Journal of
*Marine Science
and Engineering*

MDPI

Article

A Novel Algorithm for Ship Route Planning Considering Motion Characteristics and ENC Vector Maps

Qinghua He [1], Zhenyu Hou [2] and Xiaoxiao Zhu [1,*]

[1] College of Mechanical and Transportation Engineering, China University of Petroleum, Beijing 100100, China; qinghuaharry1204@gmail.com
[2] College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300399, China
* Correspondence: x.zhu@cup.edu.cn

Abstract: Global route planning is a pivotal function of unmanned surface vehicles (USVs). For ships, the safety of navigation is the priority. This paper presents the VK-RRT* algorithm as a way of designing the planned route automatically. Different from other algorithms or studies, this study employs electronic navigation chart (ENC) vector data instead of grid maps as the basis of the search, which reduces data error when converting the vector map into the grid map. In addition, Delaunay triangulation is employed to organize vector data, in which the depth value is taken as a factor to ensure the safety of the planning route. Furthermore, the initial planned route is not suitable for ship tracking as it does not consider the ship motion characteristics. Therefore, the planned route needs to be further optimized. In the final part, we also conducted experiments to verify the effectiveness and advantages of the proposed algorithm. The results show that the proposed algorithm could reduce the lengths of paths by about 23% on average and save planning time; these are largely dependent on the environment.

Keywords: route planning; ENC map; sample-based algorithm

check for updates

Citation: He, Q.; Hou, Z.; Zhu, X. A Novel Algorithm for Ship Route Planning Considering Motion Characteristics and ENC Vector Maps. *J. Mar. Sci. Eng.* **2023**, *11*, 1102. https://doi.org/10.3390/jmse11061102

Academic Editor: Alessandro Ridolfi

Received: 2 April 2023
Revised: 15 May 2023
Accepted: 18 May 2023
Published: 23 May 2023

1. Introduction

Recently, the number of studies on unmanned surface vehicles (USV) has attracted much attention in the marine technology field. In fact, one of the premises of USV is generating the ship planning route (SPR) automatically. Then, the ship can track the route and arrive at the destination safely.

The ship route planning problem is a sub-field of route planning, where various algorithms and strategies were proposed and applied.

At first, there was a large number of studies on algorithms based on graphs. Reference [1] proposed the Dijkstra algorithm to address two problems in the route planning field: (1) Refer to a data structure that consists of paths connecting any two points; (2) Find the optimal path with a minimum total length between two given nodes. Then, the Dijkstra algorithm was employed and improved to solve practice issues. To solve the route planning problem involving the length of paths, a novel algorithm was created based on the Dijkstra algorithm, which referred to fuzzy theory [2]. Except the length of paths, time costs were also considered in [3,4], where traditional Dijkstra was improved. Furthermore, there were also various studies on the ship route planning problem employing the Dijkstra algorithm. There were also researchers who created a three-dimensional Dijkstra algorithm that supported the ship to plan the motion in which the speeds and courses were determined at each second [5]. Generating global optimum solutions for ship routes was expected. Moreover, the weather was also taken as an index of the Dijkstra algorithm when planning the minimum time cost for a route [6].

Then, based on the Dijkstra algorithm, a novel method called A* was also proposed and improved. In the original A* algorithm, the path was connected by the vertices of the grid map used, which means that the length of the route was not optimal. Therefore, Theta* was applied to [7] to make up for the drawback. Moreover, to save searching time, time-efficient A* was employed in [8] and simulated successfully. To address route planning problems, researchers considered multiple factors when applying the A* algorithm. For example, collision avoidance, rules on the sea, and the motion characteristics of ships were taken as examples [9]. Moreover, some researchers even considered the space characteristics of the ship [10].

Moreover, sampling theory was also extended and applied to route planning problems. RRT (rapid-exploring random tree) is a common algorithm used in the field of path planning. The RRT algorithm is completed in probability if reaches 100% if enough time is given for exploration. The main idea is to construct the searching tree and find line segments connecting the start point to the destination. Compared to other sample-based algorithms, not only does the RRT* have their advantages but it also has a higher degree of freedom [11]. Moreover, there were several algorithms that were generated based on the RRT algorithm. For instance, the RRT* is a kind of sample-based algorithm, which optimized the length of the route given by the RRT algorithm. In reference [12], the RRT* was compared with the A* via simulations to illustrate their advantages and disadvantages. In fact, the RRT* had been improved in various ways, taking into account different optimization goals or environmental conditions. To solve the field programmable gate arrays problem, the RRT* was improved to consider the terrain, planning speed, etc., [13]. Furthermore, a novel series of improved RRT* algorithms, named Quick-RRT*, PQ-RRT*, and P-RRT* were proposed by researchers [14,15]. All of them focused on fast speed converges to improve the characteristic of the route planning module of the ship. To do so, they combined the RRT* and potential function to give a better solution with a fast speed of convergence. Except for the above algorithms, the Connect-RRT algorithm was also developed to plan routes with fast speed [16]. In fact, the Connect RRT algorithm maintains two trees: one of them started from the departure point and another began from the destination. When the two trees encountered each other during the searching process, the route planning task could be ended. Based on that, various studies imitating the principle of Connect-RRT emerged, such as RRT*-connect [17], bidirectional potential guided RRT* [18], informed RRT*-connect [19], smooth RRT-connect [20], and so on.

