

Rapid global path planning algorithm for unmanned surface vehicles in large-scale and multi-island marine environments

Dong Wang^{1,2}, Jie Zhang^{1,2}, Jiucui Jin², Deqing Liu² and Xingpeng Mao¹

¹School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, Heilongjiang, China

²First Institute of Oceanography, Ministry of Natural Resources, Qingdao, Shandong, China

ABSTRACT

A global path planning algorithm for unmanned surface vehicles (USVs) with short time requirements in large-scale and complex multi-island marine environments is proposed. The fast marching method-based path planning for USVs is performed on grid maps, resulting in a decrease in computer efficiency for larger maps. This can be mitigated by improving the algorithm process. In the proposed algorithm, path planning is performed twice in maps with different spatial resolution (SR) grids. The first path planning is performed in a low SR grid map to determine effective regions, and the second is executed in a high SR grid map to rapidly acquire the final high precision global path. In each path planning process, a modified inshore-distance-constraint fast marching square (IDC-FM²) method is applied. Based on this method, the path portions around an obstacle can be constrained within a region determined by two inshore-distance parameters. The path planning results show that the proposed algorithm can generate smooth and safe global paths wherein the portions that bypass obstacles can be flexibly modified. Compared with the path planning based on the IDC-FM² method applied to a single grid map, this algorithm can significantly improve the calculation efficiency while maintaining the precision of the planned path.

Submitted 16 February 2021

Accepted 4 June 2021

Published 29 June 2021

Corresponding authors

Jie Zhang, zhangjie@fio.org.cn

Jiucui Jin, jinjiucui@fio.org.cn

Academic editor

Alma Alanis

Additional Information and
Declarations can be found on
page 25

DOI 10.7717/peerj-cs.612

© Copyright
2021 Wang et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

Subjects Algorithms and Analysis of Algorithms, Autonomous Systems, Robotics

Keywords Unmanned surface vehicle, Global path planning, Fast marching method, Fast marching square, Time optimal

INTRODUCTION

Research on unmanned surface vehicles (USVs) has received increased attention in various military and civilian applications over recent years (Yan et al., 2010; Campbell, Naeem & Irwin, 2012; Liu et al., 2016). Robust and reliable navigation, guidance, and control (NGC) systems are required for USVs to perform a variety of complex marine missions. Path planning is an essential component of an NGC system. The main aim of path planning is to calculate a collision-free path with specific requirements. It determines the automation level of USVs and ensures the reliability and success of missions (Tan et al., 2020).

Global path planning is an important aspect of path planning. A variety of algorithms for the global path planning of USVs have been studied for different requirements. Dijkstra's algorithm (Dijkstra, 1959) is a classic graph-based algorithm that can plan the shortest global path for USVs (Xie et al., 2016; Singh et al., 2018). As an improvement of Dijkstra's

algorithm, the A* algorithm (Hart, Nilsson & Raphael, 1968) and its related improved algorithms are also commonly used in the global path planning of USVs (Campbell, Naeem & Irwin, 2012), such as the direction priority sequential selection method (Naeem, Irwin & Yang, 2012); the A* algorithm, which considers environment effects (EEA*) (Lee et al., 2015); and the angular rate-constrained Theta* (ARC-Theta*) algorithm, which considers both the angular rate and heading angle of USVs (Kim et al., 2014). These algorithms exhibit acceptable convergence and consistency; however, the planned paths must be further smoothed. The artificial potential field (APF) method (Khatib, 1986) is a classic method for both global and local path planning. The main drawback of the traditional APF method is the local minimum problem. Therefore, different studies (Guo, Gao & Cui, 2013; Song, Hao & Su, 2020) have been conducted to solve the local minimum problem for the path planning of robots or USVs. Evolutionary methods such as the genetic algorithm (GA) (Kim et al., 2017; Arzamendia et al., 2019; Wang et al., 2020), particle swarm optimization (PSO) (Song et al., 2015), and ant colony optimization (ACO) (Song, 2014; Xia et al., 2019) have also been applied in the path planning of USVs. GA is robust and adaptable, but it has the shortcomings of poor local search ability and the premature convergence phenomenon. Therefore, other methods, such as the simulated annealing algorithm, are usually introduced to improve the performance of GA (Zhang, Xu & Xie, 2019). Nevertheless, the path planned by GA still lacks consistency. Similar to GA, PSO has the advantage of strong robustness and the shortcoming of premature convergence. However, it is dependent on various parameters, and there is no specific theory for guiding the setting of these parameters for different problems. Similarly, the parameter selection of ACO is more dependent on experience, and this algorithm can easily fall into local extremum.

In some special applications, the shortest time may be an important requirement for USVs. The fast marching method (FMM) can be a solution for time-optimal global path planning. This method was first proposed by Tsitsiklis (1995), and Adalsteinsson & Sethian (1995) independently and was extended by Sethian (1999). An experimental survey for nine different FMMs is shown in Gómez et al. (2019). The path planned by the FMM is usually extremely close to the obstacles. One solution is to adjust the speed map, as exemplified by the method with an adjusted cost function (Messias et al., 2014) and the FM² method (Garrido et al., 2007). FMM-based methods have been widely used in path planning applications (Gómez et al., 2013; Amorim & Ventura, 2014; Alvarez et al., 2015; González et al., 2016). Marine applications based on FMM were introduced by Garrido, Alvarez & Moreno (2020). Interesting modifications for the FM² method have been performed, and the FMM has been subjected to a vector field considering the effects of several vector variables such as wind flow or water currents (Garrido, Alvarez & Moreno, 2020). In addition, studies on path following and obstacle avoidance and formations have also been conducted using FMM (Garrido, Alvarez & Moreno, 2020). USV formation path planning has also been performed by Liu & Bucknall (2015) and Tan et al. (2020), and an angle-guidance FM² method has been used for the Springer USV to make the generated path compliant with the dynamics and orientation restrictions of USVs (Liu & Bucknall, 2016; Liu, Bucknall & Zhang, 2017). An improved anisotropic fast marching method using a

multi-layered fast marching was proposed by [Song, Liu & Bucknall \(2017\)](#), which combines different environmental factors and provides interesting results. In addition to the global path planning, the FMM-based methods also show potential in collision avoidance of USVs ([Wang, Jin & Er, 2019](#); [Garrido, Alvarez & Moreno, 2020](#)). These successful studies have demonstrated the potential of FMM-based methods in global path planning of USVs.

The basic FMM can plan time-optimal paths for USVs. However, some of the main shortcomings are that the paths planned by the FMM are too close to obstacles, and there may be abrupt turns when the paths bypass obstacles with sharp corners. Thus, the FM^2 method is proposed to address these problems, and two dimensionless parameters are introduced to adjust the paths more flexibly ([Garrido et al., 2007](#); [Garrido, Alvarez & Moreno, 2020](#)). However, the two introduced parameters lack clear physical meaning, and the suitable adjustment is difficult to identify with specific values. Moreover, the adjustment effects of these two parameters are not common because the adjustment degree with the same parameters varies with different grid maps. Another common problem about FM-based methods is that the computational efficiency of path planning decreases sharply when the scale of the grid map is very large. Therefore, we make two improvements to address the mentioned shortcomings of the basic FM^2 method. First, we introduced an inshore-distance-constraint fast marching square (IDC- FM^2) method to improve the inshore path adjustment performance for the first time. Comparing with the basic FM^2 method, the IDC- FM^2 method applies two inshore distance parameters other than the two dimensionless parameters to adjust the paths around the obstacles. The adjustment effects for path planning of the IDC- FM^2 method are stable in different grid maps as the IDC- FM^2 method can constrain the path portions around the obstacles within the region constrained by the two inshore distance parameters. Further, to improve the computational efficiency, the algorithm that applies the IDC- FM^2 method based on two-level spatial resolution grid maps is designed.

RELATED METHODS

Environment map model

In 2D global path planning applications based on the FMM or its improved methods, discrete numerical calculations are based on cartesian grid maps. Therefore, an environment map should first be converted into a binary grid map with a suitable spatial resolution (SR). Free and open-source satellite images (such as Google satellite images) can be used as the data source for the maps in most USV applications, and the corresponding grid maps can be generated using image processing ([Shi et al., 2018](#)) combined with manual assistance. The binary cell is set as an obstacle cell when an obstacle exists at the geographic location (0 values); otherwise, it is set as a free cell (1 value).

Fast marching square method

When using FM-based methods to plan a path, the basis is the FMM. The core work of the FMM is calculating solutions of the Eikonal equation ([Tsitsiklis, 1995](#); [Adalsteinsson & Sethian, 1995](#)). The Eikonal equation describes a wave front propagation scenario from sources with the speed of a wave front given as F_x at cell x . It can be expressed as

$\|\nabla T_x\|F_x = 1$, where T_x is the arrival time of the wave from the source to the cell x and ∇ is a vector differential operator. From the perspective of time-cost, the Eikonal equation can be expressed in another form (that of [Lin, 2003](#)):

$$\|\nabla T_x\| = \tau_x \quad (1)$$

where τ_x is the time-cost at cell x and is equivalent to $1/F_x$. The solution we want to calculate is T_x , and all of them compose the arrival time map, $\mathcal{T}(x)$. All time-cost τ_x compose the time-cost function map, $\boldsymbol{\tau}(x)$.

The FMM was first proposed independently by [Tsitsiklis \(1995\)](#) and [Adalsteinsson & Sethian \(1995\)](#). The solution T_x in cell x can be interpreted as the wave arrival time from the nearest source to cell x . Three cell sets (the accepted set \mathcal{S}_A , trial set \mathcal{S}_T , and far set \mathcal{S}_F) are defined to calculate $\mathcal{T}(x)$. The set of all cells at which T_x will not change is \mathcal{S}_A . \mathcal{S}_T is the set of cells to be examined. Every T_x in \mathcal{S}_T has been previously computed but may be updated. \mathcal{S}_F is the set of all other cells with T_x , which has never been computed ([Lin, 2003](#)).

The calculation process includes initialization and loop procedures. During initialization, source cells x_0 with $T_{x_0} = 0$ are set to \mathcal{S}_T , and all other cells are set to \mathcal{S}_F with $T_x = \infty$. After the initialization, the cell x_m with the smallest T value in \mathcal{S}_T is selected and moved from \mathcal{S}_T to \mathcal{S}_A . Thereafter, the non-accepted neighbor cells of x_m are updated, including the solutions and cell sets. Considering [Tsitsiklis's](#) deduction ([Tsitsiklis, 1995](#); [Lin, 2003](#)), each of the new neighbor solutions is determined by:

$$T_x = \min(\tilde{T}_x, (T_{x_mx_i}, i = 1, 2)) \quad (2)$$

where \tilde{T}_x denotes the original solution. $T_{x_mx_i}, i = 1, 2$ are the candidate solutions for the paths that pass through the line segment x_mx_i and propagate to cell x (see [Fig. 1](#)), which are calculated by:

$$T_{x_mx_i} = \begin{cases} \frac{1}{2} \left(T_{x_m} + T_{x_i} + \sqrt{2\tau_x^2 - (T_{x_m} - T_{x_i})^2} \right), & T_{x_mx_i} > T_{x_m}, \text{ and } T_{x_mx_i} > T_{x_i} \\ \min(T_{x_m}, T_{x_i}) + \tau_x, & \text{others} \end{cases} \quad (3)$$

where T_{x_m} and T_{x_i} are the accepted solutions at cells x_m and x_i , respectively. If cell x_i is in \mathcal{S}_T , T_{x_i} is ∞ . For the cell set, if a non-accepted neighbor cell x is in \mathcal{S}_F , it moves from \mathcal{S}_F to \mathcal{S}_T . The loop procedure is performed continually until all cells are in \mathcal{S}_A . The resulting map is the arrival time map, \mathcal{T} .

