


Article

GIS-Data-Driven Efficient and Safe Path Planning for Autonomous Ships in Maritime Transportation

Xiao Hu ¹, Kai Hu ¹, Datian Tao ², Yi Zhong ^{2,*} and Yi Han ^{2,*} ¹ China Ship Development and Design Center, Wuhan 430064, China² School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; taodatian@whut.edu.cn

* Correspondence: zhongyi@whut.edu.cn (Y.Z.); hanyi@whut.edu.cn (Y.H.); Tel.: +86-27-8785-8005 (Y.Z.)

Abstract: Maritime transportation is vital to the global economy. With the increased operating and labor costs of maritime transportation, autonomous shipping has attracted much attention in both industry and academia. Autonomous shipping can not only reduce the marine accidents caused by human factors but also save labor costs. Path planning is one of the key technologies to enable the autonomy of ships. However, mainstream ship path planning focuses on searching for the shortest path and controlling the vehicle in order to track it. Such path planning methods may lead to a dynamically infeasible trajectory that fails to avoid obstacles or reduces fuel efficiency. This paper presents a data-driven, efficient, and safe path planning (ESP) method that considers ship dynamics to provide a real-time optimal trajectory generation. The optimization objectives include fuel consumption and trajectory smoothness. Furthermore, ESP is capable of fast replanning when encountering obstacles. ESP consists of three components: (1) A path search method that finds an optimal search path with the minimum number of sharp turns from the geographic data collected by the geographic information system (GIS); (2) a minimum-snap trajectory optimization formulation with dynamic ship constraints to provide a smooth and collision-free trajectory with minimal fuel consumption; (3) a local trajectory replanner based on B-spline to avoid unexpected obstacles in real time. We evaluate the performance of ESP by data-driven simulations. The geographical data have been collected and updated from GIS. The results show that ESP can plan a global trajectory with safety, minimal turning points, and minimal fuel consumption based on the maritime information provided by nautical charts. With the long-range perception of onboard radars, the ship can avoid unexpected obstacles in real time on the planned global course.

Keywords: kinematics; improved A* algorithm; path planning; GIS

Citation: Hu, X.; Hu, K.; Tao, D.; Zhong, Y.; Han, Y. GIS-Data-Driven Efficient and Safe Path Planning for Autonomous Ships in Maritime Transportation. *Electronics* **2023**, *12*, 2206. <https://doi.org/10.3390/electronics12102206>

Academic Editor: Felipe Jiménez

Received: 25 March 2023

Revised: 8 May 2023

Accepted: 10 May 2023

Published: 12 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of the global economy, according to the survey of the Baltic and International Maritime Council/International Chamber Shipping (BIMCO/ICS), the maritime industry has accounted for 80% of the world's trade and transportation [1]. Thus, the safety and efficiency of maritime transportation are of paramount importance. Current issues of maritime transportation include: (1) about 75–96% marine vessel accidents being caused by humans; (2) a severe shortage of seafarers and management personnel; (3) more than 80% of shipping costs being from fuel and labor [2]. Autonomous navigation is vital to mitigate the above issues for ships in that it can be more vigilant than humans at avoiding accidents by perceptions from heterogeneous sensors such as a camera, laser scanner, and mmWave radar [3]. Ship autonomy not only saves human labor costs but also utilizes intelligent path planning methods to achieve optimized fuel consumptions.

Realizing autonomous ships requires localization and path planning. Currently, the global positioning system (GPS) and compass have been commonly available to provide reliable location services. In contrast, path planning for ships still poses several challenges.

Existing works mainly focus on the path planning for lightweight surface vessels, which are agile and applicable to harbor patrol, marine resource exploration, etc. [4]. Ships, however, exhibit high inertia and thus a significant delay in motion control. When encountering sudden situations, e.g., encountering a large iceberg, its dynamic nature prevents agile avoidance [5]. In addition to the “shortest path”, dynamical feasibility is crucial for ship path planning [6].

Existing ship path planners are typically optimal path searching methods based on A* and its modifications [7]. Although they provide optimal paths in a given map [8], the optimality is limited in ideal maps including the occupancy grid map, Voronoi-visibility roadmap [9], risk contour map [10], etc. Their path searching is conducted by discretized heading directions without considering the ships’ dynamical constraints, making the planned path dynamically infeasible. More recent works [11,12] have taken the dynamical constraints into consideration. However, their iterative methods have high computational complexities, failing to plan in real time to avoid expected sudden risks. Researchers also proposed hybrid approaches that fuse the artificial potential field (APF) algorithm with velocity odometry and path optimization [13–15] to achieve real-time obstacle avoidance in complex maritime environments. However, the APF causes oscillations when searching for paths through narrow areas, causing frequent turning and increasing fuel consumptions and navigation risks. In addition to optimization-based methods, researchers incorporate reinforcement learning into path planning [16,17]. However, learning-based methods suffer from a trade-off between generality and accuracy. Their stochastic results cannot guarantee the safety and efficiency of ship navigation. In summary, none of the existing path planning methods meet the safety and efficiency needs when considering ship dynamics.

This paper presents ESP, a combinatorial optimized path planning approach that generates a safe, smooth, and dynamically feasible trajectory while minimizing the shipping cost. Realizing such an elegant approach poses several challenges: (1) to quantify the turning cost in optimal and dynamically feasible path searching; (2) to minimize the shipping cost in terms of fuel by formulating a minimum-snap problem, which is non-trivial in combining the dynamic model of ships; (3) to cope with sudden risks, e.g., avoid expected obstacles or enemy vessels, which requires replanning a smooth, safe, feasible and optimized path in real time.

To address the above challenges, ESP consists of three components. First, we propose A-turning, a path searching algorithm that quantifies the turning cost in order to obtain the optimal path with fewer turns. Then, we formulate the minimum-snap optimization problem subject to the dynamic constraints of ships to achieve the minimum shipping cost in terms of fuel. Finally, we propose a real-time path replanning algorithm using quasi-uniform cubic B-spline, achieving millisecond-level path replanning to cope with sudden risks.

In summary, the contributions of this paper include: (1) quantifying the turning cost and incorporating it into an optimal global path search through a modified A* algorithm; (2) formulating a minimum-snap optimization problem to generate a smooth trajectory that consumes the least fuel and satisfies the ship’s dynamic constraints; (3) enabling real-time obstacle avoidance for ships through a B-spline-based local trajectory replanner.

ESP is evaluated in a data-driven simulator implemented by MATLAB and our developed geographic information system (GIS). The simulation results demonstrate the effectiveness of ESP in generating a safe, smooth, and feasible path with minimal turns and fuel consumption. Moreover, ESP enables a quick reaction for ships to smoothly avoid unexpected obstacles by path replanning in less than 48 ms.

The rest of this paper is organized as follows. Section 2 reviews related works. Then, we elaborate on the design of ESP in Section 3. The performance evaluation in Section 4 demonstrates the effectiveness of ESP. Section 5 concludes this paper.

2. Related Works

Path planning can be divided into two steps: path searching and path optimization. Path searching involves searching for an obstacle-free path from the start to the end. Path optimization involves optimizing the searched path to meet users' specific objectives, e.g., the shortest sailing distance, minimum fuel consumption, and minimum shipping cost.

Path searching has been well-studied for decades. It can be categorized by graph-search-based and random-sampling-based path searching approaches. Graph-search-based path planning methods follow a set of steps to generate unique navigation paths. The classic algorithm, Dijkstra, expands a large number of irrelevant nodes during searching, which greatly slows down the searching process. In order to improve the searching efficiency, A*-family algorithms have been proposed. They make the searching process more purposeful to the destination by introducing heuristic functions [18–20]. These heuristic functions treat vessels as a mass point with unlimited turning and sailing speeds. Their results may have large-angle steers between consecutive path segments. However, to the best of our knowledge, the maximum speed of a ship (displacement > 320 t) is 15 knots, the maximum acceleration is 1, and the turning radius is three times the ship length. Simply considering the ship as a mass point leads to infeasible path planning, making the above heuristic solutions impractical. In addition, Yu et al. [21] proposed an A* algorithm with velocity variation and global optimization (A*-VVGO), which achieves the purpose of obstacle avoidance by changing the speed of the ship, and combines the artificial potential field method to ensure the smoothness of the path. Sang et al. [22] proposed a hybrid algorithm of an artificial potential field based on A* and local programming, which is often combined with many algorithms, such as the genetic algorithm (GA) [23], Fuzzy artificial potential field (FAPF) [24], etc. These hybrid algorithms contain various advantages. However, these methods do not consider vehicles' dynamic constraints. Tracking the paths cannot guarantee safety and smoothness. Moreover, graph-search-based methods cannot work efficiently in large environments due to the searching space being exponential to the size of the occupancy grid maps.

