamic adjustment of the weight to improve the efficiency of DWA, the distance from the USV to the target point was introduced into the evaluation function of the dynamic-window approach (DWA) algorithm. Finally, combined with the local target point selection strategy, the optimized DWA algorithm was performed for local path planning. The experimental results show the smooth and safe path planned by the fusion algorithm, which can successfully avoid dynamic obstacles and is effective and feasible in path planning for USVs.

Keywords Improved A* algorithm; Optimized DWA algorithm; Unmanned surface vehicles; Path planning; Fusion algorithm

Article Highlights

- Improved heuristic function of A* algorithm. The distance between the current node and the target node was introduced into the weight coefficient of the heuristic function in an exponentially weighted manner. And the search efficiency of the A* algorithm was improved.
- A node security expansion strategy based on the expanded 5×5 search neighborhood was proposed. This strategy optimized A* algorithm search angles and improved path security.
- Floyd algorithm was used to further eliminate redundant nodes and smooth the path. The smoothness of path was improved by removing unnecessary turning points.
- Improved evaluation function of the DWA algorithm. The distance from the end of the predicted trajectory to the target point was introduced into the evaluation function. And the planning efficiency of the DWA algorithm was improved.
- Optimized fusion strategy. A local target point selection strategy was proposed to avoid unnecessary turns. This strategy had the ability to determine if there was any obstacles between the USV and the next local target point. Local target points can be selected instead of only being updated sequentially.

✉ Chao Mi
chaomi@shmtu.edu.cn

[1] Container Supply Chain Technology Engineering Research Center, Ministry of Education, Shanghai Maritime University, Shanghai 201306, China

[2] Shanghai SMUVision Smart Technology Ltd, Shanghai 201306, China

1 Introduction

With the continuous development of the marine economy and maritime transportation, unmanned surface vehicles (USVs) are becoming more important. Autonomous surface vehicles rely on multiple sensors to interact with the environment to complete various tasks without human intervention. These vehicles have been widely used in hydrological surveying and mapping, river cleaning, and other fields. USVs also have broad application prospects (Long et al., 2023; Öztürk et al., 2022). Intelligent path planning, which can be divided into global and local path planning, is one of the key technologies that enable USVs to navigate safely and autonomously (Zhong et al., 2020; Wang et al., 2020; Chen et al., 2020).

Global path planning is based on a global static environment. The classic algorithms include the Dijkstra algorithm, A* algorithm, fast-expanding random-tree method, ant colony algorithm, and particle swarm optimization algorithm (Zhou et al., 2020). The A* algorithm is widely used in the global path planning task for robots in static environments. However, in the application of USV path planning, the path planned by the A* algorithm lies close to obstacles and has many redundant turning points. In addition, the path deviates from the USV kinematic constraints, which affects the safety of USV navigation (Song et al., 2019; Qin et al., 2023).

Most scholars use the A* algorithm to complete and improve global planning. Guan and Wang (2023) set up dangerous areas around waypoints to ensure the safety of the path planned by the A* algorithm and smooth it to ensure its suitability for USV navigation. Yin et al. (2023) optimized the node search angle, improved the heuristic function of the A* algorithm and used B-splines for path smoothing. Hu et al. (2023) introduced a risk function $D(n)$ into the cost function of the A* algorithm to ensure security through the Voronoi field algorithm (Hu et al., 2023). Many other algorithms are also used for global path planning. Mao et al. proposed a state-prediction fast exploration random-tree (spRRT) method (Mao et al., 2023). This process adds state information on USV movement to the RRT path search rule to predict whether a USV can reach the status point. The search efficiency of this algorithm is improved through the elliptic sampling domain (spRRT-Informed). However, as the RRT algorithm is based on the principle of random sampling, the path planned is not optimal. The algorithm is highly random and exhibits poor stability (Chiang and Tapia, 2018). Yang et al. (2023) proposed an improved particle swarm optimization algorithm, which enhances the algorithm's exploration capability and search accuracy through the improvement of the speed update method and inertia weight. They also adopted a particle initialization strategy to increase population diversity and address the local optimum to overcome local optima. However, the parameter selection of the intelligent optimization algorithm greatly influenced the optimal solution. The search efficiency was low, and the intelligent optimization algorithm easily fell into local optima (Bai et al., 2023). Bai et al. (2023) proposed the plant grow route (PGR) algorithm, which guides USVs in avoiding obstacles and enables them to reach the target point in accordance with the phototropism principle of plant growth. Their experiment proved the feasibility and effectiveness of the algorithm.

Local path planning involves path planning in an unknown environment based on the environmental information sensed by shipborne sensors in real time. The commonly used algorithms include the artificial potential field method, velocity obstacle, and DWA (Fox et al., 1997). In local path planning, the DWA algorithm is a widely used path planning method; it has short-range reactive dynamic obstacle-avoidance capabilities and plans a path that conforms to the kinematic constraints of robots. Guan and Wang (2023) proposed an improved DWA (IDWA) that filters a part of the predicted trajectories of DWA algorithms based on International Regulations for Preventing Collisions at Sea (COLREGs) and uses the Deep Q-network (DQN) method to train the weight coefficients of the IDWA evaluation function. The results show that IDWA can work effectively to avoid obstacles safely and allow USVs to reach their destination quickly (Guan and Wang, 2023). Zhang et al. (2021) proposed an improved dual-window dynamic-window method obstacle-avoidance algorithm based on fuzzy reasoning to solve the autonomous obstacle-avoidance problem of USVs in dense obstacle waters. Zhang et al. (2022) improved the DWA algorithm by replacing the direction angle difference with target distance and introducing the fuzzy logic control algorithm for the dynamic adjustment of the weight value of the trajectory evaluation function. Liang et al. (2021) used the DQN method to train the weight coefficient of the IDWA objective function and filter velocity space by combining it with the COLREGs. The experiment proved the effectiveness of the method. However, neural networks require the setting of a number of parameters and a large amount of diverse data. In addition, learning about these neural networks, which are difficult to converge, takes a long time to learn. Li et al. (2021) improved the DWA algorithm by considering the influence of external environmental factors and increasing the influence weight of external factors in the evaluation function. Ma et al. (2021) added pure pursuit guidance law and constant azimuth guidance law to the DWA algorithm. Guidance laws can be adaptively selected based on the state of the target in the obstacle environment. Simulation experiments proved the effectiveness of the method. However, only the DWA algorithm is prone to falling into local optima. Moreover, this algorithm cannot reach its target occasionally (Alireza et al., 2021; Tan et al., 2022).

The global and local path planning algorithms suffer from specific limitations (Niu et al., 2022). The general global path planning algorithm plans a path that contains many redundant nodes, which results in an uneven path. The path not only does not conform to the USV kinematic constraints but also cannot avoid dynamic obstacles (Lyridis, 2021). The local path planning algorithm easily falls into the local optimum and may fail in path planning, failing to meet the needs of USV path planning in complex dynamic water environments (Vagale et al., 2021). Therefore, this paper proposes a path planning algorithm that combines the improved A* algorithm and the optimized DWA to realize the path planning of USVs in complex dynamic water environments. Based on existing works, we offer the following contributions:

1) Improved heuristic function of A* algorithm. The distance from the current node to the target node was introduced into the weight coefficient of the heuristic function in an exponentially weighted manner to improve the search efficiency of the algorithm.

2) A node security expansion strategy based on the expanded 5×5 search neighborhood. This strategy reduces the redundant nodes and improves path security. Floyd algorithm was used to further eliminate redundant nodes and smooth the path.

3) Improved evaluation function of the DWA algorithm. The introduction of distance from the end of the predicted trajectory to the target point into the evaluation function

improved the planning efficiency of the DWA algorithm.

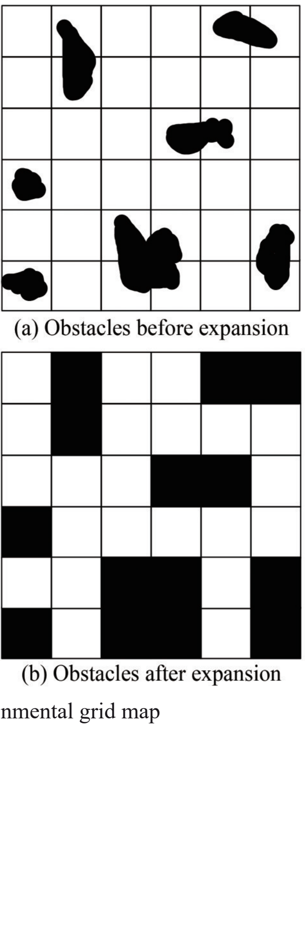
4) Optimized fusion strategy. A local target point selection strategy was proposed to avoid unnecessary turns.