The main disadvantage of the basic FMM is that the computed path is too close to obstacles and forces the vehicle to perform abrupt turns ([Garrido, Alvarez & Moreno, 2020](#)). As described in [Garrido et al. \(2007\)](#) and [Garrido, Alvarez & Moreno \(2020\)](#), a smooth path with sufficient safety distances from obstacles can be computed using the FM² method. This method applies the basic FMM twice. The procedure for computing paths is described as follows ([Garrido, Alvarez & Moreno, 2020](#)):

1. The environment is modeled as a binary grid map ([Fig. 2A](#)).
2. The FM-1st step. All obstacle cells are used as wave sources ($T = 0$), expanding several waves at the same time at a constant speed. The value of each cell in the resulting map indicates the time required for a wave to reach the closest obstacle (see [Fig. 2B](#)). This

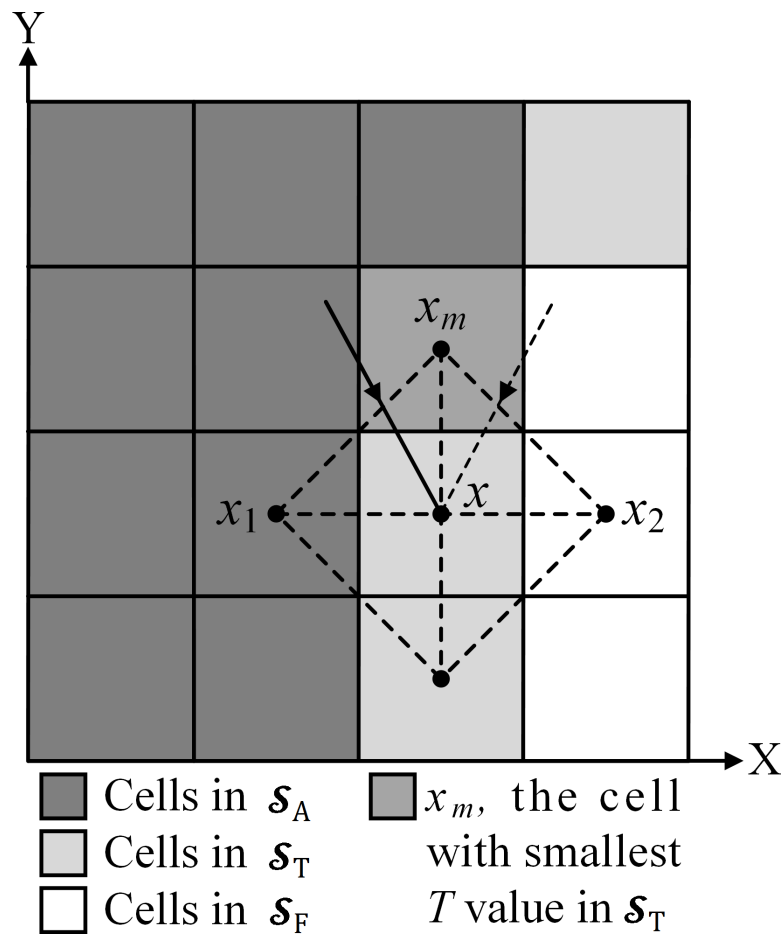


Figure 1 Diagram of updating the neighbor of x_m in a 4-neighbor scheme. Cell x is the neighbor cell to be updated. In this case, x is in \mathcal{S}_T . If x is a cell in \mathcal{S}_F , it moves from \mathcal{S}_F to \mathcal{S}_T .

Full-size DOI: 10.7717/peerjcs.612/fig-1

is proportional to the distance from the obstacles. By reversing the meaning of these values, they can be understood as the maximum admissible speed at each cell. Finally, the speed values are rescaled to fix a maximum cell value of 1.

Modifications of the speed map with two parameters, α and β , can adjust distances between the computed paths and obstacles. The value of each cell in the speed map $F_{i,j}$ is adjusted exponentially by α :

$$\text{new}F_{i,j} = F_{i,j}^\alpha \quad (4)$$

The parameter β is used to saturate the values in the speed map. It is defined within the range of 0 and 1. Every $F_{i,j}$ with a value greater than β is set to one (see Fig. 2C).

Comparing with a path without modifications, the path modified by α will be closer to obstacles when $\alpha < 1$. On the contrary, if $\alpha > 1$, the modified path will stay further away from obstacles. The parameter β allows the path to move closer to obstacles.

When β is smaller, the modified path will be closer to obstacles.

3. The FM-2nd step. The goal point is used as a unique wave source. The wave is expanded over the map until the starting point is reached. The speed at each cell (equivalent to

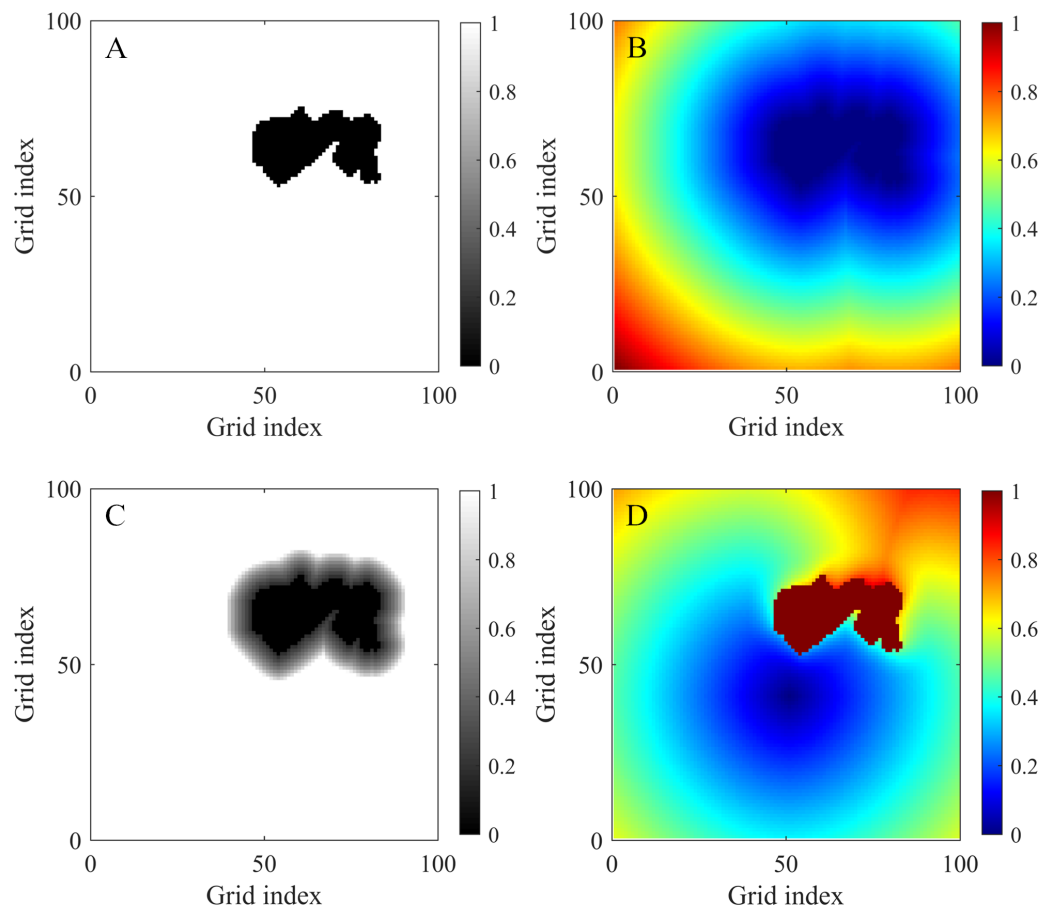


Figure 2 Maps used or generated in the process of using the FM² method. (A) Binary grid map of an obstacle environment. The black cells are obstacle cells, and the white cells are free cells. (B) Arrival time map expanding from the obstacles, which is proportional to the distance from the obstacles. (C) Speed map with a saturated value. The black areas (0 value) represent the obstacles, while the white areas (1 value) represent the areas with velocity saturation. The other values are rescaled to better demonstrate the trend. (D) Arrival time map obtained by applying the FM² method with $x_g = (50, 40)$. The path can be acquired by using the gradient descent method based on this map.

Full-size DOI: 10.7717/peerjcs.612/fig-2

$1/\tau_x$) is obtained from the modified speed map computed in the FM-1st step. The resulting arrival time map is shown in Fig. 2D.

- Finally, a gradient descent is applied over the resulting arrival time map from the starting point to the goal point. An optimal path in terms of the arrival time, smoothness, and safety is obtained.

Gradient descent method

The global path can be extracted by applying the gradient descent method. The path propagates along the gradient descent direction from the starting position P_s to the goal position P_g with a step length d . The value of d can be set to a value equal to the SR of the

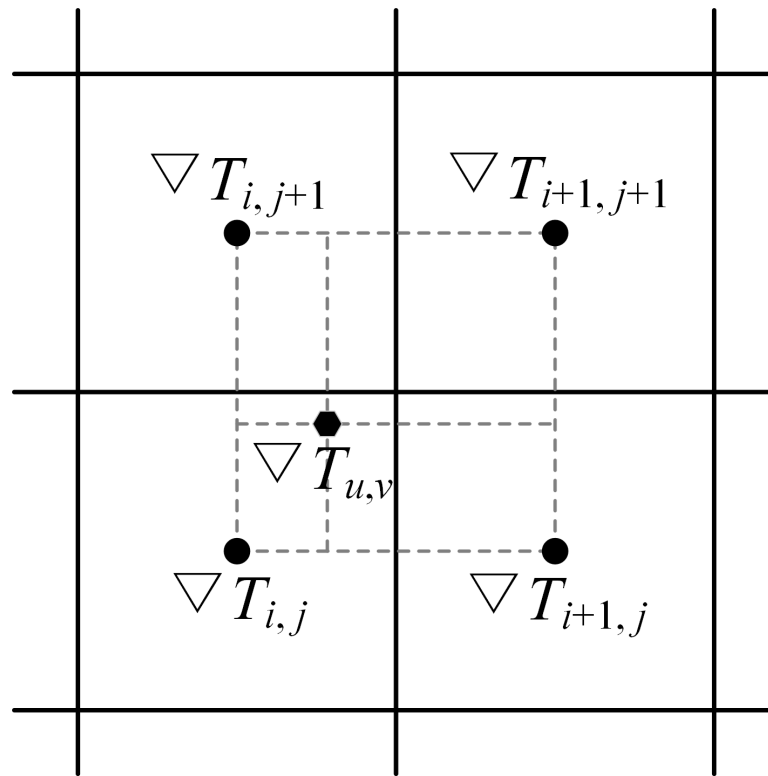


Figure 3 Modified gradient of a path waypoint calculated by the bilinear interpolation method.

Full-size  DOI: 10.7717/peerjcs.612/fig-3

grid map. The gradient of the cell $x_{i,j}$ is calculated by:

$$\nabla T_{i,j} = \left[\frac{T_{i+1,j} - T_{i-1,j}}{2} \quad \frac{T_{i,j+1} - T_{i,j-1}}{2} \right]^T \quad (5)$$

where $T_{i,j}$ represents the arrival time value at cell $x_{i,j}$ and the non-italic T in the upper-right corner represents the transpose of vector $[x \ y]$.

The path waypoints are not limited to cell centers. Generally, the gradient of a path waypoint $P_n = (u, v)$ located in the cell $x_{i,j}$ is approximated by the cell gradient $\nabla T_{i,j}$. However, a modified gradient $\nabla T_{u,v}$ calculated by the bilinear interpolation can be used to improve the path precision (see Fig. 3).

PROPOSED ALGORITHM

To improve the computational efficiency of global path planning in a large-scale and multi-island marine environment, the proposed algorithm performs path planning twice for different purposes. First, the path planning is performed in a low SR (LSR) grid map to determine an effective region. The final global path is then obtained in the second path planning within the effective region of a high SR (HSR) grid map. The relevant methods and procedures applied in the algorithm are as follows. The complete algorithm flow is summarized in the last part of this section.

Mapping of two-level SR grid maps

The two-level SR grid maps are contained in the HSR and LSR grid maps. The HSR grid map is directly obtained from the Google satellite image data. A mapping relationship between the LSR cells and the corresponding $L \times L$ HSR cell sub-blocks is established, which can be expressed as:

$$(i_L, j_L) \sim \begin{bmatrix} (i'_H, j'_H + L - 1) & \cdots & (i'_H + L - 1, j'_H + L - 1) \\ \vdots & \ddots & \vdots \\ (i'_H, j'_H) & \cdots & (i'_H + L - 1, j'_H) \end{bmatrix} \quad (6)$$

where (i_L, j_L) are the LSR cell coordinates, and (i'_H, j'_H) are the original cell coordinates of the mapped HSR cell sub-block, as shown in Fig. 4.