To address the searching efficiency problem with respect to the occupancy grid maps, random-sampling-based algorithms have been proposed to incrementally build maps by sampling. They can work in the planning of the ocean. Zhang et al. [25] proposed the adaptive hybrid dynamic step size and target attractive force-RRT (AHDSTAF-RRT), imposing the dynamic constraints of unmanned surface vehicles (USVs) to allow USVs to navigate complex aquatic environments. Webb et al. [26] proposed Kinodynamic RRT*, achieving asymptotically optimal motion planning for robots. However, these approaches suffer from slow convergence and inflexible settings of step size. Thus, Strub et al. [27] designed a heuristic function in the exploitation of random sampling with the aim that the new samples would be more likely to be closer to the destination. Xu et al. [28] proposed a simplified map-based regional sampling RRT* (SMRS-RRT*) algorithm to achieve path planning in complex environments. Dong et al. [29] proposed a path planning method based on improved RRT*-Smart, which optimizes the node sampling method by sampling in the polar coordinate system with the origin of USV, improves the search efficiency, and ensures that the navigation path follows the International Regulation for Preventing Collision at Sea. This design does not only improve the convergence speed but also improves the quality of the solution. Nevertheless, random-sampling-based methods cannot provide optimal solutions. Their results are not unique. The searched path usually contains many sharp turns, which is especially evident in open water.

Based on the path searching from graph-search-based and random-sampling-based methods, researchers tried to generate smooth trajectories. A strawman option is to use interpolation. Liang et al. [30] interpolated the trajectory with the Dobbins curve to ensure the smoothness and reduce the number of sharp turns, but the trajectory curvature was discontinuous. To solve this problem, Candeloro et al. [31] used the Fermat spiral to connect the straight line segment with the curved segment, generating the trajectory with a continuous curvature. Wang et al. [32] used B-spline interpolation to construct smooth

trajectories with sparse waypoints. It, however, does not impose the vehicle's dynamic constraints, making trajectories infeasible to be executed. To generate dynamic feasible trajectories, control-space sampling approaches [5,6,33] are simple and effective. However, such approaches lack purpose, so their sampling process could take too much time and fail to plan paths in real time. MahmoudZadeh et al. [34] combined a novel B-spline data frame and the particle swarm optimization (PSO) algorithm to establish a continuous route planning system to achieve route planning for USV ocean sampling missions. Zheng et al. [35] proposed a ship collision avoidance decision method based on improved cultural particle swarm to achieve the steering collision avoidance of a ship, but without considering the speed constraint of the ship.

3. Methods

3.1. Problem Formulation

The obstacles considered in this paper are the static obstacles in the chart and the unexpected static obstacles that appear within the detection range of the ship's radar during the actual navigation of the ship. One primary objective of our path planning is to be collision-free. Additionally, we optimize two more objectives: best stability and minimum fuel consumption. The specific objective function and constraints are given in the following subsections.

3.2. Kinematic Model

The previous studies often ignored the influence of marine environments on the ship's motion state for the ease of modeling. In order to make the planned path fit the actual sailing situation, this paper establishes the kinematics model of ships considering the ocean current.

Figure 1 illustrates the kinematic model of a ship. $O_e X_e Y_e$ denotes the world frame, which refers to the coordinate system with respect to the earth. The earth's gravity points to the positive direction of the z-axis. The x–y–z axes follow the right-hand rule. The origin of the world frame O_e is the geometrical center's initial position, the positive direction of the $O_e X_e$ axis points to east, and the positive direction of $O_e Y_e$ points to north. $O_b X_b Y_b$ denotes the local frame, which refers to the ship's body frame, O_b is used as the center of gravity of the ship, the positive direction of the $O_b X_b$ axis points to the bow, and the positive direction of the $O_b Y_b$ axis points to the port side. Ψ denotes the yaw angle, u the surge velocity, v the sway velocity, and δ the rudder angle. According to Newton's second law, considering surge, sway, and yaw, the force at the center of gravity of the ship is

$$\begin{cases} X_e = m\ddot{x} \\ Y_e = m\ddot{y} \\ N_r = I_Z\ddot{\Psi} \end{cases} \quad (1)$$

$$I_Z = \int_V (x^2 + y^2) \rho_m dV \quad (2)$$

where X_e denotes the force along the x-axis, Y_e the force along the y-axis, x , y the position of the ship's center of gravity in the world frame, m the mass of the ship, N_r the force along the z-axis, $\ddot{\Psi}$ the angular acceleration, and I_Z the moment of inertia around the z-axis. As shown in Equation (2), it depends on the volume of the ship V and the mass density ρ_m . With the yaw Ψ , we express the transformation between the world frame and the local frame as

$$\begin{bmatrix} X_b \\ Y_b \end{bmatrix} = \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix} \begin{bmatrix} X_e \\ Y_e \end{bmatrix} \quad (3)$$

Then, the forces on the surge and sway directions can be expressed as

$$\begin{cases} X_b = m(\dot{u} - vr) \\ Y_b = m(\dot{v} + ur) \end{cases} \quad (4)$$

where r denotes the yaw rate, and \dot{u} and \dot{v} denote the acceleration on the surge and sway directions, respectively. From Equations (3) and (4), we obtain the kinematic model as follows.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \quad (5)$$

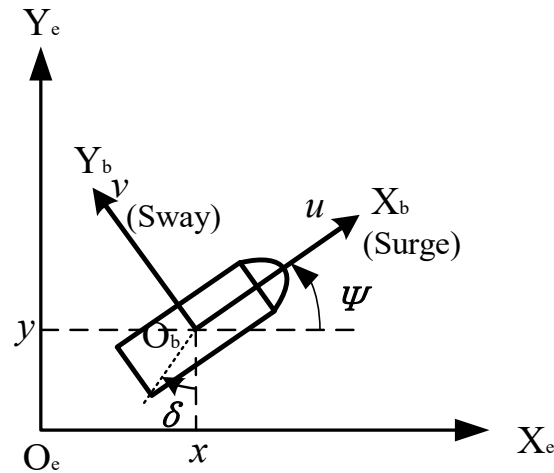


Figure 1. The kinematic model of a ship.

3.3. Dynamic Model

This paper uses the first-order K–T model to represent the hydrodynamic model of the ship, assuming that the port and starboard sides of the ship are symmetrical and the ship mass is uniformly distributed. The hydrodynamic equation can be expressed as:

$$M\dot{v} + Cv + Dv = \tau \quad (6)$$

where $v = [u, v, r]^T$,

$$M = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 \\ 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} \quad (7)$$

$$C = \begin{bmatrix} 0 & 0 & -(m - Y_{\dot{v}})v \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 \end{bmatrix} \quad (8)$$

$$D = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & 0 \\ 0 & 0 & -N_r \end{bmatrix} \quad (9)$$

$$\tau = \tau_E + \tau_r \quad (10)$$

where M denotes the inertial mass matrix, C the Coriolis centripetal force matrix, and D the drag coefficient matrix. X_u and Y_v denote the derivatives for the hydrodynamic, $X_{\dot{u}} = \frac{\partial X}{\partial \dot{u}}$, $Y_{\dot{v}} = \frac{\partial Y}{\partial \dot{v}}$, and $N_{\dot{r}} = \frac{\partial N}{\partial \dot{r}}$. τ_E denotes the force imposed by the environment and τ_r denotes the thrust of the propeller.

3.4. Ocean Circulation Model

Affected by the environment such as sea wind and ocean currents, a ship easily deviates from its course or even capsizes during sailing, resulting in property damage and

even casualties. Therefore, we consider the influence of ocean currents on the ship's motion state when planning its path.

Ocean currents are formed when seawater flows in a certain direction at a regular, relatively steady speed. It is a large-scale, aperiodic form of seawater movement. According to the characteristics of its location and time, ocean currents can be divided into uniform currents, non-uniform currents, steady currents, and unsteady currents. In offshore or seabed areas with irregular topography, the model of ocean currents is more complicated. To simplify the modeling, we assume that ocean currents are constant and uniform. Let V_c denote the ocean current speed and Ψ_c the direction of the current. Then, the velocity of the ocean currents can be expressed as

$$\mathbf{v}_c = [V_c \cos \Psi_c \quad V_c \sin \Psi_c]^T \quad (11)$$

Affected by ocean currents, the actual velocity of the ship is different from its velocity in still water. At this time $\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$, where \mathbf{v}_r is the velocity of the ship relative to the ocean current.

3.5. Optimization Objectives

When a ship sails along a trajectory, the collision-free cost function is

$$f_c = - \sum_{i=0}^n \text{Dis}(\text{Obstacle}(p_i)) \quad (12)$$

where $\text{Dis}(\text{Obstacle}(p_i))$ denotes the minimum distance from a waypoint p_i to the obstacles, which can be obtained by the Euclidean signed distance field (ESDF) [36]. The distance will be negative if a waypoint is within an obstacle.

The smoothness is determined by the sum of snaps along the trajectory. The smoothness cost can be defined as

$$f_s = \int_0^T (p^{(4)}(t))^2 dt \quad (13)$$

where $p^{(4)}(t)$ denotes the fourth-order derivative, i.e., jerk, at time t .