The rest of this paper is organized as follows. Section 2 introduces grid method modeling and discusses the improvement of the traditional A* algorithm. Section 3 proposes an optimized DWA algorithm. Section 4 presents the optimization of the fusion strategy based on the shortcomings of the traditional fusion algorithm and proposes a local target point selection strategy. Section 5 introduces the algorithm for simulation experiments and analysis of experimental values. Section 6 introduces the algorithm for the USV experiment and analysis of experimental values. Section 7 concludes our work and discusses future research issues.

2 Improved A* algorithm

2.1 Environment modeling and obstacle expansion treatment

Path planning necessitates the establishment of a map to describe environmental information on USV navigation water areas. This process includes important information on the size and location of obstacles. In this paper, the grid method was used to simplify the navigation environment into a two-dimensional grid map. Black and white grids on the map represent unnavigable and navigable water areas, respectively. To ensure the safety of USV navigation, we expanded irregular obstacles and rasterized the map. The process is shown in Figure 1.


(a) Obstacles before expansion

(b) Obstacles after expansion

Figure 1 Environmental grid map

2.2 Improved evaluation function

The A* algorithm is widely used in the path planning of robots. This heuristic search algorithm can effectively solve the global optimal path in a static environment, and its cost function is as follows:

$$F(n) = G(n) + H(n) \tag{1}$$

Aiming at the low search efficiency of the traditional A* algorithm, the evaluation function of the improved A* algorithm was modified as follows:

$$F(n) = G(n) + (e^{H(n)/L} - 1) \times H(n) \tag{2}$$

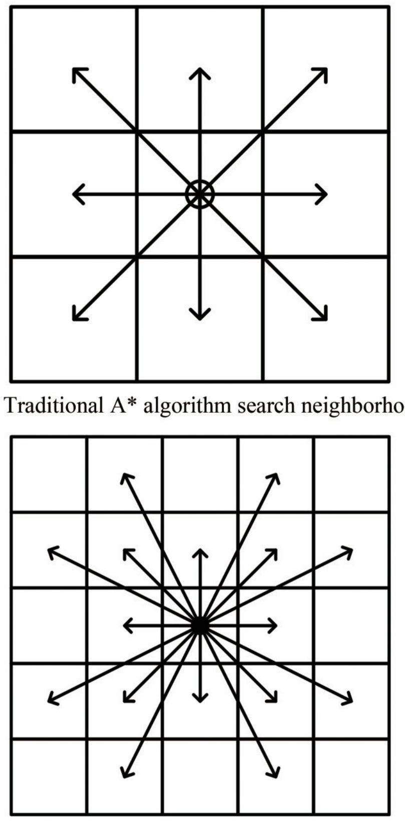
where L represents the distance from the starting point to the target point. From Eq. (2), given a length distance between the search node and the target point, the heuristic function exhibits a relatively large weight, which improves the efficiency of path planning. When the distance between the current search node and the target point is relatively short, the weight of the heuristic function decreases, which increases the search space and prevents the algorithm from falling into the local optimum. Changing the weight of the heuristic function adaptively improves the planning efficiency of the A* algorithm. In this paper, the distance between two points is Euclidean distance. Eq. (3) calculates the Euclidean distance, where (x_1, y_1) and (x_2, y_2) are the coordinates of two points.

$$H(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

2.3 Node security expansion strategy based on 5×5 neighborhood

The traditional A* algorithm has a 3×3 search neighborhood and a search angle limited to 45°. This condition results in many redundant nodes in the path, which is not conducive to the motion control of the USV. However, excessive expansion of the search neighborhood will increase the running memory of the algorithm and decrease its efficiency. In addition, the path planned by the traditional A* algorithm comes into contact with obstacles and passes through them directly, which seriously affects the safety of USV navigation. Therefore, this paper proposes a node security expansion strategy based on a 5×5 search neighborhood.

First, the neighborhood was appropriately expanded to a 5×5 search neighborhood (Figure 2(b)). The number of path nodes was reduced, and the search angle was optimized, which resulted in a smoother path. Second, nodes become unsafe with the expansion of the traditional A* algorithm. Moreover, the planned path made contact with obstacles,

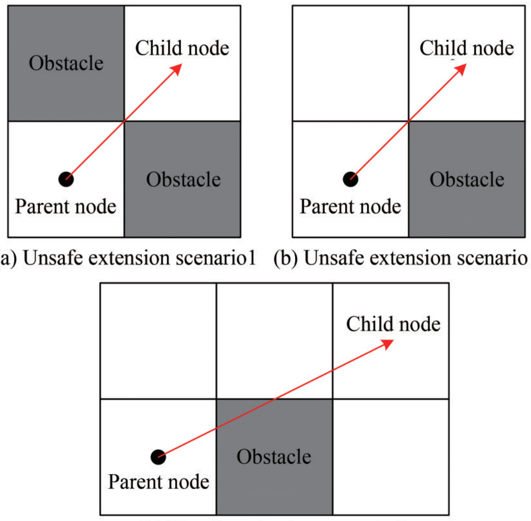

(a) Traditional A* algorithm search neighborhood

(b) Improved A* algorithm search neighborhood

Figure 2 Comparison of search neighborhood between traditional and improved A* algorithm

and the security was low. Figure 3 shows the unsafe node expansion in the traditional A* algorithm. This paper proposes a node security expansion strategy as follows. When the traditional A* algorithm explores the 3×3 search space, it initially determines whether the subnodes are obstacles and then identifies those that can be extended. The proposed node security expansion strategy directly filters out all insecure nodes in the 5×5 search space when the traditional A* algorithm explores the 3×3 neighborhood. As a result, the number of nodes that need to be explored in the 5×5 neighborhood is reduced, which improves the efficiency of the algorithm and avoids traversal of the subnodes of the 3×3 neighborhood again during the exploration of the subnodes of the 5×5 neighborhood. When the traditional A* algorithm explores node 2, it determines first whether node 2 is an obstacle. If neighbor node 2 is an obstacle, then nodes 1, 3, 10, and 11 are unsafe nodes and are not considered in the next expanded nodes (Figure 4). After the traditional A* algorithm completes the exploration of all subnodes in the 3×3 search space, all unsafe subnodes in the 5×5 search space are filtered, and the node cost value is calculated for comparison to determine the next parent node.

The improved A* algorithm combines the node security expansion strategy on the basis of neighborhood expansion.



(a) Unsafe extension scenario1 (b) Unsafe extension scenario 2

(c) Unsafe extension scenario 3

Figure 3 Three scenarios of unsafe extension

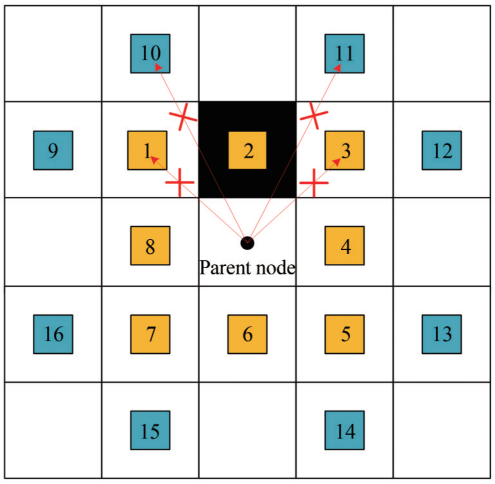


Figure 4 Node security extension (red crosses indicate that the node cannot be extended)

It prevents the planned path from coming into contact with the edge of obstacles or passing through obstacles to ensure the safety of USV navigation. The enhanced A* algorithm not only optimizes the search angle but also improves the smoothness and safety of the path.

2.4 Path smoothing optimization

The high number of redundant inflection points in the path planned by the A* algorithm increases the difficulty of USV path tracking control. Therefore, a smooth path must be obtained. The Floyd algorithm was used to further smooth the path. Figure 5 shows the concept of the Floyd path smoothing algorithm. *A*, *B*, *C*, and *D* refer to path nodes, and the original path is *A*→*B*→*C*→*D*. *AB*, *AC*, and *AD* connect in turn. Obstacles are lacking in *AC* but are present in *AD*. Thus, point *B* is a redundant node. Point *B* was removed from the original path point. As a result, the smoothed path from A to point D is $A \to C \to D$. The specific implementation steps are as follows:

Step 1: Record the set of all nodes in the path planned by the A* algorithm as $P=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$. (x_1, y_1) is the starting point; (x_n, y_n) is the target point.