To map the LSR grid map to a sub-block of the HSR grid map, the following is used:

$$\begin{cases} i'_H = i_{Ho} + Li_L \\ j'_H = j_{Ho} + Lj_L \end{cases} \quad (7)$$

To map from a sub-block of the HSR grid map to a cell of the LSR grid map, we use:

$$\begin{cases} i_L = f_{\text{floor}}\left(\frac{i_H - i_{Ho}}{L}\right) \\ j_L = f_{\text{floor}}\left(\frac{j_H - j_{Ho}}{L}\right) \end{cases} \quad (8)$$

where (i_H, j_H) , $i'_H \leq i_H \leq i'_H + L - 1$, $j'_H \leq j_H \leq j'_H + L - 1$ are the HSR cell coordinates. (i_{Ho}, j_{Ho}) are the original cell coordinates of the original HSR cell sub-block (see Fig. 4), which are determined by:

$$\begin{cases} i_{Ho} = \left(i_{Hg} - f_{\text{floor}}\left(\frac{L}{2}\right)\right) \% L \\ j_{Ho} = \left(j_{Hg} - f_{\text{floor}}\left(\frac{L}{2}\right)\right) \% L \end{cases} \quad (9)$$

where (i_{Hg}, j_{Hg}) are the HSR cell coordinates of the goal cell x_g , the function $f_{\text{floor}}(m)$ indicates a rounding down of m , and $a \% b$ indicates the modulo operation.

The SR of the LSR grid map is:

$$D_{LRes} = LD_{HRes} \quad (10)$$

where D_{HRes} is the SR of the HSR grid map. In general, L is defined as:

$$L \leq f_{\text{round}}\left(\frac{D_{Th}}{2D_{HRes}}\right) \quad (11)$$

where D_{Th} is the distance threshold of the virtual obstacle influence, and the function $f_{\text{round}}(m)$ indicates a rounding of m . The cell numbers of the LSR grid map in the X-axis and Y-axis directions are:

$$M_L = f_{\text{floor}}\left(\frac{M_H - i_{Ho}}{L}\right) \quad (12)$$

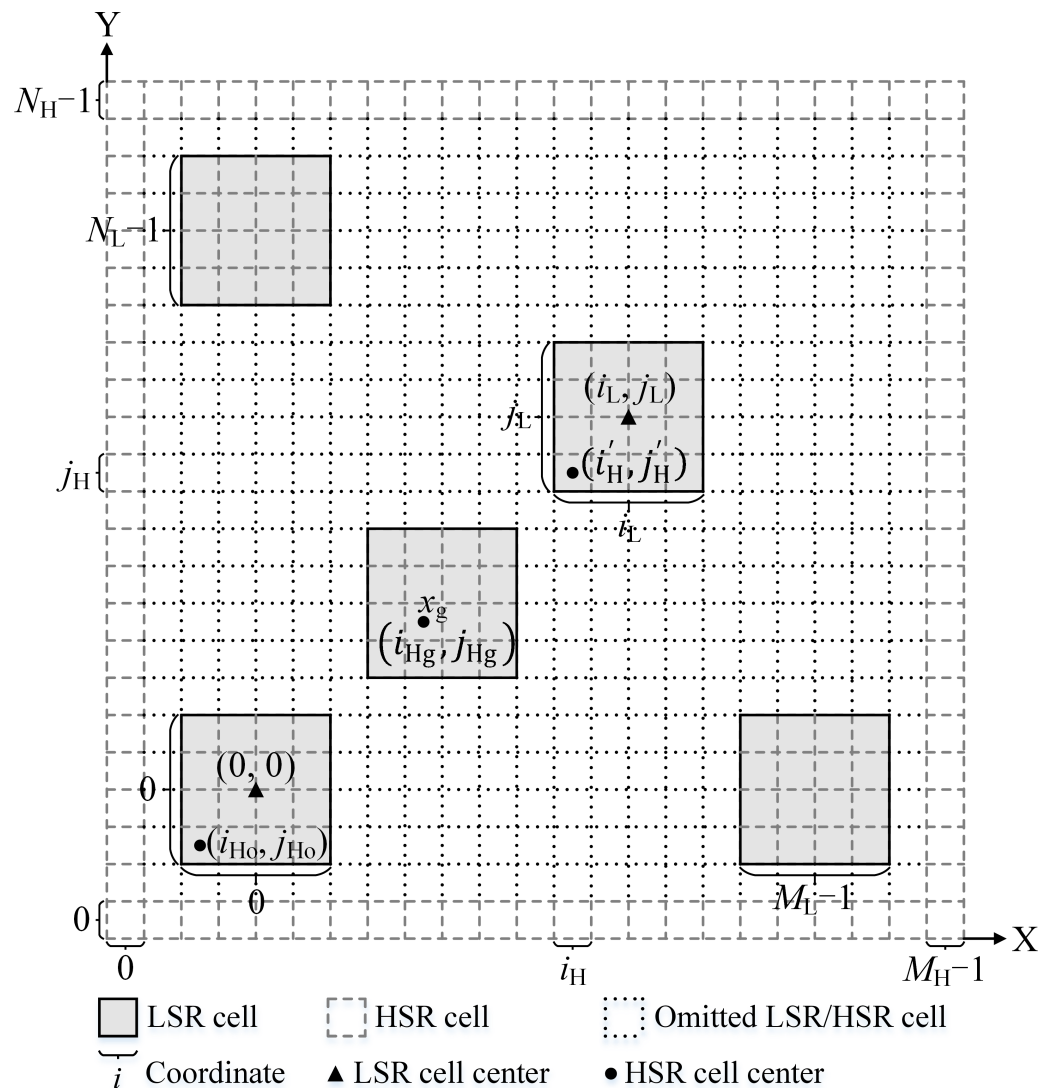


Figure 4 Mapping between an LSR grid map and an HSR grid map. Here, a case with $L = 4$ is shown. (i_{Ho}, j_{Ho}) is adaptively fine-tuned based on x_g . Therefore, x_g is set as the center (L is odd) or the quasi-center (L is even) cell of an HSR cell sub-block.

Full-size DOI: 10.7717/peerjcs.612/fig-4

$$N_L = f_{\text{floor}}\left(\frac{N_H - j_{Ho}}{L}\right) \quad (13)$$

where M_H and N_H are the cell numbers of the HSR grid map in the X-axis and Y-axis directions, respectively.

Based on the mapping relationship, an LSR cell type is determined by comparing the setting threshold Γ and obstacle cell proportion γ in the HSR cell sub-block. If $\gamma > \Gamma$, the LSR cell is set as an obstacle cell; otherwise, it is set as a free cell. Γ prefers to select a small value to retain as much obstacle information as possible.

IDC-FM² method

In the basic FM² method, the path is improved using a modified speed map. Two parameters, α and β , are used to modify the speed map. These modifications ensure the flexibility of the computed path. However, these two parameters have no clear physical meaning. The normalization of the speed values in the FM-1st step also results in several shortcomings. For example, all the speed values in the speed map must be calculated. When the map range changes slightly, the entire speed map rescales and differs in the same locations of the original map, and the parameters must be adjusted.

An IDC-FM² method is proposed to address these shortcomings. In this method, a time-cost weighting function, w_x , is introduced to adjust the speed map. Three inshore-distance parameters (the distance threshold D_{Th} , as well as the virtual strong and weak constraint distance parameters, D_{sc} and D_{wc} , respectively) and two weighting function values with respect to D_{sc} and D_{wc} (w_{sc} and w_{wc} , respectively) are used to determine the function w_x , and D_{Th} is used to achieve the same function as the speed saturation. Although five parameters are set, only two inshore-distance parameters, D_{Th} and D_{sc} , must be considered, while suitable values of w_{sc} and w_{wc} can be selected as constant values. Based on this method, the path portion around an obstacle is constrained in a region determined by D_{Th} and D_{sc} .

Time-cost weighting function

The time-cost weighting function w_x is introduced as a concept of the virtual obstacle influence. When a USV is far away from obstacles, there is no obstacle influence. As the USV approaches, the virtual obstacle influence increases slowly at first and the amplitude increases gradually. The obstacle influence increases dramatically when the USV is close to within a certain degree.

Similar to the FM-1st step in the basic FM² method, all obstacle cells are set as source cells to first calculate the arrival time map. The difference here is that a threshold $T_{Th} = \tau D_{Th}/D_{Res}$ could be set to speed up the calculation, where D_{Res} is the SR of the map, and τ is the unified time-cost which can take 1 as the value. In the loop procedure, if the smallest T value of a cell in S_T is larger than or equal to T_{Th} , all T values of non-accepted cells can be set as T_{Th} directly. Based on the resulting arrival time map, \mathcal{T}_{sat} , with a saturation threshold, the time-cost weighting function map, W' , is designed as:

$$w'_x = \begin{cases} \infty, & T_x = 0 \\ 1 + a \left(\frac{T_{Th}}{T_x} - 1 \right)^b, & T_x > 0 \end{cases} \quad (14)$$

where w'_x and T_x are the values of the cell x in W' and \mathcal{T}_{sat} , respectively, and a and b are two positive coefficients to be determined.

When a free cell, x , is within the virtual influenced scope of obstacles (i.e., $0 < D_x \leq D_{Th}$, where D_x is the closest distance to obstacles), then:

$$D_x \approx D_{Res} \frac{T_x}{\tau} \quad (15)$$

Therefore, w_x with respect to D_x is used to approximate w'_x :

$$w_x(D_x) = \begin{cases} \infty, & D_x = 0 \\ 1 + a \left(\frac{D_{Th}}{D_x} - 1 \right)^b, & 0 < D_x \leq D_{Th} \\ 1, & D_x > D_{Th} \end{cases} \quad (16)$$

where $w_x(D_x)$ is the value in cell x of the approximate time-cost weighting function map, \mathbf{W} . The time-cost value of each cell in the time-cost map, which is the inversion of the speed map, is then adjusted to:

$$new\tau_x = w_x(D_x)\tau_x \quad (17)$$

Equation 16 implies that w_x increases when cell x is closer to obstacles. The planned path tends to select points with a lower arrival time-cost. Therefore, when the path is close to obstacles, it will be farther away from obstacles compared to the path planned by the basic FMM.

Parameters for the time-cost weighting function

Five parameters, D_{Th} , D_{sc} , D_{wc} , w_{sc} , and w_{wc} , are introduced to determine the coefficients a and b , which are determined by:

$$b = \frac{\ln(w_{sc} - 1) - \ln(w_{wc} - 1)}{\ln(1 - e_{sc}) - \ln(1 - e_{wc}) + \ln e_{wc} - \ln e_{sc}} \quad (18)$$

$$a = (w_{sc} - 1) \left(\frac{e_{sc}}{1 - e_{sc}} \right)^b \quad (19)$$

where $e_{sc} = D_{sc}/D_{Th}$, $e_{wc} = D_{wc}/D_{Th}$, $D_{sc} < D_{wc} < D_{Th}$, and $w_{sc} > w_{wc} > 1$.