The fuel consumption depends on the sailing speed. We use the exponential distribution model proposed in [37] as follows to describe the relationship between fuel consumption and speed.

$$FCPH = 0.128e^{0.243V} \quad (14)$$

where $V = \sqrt{v^2 + u^2}$. Thus, we define the cost function of fuel consumption as

$$f_o = \sum_{i=1}^{n-1} t_i \cdot FCPH_i \quad (15)$$

where t_i denotes the time duration between waypoint p_i and p_{i+1} and $FCPH_i$ the fuel consumption per hour between waypoint p_i and p_{i+1} .

Combining the collision-free cost, the smoothness cost, and the fuel consumption cost, we obtain the overall optimization objective function

$$F = \min\{f_c + f_s + f_o\} \quad (16)$$

subject to

$$|u| \leq u_{max} \quad (17)$$

$$|v| \leq v_{max} \quad (18)$$

$$|r| \leq r_{max} \quad (19)$$

where u_{max} , v_{max} , and r_{max} are the maximum surge velocity, the maximum sway velocity, and the maximum yaw rate, respectively.

3.6. Occupancy Grid Map Construction

In order to apply the environmental information provided by the electronic chart for path planning, it is necessary to process the chart into a binary occupancy grid map as shown in Figure 2. In this process, we first set an appropriate binarization threshold that converts an RGB image into a binary image. Such a threshold [38] is vital to constructing an accurate grid map that ensures the feasibility of path planning. If the threshold is inappropriate, as shown in Figure 2c, then a shoal is identified as a passable area, greatly increasing the navigation risk. Second, the grid size determines the resolution of path planning. Too large a grid cannot capture the subtle details of environments, e.g., small obstacles, resulting in unsafe path searching. On the other hand, too small a grid greatly increases the search space, reducing the computation speed.

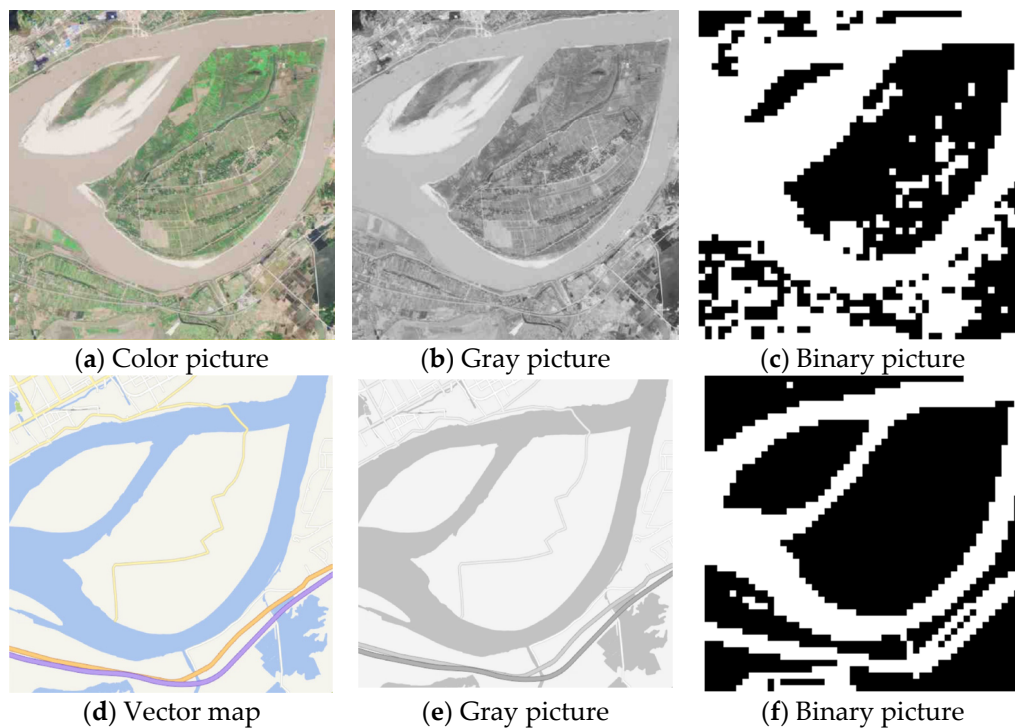


Figure 2. Converting an electronic chart into an occupancy grid map, where the black areas in the grid map represent obstacles and the white areas represent passable areas.

Considering the maneuverability of a ship, this paper chooses the minimum turning radius as the criterion to measure the grid size. The ship's minimum turning radius can be measured through the ship's maneuverability experiment. The turning radius depends on the sailing speed and water flow velocity. According to [39], in our simulation, we set the minimum turning radius of the ship as three times the length of the ship. Finally, we map the geographic chart represented in terms of longitude and latitude to the grid map using the following equation:

$$\begin{cases} lon_{(i,j)} = tlLon + \frac{|brLon - tlLon|}{w} \cdot (i - 1) \\ lat_{(i,j)} = brLat + \frac{|tlLat - brLat|}{h} \cdot (j - 1) \end{cases} \quad (20)$$

where $lon_{(i,j)}$ and $lat_{(i,j)}$ denote the longitude and latitude of position (i, j) . $tlLon$ and $tlLat$ denote the longitude and latitude at the top-left corner of the selected area. $brLon$ and

$brLat$ denote the longitude and latitude at the bottom-right corner of the selected area. w and h denote the width and height of the electronic chart.

In our simulated forward exploration, we use eight discretized directions in the grid map to search for paths as shown in Figure 3.

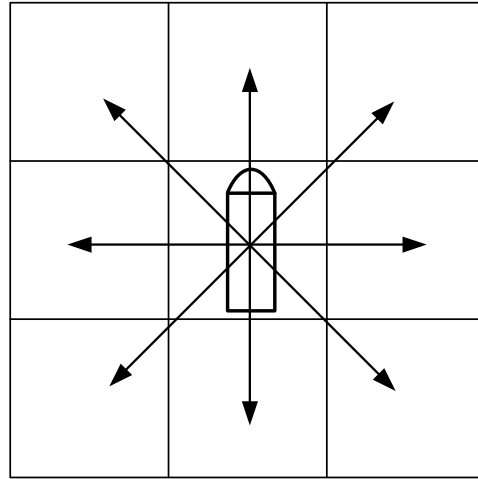


Figure 3. 8 discretized directions in path searching for autonomous ships.

3.7. Quantification of Turning Cost

The A* algorithm uses Equation (18) as the evaluation function to obtain a path with the shortest distance.

$$f(N_i) = g(N_i) + h(N_i) \quad (21)$$

where $f(N_i)$ denotes the estimated cost from the starting point to the target point, $g(N_i)$ the actual cost from the initial node to node N_i , and $h(N_i)$ the estimated cost of the best path from node N_i to the target node.

However, in path searching for ships, the turning is much more difficult than for cars or aerial vehicles. Thus, we have to add the cost of turning in order to better evaluate the path. Here we use the diagonal distance to compute the turning cost as shown in Figure 4.

The yaw ϕ_i between waypoint p_i and p_{i+1} can be computed as follows.

$$\phi_i = \arctan\left(\left|\frac{p_{i+1}(y) - p_i(y)}{p_{i+1}(x) - p_i(x)}\right|\right) \quad (22)$$

where $p_i(x), p_i(y)$ denote the coordinates of waypoint p_i , $p_{i+1}(x), p_{i+1}(y)$ the coordinates of waypoint p_{i+1} , $p_{i-1}(x), p_{i-1}(y)$ the coordinates of waypoint p_{i-1} . Preventing collisions is still of the highest priority in path planning. Thus, it is not reasonable to simply pursue the minimum turning cost in planning. We add a penalty to the turning cost:

$$c(N_i) = \varepsilon \cdot \max(0, \Delta\phi_i - \phi) \quad (23)$$

where ϕ is the penalty threshold of the yaw and ε the penalty coefficient. Empirically, we set $\phi = 30^\circ$ and $\varepsilon = 0.8$. $\Delta\phi_i$ is computed as follows.

$$\Delta\phi_i = \left| \arctan\left(\frac{p_{i+1}(y) - p_i(y)}{p_{i+1}(x) - p_i(x)}\right) - \arctan\left(\frac{p_i(y) - p_{i-1}(y)}{p_i(x) - p_{i-1}(x)}\right) \right| \quad (24)$$

Finally, the new evaluation function of our path searching algorithm is defined as:

$$f(N_i) = g(N_i) + h(N_i) + c(N_i) \quad (25)$$

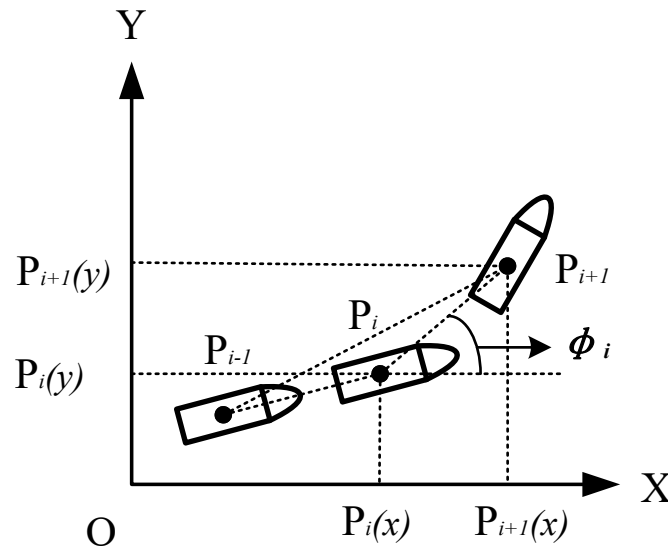


Figure 4. An illustration of how to compute the turning cost.