Step 2: Connect (x_1, y_1) and (x_2, y_2), (x_2, y_2), and (x_3, y_3) and assess the collinearity of the two lines. If the two lines are collinear, remove node (x_2, y_2) in the P set. Otherwise, determine whether the connecting lines between (x_2, y_2) and (x_3, y_3) and between (x_3, y_3) and (x_4, y_4) are collinear. After traversing all path nodes, the remaining nodes in the P set are no longer collinear, and the remaining nodes are renumbered in sequence.

Step 3: Take (x_1, y_1) as the starting point, connect (x_2, y_2), (x_3, y_3), $\cdots$ in turn, and determine whether each connecting line passes through obstacles. If the connecting line between (x_1, y_1) and (x_i, y_i) is in contact with obstacles, then (x_2, y_2), $\cdots$, (x_{i-2}, y_{i-2}) are redundant nodes. The redundant nodes are removed from the P set, and (x_{i-1}, y_{i-1}) is retained in the P set. In addition, set (x_i, y_i) as the starting point, and connect (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2}), ... Perform continuous iteration until no more obstacles exist between the starting and end points. At this stage, the remaining nodes of the P set are key nodes.

Step 4: Connect the remaining nodes in the P set with a straight line and obtain the smoothed path.

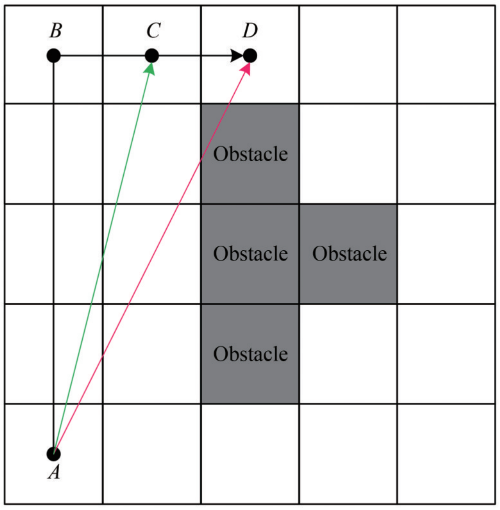


Figure 5 Schematic of floyd principle

After the path was smoothed (Figure 6), the path length was reduced from 38 to 30.332 5 (20.18%) and the number of turning nodes from 24 to 5. However, the smoothness of the path remained insufficient. In addition, the improved A* algorithm neither considers the kinematic constraints of the USV nor avoids dynamic obstacles, and thus, it cannot meet the requirements of USV path planning.

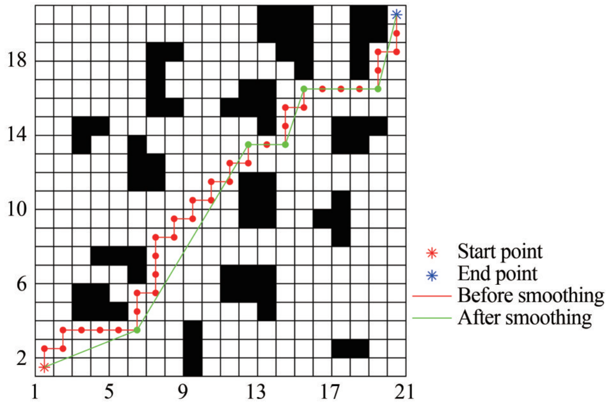


Figure 6 Comparison of paths before and after smoothing

3 Improved DWA algorithm

3.1 USV kinematics model and trajectory deduction

In this paper, in consideration of the underactuated and high inertial motion characteristics of USVs and given that USVs can only move forward or rotate, we adopted a nonholonomic motion model to describe the motion characteristics and control limitations of USVs (Zhang et al., 2021; Xu et al., 2024). The path planned by the algorithm is consistent with the actual navigation path and conducive to USV motion control. Figure 7 shows the established nonholonomic model.

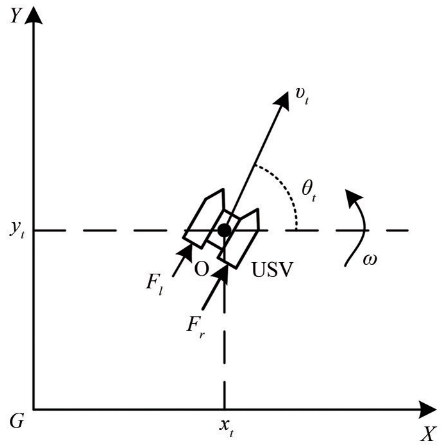


Figure 7 USV kinematics model

G-XY in the figure refers to the geodetic coordinate. x_t, y_t, and θ_t represent the X-axis coordinate, Y-axis coordinate, and heading angle of the USV at time t, respectively. v_t and ω_t represent the linear velocity and rotational angular velocity of the USV at time t, respectively. F_l and F_r, respectively, denote the thrust of the left and right sides of the hull. As the interval time Δt is extremely short, the trajectory of the USV from time t to time $t+\Delta t$ is approximated as a straight line. The USV trajectory derivation formula can be obtained as follows:

$$\begin{cases} x_{t+\Delta t} = x_t + \upsilon_t \Delta t \cos(\theta_t) \\ y_{t+\Delta t} = y_t + \upsilon_t \Delta t \sin(\theta_t) \\ \theta_{t+\Delta t} = \theta_t + \omega_t \Delta t \end{cases} \tag{4}$$

3.2 DWA algorithm

DWA is a classic local path planning algorithm, and it is mainly divided into two processes:

1) First, calculate the dynamic velocity window based on the local environmental information on the navigation water area and kinematic constraints of the USV. In the dynamic velocity window, use a certain linear velocity and angular velocity resolution to sample multiple sets of linear velocity υ and angular velocity ω based on and then simulate the trajectory of the USV with multiple sets of the sampled velocity in period T.

2) Use the trajectory evaluation function to evaluate the simulated multiple trajectories. Next, select the speed (υ_{best}, ω_{best}) corresponding to the optimal trajectory, with the highest score considered the best sailing speed for the USV.

Velocity sampling and trajectory prediction: Velocity can be limited within a certain range based on a USV's kinematic constraints and local environmental constraints. Velocity sampling was performed within the velocity window range, and trajectory prediction was performed based on the sampling speed (Figure 8). In consideration of a USV's kinematic constraints and navigation safety, velocity constraints can be abstracted as Eq. 5:

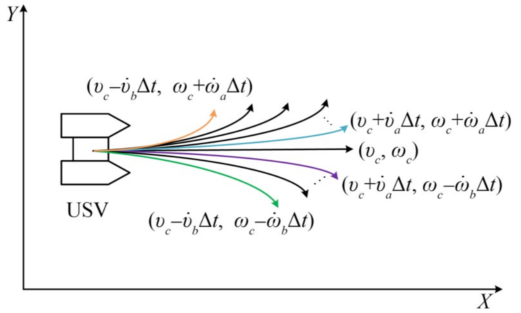

Y
X
USV
$(\upsilon_c - \dot{\upsilon}_b \Delta t,\ \omega_c + \dot{\omega}_a \Delta t)$
$(\upsilon_c + \dot{\upsilon}_a \Delta t,\ \omega_c + \dot{\omega}_a \Delta t)$
$(\upsilon_c,\ \omega_c)$
$(\upsilon_c + \dot{\upsilon}_a \Delta t,\ \omega_c - \dot{\omega}_b \Delta t)$
$(\upsilon_c - \dot{\upsilon}_b \Delta t,\ \omega_c - \dot{\omega}_b \Delta t)$