The parameter D_{Th} determines the largest virtual influence scope of the obstacles. The suggestion for the selection of D_{Th} is that there should be a sufficiently safe buffer area for obstacles. The parameter D_{sc} is very important for the safety of the USV. First, it is suggested that this parameter satisfies:

$$D_{sc} \geq v_{U,\max} t_r - \frac{v_{U,\max}^2}{2a_d} \quad (20)$$

where $v_{U,\max}$ is the maximum speed of the USV, t_r is the reaction time of the thruster, and a_d is the negative maximum acceleration of the USV under braking. In an unknown environment, it is better to select a slightly larger value to ensure safety, which is similar in environments with shallow and reef waters. This value may be relatively small in deep and reef-free waters to improve path efficiency. D_{wc} is a parameter that acts as the adjusted constraint distance. In a non-channel ocean area, the path portions around obstacles will be limited to the regions between D_{wc} and D_{Th} . The D_{wc} value can be set independently or determined jointly by D_{Th} and D_{sc} . In this study, it is set as:

$$D_{wc} = D_{Th} - \frac{\sqrt{2}}{2}(D_{Th} - D_{sc}) \quad (21)$$

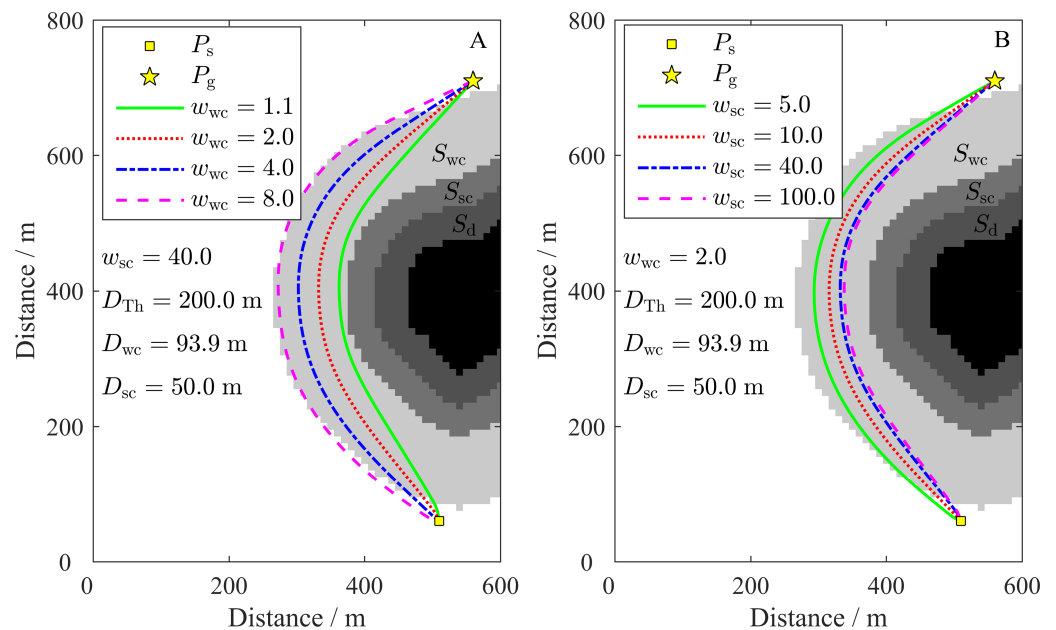


Figure 5 Influence of w_{wc} and w_{sc} on the path. (A) Influence of w_{wc} on the path. The black region is the obstacle, while the regions with different grayscales from dark to light are the danger region, S_d ; the strong constraint region, S_{sc} ; and the weak constraint region, S_{wc} . (B) Influence of w_{sc} on the path.

Full-size [DOI: 10.7717/peerjcs.612/fig-5](https://doi.org/10.7717/peerjcs.612/fig-5)

The area around the obstacles is divided into four parts by the three inshore-distance parameters: D_{sc} , D_{wc} , and D_{Th} . The three parts extending from an obstacle to the outside region are the danger region, S_d ; the strong constraint region, S_{sc} ; and the weak constraint region, S_{wc} (see Fig. 5). The influences of w_{wc} and w_{sc} on the path are shown in Fig. 5. As shown in Fig. 5A, the path will be located far from the obstacle until it is near the boundary of S_{wc} when w_{wc} is large. In contrast, the path is closer to the obstacle when w_{sc} is large (see Fig. 5B). The minimum degree of the path close to S_{sc} is mainly determined by w_{wc} , which can be inferred by further comparing the paths in Fig. 5B with the closest path to the obstacle ($w_{wc} = 1.1$) in Fig. 5A. However, regardless of the change in w_{wc} and w_{sc} ($w_{sc} > w_{wc} > 1$), the path portions bypassing obstacles will always be within S_{wc} or around the outside boundary of S_{wc} in non-channel ocean areas. In channel areas where there is no S_{wc} , the path will follow the quasi-centerline of the channel if it passes through the channel. In this study, the values of $w_{sc} = 40.0$ and $w_{wc} = 2.0$ were selected and fixed. The paths can be flexibly modified by the inshore-distance parameters D_{Th} and D_{sc} .

Determination of effective regions within the HSR grid map

The effective region within the HSR grid map is important for improving the computational efficiency of the proposed algorithm. It is acquired based on the mapping between the LSR and HSR grid maps when the corresponding region within the LSR grid map is determined.

Two effective regions that are respectively used in the FM-1st and FM-2nd steps of the IDC-FM² method must be determined. The low-precision initial path, ℓ_{ini} , is obtained first based on the LSR arrival time map \mathcal{T}_L . Two effective regions within the LSR grid map,

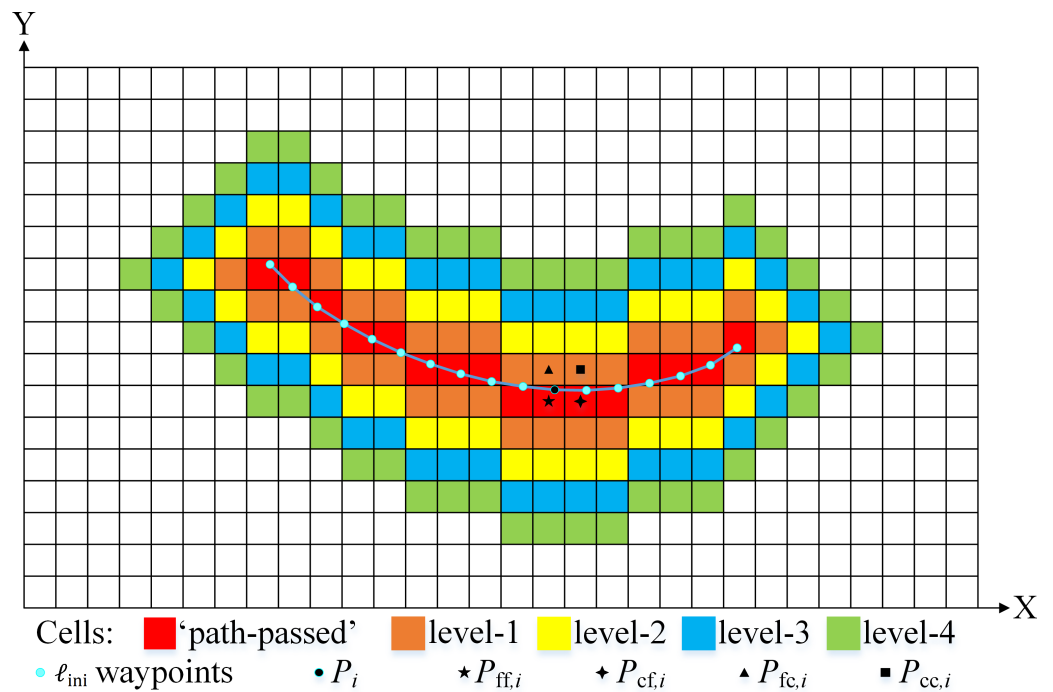


Figure 6 Effective region within the LSR grid map. The case shown uses $k = 4$, and $P_{ff,i} = (f_{\text{floor}}(x_i), f_{\text{floor}}(y_i))$, $P_{cf,i} = (f_{\text{ceil}}(x_i), f_{\text{floor}}(y_i))$, $P_{cc,i} = (f_{\text{ceil}}(x_i), f_{\text{ceil}}(y_i))$, and $P_{fc,i} = (f_{\text{floor}}(x_i), f_{\text{ceil}}(y_i))$ are the four neighbor cells of P_i . $P_{ff,i}$ is the “path-passed” cell with respect to P_i in this case.

Full-size DOI: 10.7717/peerjcs.612/fig-6

$S_{\text{LER}_{1\text{st}}}$ and $S_{\text{LER}_{2\text{nd}}}$, are determined by expanding the cells around the “path-passed” cells of ℓ_{ini} . As shown in Fig. 6, the nearest neighbor cell among the four neighbor cells of the waypoint $P_i = (x_i, y_i)$ is defined as the “path-passed” cells. When all “path-passed” cells are determined, the $\kappa_{2\text{nd}} = \kappa$ levels of the cells are extended to obtain $S_{\text{LER}_{2\text{nd}}}$. Considering $S_{\text{LER}_{1\text{st}}}$, there are three different situations:

- **Situation 1:** there is no cell with $w_x > 1$ in $S_{\text{LER}_{2\text{nd}}}$;
- **Situation 2:** there is at least one cell with $w_x > 1$ and no obstacle cell in $S_{\text{LER}_{2\text{nd}}}$.
- **Situation 3:** there is at least one obstacle cell in $S_{\text{LER}_{2\text{nd}}}$.

In **Situations 1** and **3**, $\kappa_{1\text{st}} = \kappa$. In **Situation 2**, $S_{\text{LER}_{2\text{nd}}}$ is extended continuously with $\Delta\kappa$ levels of cells until there is at least one obstacle cell for the first time, and $\kappa_{1\text{st}} = \kappa + \Delta\kappa$. Finally, two corresponding effective regions within the HSR grid map, $S_{\text{HER}_{1\text{st}}}$ and $S_{\text{HER}_{2\text{nd}}}$, are determined and used in the different FM steps of the IDC-FM² method to obtain the HSR arrival time map \mathcal{T}_H .

The extended parameter κ affects the paths (recorded as ℓ_κ) based on the proposed algorithm. When κ is not adequately large, these paths may not be consistent with the reference path ℓ_{ref} , which is acquired by applying the IDC-FM² method in the HSR grid map directly. A navigable global path is used as a case to compare the consistency between ℓ_κ and ℓ_{ref} . The basic parameters used in the proposed algorithm are listed in Table 1. Using ℓ_{ref} as the reference, the distances between the corresponding waypoints of ℓ_κ ($\kappa = 3, \dots, 6$)

Table 1 Basic parameter values for the proposed algorithm.

Parameter	Value	Parameter	Value	Parameter	Value
L	8	Γ	0.2	D_{Th} (m)	200
D_{sc} (m)	50	w_{sc}	40.0	w_{wc}	2.0

and ℓ_{ref} are shown in Fig. 7. It is shown that the largest distance decreases when κ increases, and the distance becomes 0 when $\kappa \geq 7$. Other cases were tested and similar results were obtained. These results indicate that there is good consistency when κ is sufficiently large (such as $\kappa = 10$). $\Delta\kappa$ is a variable parameter that was determined based on three different cases.

Algorithm flow

The proposed algorithm is improved based on the basic FM²-based algorithm. The basic FM²-based algorithm is executed directly on a single grid map. The algorithm flow of this basic algorithm is shown in Fig. 8 in the form of main data stream and used methods. This algorithm flow has two main steps. In *Step S1*, an arrival time map with obstacle sources and a saturation threshold, \mathcal{T}_{sat} , is obtained by first applying the FMM. Then, the approximate time-cost weighting function map, W , is calculated and used to adjust the time-cost map. Based on the adjusted time-cost map, an arrival time map with the goal point source, \mathcal{T} , is obtained by applying the FMM again. The entire process is used to obtain \mathcal{T} by applying the IDC-FM² method. Finally, the global path is acquired based on \mathcal{T} by applying the gradient descent method in *Step S2*.

The main data stream in the proposed algorithm and the methods used to obtain them are shown in Fig. 9. The main flow is as follows:

1. *Step T1*. Obtain the two-level SR grid maps;
2. *Step T2*. Obtain the LSR arrival time map \mathcal{T}_L in the LSR grid map;
3. *Step T3*. Determine the effective regions within the HSR grid map (S_{HER_1st} and S_{HER_2nd});
4. *Step T4*. Obtain the HSR grid arrival time map \mathcal{T}_H with the effective region constraint in both FM steps of the IDC-FM² method. There are several adjustments in both FM steps compared to the basic FMM performance. For the FM-1st step, all w_x values within S_{HER_1st} need not be adjusted, and this step can be ignored in **Situation 1**. In **Situation 2** and **Situation 3**, all free cells out of S_{HER_1st} are set to S_A with $T = \infty$ in the initialization of this step. In the FM-2nd step, all free cells out of S_{HER_2nd} are set to S_A with $T = \infty$ in the initialization;
5. *Step T5*. Acquire the final high precision global path by applying the gradient descent method based on \mathcal{T}_H .