The improved A* algorithm is shown in Algorithm 1.

Algorithm 1. Improved A* algorithm

Input: Start node X_{start} , end node X_{end}
 1: OPEN_list := X_{start} , where $f(X_{start}) = h(X_{start})$
 2: CLOSE_list := { }
 3: **while** OPEN_list is not empty **do**
 4: current node X_n := the node in the OPEN_list with the lowest $f(X)$
 5: **if** $X_n = X_{end}$ **break**
 6: Remove X_n from OPEN_list and add it to CLOSE_list
 7: **for each** adjacent node, X_i of X_n **do**
 8: **if** $f(X_i) = 0 \parallel X_i \in \text{CLOSE_list}$ **continue**
 9: **if** $X_i \notin \text{OPEN_list}$
 10: add X_i into OPEN_list
 11: the parent node of X_i =, $X_{i>parent} = X_n$
 12: calculate $f(X_i)$, $g(X_i)$, $h(X_i)$ and $c(X_i)$
 13: **if** $X_i \in \text{OPEN_list}$
 14: calculate $f(X_i)$ via (25)
 15: Resort and keep OPEN_list sorted by f value
 16: $X_p = X_{end}$
 17: Path_list := X_p
 18: **while** $X_p \neq X_{start}$ **do**
 19: $X_p = X_{p.parent}$
 20: Path_list = {Path_list, X_p }
Return: Path_list

3.8. Global Trajectory Optimization

The previous path searching gives us a discrete path with the minimum cost. However, the path does not consider the dynamic feasibility with respect to time, velocity, and acceleration. This part requires a further step to optimize the searched path into a smooth trajectory. The trajectory can be defined as a n -order polynomial.

$$p(t) = p_0 + p_1t + p_2t^2 + \dots + p_nt^n = \sum_{i=0}^n p_it^i \quad (26)$$

where p_0, p_1, \dots, p_n are the coefficients of this trajectory. We denote $P = [p_0, p_1, \dots, p_n]^T$, and then Equation (23) can be rewritten as

$$p = [1, t, t^2, \dots, t^n] \cdot P \quad (27)$$

Then, Equation (14) can be expressed as

$$\begin{aligned} & \int_0^T (p^{(4)}(t))^2 dt \\ &= \sum_{i=1}^k \int_{t_{i-1}}^{t_i} (p^{(4)}(t))^2 dt \\ &= \sum_{i=1}^k \int_{t_{i-1}}^{t_i} \left([0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4}] \cdot p \right)^T \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] \cdot p dt \\ &= \sum_{i=1}^k p^T \int_{t_{i-1}}^{t_i} \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right]^T \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] dt \cdot p \quad (28) \end{aligned}$$

Let

$$Q_i = \int_{t_{i-1}}^{t_i} \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right]^T \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] dt \quad (29)$$

We have

$$\int_0^T (p^{(4)}(t))^2 dt = \sum_{i=1}^k p^T Q_i p \quad (30)$$

However, the polynomial expression cannot explicitly control the shape of the trajectory. To gain better control, we choose the Bezier curve using Bernstein polynomials. The k -th segment of the trajectory can be expressed as

$$B_k(t) = \sum_{i=0}^n c_k^i b_n^k(t) \quad (31)$$

where $b_n^k(t) = \binom{n}{k} \cdot t^k \cdot (1-t)^{n-k}$, $t \in [0,1]$ c_k^i denotes the control point of the k -th segment of the Bezier curve.

Since the trajectory must pass through the first and last control points, it can satisfy the positional constraints of the initial and final states. In addition, based on the hodograph of the Bezier curve, we impose constraints on the velocity and acceleration of the trajectory, ensuring the multi-order continuity of the trajectory.

3.9. Real-Time Obstacle Avoidance

In a static chart, a ship can navigate safely along the aforementioned global trajectory. However, the marine environment is complex and changeable. Ships need to deal with unexpected risks when sailing, e.g., avoiding islands and reefs. As shown in Figure 5, if the ship maintains the planned global trajectory, it will collide with a temporary obstacle. To address this issue, we perform local path planning based on the B-spline curve. The advantage of the B-spline trajectory is that it can change the curve locally by adjusting few control points, while any control point of a Bezier curve will change the shape of the whole trajectory. Moreover, it guarantees that the locally replanned trajectory still satisfies the ship's kinematic and dynamic constraints. This not only achieves the goal of real-time obstacle avoidance, but also satisfies all optimization objectives.

4. Simulation and Results

We conduct the evaluation of ESP by implementing a data-driven simulator in MATLAB. All simulation experiments are run on a quad-core 2.40 GHz Intel i5-1135G7 processor and 16 GB RAM. We input the data from the database of our developed GIS. The data include image and vector maps, longitude and latitude coordinates, and ship route information. During the simulation, we use environmental data such as shoals and whirlpools that are not currently marked in the GIS database to evaluate the effectiveness of ESP. The simulated settings are listed in Table 1. Specifically, the ship length is 30 m, the maximum turning radius is 3 times the ship length. The maximum velocity is 7.7 m/s. The maximum acceleration and jerk are 1 m/s^2 and 10 m/s^3 [42], respectively.

Table 1. Simulation parameters.

| Parameter | Value |
|----------------------|--------------------|
| Length of the ship | 30 m |
| Maximum velocity | 7.7 m/s |
| Maximum acceleration | 1 m/s^2 |
| Maximum jerk | 10 m/s^3 |

We simulate ESP in two scenes using the GCJ-02 coordinate system. In both scenes, the velocities and accelerations at the start and the end are 0. The simulated occupancy grid map is of size 50×50 . We compare the path searching results of ESP with the A* algorithm as shown in Figure 7. The results show that ESP effectively reduces the number of turning points and the planned path is safe.

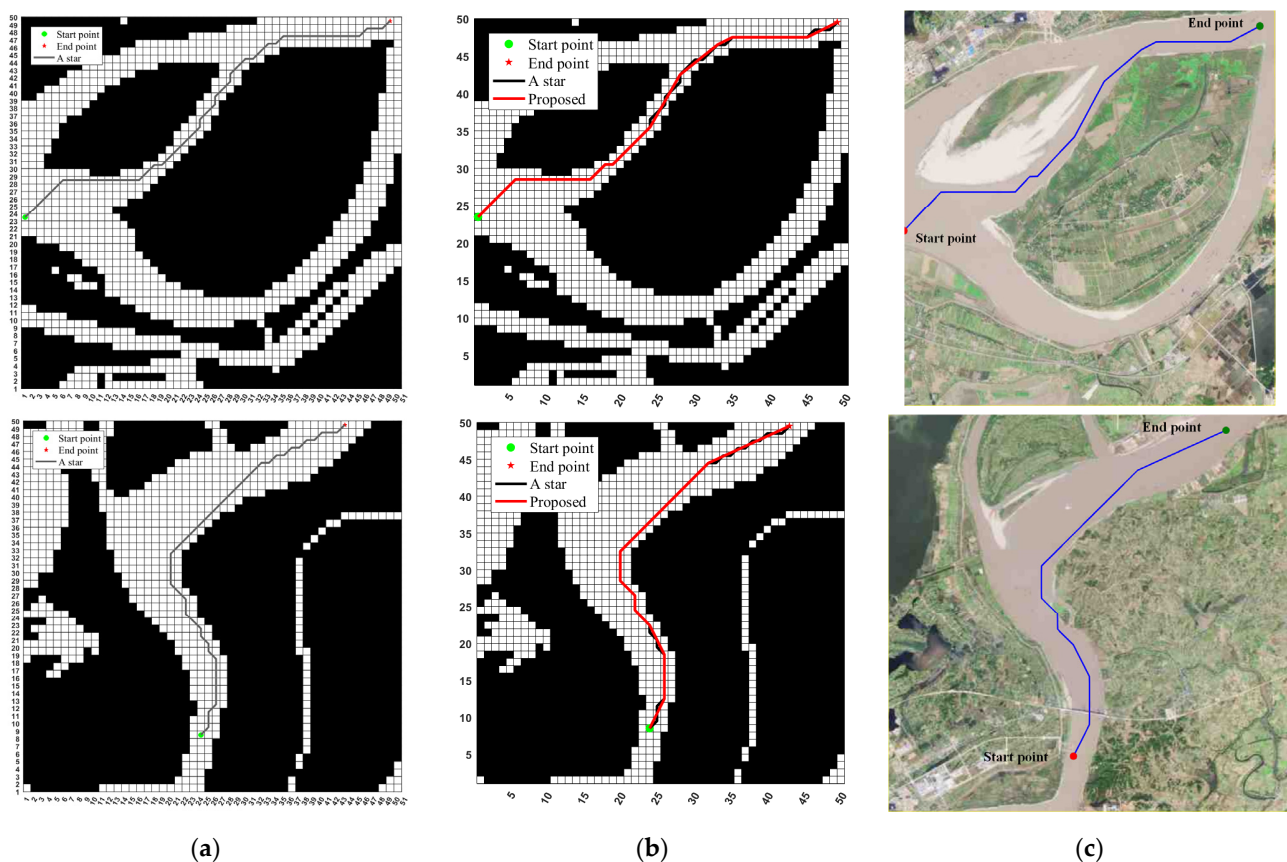


Figure 7. In (a), the black line represents the searched path by A*. The green dot denotes the start and the red star denotes the goal. In (b), the red line represents the searched path by ESP, considering the turning cost. (c) shows the searched path by ESP in the chart.