RRT and its extended methods in the marine technology field (ship route planning) was applied to generate planned routes in a canal [21]. In such area, the current had to be considered, as is reasonable, as it would affect the navigation safety of ships. Moreover, the rule of local governments regarding canal navigation shall be also considered to avoid collisions between surface vehicles. To make up for the disadvantages of the RRT algorithm, reference [22] gave a hybrid step size and target attractive force RRT algorithm, which mainly improved the accuracy of the planned route in narrow waters. Moreover, the RRT algorithm was developed to have abilities of enhanced adaptability in [23]. Furthermore, it was necessary to include regulations when considering the route planning algorithm. Based on that, reference [24] gave a novel algorithm that combined the rule on the sea. Moreover, reference [25] gave the Bi-RRT algorithm, which can make decisions when faced with obstacles to prevent collisions.

Recently, with the development of artificial intelligence, many researchers have started using the reinforce leaning theory to address route planning problems [26–28]. For instance, the algorithm called deep reinforcement learning was provided in [29] to solve collision avoidance problem and optimize the length of the route at the same time. Moreover, the reinforcement learning algorithm was also engaged in unsupervised learning [30].

Different from robots and other vehicles, the navigation safety of ships involves two factors: water depth and obstacles. It means that the SPR could not cross land, reefs, and shallow water areas. However, few of the above studies consider both factors. Instead, they paid more attention to avoiding obstacles. Moreover, it is well-known that SPR is a polyline. However, the ship could not follow polylines accurately because of kinetic constraints [31]. Therefore, it is essential to consider ship motion characteristics in waypoint areas in optimization. In addition, most of the studies reviewed employed grid maps as configuration spaces. It means that converting from ENC or other sources of charts to raster maps should be conducted in advance. Errors occurred during such processes.

To make up for the above drawbacks, this study proposes a novel approach called the VK-RRT* algorithm. The main contributions of it are as follows:

(1) Unlike in most research, ENC vector data are utilized instead of simulated environmental data or raster map data. The use of vector data could reduce errors emerging from the raster map data process and could also accelerate the application of the path planning algorithm in navigational practice;

(2) We propose a novel strategy for the implementation of the path planning algorithm to account for ship kinetic constraints. Under constant speed, turning trajectories in the form of arcs are predicted and checked for safety in the path planning process. The strategy could largely decrease the pressure of controllers to track the planed path in the turning area;

(3) Compared to RRT and RRT* algorithms, the VK-RRT* method could give solutions faster.

The remainder of this study is arranged as follows: Section 2 presents the preliminary knowledge involved in this study. Section 3 gives RRT, RRT*, and the proposed VK-RRT* algorithm descriptions in detail for readers to compare and understand. Section 4 presents two case studies to illustrate the advantages and effectiveness of the proposed algorithm. Section 5 gives some conclusions and drawbacks, which need to be addressed in near future.

2. Preliminary Knowledge

In this section, we give the definition of the route planning of ships, motion constrains considered, and the Delaunay triangulation explanations used in our algorithms.

2.1. Problem Definition

This section presents the ship route planning problem that is investigated in this study. Let $\chi \subseteq R^d$ be the configuration space where ships navigate where $d \geq 2$. In addition, there are two sub-spaces χ_{obs} and χ_{free} in χ, which represent the obstacle space and the free space of navigation, respectively. p_{init} represents the initial position, whereas p_{goal} is the position of the destination that is to be reached. Path σ is a continuous function $[0,1] \mapsto \chi$ and it has bounded variation. Moreover, σ is free of collision when $\sigma(\tau) \in \chi_{free}$, $\tau \in [0,1]$.

Ship route planning problem: It is designed to safely find a proper or optimal σ guiding ship from the departure position to the destination.

Additionally, the three sub-problems are defined as follows:

Sub-problem 1. Considering a triplet $\left\{p_{init}, p_{goal}, \chi_{obstacle}\right\}$, generate a feasible route if there is one. Otherwise, report failure.

For ship navigation, the shorter route means less cost. Therefore, this study also considers route optimization in the process of planning. Let Σ be the set of all routes and $\Sigma_{feasible}$ be the set of all feasible routes.

Sub-problem 2. Considering a triplet $\left\{p_{init}, p_{goal}, \chi_{obstacle}\right\}$, define a cost function c and find a feasible route σ_f, such as $c(\sigma_f) = \min(c(\sigma) : \sigma \in \Sigma_{feasible})$. Report failure if there is no σ_f.

Sub-problem 3. Considering a triplet $\left\{p_{init}, p_{goal}, \chi_{obstacle}\right\}$, define a cost function c and find a feasible route σ_k, such as $c(\sigma_k) = \min(c(\sigma) : \sigma \in \Sigma_{feasible})$, and find out if it is feasible to follow σ_k, considering ship motion characteristics. Report failure if there is no σ_k.

2.2. ENC Vector Data

To address the above problems, electronic nautical chart (ENC) vector data are employed to generate configuration space χ without rasterization. This type of chart could be used in the marine field, such as in navigation, fishing, etc.

According to the International Hydrographic Organization (IHO) S57 standard [32], ENC data consist of a certain amount of features that play a vital role in navigation. However, in this study, sound/depth in χ_{free}, land, reef or other static obstacles instead of all the features in ENC are employed in χ_{obs} to generate a planning route. In short, this study only considers the effects of depth and obstacle positions on the route planning problem.

Before implementing the ENC vector data, it is important to clarify the types of data. In ENC, an object is defined as an identifiable set of information. An object may have attributes and may be related to other objects. Feature objects have descriptive attributes but no geometry (i.e., information about the shape and position of a real-world entity). Spatial objects may have descriptive attributes and must have geometry. To be specific, there are three elements that represent geometry: point, line, and area.