Figure 8 Schematic of dynamic-window trajectory prediction

$$\begin{cases} \upsilon_m = \left\{ \upsilon \in \left[\upsilon_{\min}, \upsilon_{\max} \right], \omega \in \left[\upsilon_{\min}, \upsilon_{\max} \right] \right\} \\ \upsilon_d = \left\{ (\upsilon, \omega) \middle| \begin{matrix} \upsilon \in \left[\upsilon_c - \dot{\upsilon}_b \Delta t, \upsilon_c + \dot{\upsilon}_a \Delta t \right] \\ \omega \in \left[\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t \right] \end{matrix} \right\} \\ \upsilon_a = \left\{ (\upsilon, \omega) \middle| \upsilon \leqslant \sqrt{2 \text{dist}(\upsilon, \omega) \dot{\upsilon}_b} \wedge \omega \leqslant \sqrt{2 \text{dist}(\upsilon, \omega) \dot{\omega}_b} \right\} \end{cases} \tag{5}$$

where υ_m refers to the maximum and minimum velocity limit window of USV navigation. Given the limitations of acceleration and deceleration performance of USV, υ_d indicates the velocity window that USV can actually achieve during a simulation period T. υ_a corresponds to the allowable velocity window, that is, the set of speeds at which the USV can safely stop sailing upon the detection of an obstacle. $\upsilon_{\max}$, $\upsilon_{\min}$, $\omega_{\max}$, and $\omega_{\min}$ represent the maximum and minimum values of USV linear velocity and angular velocity, respectively. υ_c, ω_c, υ_b, ω_b, υ_a, and ω_a stand for the current linear velocity, current angular velocity, maximum deceleration of linear velocity, maximum deceleration of angular velocity, maximum acceleration of linear velocity, and maximum acceleration of angular velocity, respectively. Δt is the interval.

After calculation of the velocity limits υ_m, υ_d, and υ_a, the intersection of the three velocity ranges, which is the desirable velocity window range, was denoted as υ_r and computed as follows:

$$\upsilon_r = \upsilon_m \cap \upsilon_d \cap \upsilon_a \tag{6}$$

Evaluation of simulated trajectory: The optimal path was selected based on the evaluation function in simulated trajectories. The evaluation function is designed to ensure that the USV selects the optimal path, sails to the target point without collision, and allows the trajectory to conform to the USV's kinematic characteristics. The evaluation function is determined as follows:

$$G(\upsilon, \omega) = \sigma\left(\alpha \cdot \text{yaw}(\upsilon, \omega) + \beta \cdot \text{dist}(\upsilon, \omega) + \gamma \cdot \text{velocity}(\upsilon, \omega)\right) \tag{7}$$

where yaw(υ, ω) refers to the azimuth evaluation function, which represents the deviation angle between the current pose of the USV and the target point, velocity (υ, ω) indicates the speed evaluation function, dist (υ, ω) represents the closest distance to the obstacle. σ denotes a smooth function used for normalized processing; α, β, and γ are the azimuth and closest distance to the obstacle and the weighting coefficient of velocity, respectively.

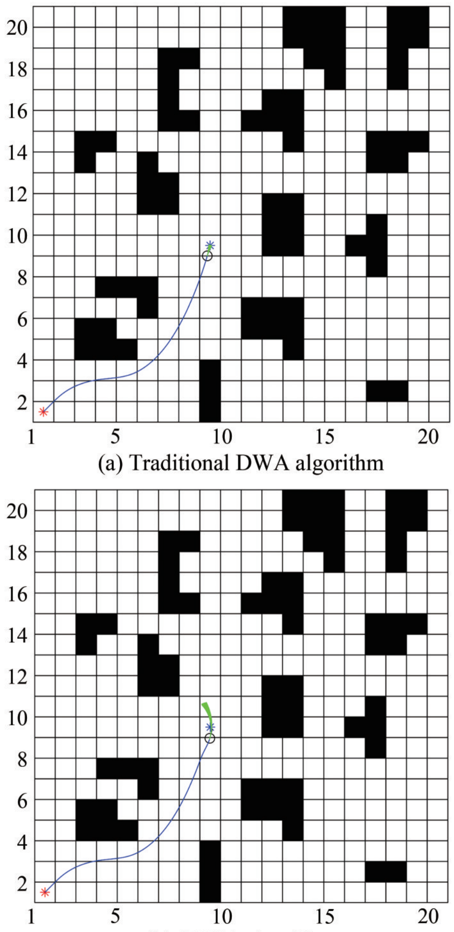
3.3 Optimization of evaluation function

When the path planned by the traditional DWA algorithm is near the target point, the heading angle accounts for too much in the evaluation function, and the velocity decreases, which affects the efficiency of the algorithm. To improve the navigation efficiency of the DWA algorithm, improved the evaluation function by considering adaptive weights rather than fixed ones. We also introduced the distance length between the end of the simulated trajectory and the local target point. When the path was near the local target point, the scoring weight of the heading angle was reduced appropriately, and the velocity weight was increased. This condition improved the efficiency of path planning. The improved evaluation function is calculated as follows:

$$G(\upsilon,\omega)=\sigma\left(\alpha\cdot \mathrm{yaw}(\upsilon,\omega)\cdot\left(\frac{\mathrm{length}(\upsilon,\omega)}{2R}\right)^{3}+\beta\cdot\mathrm{dist}(\upsilon,\omega)+\gamma\cdot\mathrm{velocity}(\upsilon,\omega)\cdot\left(2-\left(\frac{\mathrm{length}(\upsilon,\omega)}{2R}\right)^{3}\right)\right)\tag{8}$$

where length (υ, ω) represents the distance between the end of the simulated trajectory and the target point, and R denotes the radius of the obstacle. In calculations, the maximum values of dist (υ, ω) and length (υ, ω) were set. If the maximum value of dist (υ, ω) is not limited, the simulated trajectory without obstacles attains an extremely high score and ignores the heading angle and velocity. If the maximum value of length (υ, ω) is not set, when the USV is distant from the target point, the speed is extremely low. Therefore, dist $(\upsilon, \omega)\in(0, 2R)$ and length $(\upsilon, \omega)\in(0, 2R)$.

Figures 9(a) and 9(b) show the paths of the traditional DWA and IDWA, respectively. The comparison of experimental data in Table 1 reveals that the path lengths were the same. However, the planning time was reduced by approximately 17.19%, which proves that the improved evaluation function effectively reduced the planning time of the traditional DWA algorithm and thus increased its efficiency.



(a) Traditional DWA algorithm

(b) IDWA algorithm

Figure 9 Path comparison of traditional DWA algorithm and IDWA algorithm

Table 1 Comparison of algorithms

Algorithm	Cost time (s)	Path length (m)
Traditional DWA	15.499 047	11.727
Improved DWA	12.833 989	11.726

4 Fusion algorithm

First, the reference point of the traditional DWA algorithm is the only final target point. Therefore, the algorithm is prone to fall into a local optimum, which results in an unreachable target point and failure of the algorithm to plan the path. Second, the path planned by the improved A* algorithm is still not smooth enough near the key inflection point, and the kinematic constraints of the USV are disregarded, all of which result in a large difference between the planned path and the actual navigation path of the USV. Such a condition is not conducive to the path tracking control of the USV, which affects the safety of navigation. Aiming at the disadvantages of the above two algorithms, we proposed a local target point selection strategy to fuse the improved A* algorithm and DWA algorithm. Figure 10 displays the flow chart. First, the key inflection points obtained using the improved A* algorithm were extracted as local target points of the IDWA algorithm. Then, the IDWA algorithm was used for path planning. However, unlike traditional fusion algorithms, the fusion algorithm proposed in this paper simultaneously calculates and determines the presence of an obstacle between the USV and the next local target point. If no obstacle is present, the current local target point is switched to the next one directly. The path planned by the fusion algorithm can not only ensure the global optimum but also avoid dynamic obstacles. In addition, the path avoids unnecessary bending and attains a desirable smoothness, which is beneficial to the follow-up path tracking control.