Figure 8 shows the performance of RRT. Due to the randomness of RRT, the planned paths have many unnecessary turning points, which is detrimental to the safe navigation of ships.

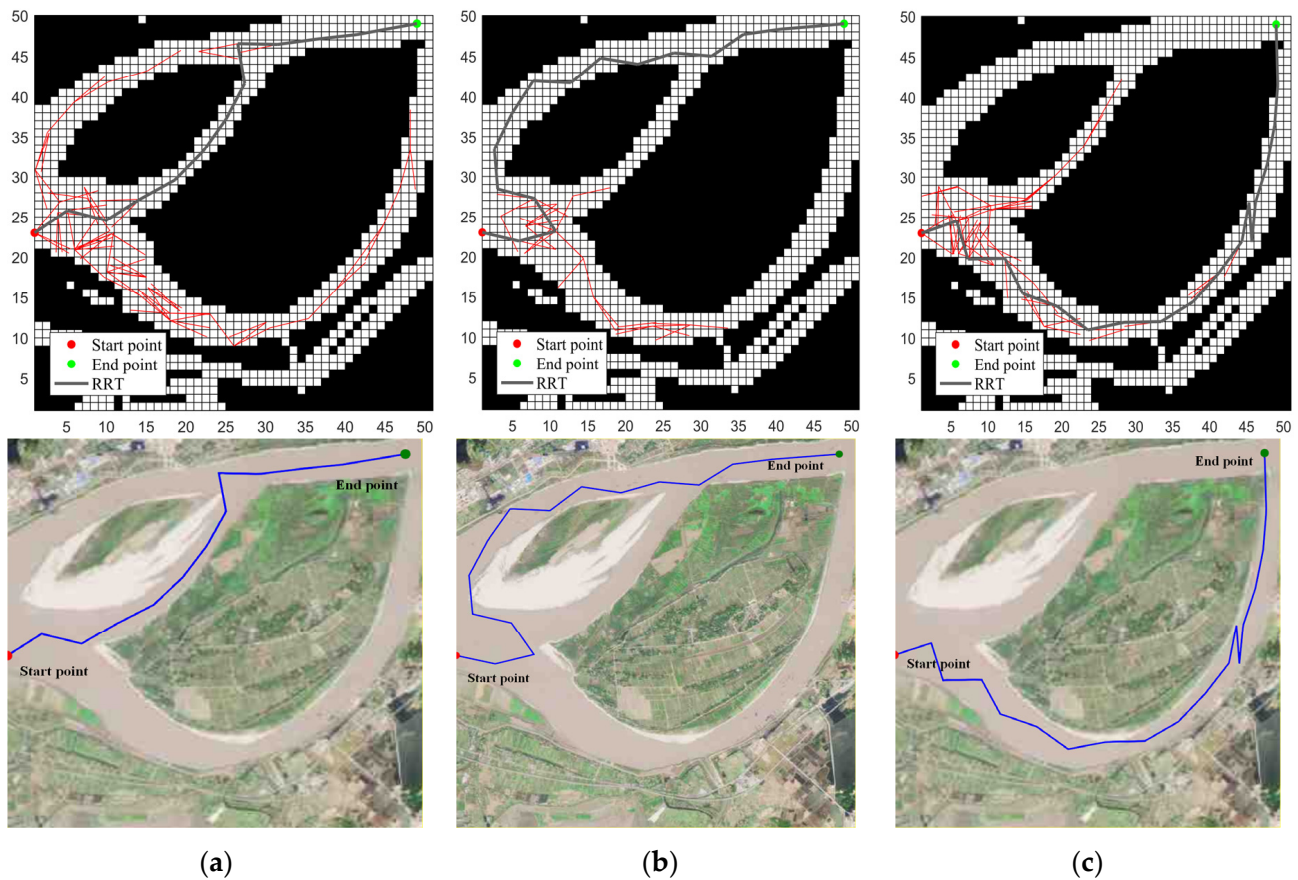


Figure 8. The results of the RRT algorithm. (a–c) are the three stochastic planning results of the RRT algorithm. The solid black line is the planned route, the solid red line records the sampling process of the algorithm, and the solid blue line shows the planning results.

Figure 9 shows the result of RRT*. RRT* needs to rewire parents to find asymptotically optimal paths. The result will be close to the optimal solution with more iterations.

Figure 10 shows the optimized trajectory of our proposed ESP. It can be seen that the optimized trajectory (the green curve in Figure 10a) meets the requirements of safety, feasibility, and smoothness.

Table 2 shows the numeric comparison in Scene 1 among ESP, A* [12], RRT [43] and RRT* [44] in terms of computation time, number of turns, and fuel consumption. It can be seen that due to the need to measure the turning cost of the path, the searching time of ESP is 0.105 s longer than that of the A* algorithm, and the algorithm's operating efficiency is reduced by 36.71%. Nevertheless, it is still 1.771 s shorter than that of the RRT algorithm, and 6.021 s shorter than that of the RRT* algorithm. The efficiency of the algorithm is improved by 4.35 times and 15.42 times, respectively. In addition, the number of turns of ESP is obviously less than that of the A* algorithm, which reduces the number of large-angle steers to 8 and improves the safety of ship navigation. The fuel consumption of ESP is 164.6008 kg less than that of the A* algorithm, 387.1543 kg less than that of the RRT algorithm, and 24.3311 kg less than that of the RRT* algorithm.

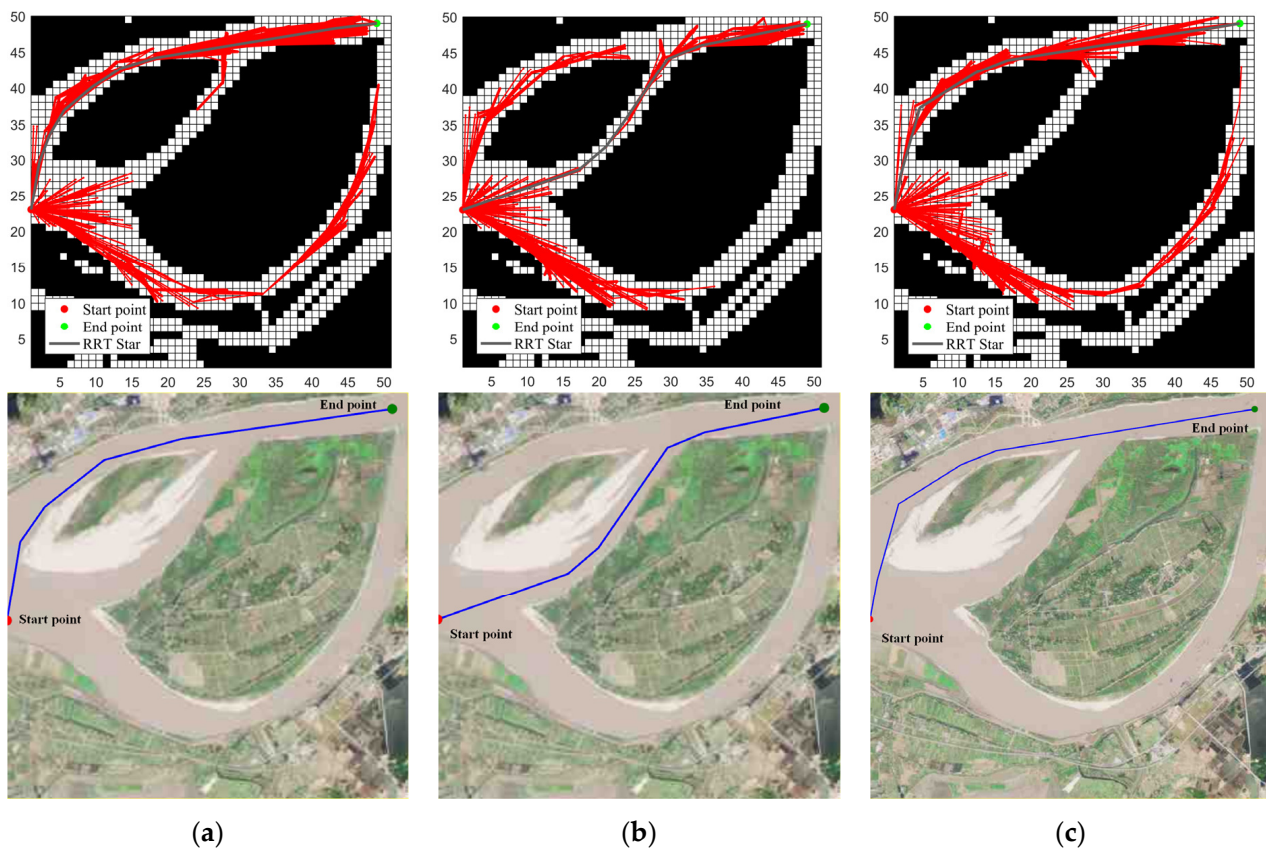


Figure 9. Three different searched paths from the RRT* algorithm. (a–c) are the three stochastic planning results of the RRT* algorithm. The solid black line is the planned route, the solid red line records the sampling process of the algorithm, and the solid blue line shows the planning results.