RESULTS AND DISCUSSION

Simulation environments

Two surrounding spatial areas of Zhucha Island and Changhai County were selected as the simulation environments. The original maps adopted Google satellite images, as shown in

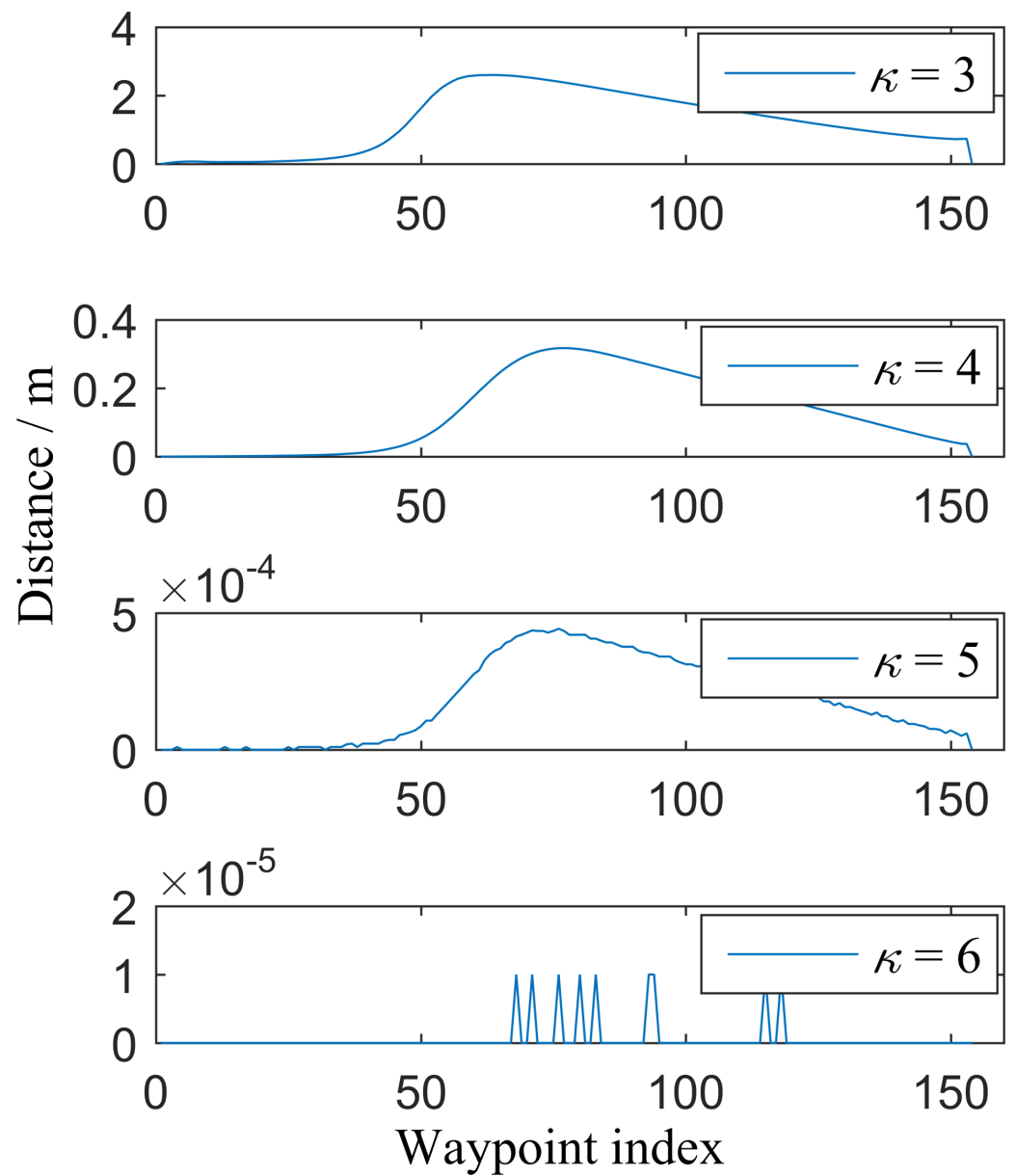


Figure 7 Distances between the corresponding waypoints of l_{ref} and l_k ($k = 3, \dots, 6$). l_{ref} is the path obtained by applying the IDC-FM² method based on the single HSR grid map directly, while l_k is the path obtained by applying the proposed algorithm based on two-level SR grid maps.

Full-size DOI: [10.7717/peerjcs.612/fig-7](https://doi.org/10.7717/peerjcs.612/fig-7)

Fig. 10. The spatial resolutions in the longitude and latitude directions are approximately 4.8 m and 3.9 m, respectively. Temporary binary grid maps are obtained based on the image processing method (Shi et al., 2018) combined with manual assistance, and then the HSR grid maps with $D_{\text{HRes}} = 10\text{m}$ in both the longitude and latitude directions are determined by resampling processing. The corresponding binary grid maps are presented in Fig. 11. Their ranges are $7\text{ km} \times 7\text{ km}$ and $64\text{ km} \times 48\text{ km}$, respectively.

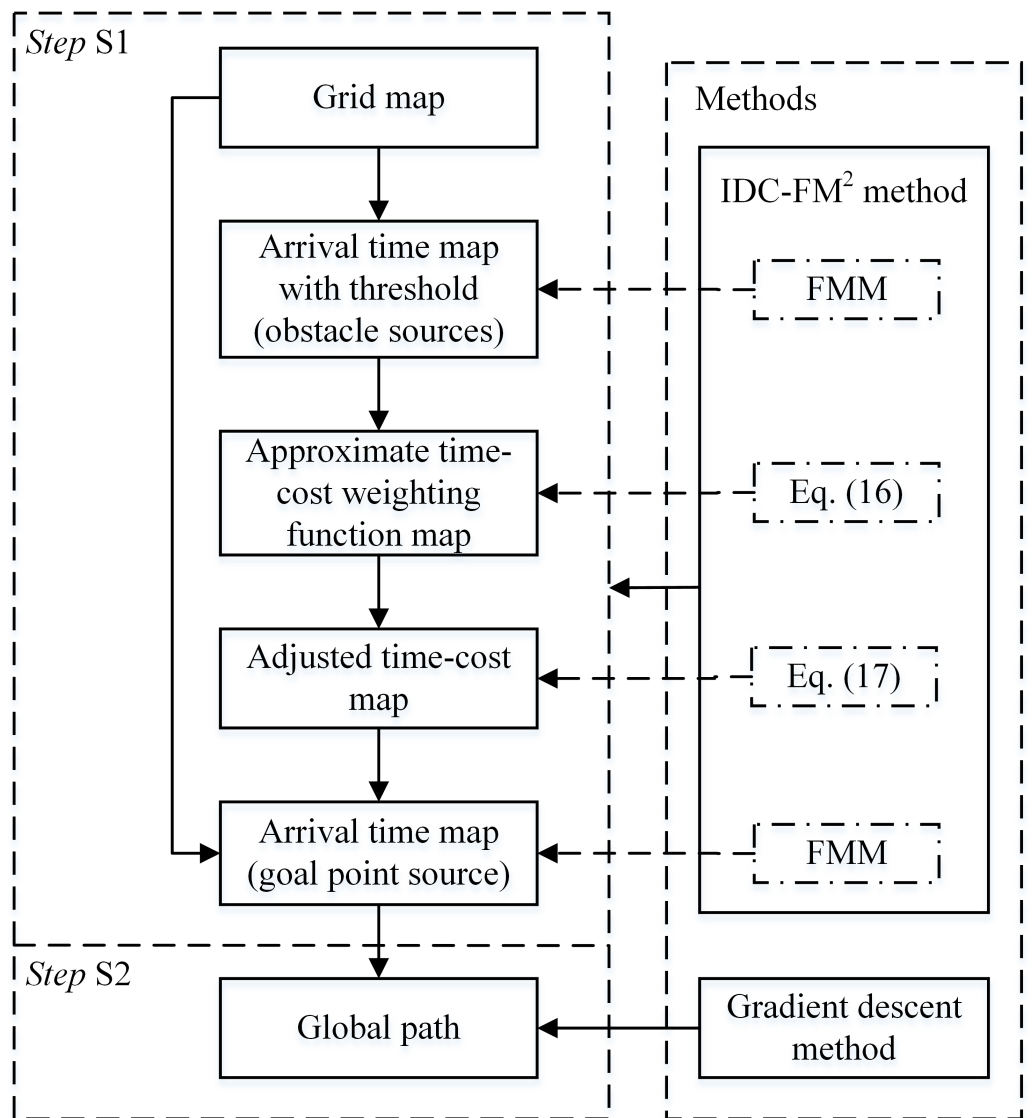


Figure 8 Algorithm flow of the basic algorithm applying the IDC-FM² method on a single grid map. This flow is expressed in the form of main data stream and used methods.

Full-size DOI: 10.7717/peerjcs.612/fig-8

Inshore-distance-constraint performances

A path planning from $P_s = [4.3 \text{ km}, 2.8 \text{ km}]$ to $P_g = [3.6 \text{ km}, 2.0 \text{ km}]$ is used as a typical case to analyze the inshore-distance-constraint performance of the IDC-FM² method using the inshore-distance parameters D_{Th} and D_{sc} . As shown in Fig. 12, the path planned based on the basic FMM, ℓ_{FMM} , is very close to the islands when this path bypasses them. For comparison, all paths planned by the proposed algorithm, ℓ_1 to ℓ_4 , are located away from islands by a certain distance. They are therefore significantly better choices from a safety perspective.

Paths ℓ_1 to ℓ_4 are acquired using different inshore-constraint distance parameters (D_{Th} and D_{sc} , see Table 2 wherein D_{wc} is calculated based on D_{Th} and D_{sc}) in the IDC-FM²

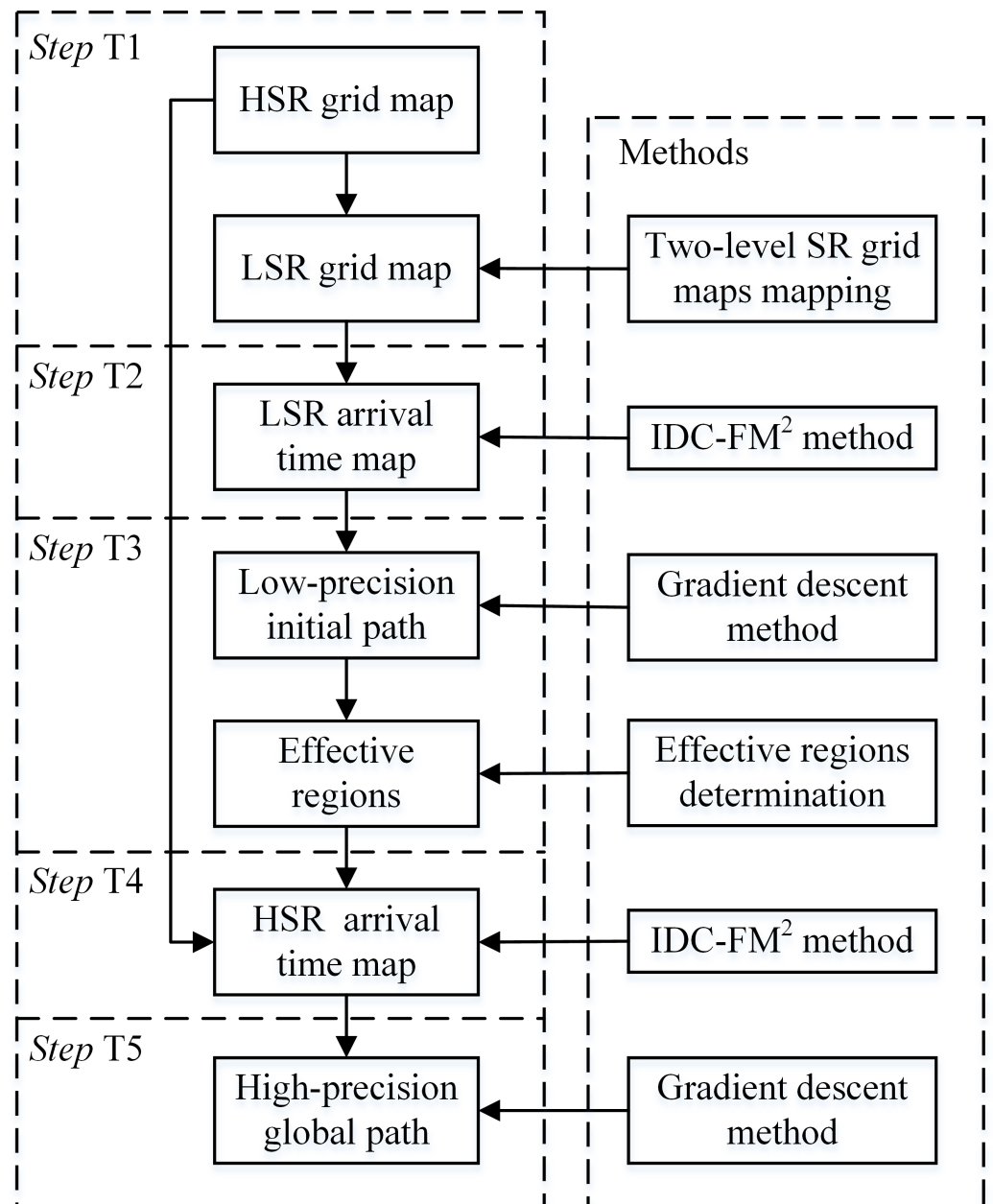


Figure 9 Algorithm flow of the proposed algorithm performed on two-level SR grid maps. This flow is expressed in the form of main data stream and used methods.