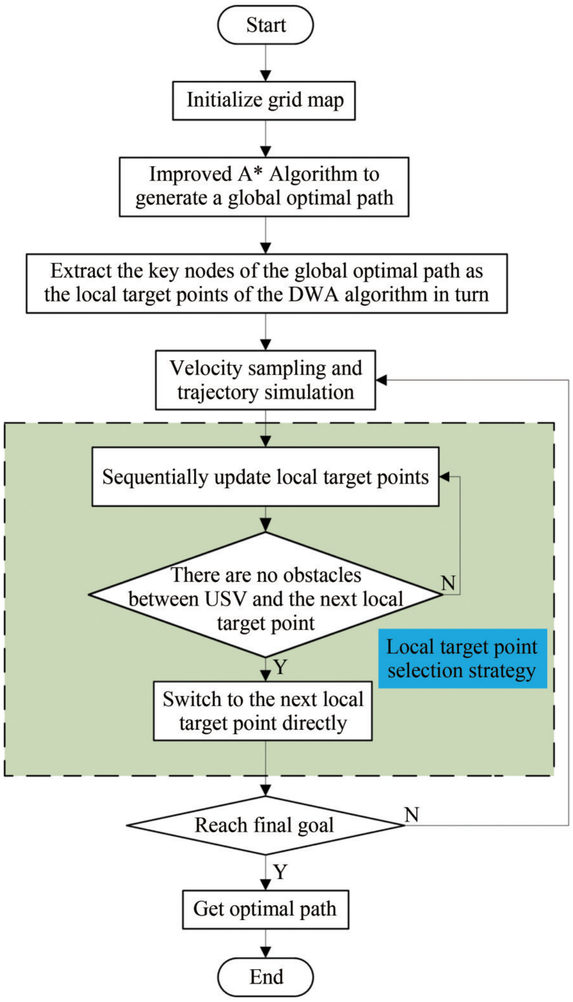
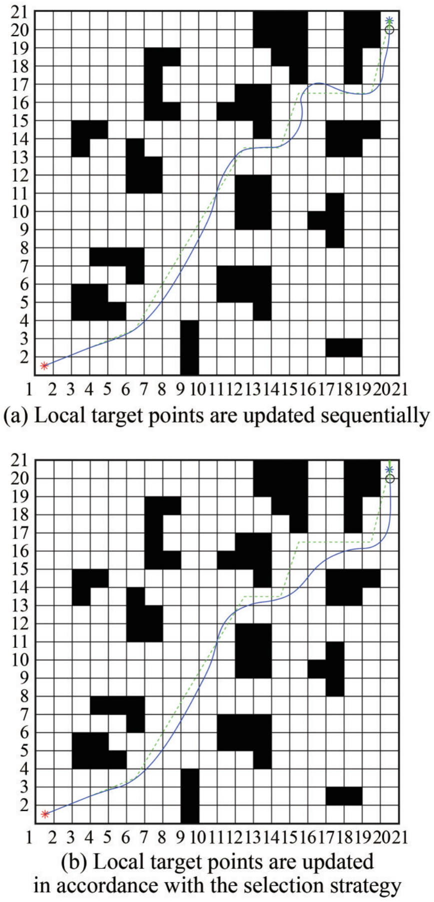

Start
Initialize grid map
Improved A* Algorithm to generate a global optimal path
Extract the key nodes of the global optimal path as the local target points of the DWA algorithm in turn
Velocity sampling and trajectory simulation
Sequentially update local target points
There are no obstacles between USV and the next local target point
N
Y
Local target point selection strategy
Switch to the next local target point directly
Reach final goal
N
Y
Get optimal path
End

Figure 10 Flow chart of the fusion algorithm

Figure 11(a) shows the path of local target points updated in sequence, and Figure 11(b) displays the path of local target points updated in accordance with the selection strategy. The comparison revealed that the path that combines the local target point selection strategy is better than that updated in sequence.

5 Simulation experiment and analysis

Experiments were carried out to verify the effectiveness and feasibility of the optimization algorithm in this paper. The running environment comprised the following: MATLAB R2021a, Windows 11 64-bit, AMD Ryzen 75 800 H processor, and 16 GB memory. A grid map was used to simulate the USV navigation environment information. The black and white parts in the map represent the obstacle and non-obstacle areas, respectively. The red * denotes the starting point, and its coordinates were (1.5, 1.5). The blue * corresponds to the final goal, and its coordinates were (20.5, 20.5). The weights of evaluation functions α, β, and γ were 0.05, 0.2, and 0.1, respectively. The velocity resolution was 0.01 m/s, the angular velocity resolution was 1°, and the simulation time period T was 3 s. The kinematic parameters of the USV were set, as shown in Table 2.



(a) Local target points are updated sequentially

(b) Local target points are updated in accordance with the selection strategy

Figure 11 Path comparison of traditional and improved fusion strategies

Table 2 Kinematic parameters of USV

Parameters	Value
Maximum linear velocity (m/s)	1
Maximum angular velocity ((°)/s)	20
Maximum linear acceleration (m/s^2)	0.2
Maximum angular acceleration ((°)/s^2)	50

To verify the efficiency of the improved A* algorithm, we performed simulation experiments 10 times. Table 3 contains the average values of the 10 experimental data. Figure 12 shows the comparison of the path planning of the two algorithms. The experimental results in Table 3 reveal that the improved A* algorithm shortened the path length by 20.18%, reduced the path planning time by 28.1%, and decreased the number of turning nodes from 8 to 5.

Table 3 Data comparison between traditional and improved A* algorithms

Algorithm	Cost time (s)	Path length (m)	Number of turning nodes
Traditional A*	0.118 029	38	8
Improved A*	0.079 533	30.332 5	5

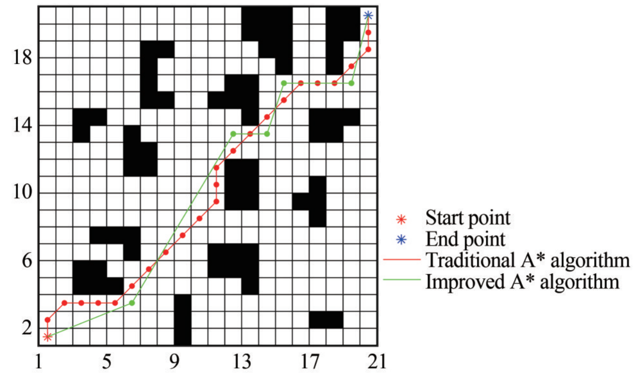
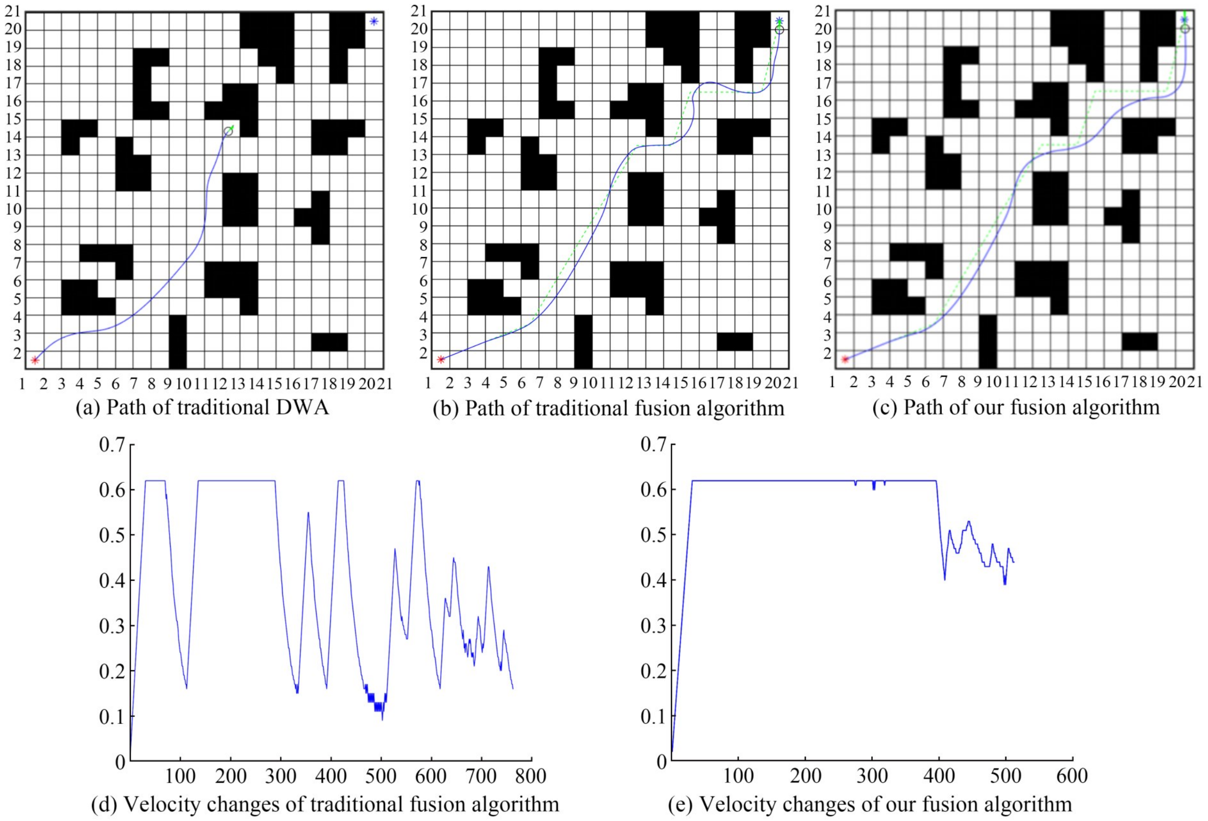