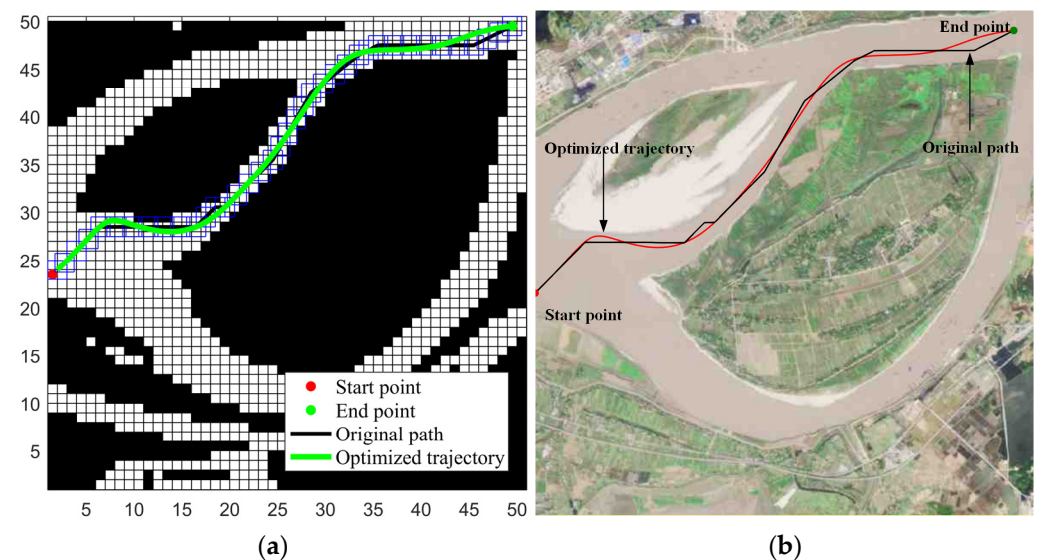
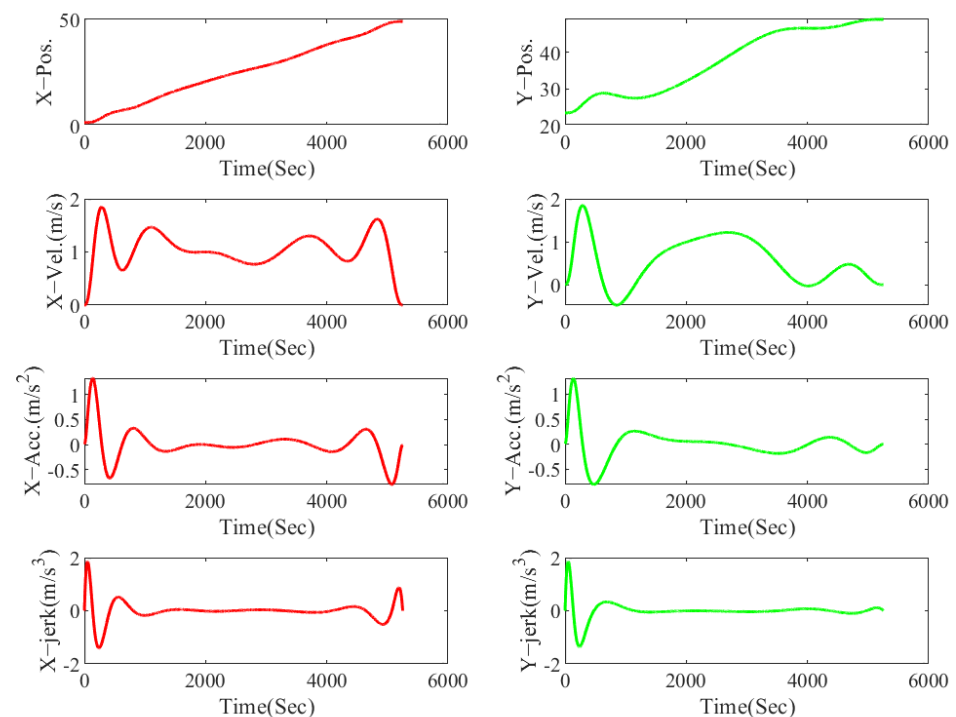


Figure 10. (a) The blue rectangles represent the sailing corridor. (b) The black line is the searched path. The optimized trajectory is shown as the red curve.

To evaluate the smoothness, Figure 11 shows the generated positions, velocities, accelerations, and jerks. All these curves are continuous and satisfy the ship's dynamic constraints.

Table 2. Performance comparison in Scene 1 with 1000 trials.

| Methods | Computation Time (s) | | No. of Turns | Average Fuel Consumption (kg) |
|-----------|----------------------|--------------|--------------|-------------------------------|
| A* [12] | Average | 0.286 | 18 | 921.7465 |
| | Max | 0.327 | | |
| RRT [43] | Average | 2.162 | 11 | 1144.3000 |
| | Max | 3.534 | | |
| RRT* [44] | Average | 6.412 | 4 | 781.4768 |
| | Max | 7.282 | | |
| ESP | Average | 0.391 | 8 | 757.1457 |
| | Max | 0.532 | | |

**Figure 11.** The optimized smooth trajectory in terms of position, velocity, acceleration, and jerk.

To highlight the effectiveness of ESP, Figure 12 compares the result of ESP with that which did not consider the ship's dynamic constraints. The blue line denotes the searched path. The green line shows that Bezier curve without considering the dynamic constraints. The red line is the optimized trajectory from ESP.

From the enlarged part, we can see that there are knots in the result without considering the dynamic constraints, making sailing control very difficult. In contrast, ESP generates a smooth and continuous trajectory with the lowest number of turns, which is safer.

Figures 13 and 14 show the collision avoidance in Scene 1 and 2. In Scene 1, ESP generates a feasible and smooth local trajectory (the orange line) that avoids the unexpected circular obstacle. In Scene 2, the original planned trajectory is very close to the shoal, increasing the risk of the ship running aground. The local replanned trajectory effectively solves the problem. In both scenarios, the goal of safely avoiding temporarily appearing static obstacles is achieved.

The computation times of the local replanning in both scenes are listed in Table 3. Over 1000 trials, the best calculation time is 48 ms, and the maximum computation time is 265 ms. The mean computation time is 192 ms in Scene 1 and 194 ms in Scene 2, ensuring real-time processing.

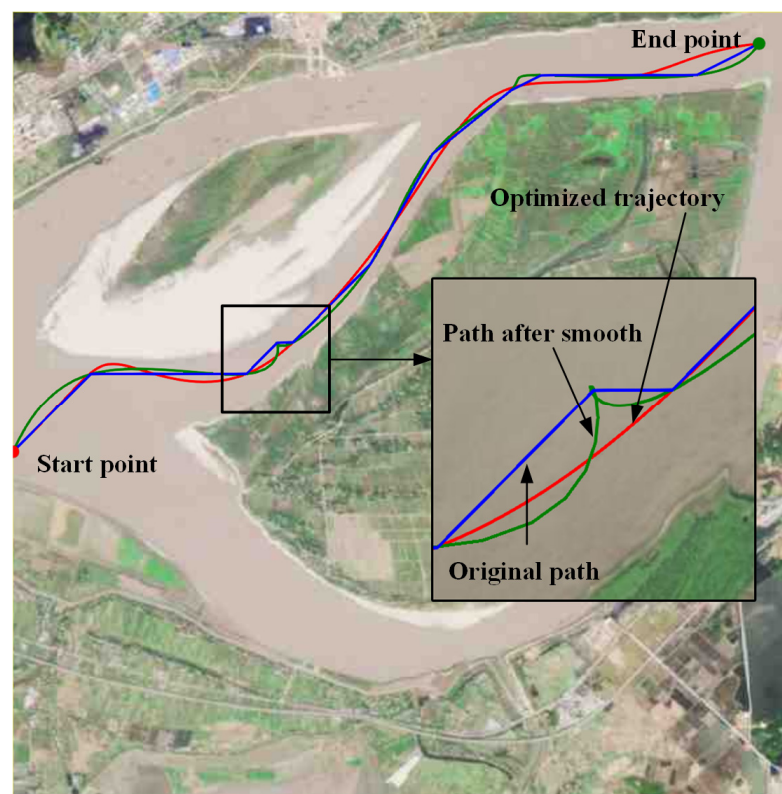


Figure 12. ESP vs. w/o considering the dynamic constraints.