In the proposed algorithm, ENC data are employed to form the configuration space χ. Obstacle areas make up χ_{obs} while depth points compose χ_{free}.

2.3. Ship Motion Characteristics

The motion characteristics of a vessel refer to the physical phenomena that describe the vessel's movements and behaviors on a waterway. These characteristics are influenced by several factors, including the physical characteristics of a vessel, such as its size, shape, weight, speed, and the hydrodynamic conditions of the water through which it moves. In particular, the motion of a vessel can be characterized by its heave (vertical motion), pitch (rotational motion around the transverse axis), roll (rotational motion around the longitudinal axis), and yaw (rotational motion around the vertical axis). These motions are indicative of the vessel's dynamic behavior and are influenced by external forces, such as wind, waves, and currents. A thorough understanding of the motion characteristics of a vessel is essential for ensuring the safe and efficient operation of a marine craft.

For route planning control problems, the turning constraint are usually taken into consideration. To be specific, when a ship alters its course, its trajectory is an arc. While most routes designed by researchers are presented as broken lines, straight line segments are connected in sequence. It is possible that ships ground in waypoint areas. Therefore, we have to take ship motion into account for safety purposes.

In order to simulate the ship motions, the MMG model is applied in this paper and Table 1 gives explanation of the variables used in Equations (1) and (2).

$$\begin{cases} (m+m_x)\dot{u} - (m+m_y)vr = X_{\text{H}} + X_{\text{P}} + X_{\text{R}} + X_{\text{wind}} + X_{\text{wave}} \\ (m+m_y)\dot{v} + (m+m_x)ur = Y_{\text{H}} + Y_{\text{P}} + Y_{\text{R}} + Y_{\text{wind}} + Y_{\text{wave}} \\ (I_{xx}+J_{xx})\dot{p} = K_{\text{H}} + K_{\text{P}} + K_{\text{R}} + K_{\text{wind}} + K_{\text{wave}} \\ (I_{zz}+J_{zz})\dot{r} = N_{\text{H}} + N_{\text{P}} + N_{\text{R}} + N_{\text{wind}} + N_{\text{wave}} \end{cases}, \tag{1}$$

$$\begin{cases} \dot{x} = u\cos(\psi) - v\cos(\varphi)\sin(\psi) \\ \dot{y} = u\sin(\psi) + v\cos(\varphi)\cos(\psi) \\ \dot{\varphi} = p \\ \dot{\psi} = r\cos(\varphi) \end{cases}, \tag{2}$$

Table 1. Explanation of variables.

Variables	Explanation	Variables	Explanation
m	mass	I_{xx}	Roll moment of inertia
m_x, m_y	additional mass	I_{zz}	yaw moment
x, y	position	J_{xx}	additional roll moment
ψ, φ	course and roll angle	J_{zz}	additional yaw moment
X, Y, K, N with H, P, R, wind, wave subscripts		Forces and moments	

Reference [33] gives more information about this model.

2.4. Delaunay Triangulation

The Delaunay triangulation is a geometric algorithm that constructs a triangulation using a set of points in a plane, where each triangle in the triangulation satisfies the Delaunay criterion; no point in the point set is inside the circumcircle of any triangle. This feature ensures that the triangulation avoids forming skinny triangles, which is advantageous for various applications, such as mesh generation and image processing.

In practice, there are many situations that we need in order to design routes using discrete points. The Delaunay triangulation (DT) method is such a method that can be used to solve the above problem. In electronic nautical charts, the depth points are unordered and discrete. The DT method could reconstruct them as the initial map for route planning.

The principle of DT is simple: for a given set P, any two points could construct an edge of DT map when there is a circle above two points and there is no other point in the circle [34].

It should be made clear that different ways of generating maps largely affect the planed route. In this paper, we only discuss the effectiveness of the proposed algorithm.

3. Route Planning Algorithm

In this section, we propose the VK-RRT* algorithm for the route planning problem of USV. Based on traditional RRT and considering motion constraints, the VK-RRT* could be applied using data from the ECDIS platform, which has great potential for commercial realisation.

3.1. Rapid-Exploring Random Tree

Rapid-exploring random tree (RRT) is a conventional algorithm in the field of path planning. It gives feasible solutions faster than other methods, such as A*, genetic algorithm (GA). Therefore, in this study, we improve RRT to VK-RRT* to produce a planned route quickly. In fact, RRT is a data structure. It will be sampled in the configuration space χ, and connect adjacent points constructing tree Γ.

Figure 1 shows the working principles of the RRT algorithm. The function *sample()* will randomly find a point from space χ; nearest() is responsible for selecting the nearest node from X_{rand} in tree Γ; steer() gives the X_{new} according to X_{rand} and $X_{nearest}$.