Figure 12 Path comparison between traditional and improved A* algorithms

Path planning algorithms were generated to verify the effectiveness of the fusion algorithm in this paper. Figure 13(a) shows the path of the traditional DWA algorithm, and Figure 13(b) displays the path of the fusion of improved A* and traditional DWA algorithms. Figure 13(c) illustrates the path of the fusion of improved A* and IDWA algorithms. The traditional DWA algorithm suffers from having an unreachable target point and an unplanned path to the final target point. Although the fusion algorithm of improved A* and traditional DWA algorithms plans the path successfully, the path contains unnecessary twists and turns. The fusion algorithm proposed in this paper avoids such twists and turns and offers a better path. Compared with the fusion algorithm of improved A* and traditional DWA algorithms, our algorithm reduces time by 9.9% and the path length by 7.54%. The running results are filled in Table 4.

To verify the obstacle-avoidance performance of the fusion algorithm introduced in this paper, we added static (red) and dynamic obstacles (pink) to the environment (Figure 14(a)). The USV avoided the obstacles, as shown in Figure 14(b) – 14(c). Figures 15(a) – 15(c) display the changes in the USV linear velocity, angular velocity, and heading angle during the process, respectively. After the addition of dynamic and static obstacles to the preplanned



(a) Path of traditional DWA (b) Path of traditional fusion algorithm (c) Path of our fusion algorithm

(d) Velocity changes of traditional fusion algorithm (e) Velocity changes of our fusion algorithm

Figure 13 Path and velocity comparison of traditional and our fusion algorithms

Table 4 Comparison of the performances of different algorithms

Algorithm	Cost time (s)	Path length (m)
Traditional DWA	∞	–
Traditional fusion algorithm	80.12	30.384
Our fusion algorithm	72.23	28.093

path of the improved A* algorithm, the fusion algorithm still successfully planned the path from the starting point to the final target point, and the path avoided all obstacles successfully.

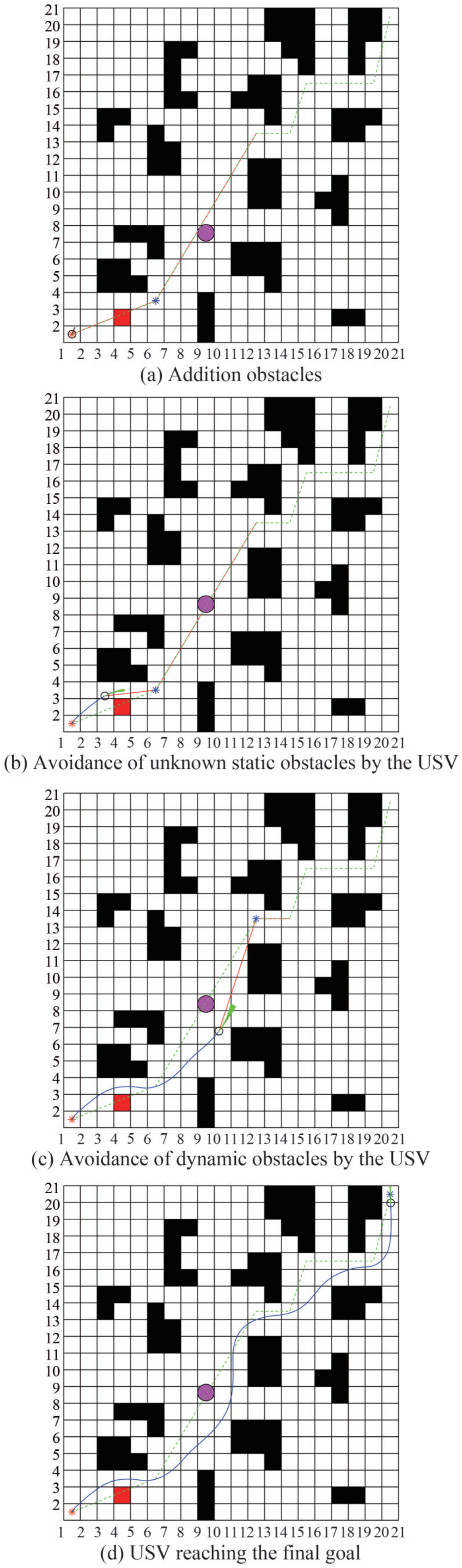
6 Real ship experiment

This study used a self-developed double-body differential propulsion unmanned boat with dimensions of 500 cm× 400 cm×200 cm (length×width×height) as the experimental platform (Figure 16). The unmanned boat uses the GNSS RTK module (TAU1308) and IMU (HWT901B-485) for location and posture measurement, respectively. The lidar (WLR-716) and ultrasonic sensors detect obstacles. The boat covers the longest detection distance of 25 m and has a detection angle of 270°. The camera functions in collecting images of the water area in front of the boat and monitors sailing conditions of the unmanned boat. The control box calculates and processes the sensor data. Then, the boat returns the processed data to the shore-based system through its 5G routing module.

To verify the feasibility and effectiveness of the algorithm introduced in this paper, we loaded the algorithm into an unmanned boat. An obstacle-avoidance path planning experiment was performed in the lake area of Luxun's Park in Shanghai under sunny and light wind weather conditions.

First, to verify whether the unmanned boat can sail in the preset path, we conducted an obstacle-free navigation experiment. Figure 17(a) shows the schematic of the experiment. The experimental path contained four key inflection points. The navigation direction was Point 1→2→3→4→1. Table 5 displays the latitude and longitude coordinates of the waypoints based on WSG-84. The unmanned boat can sail based on the set route from the actual navigation track in Figure 17(b).

on obstacle avoidance, we conducted the obstacle-avoidance experiment. Figure 18(a) shows the schematic of the experimental scene of obstacle avoidance. Figure 18(b) displays the static and dynamic obstacles. We used a small stationary cruise boat to add an unknown static obstacle and a dynamic obstacle on the preplanned path. Moreover, we used another USV as a dynamic obstacle named USV-obstacle (Figure 18(b)). The distance between two USVs was calculated based on the location that they uploaded. When the distance is less than 25 m, the lidar can detect USV-obstacle. At this point, we ran the control procedures