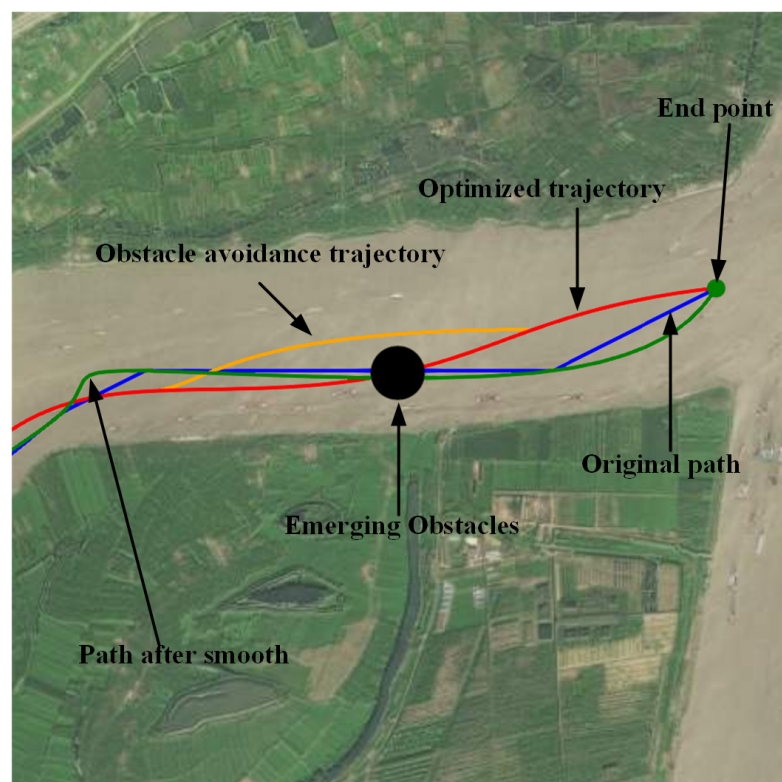


Figure 13. Avoid the unexpected circular obstacle in Scene 1.

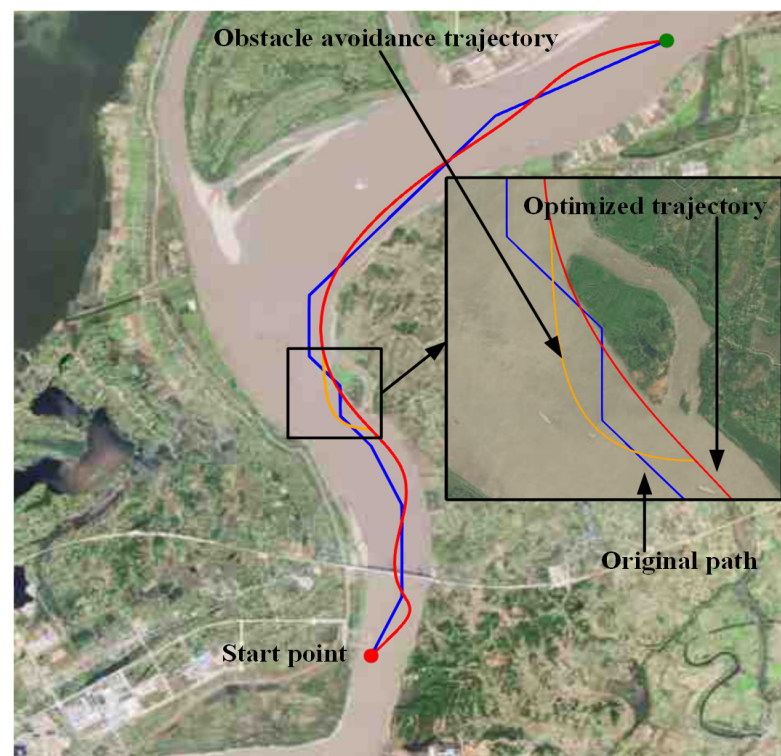


Figure 14. Avoid the shoal to reduce the risk of ship running aground in Scene 2.

Table 3. Computation time of the local replanning in both scenes with 1000 trials.

| Test Scene | Max Computation Time (s) |
|------------|--------------------------|
| Scene 1 | 0.233 |
| Scene 2 | 0.265 |

Avoiding multiple obstacles requires multiple iterations of the above process, and the response time will be multiplied by the number of iterations.

5. Conclusions

This paper proposes a GIS-data-driven method for the efficient and safe path planning of autonomous ships in maritime transportation, which makes up for the shortcomings of existing methods that ignore the motion dynamic limitations of ships in order to achieve the shortest path, leading to sudden changes in the planned route and thus lacking practical applicability. To this end, we propose ESP, a new path planner that provides comfortable sailing while saving fuel. The key intuition of our proposal is to reduce the expensive turning in path searching. The expensiveness comes from the inertia exhibited by the huge weight of the ship. To realize the above intuitive idea, we design three components for ESP. First, we quantify the ship's turning cost based on its kinematic and dynamic model and develop a modified A* path search algorithm. Second, we formulate an optimization problem subject to dynamic ship constraints and environment constraints to produce a safe and smooth trajectory that consumes minimal fuel. Finally, we use the B-spline representation to perform real-time local replanning, enabling autonomous ships to quickly respond to unexpected risks while maintaining the previous optimization objectives. The data-driven experiments demonstrate the effectiveness of ESP. However, we currently only consider the effect of ocean currents on the dynamic ship model. In future, we will consider the influence of other environmental factors, e.g., sea wind, to build a more robust model for autonomous ships. The avoidance of dynamic obstacles and the real-time avoidance of multiple static obstacles will also be investigated on this basis.

Author Contributions: Conceptualization, X.H. and K.H.; methodology, K.H., X.H. and D.T.; software, D.T., X.H. and K.H.; validation, X.H., K.H. and D.T.; formal analysis, X.H., K.H. and D.T.; investigation, D.T., Y.Z. and Y.H.; resources, X.H., K.H., D.T. and Y.H.; data curation, X.H., K.H. and D.T.; writing—original draft preparation, X.H. and D.T.; writing—review and editing, D.T. and X.H.; visualization, D.T. and K.H.; supervision, X.H., K.H. and Y.Z.; project administration, X.H. and Y.Z.; funding acquisition, Y.Z. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Research Project of Wuhan University of Technology Chongqing Research Institute (No. YF2021-06). This work was also supported by a grant from the National Natural Science Foundation of China (Grant No. 61801341).

Data Availability Statement: The simulation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tang, L.; Bhattacharya, S. Revisiting the Shortage of Seafarer Officers: A New Approach to Analysing Statistical Data. *WMU J. Marit. Affairs* **2021**, *20*, 483–496. [\[CrossRef\]](#)
2. Wilson, D.D. Evaluating Jamaica’s position as a seafarer-supplying country for cruise and cargo. *Worldw. Hosp. Tour. Themes* **2022**, *14*, 124–136. [\[CrossRef\]](#)
3. Lee, S.-M.; Roh, M.-I.; Kim, K.-S.; Jung, H.; Park, J.J. Method for a Simultaneous Determination of the Path and the Speed for Ship Route Planning Problems. *Ocean Eng.* **2018**, *157*, 301–312. [\[CrossRef\]](#)
4. Liu, Y.; Bucknall, R. A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica* **2018**, *36*, 1019–1047. [\[CrossRef\]](#)
5. Stankiewicz, P.; Kobilarov, M. A Primitive-Based Approach to Good Seamanship Path Planning for Autonomous Surface Vessels. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China, 30 May–5 June 2021; pp. 7767–7773.
6. Du, Z.; Wen, Y.; Xiao, C.; Zhang, F.; Huang, L.; Zhou, C. Motion Planning for Unmanned Surface Vehicle Based on Trajectory Unit. *Ocean Eng.* **2018**, *151*, 46–56. [\[CrossRef\]](#)
7. Vettor, R.; Guedes Soares, C. Development of a Ship Weather Routing System. *Ocean Eng.* **2016**, *123*, 1–14. [\[CrossRef\]](#)
8. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. A Constrained A* Approach towards Optimal Path Planning for an Unmanned Surface Vehicle in a Maritime Environment Containing Dynamic Obstacles and Ocean Currents. *Ocean Eng.* **2018**, *169*, 187–201. [\[CrossRef\]](#)
9. Niu, H.; Savvaris, A.; Tsourdos, A.; Ji, Z. Voronoi-Visibility Roadmap-Based Path Planning Algorithm for Unmanned Surface Vehicles. *J. Navig.* **2019**, *72*, 850–874. [\[CrossRef\]](#)
10. Jeong, M.-G.; Lee, E.-B.; Lee, M.; Jung, J.-Y. Multi-Criteria Route Planning with Risk Contour Map for Smart Navigation. *Ocean Eng.* **2019**, *172*, 72–85. [\[CrossRef\]](#)
11. Yang, J.-M.; Tseng, C.-M.; Tseng, P.S. Path Planning on Satellite Images for Unmanned Surface Vehicles. *Int. J. Nav. Archit. Ocean. Eng.* **2015**, *7*, 87–99. [\[CrossRef\]](#)
12. Song, R.; Liu, Y.; Bucknall, R. Smoothed A* Algorithm for Practical Unmanned Surface Vehicle Path Planning. *Appl. Ocean. Res.* **2019**, *83*, 9–20.
13. Zhang, L.; Mou, J.; Chen, P.; Li, M. Path Planning for Autonomous Ships: A Hybrid Approach Based on Improved APF and Modified VO Methods. *J. Mar. Sci. Eng.* **2021**, *9*, 761.
14. Liu, X.; Li, Y.; Zhang, J.; Zheng, J.; Yang, C. Self-Adaptive Dynamic Obstacle Avoidance and Path Planning for USV Under Complex Maritime Environment. *IEEE Access* **2019**, *7*, 114945–114954. [\[CrossRef\]](#)
15. Lazarowska, A. A Discrete Artificial Potential Field for Ship Trajectory Planning. *J. Navig.* **2020**, *73*, 233–251. [\[CrossRef\]](#)
16. Chen, C.; Chen, X.-Q.; Ma, F.; Zeng, X.-J.; Wang, J. A Knowledge-Free Path Planning Approach for Smart Ships Based on Reinforcement Learning. *Ocean Eng.* **2019**, *189*, 106299.
17. Wang, C.; Zhang, X.; Li, R.; Dong, P. Path Planning of Maritime Autonomous Surface Ships in Unknown Environment with Reinforcement Learning. In Proceedings of the ICCSIP 2018: International Conference on Cognitive Systems and Signal Processing, Beijing, China, 29 November–1 December 2018; Springer: Singapore, 2019; pp. 127–137.
18. Yu, L.; Qian, T.; Ye, X.; Zhou, F.; Luo, Z.; Zou, A. Research on obstacle avoidance strategy of USV based on improved grid method. In Proceedings of the 4th International Symposium on Power Electronics and Control Engineering (ISPECE 2021), Nanchang, China, 16–19 September 2021; Volume 12080, pp. 172–176.
19. Xie, L.; Xue, S.; Zhang, J.; Zhang, M.; Tian, W.; Haugen, S. A Path Planning Approach Based on Multi-Direction A* Algorithm for Ships Navigating within Wind Farm Waters. *Ocean Eng.* **2019**, *184*, 311–322. [\[CrossRef\]](#)
20. Wang, H.; Qi, X.; Lou, S.; Jing, J.; He, H.; Liu, W. An Efficient and Robust Improved A* Algorithm for Path Planning. *Symmetry* **2021**, *13*, 2213. [\[CrossRef\]](#)