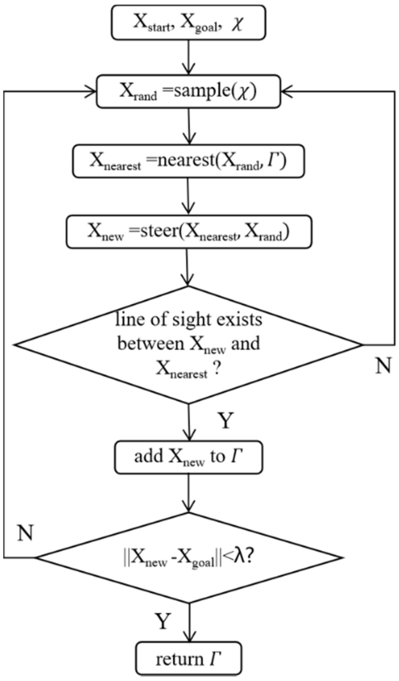

X_{start}, X_{goal}, χ
X_{rand} =sample(χ)
$X_{nearest}$ =nearest(X_{rand}, Γ)
X_{new} =steer($X_{nearest}$, X_{rand})
line of sight exists between X_{new} and $X_{nearest}$?
N
Y
add X_{new} to Γ
N
$||X_{new}$ -$X_{goal}||<\lambda$?
Y
return Γ

Figure 1. Flow chart of rapid-exploring random tree.

Theoretically, the RRT algorithm could find feasible paths connecting departure points and destinations if time cost is not considered, which means that the algorithm reaches 100%. However, the RRT algorithm cannot be used in practical applications because the given path is in the form of a polyline, making it impossible to track ships.

Therefore, the RRT* algorithm has been proposed by previous researchers to improve it. Figure 2 describes the RRT* algorithm. The difference between RRT and RRT* is that there is a rewire function in the RRT* algorithm. It optimizes the relationships of nodes in Γ to reduce the account of nodes.

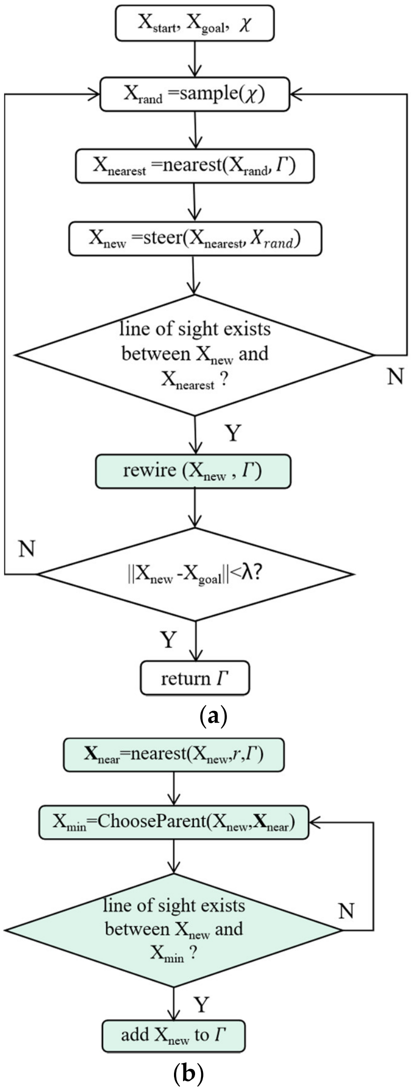

X_{start}, X_{goal}, χ
X_{rand} =sample(χ)
$X_{nearest}$ =nearest(X_{rand}, Γ)
X_{new} =steer($X_{nearest}$, X_{rand})
line of sight exists between X_{new} and $X_{nearest}$?
N
Y
rewire (X_{new} , Γ)
N
$||X_{new}$ -$X_{goal}||$<λ?
Y
return Γ
(a)
$\mathbf{X}_{near}$=nearest(X_{new},r,Γ)
X_{min}=ChooseParent(X_{new},$\mathbf{X}_{near}$)
line of sight exists between X_{new} and X_{min} ?
N
Y
add X_{new} to Γ
(b)

Figure 2. Flow chart of RRT* algorithm. (**a**) Outlines of RRT* algorithm, (**b**) Rewire function of RRT* algorithm.

3.2. VK-RRT* Algorithm

Taking ship motion characteristics into account, we propose the VK-RRT* algorithm.

Before introducing the algorithm, the usage of the Delaunay triangulation is illustrated. The depth points $E_i \in \mathbf{E}$ and obstacle positions are extracted from the electronic nautical chart. There are five attributes of E_i: (φ,λ) represents the latitude and longitude; d denotes the depth value; η is minimum depth of around E_i; ε means the minimum distance from adjacent objects. Similarly, to enlarge the waypoints data set, we also created $\mathrm{C}_i \in \mathbf{C}$ to represent the nodes in the center of every triangular (Figures 3 and 4).

By obtaining data sets of depth points and obstacles, Delaunay triangulation is applied for generating the DT map, which is represented as χ. It is noted that there are two areas in space χ: free space χ_{free} and obstacle space χ_{obs}. **E** and **C** make up the configuration space χ_{free}.

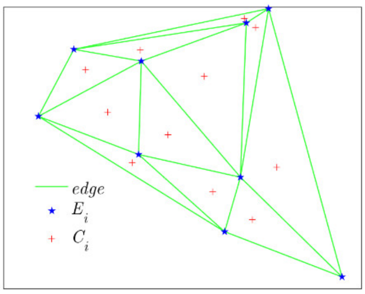
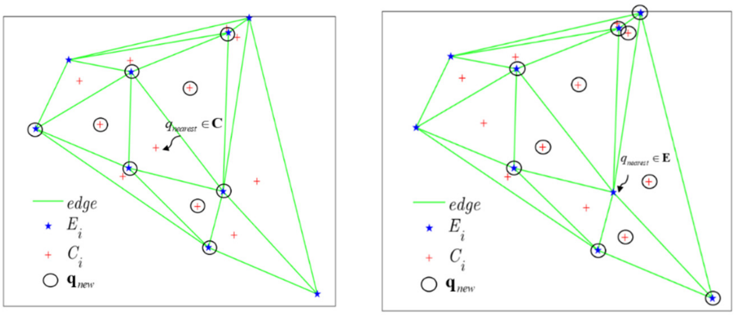


Figure 3. Sketch map of Delaunay triangle constructing **E**.