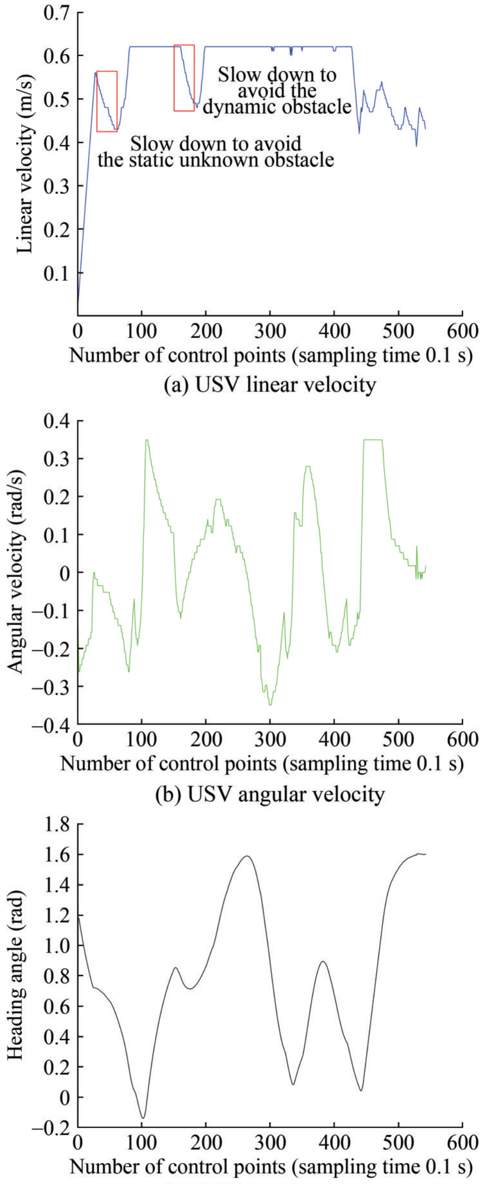
(a) Addition obstacles

(b) Avoidance of unknown static obstacles by the USV

(c) Avoidance of dynamic obstacles by the USV

(d) USV reaching the final goal

Figure 14 Fusion algorithm for avoiding obstacles



(a) USV linear velocity

(b) USV angular velocity

(c) USV heading angle

Figure 15 Changes of USV motion parameters

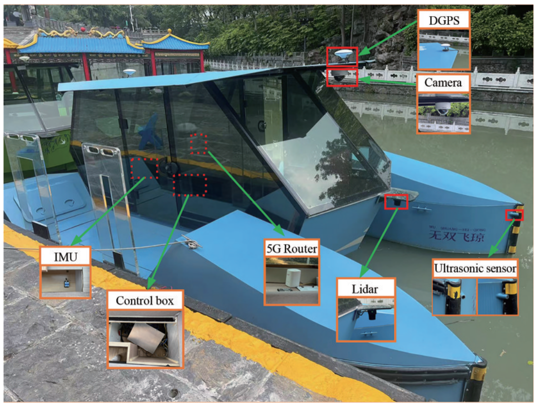
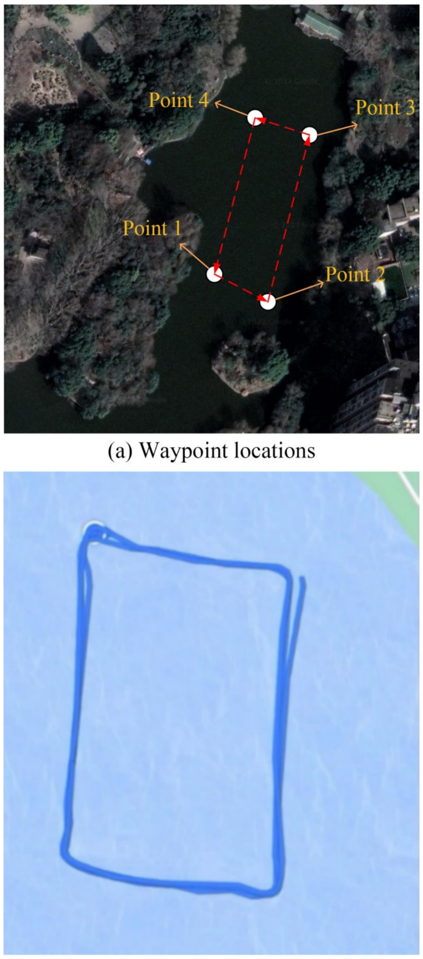


Figure 16 Experimental platform for unmanned boats (the dotted red box indicates that the installation location is inside the boat)



(a) Waypoint locations

(b) Navigation trajectory

Figure 17 Obstacle-free navigation experiment to verify the performance of the fusion algorithm

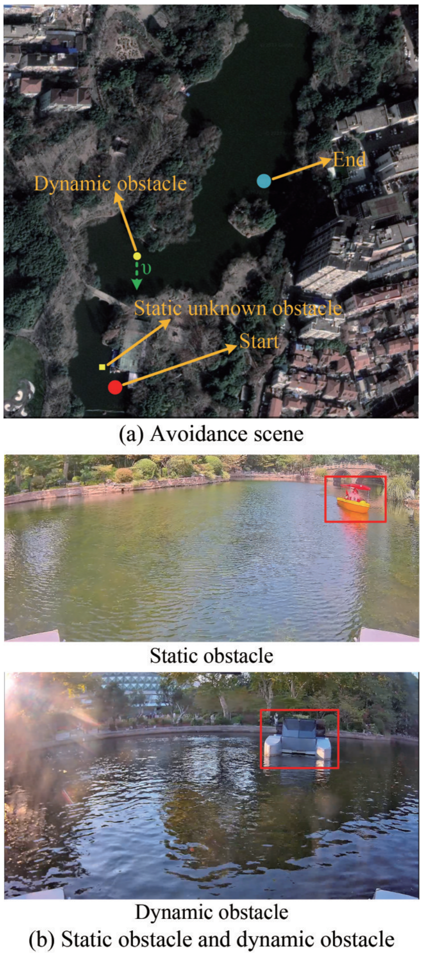
Table 5 Waypoint locations

Waypoint number	No. 1	No. 2	No. 3	No. 4
Longitude (°E)	121.480 000	121.480 300	121.480 480	121.480 210
Latitude (°N)	31.274 800	31.274 690	31.275 620	31.275 620

that would enable the USV obstacle to follow the set path at the speed of 0.5 m/s. The initial course of USV-obstacle was 164°.

Figure 20 shows the paths of the traditional and improved A* algorithms. The data in Table 8 reveal that compared with the traditional A* algorithm, the improved A* algorithm reduced the time by approximately 13.1% and the path length by 20%, and the number of turning nodes decreased from 15 to 4. These findings prove that the improved A* algorithm, which produced a shorter and smoother path, is more efficient than the traditional one.

The latitude and longitude were recorded during the navigation and processed by MATLAB. The USV navigation trajectory was drawn on a grid map (Figure 21). The



(a) Avoidance scene

Static obstacle

Dynamic obstacle

(b) Static obstacle and dynamic obstacle

Figure 18 Schematic of obstacle avoidance scene and obstacles

Table 6 Start and end point locations

Key point	Start	End
Longitude (°E)	121.479 113	121.480 300
Latitude (°N)	31.273 212	31.274 690

Table 7 Setting of obstacles in the scene

Obstacle	Static obstacle	Dynamic obstacle
Start location (Lon, Lat)	(121.478 959, 31.273 465)	(121.479 346, 31.274 285)
End location (Lon, Lat)	–	(121.479 259, 31.274 016)
Velocity (m/s)	0	0.5
Initial course	–	164°

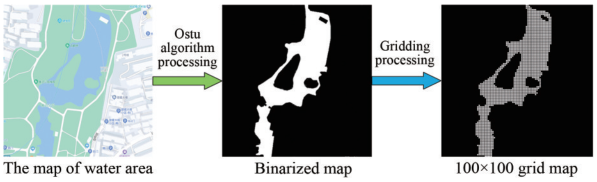


Figure 19 Map gridding process

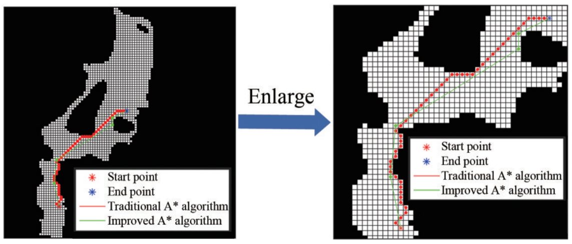
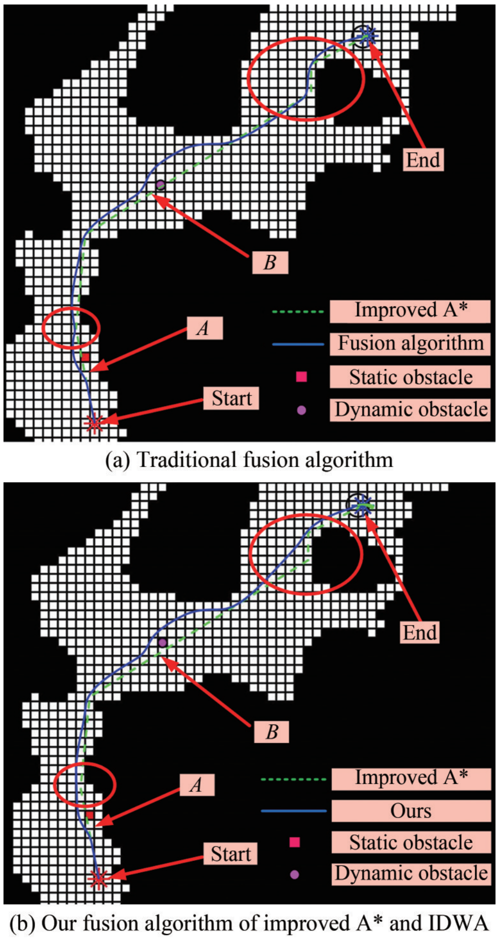


Figure 20 Comparison of paths obtained using traditional and improved A* algorithms

Table 8 Comparison of the performances of traditional and improved A* algorithms

Algorithm	Cost time (s)	Path length (m)	Number of turning nodes
Traditional A*	12.160 083	72.828 4	15
Improved A*	10.566 424	58.258 1	4



(a) Traditional fusion algorithm

(b) Our fusion algorithm of improved A* and IDWA

Figure 21 Path comparison between traditional and our fusion algorithm

unmanned boat first followed the path preplanned by the improved A*. When it encountered the unknown static obstacle at point *A*, an evident left-turn avoidance action was observed. After complete avoidance, the unmanned boat returned to the preplanned path with an arc trajectory.