21. Yu, K.; Liang, X.; Li, M.; Chen, Z.; Yao, Y.; Li, X.; Zhao, Z.; Teng, Y. USV Path Planning Method with Velocity Variation and Global Optimisation Based on AIS Service Platform. *Ocean Eng.* **2021**, *236*, 109560. [\[CrossRef\]](#)
22. Sang, H.; You, Y.; Sun, X.; Zhou, Y.; Liu, F. The Hybrid Path Planning Algorithm Based on Improved A* and Artificial Potential Field for Unmanned Surface Vehicle Formations. *Ocean Eng.* **2021**, *223*, 108709.
23. Han, S.; Xiao, L. An Improved Adaptive Genetic Algorithm. *SHS Web Conf.* **2022**, *140*, 01044. [\[CrossRef\]](#)
24. Wang, N.; Xu, H.; Li, C.; Yin, J. Hierarchical Path Planning of Unmanned Surface Vehicles: A Fuzzy Artificial Potential Field Approach. *Int. J. Fuzzy Syst.* **2021**, *23*, 1797–1808. [\[CrossRef\]](#)
25. Zhang, Z.; Wu, D.; Gu, J.; Li, F. A Path-Planning Strategy for Unmanned Surface Vehicles Based on an Adaptive Hybrid Dynamic Stepsize and Target Attractive Force-RRT Algorithm. *J. Mar. Sci. Eng.* **2019**, *7*, 132. [\[CrossRef\]](#)
26. Chen, L.; Mantegh, I.; He, T.; Xie, W. Fuzzy Kinodynamic RRT: A Dynamic Path Planning and Obstacle Avoidance Method. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 188–195.
27. Strub, M.P.; Gammell, J.D. Adaptively Informed Trees (AIT*): Fast Asymptotically Optimal Path Planning through Adaptive Heuristics. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3191–3198.
28. Zong, C.; Han, X.; Zhang, D.; Liu, Y.; Zhao, W.; Sun, M. Research on Local Path Planning Based on Improved RRT Algorithm. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2021**, *235*, 2086–2100. [\[CrossRef\]](#)
29. Yu, J.; Yang, M.; Zhao, Z.; Wang, X.; Bai, Y.; Wu, J.; Xu, J. Path Planning of Unmanned Surface Vessel in an Unknown Environment Based on Improved D*Lite Algorithm. *Ocean Eng.* **2022**, *266*, 112873. [\[CrossRef\]](#)
30. Liang, X.; Jiang, P.; Zhu, H. Path Planning for Unmanned Surface Vehicle with Dubins Curve Based on GA. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 5149–5154.
31. Candeloro, M.; Lekkas, A.M.; Sørensen, A.J. A Voronoi-Diagram-Based Dynamic Path-Planning System for Underactuated Marine Vessels. *Control Eng. Pract.* **2017**, *61*, 41–54. [\[CrossRef\]](#)
32. Wang, N.; Zhang, Y.; Ahn, C.K.; Xu, Q. Autonomous Pilot of Unmanned Surface Vehicles: Bridging Path Planning and Tracking. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2358–2374. [\[CrossRef\]](#)
33. Zhou, C.; Gu, S.; Wen, Y.; Du, Z.; Xiao, C.; Huang, L.; Zhu, M. Motion Planning for an Unmanned Surface Vehicle Based on Topological Position Maps. *Ocean Eng.* **2020**, *198*, 106798. [\[CrossRef\]](#)
34. MahmoudZadeh, S.; Abbasi, A.; Yazdani, A.; Wang, H.; Liu, Y. Uninterrupted Path Planning System for Multi-USV Sampling Mission in a Cluttered Ocean Environment. *Ocean Eng.* **2022**, *254*, 111328. [\[CrossRef\]](#)
35. Zheng, Y.; Zhang, X.; Shang, Z.; Guo, S.; Du, Y. A decision-making method for ship collision avoidance based on improved cultural particle swarm. *J. Adv. Transp.* **2021**, *2021*, 8898507. [\[CrossRef\]](#)
36. Zhou, X.; Wang, Z.; Ye, H.; Xu, C.; Gao, F. EGO-Planner: An ESDF-Free Gradient-Based Local Planner for Quadrotors. *IEEE Robot. Autom. Lett.* **2021**, *6*, 478–485. [\[CrossRef\]](#)
37. Zhou, M.; Jin, H. Development of a Transient Fuel Consumption Model. *Transp. Res. Part D Transp. Environ.* **2017**, *51*, 82–93. [\[CrossRef\]](#)
38. Wang, W.; Tu, A.; Bergholm, F. Improved Minimum Spanning Tree Based Image Segmentation with Guided Matting. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 211–230.
39. Min, B.; Zhang, X. Concise Robust Fuzzy Nonlinear Feedback Track Keeping Control for Ships Using Multi-Technique Improved LOS Guidance. *Ocean Eng.* **2021**, *224*, 108734. [\[CrossRef\]](#)
40. Barraquand, J.; Kavraki, L.; Latombe, J.C.; Li, T.Y.; Motwani, R.; Raghavan, P. A random sampling scheme for path planning. In Proceedings of the Robotics Research: The Seventh International Symposium, Herrsching, Germany, 21–24 October 1995; Springer: London, UK, 1996; pp. 249–264.
41. Li, Z.; Li, L.; Zhang, W.; Wu, W.; Zhu, Z. Research on Unmanned Ship Path Planning Based on RRT Algorithm. *J. Phys. Conf. Ser.* **2022**, *2281*, 012004. [\[CrossRef\]](#)
42. Tran, N.K.; Lam, J.S.L. Effects of Container Ship Speed on CO₂ Emission, Cargo Lead Time and Supply Chain Costs. *Res. Transp. Bus. Manag.* **2022**, *43*, 100723. [\[CrossRef\]](#)
43. Kaur, A.; Prasad, M.S. Path Planning of Multiple Unmanned Aerial Vehicles Based on RRT Algorithm. In *Advances in Interdisciplinary Engineering*; Kumar, M., Pandey, R.K., Kumar, V., Eds.; Springer: Singapore, 2019; pp. 725–732.
44. Zhang, X.; Chen, X. Path Planning Method for Unmanned Surface Vehicle Based on RRT* and DWA. In Proceedings of the ICMTEL 2021: International Conference on Multimedia Technology and Enhanced Learning, Virtual Event, 8–9 April 2021; Fu, W., Xu, Y., Wang, S.-H., Zhang, Y., Eds.; Springer: Cham, Switzerland, 2021; pp. 518–527.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.