(**a**) (**b**)

Figure 4. Methods to generate the new node $\mathbf{q}_{new}$. (**a**) Method when the nearest node is in the triangular; (**b**) Method when the nearest node is on the vertex of a triangular.

Figure 5 describes the working principle of VK-RRT*. Except for the data process, the rewire module is different from traditional RRT*, as it considers ship motion characteristics.

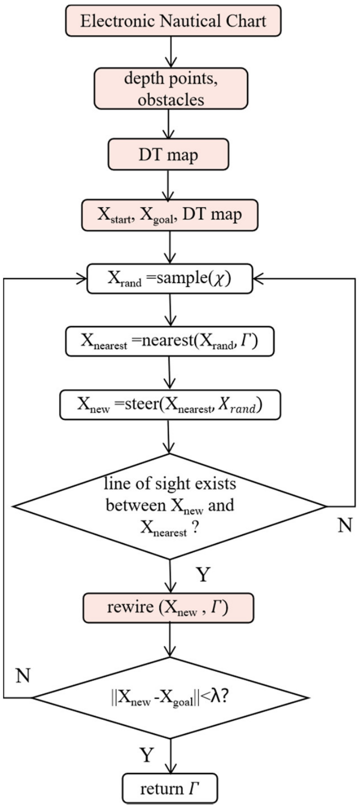

Electronic Nautical Chart
depth points, obstacles
DT map
X_{start}, X_{goal}, DT map
X_{rand} =sample(χ)
$X_{nearest}$ =nearest(X_{rand}, Γ)
X_{new} =steer($X_{nearest}$, X_{rand})
line of sight exists between X_{new} and $X_{nearest}$?
N
Y
rewire (X_{new} , Γ)
N
$\|X_{new}$ -$X_{goal}\|<\lambda$?
Y
return Γ

Figure 5. Flow chart of VK-RRT*.

As shown in Figure 6, the dark blue line is the route given by RRT*. However, it is impossible for surface ships to track the polyline as the large inertia.

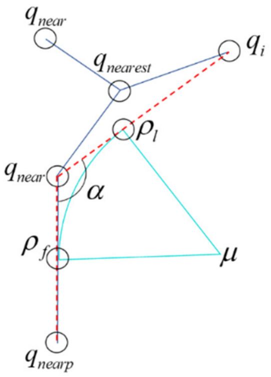


Figure 6. Route optimization considering ship motions.

Moreover, ships may encounter dangers in waypoint areas, as such in areas that have not been checked for safety in the route planning stage.

In fact, when the ship changes its course, the trajectory will be an arc line. Therefore, arc line segments in waypoint areas are designed. Navigating along the arc line, the ship only needs to give a fixed rudder angle to achieve tracking.

For example, Figure 7 shows the sketch map. The route $q_{nearq} \rightarrow q_{near} \rightarrow q_{nearest}$ is not suitable for a ship to track. Instead, we used $q_{nearq} \rightarrow \rho_f \rightarrow \rho_l \rightarrow q_i$ to replace it. The new route is determined by using Equations (1) and (2).

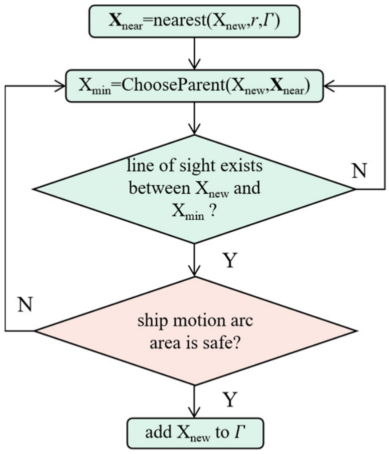

$\mathbf{X}_{near}$=nearest(X_{new},r,Γ)
X_{min}=ChooseParent(X_{new},$\mathbf{X}_{near}$)
line of sight exists between X_{new} and X_{min} ?
N
Y
N
ship motion arc area is safe?
Y
add X_{new} to Γ

Figure 7. Rewire function of VK-RRT*.

Specifically, the main task is to determine the distance of $\rho_f \rightarrow q_{near}$: ν. We suggest the following equations

$$\nu = \frac{R}{\tan((DN(q_{nearp} \rightarrow q_{near}) - DN(q_{near} \rightarrow q_i))/2)}, \tag{3}$$

where DN is the function used to measure the direction of a vector from the north; R is the turning radius determined by Equations (1) and (2).

Then, we can obtain the latitude and longitude of ρ_f and ρ_l.

$$\begin{cases} P_{\rho_f}(x,y) = & q_{nearp}(x,y) + (1 - \frac{\nu}{\|q_{near}(x,y) - q_{nearp}(x,y)\|}) \\ & \cdot(q_{near}(x,y) - q_{nearp}(x,y)) \\ P_{\rho_l}(x,y) = & q_{near}(x,y) + (1 - \frac{\nu}{\|q_{near}(x,y) - q_i(x,y)\|}) \\ & \cdot(q_i(x,y) - q_{near}(x,y)) \end{cases}, \tag{4}$$

In detail, VK-RRT* could plan the route that satisfies the requirement of ship motion constraints.