Full-size DOI: 10.7717/peerjcs.612/fig-9

method in which $w_{sc} = 40.0$ and $w_{wc} = 2.0$. These paths are clearly adjusted by these distance parameters. When the distance constraints (D_{Th} and D_{sc}) are small, the path (such as ℓ_1 , see Fig. 12) will be a somewhat close to the islands. The estimated quasi-closest distance from ℓ_1 to an island is approximately 38 m, which is shorter than half of the shortest channel width ($d_{hscw} \approx 101\text{m}$). In this situation, ℓ_1 is always outside of its S_{sc} (<28.2 m) value, and the path portions around the islands are located in the region of

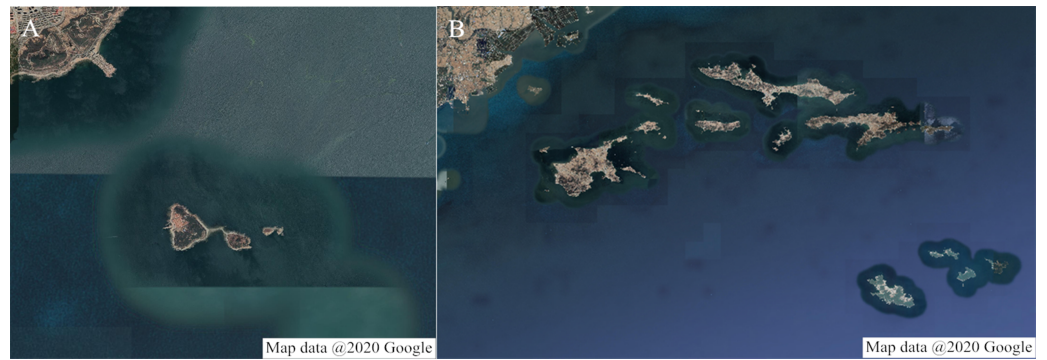


Figure 10 Google satellite images of the simulation environments. (A) Surrounding areas of Zhucha Island in Qingdao, Shandong Province, China. Map data @ 2020 Google. (B) Surrounding areas of Changhai County in Liaoning Province, China. Map data @ 2020 Google.

Full-size DOI: 10.7717/peerjcs.612/fig-10

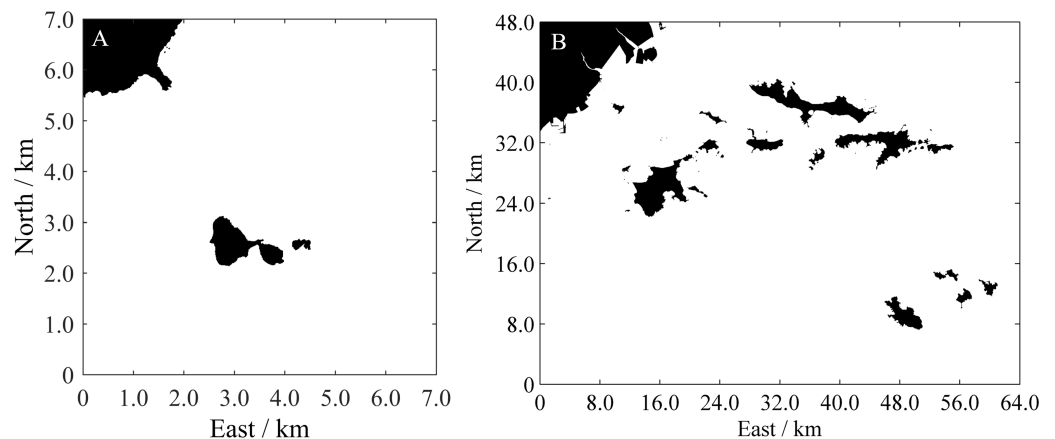


Figure 11 Binary grid maps of the simulation environments. (A) Binary grid map of the environment shown in Fig. 10A. The black cells indicate obstacle areas, while the white cells indicate the navigable areas for USVs. (B) Binary grid map of the environment shown in Fig. 10B.

Full-size DOI: 10.7717/peerjcs.612/fig-11

its S_{wc} (28.2 m to 60.0 m). When the distance constraints increase, the paths (such as ℓ_2 and ℓ_3 , see Fig. 12) in the non-channel areas will be outside of S_{sc} . The estimated quasi-closest distances (approximately 115 m and 128 m, respectively) are larger than d_{hscw} . However, ℓ_2 and ℓ_3 will be along the quasi-midline of the channel in the channel area, regardless of whether d_{hscw} is greater or less than D_{wc} for these two paths because the weighting time-cost will be smaller than the path around the islands (as in ℓ_4 , see Fig. 12). If the distance constraints are further increased, ℓ_4 will be selected for safety, although more time will be required. These results indicate that the path based on the proposed algorithm can be adjusted by setting different inshore-constraint distance parameters to meet safety requirements. Additionally, when the inshore-constraint distance parameters are determined, the path will cross a channel when its width is adequately large; otherwise, the path will bypass around the islands.

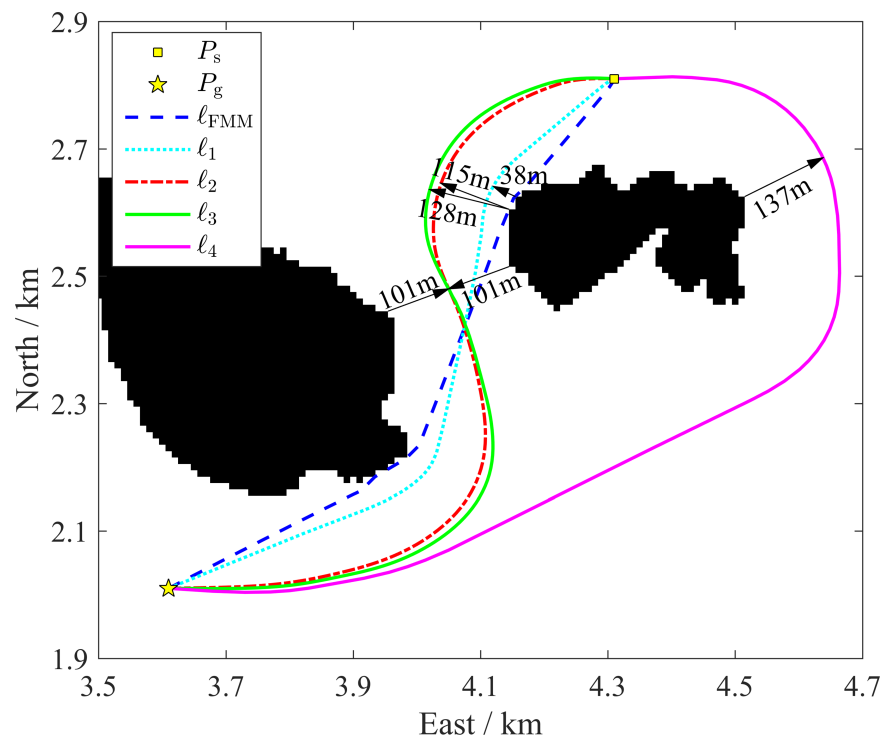


Figure 12 Path based on the basic FMM (l_{FMM}) and paths based on the proposed algorithm with different parameter settings of w_x (l_1 to l_4). The indicated distances are the estimated quasi-closest distances between the islands and path segments.

Full-size DOI: [10.7717/peerjcs.612/fig-12](https://doi.org/10.7717/peerjcs.612/fig-12)

Table 2 Inshore-constraint distance parameters for the determination of w_x .

Path	D_{Th} (m)	D_{wc} (m)	D_{sc} (m)
l_1	60.0	28.2	15.0
l_2	200.0	79.8	30.0
l_3	200.0	104.5	65.0
l_4	200.0	118.7	85.0

One classical approach to plan collision-free paths is applying the FMM within the map with inflated obstacles. For comparison, paths planned by the classical approach and the IDC-FM² method are shown in Fig. 13. The starting and goal positions are $P_s = [4.15 \text{ km}, 2.74 \text{ km}]$ to $P_g = [4.33 \text{ km}, 2.40 \text{ km}]$ and the island obstacles have been inflated by 94.0 m. As shown in Fig. 13, the path planned by the classical approach, $l_{FMM+\text{inflated}}$, is obviously influenced by the shape of the inflated obstacle. In some path turns which are marked by ellipses in Fig. 13, they are both affected by the shape of the inflated obstacle and the time-value gradient calculation of cells adjacent to the inflated obstacle. A smooth path, $l_{FMM+\text{inflated}+\text{smooth}}$, is also shown in Fig. 13. Although it removes abrupt turn waypoints, the path is still not very smooth. As comparisons, paths planned based on the IDC-FM² method, l_1 and l_2 , are obviously smoother than $l_{FMM+\text{inflated}+\text{smooth}}$. In addition, when using the proposed rapid path planning algorithm based on two-level SR grid maps to

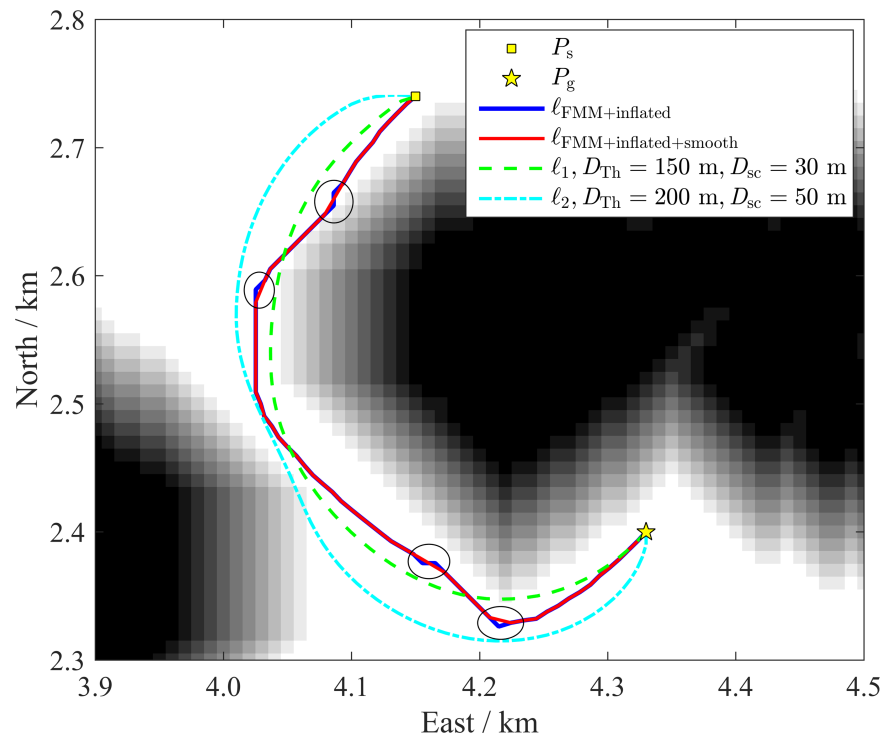


Figure 13 Paths planned by a classical approach applying the FMM within the map with inflated obstacles and the IDC-FM² method. When using the classical approach, the obstacle is inflated by 94.0 m. The black cells indicate the raw obstacle, the white cells indicate the free areas, while the cells with different gray scales indicate the inflated obstacles.

Full-size  DOI: [10.7717/peerjcs.612/fig-13](https://doi.org/10.7717/peerjcs.612/fig-13)

improve the computational efficiency, the channel will be mapped as obstacle cells when mapping the HSR grid map to the LSR grid map in this case (taking the mapping parameter $L = 8$ as shown in Table 1), because of the influence of the inflated obstacle. This unexpected mapping will result in the loss of the path through the channel.