The boat avoided the dynamic obstacle successfully by turning left at point B and reached the target point safely. In the red circle, the fusion algorithm introduced in this paper planned a remarkably smooth path, which proves its feasibility and effectiveness. Therefore, the introduced algorithm can complete the USV path planning task well.

7 Conclusion

A fusion algorithm that combines the improved A* algorithm and the IDWA algorithm with an improved fusion strategy was proposed in this paper. First, in consideration of the safety of USV navigation and the smoothness of the path, the node expansion strategy was optimized on the basis of the enlarged 5×5 searching neighborhood. Second, the heuristic function of A* was improved. Next, redundant nodes of the path were removed, and the evaluation function of the DWA algorithm and fusion strategy were improved. Experiments have proven that compared with the fusion algorithm of improved A* and traditional DWA algorithms with a traditional fusion strategy, the fusion algorithm proposed in this paper reduced the path length by 7.54% and the time by 9.9%, prevented unnecessary turns, and obtained a smoother path. In addition, the USV can avoid static and dynamic obstacles.

Although the algorithm introduced in this paper performs better than the common traditional path planning algorithm, certain issues need to be considered in future research.

1) Three-dimensional environmental factors, such as wind, waves, and ocean currents, must be considered and combined with COLREGs during path planning to improve the applicability of USVs in oceans.

2) The obstacle-avoidance problem of multiple USVs must be considered in path planning.

Funding Supported by the EDD of China (No. 80912020104) and the Science and Technology Commission of Shanghai Municipality (No. 22ZR1427700 and No. 23692106900).

Competing interest The authors have no competing interests to declare that are relevant to the content of this article.

Reference

Alireza M, Vincent D, Tony W (2021) Experimental study of path planning problem using EMCOA for a holonomic mobile robot. Journal of Systems Engineering and Electronics 32(6): 1450-1462. https://doi.org/10.23919/jsee.2021.000123

Bai X, Li B, Xu X, Xiao Y (2023) USV path planning algorithm based on plant growth. Ocean Engineering 273: 113965. https://doi.org/10.1016/j.oceaneng.2023.113965

Chen P, Huang Y, Papadimitriou E, Mou J, Van Gelder P (2020) Global path planning for autonomous ship: A hybrid approach of fast marching square and velocity obstacles methods. Ocean Engineering 214: 107793. https://doi.org/10.1016/j.oceaneng.2020.107793

Chiang HTL, Tapia L (2018) COLREG-RRT: An RRT-based COLREGS-compliant motion planner for surface vehicle navigation. IEEE Robotics and Automation Letters 3(3): 2024-2031. https://doi.org/10.1109/lra.2018.2801881

Fox D, Burgard W, Thrun S (1997) The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine 4(1): 23-33. https://doi.org/10.1109/100.580977

Guan W, Wang K (2023) Autonomous collision avoidance of unmanned surface vehicles based on improved A-Star and dynamic window approach algorithms. IEEE Intelligent Transportation Systems Magazine 15(3): 36-50. https://doi.org/10.1109/mits.2022.3229109

Hu S, Tian S, Zhao J, Shen R (2023) Path planning of an unmanned surface vessel based on the improved A-Star and dynamic window method. Journal of Marine Science and Engineering 11(5): 1060. https://doi.org/10.3390/jmse11051060

Li W, Wang L, Fang D, Li Y, Huang J (2021) Path planning algorithm combining A* with DWA. Syst. Eng. Electron 43: 3694-3702. DOI: 10.12305/j.issn.1001-506X.2021.12.33

Liang C, Zhang X, Watanabe Y, Deng Y (2021) Autonomous collision avoidance of unmanned surface vehicles based on improved A Star and minimum course alteration algorithms. Applied Ocean Research 113: 102755. https://doi.org/10.1016/j.apor.2021.102755

Long Y, Liu S, Qiu D, Li C, Guo X, Shi B, AbouOmar MS (2023) Local path planning with multiple constraints for USV based on improved bacterial foraging optimization algorithm. Journal of Marine Science and Engineering 11(3): 489. https://doi.org/10.3390/jmse11030489

Lyridis DV (2021) An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints. Ocean Engineering 241: 109890. https://doi.org/10.1016/j.oceaneng.2021.109890

Ma J, Jin J, Liu D, Liu L, Wang D, Chen M (2021) An improved adaptive dynamic window algorithm for target tracking of unmanned surface vehicles. 2021 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), 1-5. https://doi.org/10.1109/icspcc52875.2021.9564615

Mao S, Yang P, Gao D, Bao C, Wang Z (2023) A motion planning method for unmanned surface vehicle based on improved RRT algorithm. Journal of Marine Science and Engineering 11(4): 687. https://doi.org/10.3390/jmse11040687

Niu Y, Zhang J, Wang Y, Yang H, Mu Y (2022) A review of path planning algorithms for USV. Lecture Notes in Electrical Engineering, 263-273. https://doi.org/10.1007/978-981-16-9492-9_27

Öztürk Ü, Akdağ M, Ayabakan T (2022) A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective. Ocean Engineering 251: 111010. https://doi.org/10.1016/j.oceaneng.2022.111010

Qin H, Shao S, Wang T, Yu X, Jiang Y, Cao Z (2023) Review of autonomous path planning algorithms for mobile robots. Drones 7(3): 211. https://doi.org/10.3390/drones7030211

Song R, Liu Y, Bucknall R (2019) Smoothed A* algorithm for practical unmanned surface vehicle path planning. Applied Ocean Research 83: 9-20. https://doi.org/10.1016/j.apor.2018.12.001

Tan Z, Wei N, Liu Z (2022) Local path planning for unmanned surface vehicle based on the improved DWA algorithm. 2022 41st Chinese Control Conference (CCC), 3820-3825. https://doi.org/10.23919/ccc55666.2022.9901807

Vagale A, Oucheikh R, Bye RT, Osen OL, Fossen TI (2021) Path planning and collision avoidance for autonomous surface vehicles I: a review. Journal of Marine Science and Technology 26(4): 1292-1306. https://doi.org/10.1007/s00773-020-00787-6

Wang N, Xu H, Li C, Yin J (2020) Hierarchical path planning of unmanned surface vehicles: A fuzzy artificial potential field approach. International Journal of Fuzzy Systems 23(6): 1797-1808. https://doi.org/10.1007/s40815-020-00912-y

Xu D, Yang J, Zhou X, Xu H (2024) Hybrid path planning method for USV using bidirectional A* and improved DWA considering the manoeuvrability and COLREGs. Ocean Engineering 298: 117210. https://doi.org/10.1016/j.oceaneng.2024.117210

Yang Z, Li N, Zhang Y, Li J (2023) Mobile robot path planning based on improved particle swarm optimization and improved dynamic window approach. Journal of Robotics 23: 1-16. https://doi.org/10.1155/2023/6619841

Yin X, Cai P, Zhao K, Zhang Y, Zhou Q, Yao D (2023) Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms. Sensors 23(8): 4102. https://doi.org/10.3390/s23084102

Zhang J, Zhao H, Wang N, Guo C (2021) Fuzzy dual-window DWA algorithm for USV in dense obstacle conditions. Chinese Journal of Ship Research 16(6): 10. DOI: 10.19693/j.issn.1673-3185. 02095

Zhong X, Tian J, Hu H, Peng X (2020) Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in Large-Scale dynamic environment. Journal of Intelligent & Robotic Systems 99(1): 65-77. https://doi.org/10.1007/s10846-019-01112-z

Zhou C, Gu S, Wen Y, Du Z, Xiao C, Huang L, Zhu M (2020) The review unmanned surface vehicle path planning: Based on multi-modality constraint. Ocean Engineering 200: 107043. https://doi.org/10.1016/j.oceaneng.2020.107043