4. Simulation and Analysis

In order to verify the effectiveness and advantages of VK-RRT*, we carried out two simulations. Moreover, in our tests, we used the ships described in [33]. In the first scenario, we aimed to examine the ability of the proposed algorithm to plan route passing through narrow channels. In the second scenario, we tried to examine it in an open area. In each experiment, we compared the proposed algorithm with the RRT and RRT* algorithms to present the advantages of our algorithm. Both experiments have been conducted on ECDIS platform, which means that the frame we propose could be applied after professional packaging.

Moreover, in the process of choosing new points, we could add goal orientation to accelerate the search. This kind of strategy has been extensively studied in [35–40].

$$q_{rand}\begin{cases} q_{goal}, & \omega > \omega_d \\ x \in \chi_{free}, & \omega \leq \omega_d \end{cases}, \tag{5}$$

where ω is a random number, ω_d is set parameter of the adjustment.

4.1. Simulation in Islands

In this simulation, we use the electronic nautical chart numbered "US5M16IM.000", shown in Figure 8.

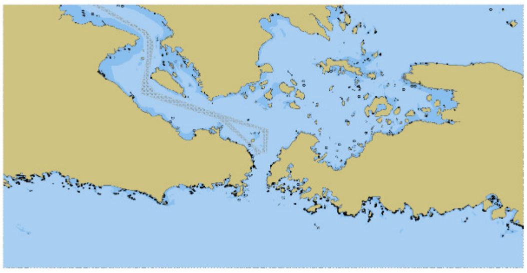
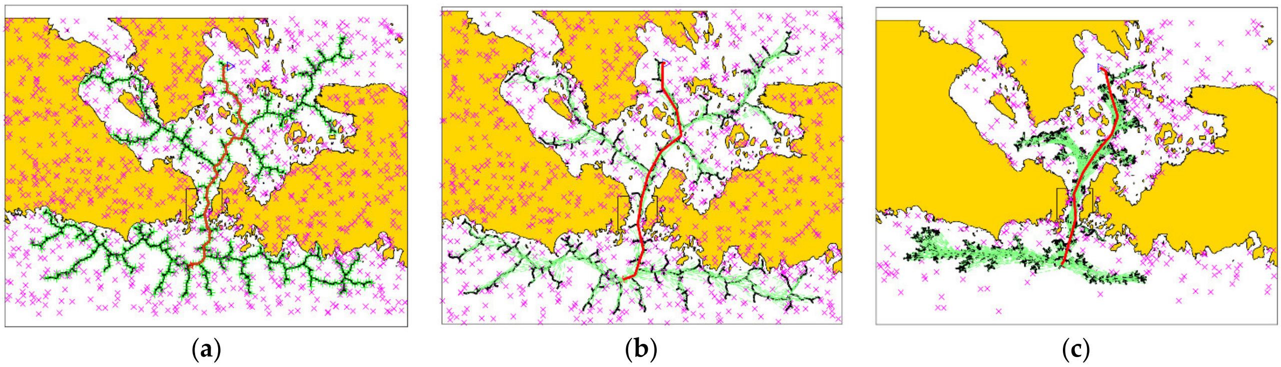


Figure 8. Electronic nautical chart numbered "US5M16IM.000".

First, the positions of the departure point and destination are (89.909° W, 45.924° N) and (83.853° W, 46.924° N) separately. Moreover, we assume that the ship will steer in a rudder angle of $\delta = 15°$ altering its course. In this experiment, the ability to search the route passing through narrow channels is mostly validated.

The results are illustrated in Figures 9–13, in which the red line is the planned route given by algorithms with different strategies; the blue lines construct the rapid-explore random tree, and the black lines represent the search process.



(**a**) (**b**) (**c**)

Figure 9. Planned routes without goal orientation in islands areas: (**a**) Given by the RRT algorithm; (**b**) Given by the RRT* algorithm; (**c**) Given by the proposed algorithm.

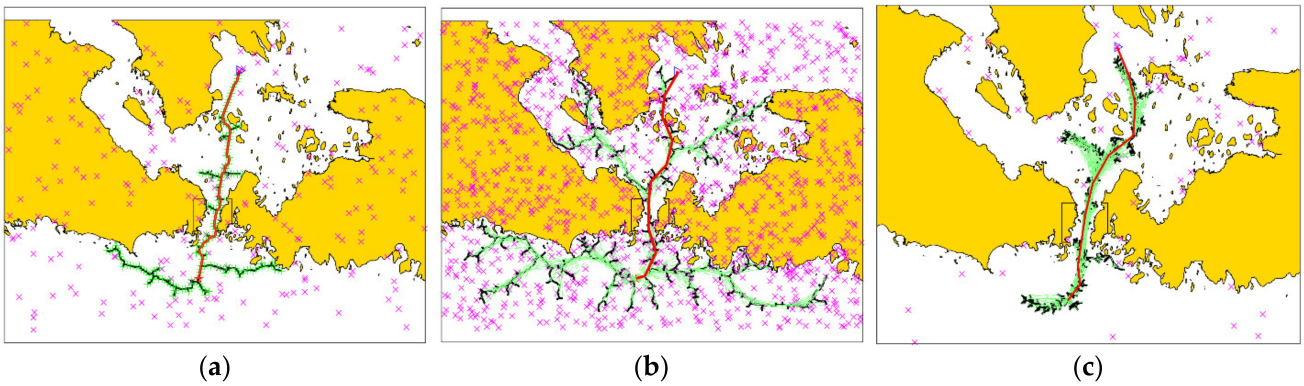


Figure 10. Planned routes with goal orientation in island areas: (**a**) Given by the RRT algorithm; (**b**) Given by the RRT* algorithm; (**c**) Given by the proposed algorithm.