Global path planning in a large-scale and complex multi-island environment

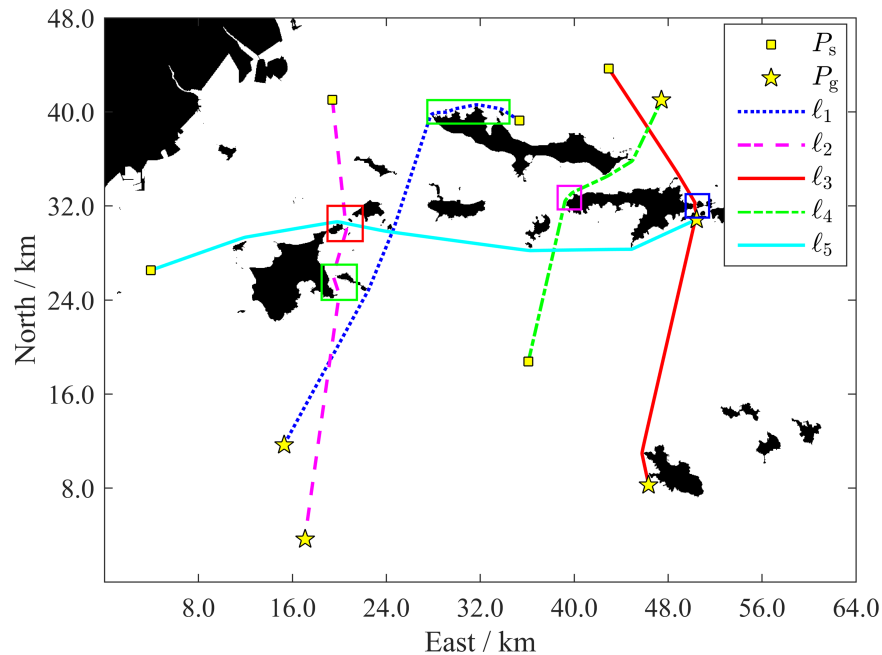
To verify the path planning ability and the computational efficiency improvement of the proposed algorithm in a large-scale and relatively complex multi-island environment, the simulation environment shown in Fig. 11B is selected, and the long-length path cases of USVs around the islands or across channels are investigated.

Path planning cases

Five typical path cases were selected to demonstrate the performance of the proposed algorithm. The P_s and P_g groups are listed in Table 3. When determining the inshore-distance parameter D_{sc} , a USV with $v_{U,max} = 6\text{m/s}$, $t_r = 2\text{s}$, $a_d = -1\text{m/s}^2$ is considered as a case. Based on Eq. (20), D_{sc} is suggested to select a value larger than 30 m. Therefore, a larger value $D_{sc} = 50\text{m}$ is used for the path planning of the selected five path cases. Another

Table 3 Quantified information of the five typical paths.

Path	P_s (km)	P_g (km)	Length (km)
l_1	[35.34, 39.25]	[15.31, 11.65]	38.84
l_2	[19.42, 41.02]	[17.10, 3.63]	37.90
l_3	[42.96, 43.67]	[46.34, 8.24]	38.20
l_4	[36.11, 18.77]	[47.44, 41.01]	26.47
l_5	[3.95, 26.52]	[50.45, 30.83]	47.82

**Figure 14** Five typical paths in the surrounding area of Changhai County. Several portions of these paths marked by rectangles are amplified to show more details in Fig. 15.

Full-size DOI: 10.7717/peerjcs.612/fig-14

inshore-distance parameter $D_{Th} = 200$ m is selected empirically. The main parameters used in these path planning cases are shown in Table 1.

As shown in Fig. 14, all paths successfully bypass the islands. The enlarged views (see Fig. 15, every path is displayed with a solid line) provide further details. Every path maintains a relatively safe distance when close to an island, and smooth when turning around the island. For a narrow channel of a certain degree (usually wider than $2D_{LRes}$), the path is planned along the quasi-midline of the channel to ensure that it is as safety as possible. The path lengths range from about 26.52 km to 47.86 km (see Table 3). These lengths can cover the range of most applications of current small-and medium-sized USVs. These results show the effective path planning ability of the proposed algorithm in a large-scale and complex multi-island environment.

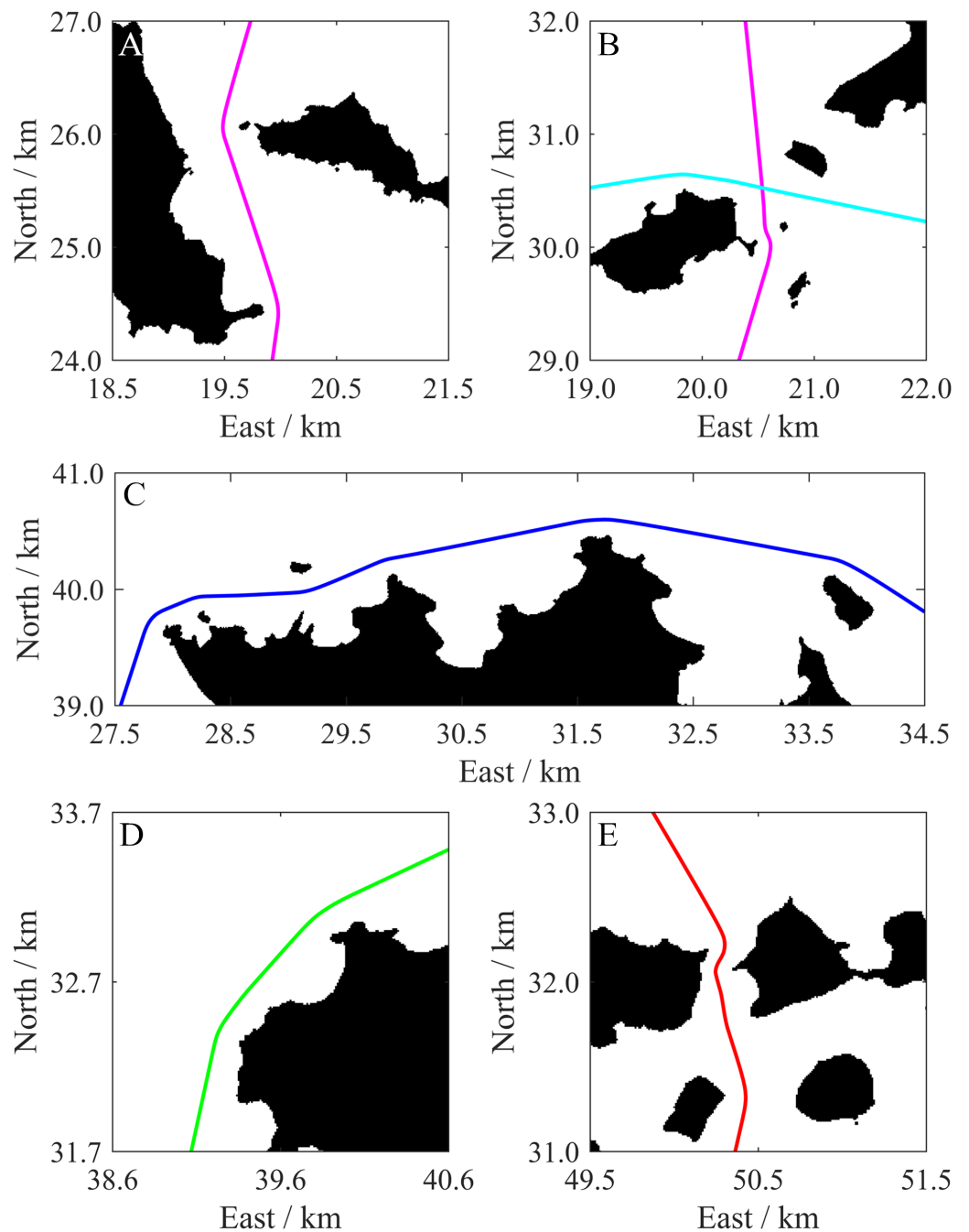


Figure 15 Enlarged views of path portions. (A)–(E) show the portions that are indicated by rectangles in Fig. 14. Detailed information can be acquired from (A) to (E). For example, all paths maintain a relatively safe distance from each island, the paths are smooth even when they turn around islands, and paths across narrow channels can be planned along the quasi-midline of the channels.

Full-size  DOI: [10.7717/peerjcs.612/fig-15](https://doi.org/10.7717/peerjcs.612/fig-15)

Table 4 Planning times of five typical path cases for the compared algorithm based on a single HSR grid map and the proposed algorithm based on two-level SR grid maps.

Path	Average planning times (s)	
	Compared algorithm	Proposed algorithm
ℓ_1	29.21	1.94
ℓ_2	24.65	1.95
ℓ_3	30.25	1.97
ℓ_4	16.44	1.76
ℓ_5	40.40	2.24

Computational efficiency improvement

To verify the computational efficiency improvement of the proposed algorithm, the time spent on the five typical paths (see Table 3) based on the proposed algorithm was calculated. For comparison, the time spent on the basic algorithm which is executed directly on a single HSR grid map by applying the IDC-FM² method was also calculated as a reference. The C algorithm code was tested on a computer with a Core i5-6300U CPU and 8G memory, which runs a 64-bit Win7 operating system.

Path planning for every typical path was repeated 10 times. Every planning time starts from the HSR grid map reading (in Steps S1 and T1) and ends when the global path has been calculated (in Steps S2 and T5). Because the path planning is performed on a Windows operating system, which involves multitasking, there may be many factors influencing the planning time, such as the variable CPU usage, random CPU hit rate to the cache, and possible thread scheduling. Therefore, the average planning time is used to evaluate the computational efficiency. The average planning time results of the five planned paths are presented in Table 4. These results indicate that the computational efficiency of the proposed algorithm based on two-level grid maps is significantly higher than the algorithm based on a single HSR grid map. The time for all cases is approximately 2 s, indicating that this method can be used in less demanding real-time planning applications. This can effectively improve the practicality of the proposed algorithm.

When planning a path by applying the basic FM²-based algorithm, if the grid map scale is very large, two aspects of calculations will severely increase compared with small-scale maps. First, as the number of cells increases significantly, the calculations for free cells increase. Second, S_T changes in every iteration of calculations. The overall scale of S_T clearly increases, resulting in the increase in sorting time of the adopted priority heap struct. This compared algorithm, whose algorithm flow is shown in Fig. 8, is used directly on a single HSR grid map. The valid scale of the HSR grid map is the number of free cells. For comparison, the proposed algorithm determines the effective regions within the HSR grid map first and then plans a path within the determined effective regions. As shown in Fig. 9, Steps T1–T3 complete the determination of effective regions, and Steps T4–T5 realize the path planning. The same function of the path planning is also realized by Steps S1–S2, as shown in Fig. 8. Comparing the path planning steps of the compared and proposed algorithms, the numbers of free cells in effective regions are much less than the ones in the HSR grid map. This indicates that the calculations for free cells and the overall

scale of \mathcal{S}_T will greatly decrease. Therefore, the planning time of the proposed algorithm reduces. Certainly, there are more steps in the proposed algorithm. Planning time spent on these steps (*Steps* T1–T3) mainly depends on the LSR mapping parameter L . When the planning time spent on the compared algorithm, t_{ref} , is large enough, the planning time spent on *Step* T2 can reduce to less than t_{ref}/L^2 (in the cases shown in [Table 4](#), this value is approximately $t_{\text{ref}}/(2L^2)$). The planning time spent on *Steps* T1 and T3 is often much less than t_{ref} . Therefore, the increased planning time spent on *Steps* T1–T3 is much less than the saved planning time spent on *Steps* T4–T5.