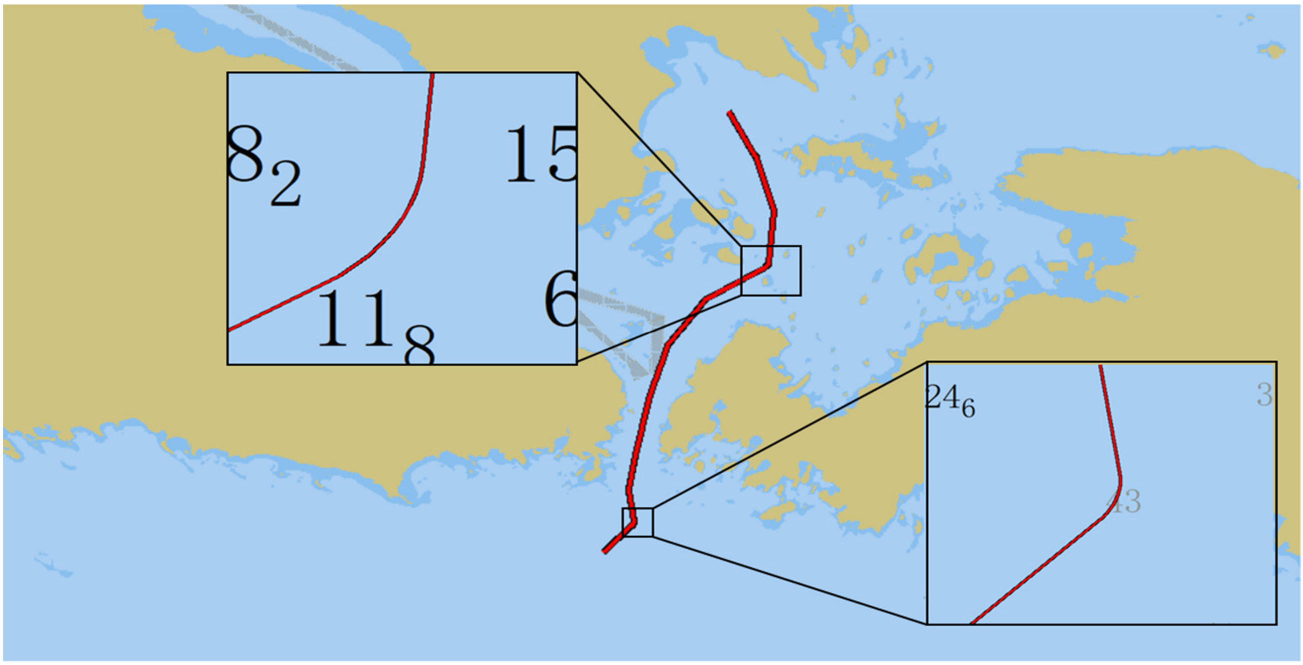


Figure 11. Overview of the planned route designed by the proposed algorithm on electric nautical chart.

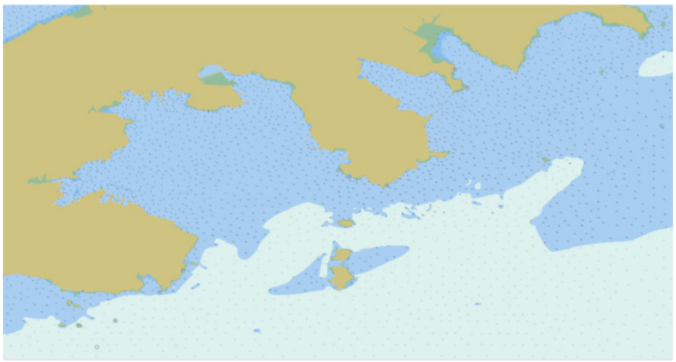


Figure 12. Electronic nautical chart of Dalian port.

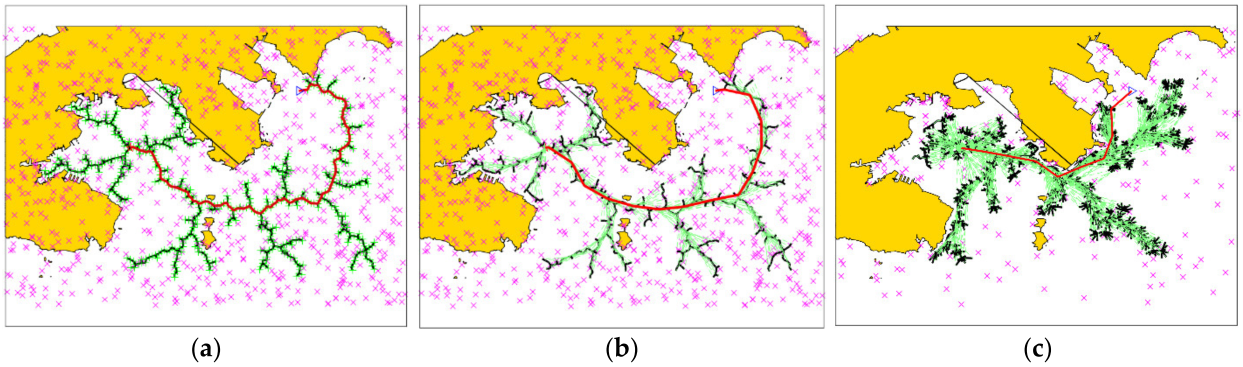


Figure 13. Planned routes without goal orientation in the port: (**a**) Given by the RRT algorithm; (**b**) Given by the RRT* algorithm; (**c**) Given by the proposed algorithm.