CONCLUSIONS

In some special USV applications such as sea rescue, it requires the USV to reach a goal position as soon as possible. FMM is a suitable global path planning method which can plan a time-optimal global path. However, the planned paths are not safe enough when they bypass obstacles, because of that they are too close to the obstacles. Therefore, FMM should be improved for safety considerations. One classical approach is applying the FMM within the map with inflated obstacles. Although the planned paths are safe by inflating the obstacle size, there may be abrupt turns when the obstacles have sharp corners, and the planned paths are influenced by shapes of inflated obstacles and may be not very smooth. Such paths are usually unfriendly for USVs. A rapid global path planning algorithm applying an IDC-FM² method in two-level SR grid maps is proposed for USV applications with short time requirements in large-scale and complex multi-island environments. This algorithm can acquire a continuous, smooth, quasi-time-optimal path while maintaining a safe distance around obstacles when bypassing obstacles. When the path is near obstacles, it is limited to a safe area determined by two inshore-distance parameters. By adjusting these two inshore-distance parameters, the path can be modified flexibly. Although the time optimality is missed by using the IDC-FM² method, the safety with a higher priority in most applications has been ensured initially while the optimal loss of time remains at a certain level. The two-path planning process based on two-level SR grid maps improves the computational efficiency compared with the basic FM²-based method. The planning time on the order of seconds is acceptable in many global path planning applications of USVs. Meanwhile, the planning time of this order of magnitude is typically short enough for USVs in most situations with the requirement to replan the path. This indicates the potential of replanning by using the proposed algorithm from the perspective of planning time. As a comparison, when using the classical approach which applies the FMM within the map with inflated obstacles, narrow channels will be easier to map as obstacles when mapping the HSR grid map to the LSR grid map in the rapid planning process based on two-level SR grid maps. This shortcoming may result in the loss of some possible paths through channels.

On the other hand, there are still some challenges when using the proposed algorithm. For example, how to accurately obtain the location information of newly detected obstacles and how to add this information into the grid map when replanning are the common challenges. The IDC-FM² method also needs to be modified to plan the path meeting the

dynamic characteristics of a USV. Otherwise, the USV may not follow the replanned path successfully at the beginning of the path.

One shortcoming of the proposed algorithm is that paths through channels that are very narrow but still passable in reality may be missed because of the first path planning process in the LSR grid map. The introduction of environmental effects into the proposed algorithm is an important task to be performed in future works. Marine experiments should also be conducted to verify the validity of the algorithm.

ACKNOWLEDGEMENTS

We would like to thank Editage for English language editing.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the National Key R&D Program of China (No. 2017YFC14052**). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
National Key R&D Program of China: 2017YFC14052**.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Dong Wang conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Jie Zhang, Deqing Liu and Xingpeng Mao analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.
- Jiucai Jin conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:
The raw data is available in the [Supplemental Files](#).

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj-cs.612#supplemental-information>.

REFERENCES

- Adalsteinsson D, Sethian JA. 1995.** A fast level set method for propagating interfaces. *Journal of Computational Physics* **118**:269–277 DOI [10.1006/jcph.1995.1098](https://doi.org/10.1006/jcph.1995.1098).
- Alvarez D, Gómez J, Garrido S, Moreno L. 2015.** 3D robot formations path planning with fast marching square. *Journal of Intelligent & Robotic Systems* **80**:507–523 DOI [10.1007/s10846-015-0187-1](https://doi.org/10.1007/s10846-015-0187-1).
- Amorim D, Ventura R. 2014.** Towards efficient path planning of a mobile robot on rough terrain. In: *2014 IEEE international conference on autonomous robot systems and competitions (ICARSC)*. Espinho, Portugal: IEEE, 22–27.
- Arzamendia M, Gregor D, Reina DG, Toral SL. 2019.** An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of Ypacarai Lake. *Soft Computing* **23**:1723–1734 DOI [10.1007/s00500-017-2895-x](https://doi.org/10.1007/s00500-017-2895-x).
- Campbell S, Naeem W, Irwin GW. 2012.** A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control* **36**:267–283 DOI [10.1016/j.arcontrol.2012.09.008](https://doi.org/10.1016/j.arcontrol.2012.09.008).
- Dijkstra EW. 1959.** Communication with an automatic computer doctor. Ph.D. thesis, University of Amsterdam.
- Garrido S, Alvarez D, Moreno LE. 2020.** Marine applications of the fast marching method. *Frontiers in Robotics and AI* **7**:1–12 DOI [10.3389/frobt.2020.00002](https://doi.org/10.3389/frobt.2020.00002).
- Garrido S, Moreno L, Blanco D, Martin F. 2007.** FM2: A real-time fast marching sensor-based motion planner. In: *2007 IEEE/ASME international conference on advanced intelligent mechatronics*. Zurich, Switzerland: IEEE, 1–6.
- Gómez JV, Alvarez D, Garrido S, Moreno L. 2019.** Fast methods for Eikonal equations: an experimental survey. *IEEE Access* **7**:39005–39029 DOI [10.1109/ACCESS.2019.2906782](https://doi.org/10.1109/ACCESS.2019.2906782).
- Gómez JV, Lumbier A, Garrido S, Moreno L. 2013.** Planning robot formations with fast marching square including uncertainty conditions. *Robotics and Autonomous Systems* **61**:137–152 DOI [10.1016/j.robot.2012.10.009](https://doi.org/10.1016/j.robot.2012.10.009).
- González V, Monje CA, Moreno L, Balaguer C. 2016.** Fast marching square method for UAVs mission planning with consideration of Dubins model constraints. *IFAC-PapersOnLine* **49**:164–169 DOI [10.1016/j.ifacol.2016.09.029](https://doi.org/10.1016/j.ifacol.2016.09.029).
- Guo J, Gao Y, Cui G. 2013.** Path planning of mobile robot based on improved potential field. *Information Technology Journal* **12**:2188–2194 DOI [10.3923/itj.2013.2188.2194](https://doi.org/10.3923/itj.2013.2188.2194).
- Hart PE, Nilsson NJ, Raphael B. 1968.** A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* **4**:100–107 DOI [10.1109/TSSC.1968.300136](https://doi.org/10.1109/TSSC.1968.300136).
- Khatib O. 1986.** Real-time obstacle avoidance for robot manipulator and mobile robots. *The International Journal of Robotics Research* **5**:90–98 DOI [10.1177/027836498600500106](https://doi.org/10.1177/027836498600500106).

- Kim H, Kim D, Shin JU, Kim H, Myung H. 2014.** Angular rate-constrained path planning algorithm for unmanned surface vehicles. *Ocean Engineering* **84**:37–44 DOI [10.1016/j.oceaneng.2014.03.034](https://doi.org/10.1016/j.oceaneng.2014.03.034).
- Kim H, Kim SH, Jeon M, Kim J, Song S, Paik KJ. 2017.** A study on path optimization method of an unmanned surface vehicle under environmental loads using genetic algorithm. *Ocean Engineering* **142**:616–624 DOI [10.1016/j.oceaneng.2017.07.040](https://doi.org/10.1016/j.oceaneng.2017.07.040).
- Lee T, Kim H, Chung H, Bang Y, Myung H. 2015.** Energy efficient path planning for a marine surface vehicle considering heading angle. *Ocean Engineering* **107**:118–131 DOI [10.1016/j.oceaneng.2015.07.030](https://doi.org/10.1016/j.oceaneng.2015.07.030).
- Lin Q. 2003.** Enhancement, extraction, and visualization of 3D volume data doctor. Ph.D. thesis, Linköping University.
- Liu Y, Bucknall R. 2015.** Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering* **97**:126–144 DOI [10.1016/j.oceaneng.2015.01.008](https://doi.org/10.1016/j.oceaneng.2015.01.008).
- Liu Y, Bucknall R. 2016.** The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method. *Applied Ocean Research* **59**:327–344 DOI [10.1016/j.apor.2016.06.013](https://doi.org/10.1016/j.apor.2016.06.013).
- Liu Y, Bucknall R, Zhang X. 2017.** The fast marching method based intelligent navigation of an unmanned surface vehicle. *Ocean Engineering* **142**:363–376 DOI [10.1016/j.oceaneng.2017.07.021](https://doi.org/10.1016/j.oceaneng.2017.07.021).
- Liu Z, Zhang Y, Yu X, Yuan C. 2016.** Unmanned surface vehicles: an overview of developments and challenges. *Annual Reviews in Control* **41**:71–93 DOI [10.1016/j.arcontrol.2016.04.018](https://doi.org/10.1016/j.arcontrol.2016.04.018).
- Messias J, Ventura R, Lima P, Sequeira J, Alvito P, Marques C, Carriço P. 2014.** A robotic platform for edutainment activities in a pediatric hospital. In: *2014 IEEE international conference on autonomous robot systems and competitions (ICARSC)*. Espinho, Portugal: IEEE, 193–198.
- Naeem W, Irwin GW, Yang A. 2012.** COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* **22**:669–678 DOI [10.1016/j.mechatronics.2011.09.012](https://doi.org/10.1016/j.mechatronics.2011.09.012).
- Sethian JA. 1999.** *Level set methods and fast marching methods*. Cambridge: Cambridge University Press.
- Shi B, Su Y, Zhang H, Liu J, Wan L. 2018.** Obstacles modeling method in cluttered environments using satellite images and its application to path planning for USV. *International Journal of Naval Architecture and Ocean Engineering* **11**:202–210 DOI [10.1016/j.ijnaoe.2018.04.001](https://doi.org/10.1016/j.ijnaoe.2018.04.001).
- Singh Y, Sharma S, Sutton R, Hatton D. 2018.** Towards use of Dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field. *TransNav the International Journal on Marine Navigation and Safety of Sea Transportation* **12**:125–131 DOI [10.12716/1001.12.01.14](https://doi.org/10.12716/1001.12.01.14).

- Song C.** 2014. Global path planning method for USV system based on improved ant colony algorithm. *Applied Mechanics and Materials* **568–570**:785–788 DOI [10.4028/www.scientific.net/AMM.568-570.785](https://doi.org/10.4028/www.scientific.net/AMM.568-570.785).
- Song J, Hao C, Su J.** 2020. Path planning for unmanned surface vehicle based on predictive artificial potential field. *International Journal of Advanced Robotic Systems* **17**:1–13 DOI [10.1177/1729881420918461](https://doi.org/10.1177/1729881420918461).
- Song R, Liu Y, Bucknall R.** 2017. A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment. *Ocean Engineering* **129**:301–317 DOI [10.1016/j.oceaneng.2016.11.009](https://doi.org/10.1016/j.oceaneng.2016.11.009).
- Song L, Mao Y, Xiang Z, Zhou Y, Du K.** 2015. A study on path planning algorithms based upon particle swarm optimization. *Journal of Information & Computational Science* **12**:673–680 DOI [10.12733/jics20105310](https://doi.org/10.12733/jics20105310).
- Tan G, Zou J, Zhuang J, Wan L, Sun H, Sun Z.** 2020. Fast marching square method based intelligent navigation of the unmanned surface vehicle swarm in restricted waters. *Applied Ocean Research* **95**:102018 DOI [10.1016/j.apor.2019.102018](https://doi.org/10.1016/j.apor.2019.102018).
- Tsitsiklis JN.** 1995. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* **40**:1528–1538 DOI [10.1109/9.412624](https://doi.org/10.1109/9.412624).
- Wang N, Jin X, Er J.** 2019. A multilayer path planner for a USV under complex marine environments. *Ocean Engineering* **184**:1–10 DOI [10.1016/j.oceaneng.2019.05.017](https://doi.org/10.1016/j.oceaneng.2019.05.017).
- 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* DOI [10.1007/s40815-020-00912-y](https://doi.org/10.1007/s40815-020-00912-y).
- Xia G, Han Z, Zhao B, Liu C, Wang X.** 2019. Global path planning for unmanned surface vehicle based on improved quantum ant colony algorithm. *Mathematical Problems in Engineering* **2019**:1–10 DOI [10.1155/2019/2902170](https://doi.org/10.1155/2019/2902170).
- Xie S, Wu P, Liu H, Yan P, Li X, Luo J, Li Q.** 2016. A novel method of unmanned surface vehicle autonomous cruise. *Industrial Robot: An International Journal* **43**:121–130 DOI [10.1108/IR-05-2015-0097](https://doi.org/10.1108/IR-05-2015-0097).
- Yan R, Pang S, Sun H, Pang Y.** 2010. Development and missions of unmanned surface vehicle. *Journal of Marine Science and Application* **9**:451–457 DOI [10.1007/s11804-010-1033-2](https://doi.org/10.1007/s11804-010-1033-2).
- Zhang W, Xu Y, Xie J.** 2019. Path planning of USV based on improved hybrid genetic algorithm. In: *2019 European navigation conference (ENC)*. Warsaw, Poland: IEEE, 1–7.