Specifically, in Figure 9a, the RRT algorithm gives the planning route in 170 s. It sampled the entire area and constructed bigger trees compared to other algorithms. The length of the route is 22.62 nautical miles, with 84 waypoints. Then, in Figure 9b, the RRT* spends 167.24 s to plan the route. Because of its advance, the length of the route is 19.65, which is shorter than that given by the RRT algorithm. The result of the proposed algorithm is shown in Figure 9c. The time cost has been reduced significantly and it only took 129.41 s to provide the planned route. The length of that route is also the shortest: 18.72 nautical miles. We can also observe that in the proposed algorithm, most of nodes searched were lump together, which improves search effectiveness.

Figure 10 illustrates the results of comparisons of the RRT, RRT*, and proposed algorithms considering goal orientations. The time consumed using the above three algorithms is reduced significantly compared to the algorithms without goal orientation. Their time costs are 41.47 s, 161.72 s, and 39.81 s, respectively. The lengths of the routes are 19.55 nm, 18.91 nm, and 18.71 nm.

Moreover, we should also focus on the characteristics of the waypoint areas. In such places, ships will alter their courses, which could not be tracked by the route planned by the RRT and RRT* algorithms, as both of them create a polyline-based route.

Figure 11 shows the planned route on an electronic nautical chart. It is noticeable that the planned route consists of straight-line segments and arc-line segments. In waypoint areas, the route guides ship navigation from a straight line segment to another line segment in an arc-shaped trajectory. This kind of route takes ship motion characteristics into account. The arc line segments are checked for safety in the search process to could ensure safety.

To be specific, in navigation practice, the planned route was not considered by route optimization in waypoints areas. In previous studies, many researchers have used different kinds of technologies to smooth out the route, making it smoother to track. However, for ship control engineers, it is difficult to design a controller that could track curves well, especially considering the large inertia of ships. Instead, our strategy could let ship navigate at a fix rudder angle to track arc lines as they are designed in reference to ship turning trajectories.

4.2. Simulation in the Port

Here, we provide another simulation in the port area to further validate the feasibility of the proposed algorithm. The area is shown as Figure 12: Dalian Port, China.

In this scenario, the departure point and destination are set as (121.727° E, 38.964° N) and (121.944° E, 39.02° N), respectively. The other parameters of the simulation are the same as those of the first simulation.

Figures 13 and 14 demonstrate the search process of the RRT, RRT*, and proposed algorithms with or without goal orientation strategy. Without considering goal orientation (shown in Figure 12), the time cost of RRT, RRT*, and proposed algorithms is 97.08 s, 87.59 s, and 34.01 s, respectively. It is clear that the proposed algorithm has advantages in the time cost index. Moreover, the lengths of the routes of these algorithms are 32.79 nm, 27.45 nm, and 17.57 nm, respectively.

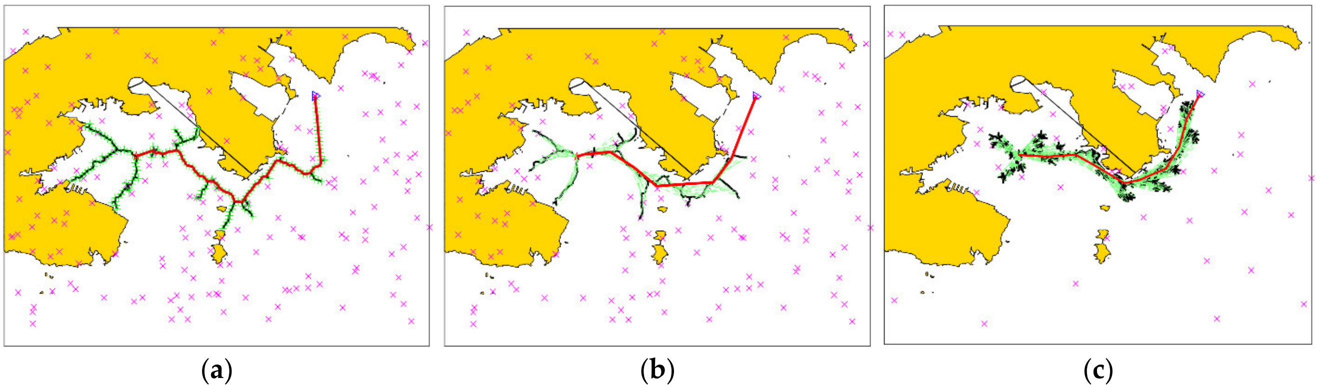

(a) (b) (c)

Figure 14. Planned routes with goal orientation in island areas: (**a**) Given by the RRT algorithm; (**b**) Given by the RRT* algorithm; (**c**) Given by the proposed algorithm.

Figure 14 illustrates the planned route given by three algorithms with goal orientation. Their time costs are 26.02 s, 19.94 s, and 9.49 s, respectively. The lengths of the routes given by the RRT, RRT*, and the proposed algorithms are 22.98 nm, 17.73 nm, and 17.22 nm. For the proposed algorithm, the goal orientation strategy only affects the time cost but has little impact on the length of the route.

Similarly, we also show the planned route on the electronic nautical chart for the convenience of analyzing waypoint areas.

We can see that in waypoint areas in Figure 15, the planned route is optimized to adapt ship motion characteristics.

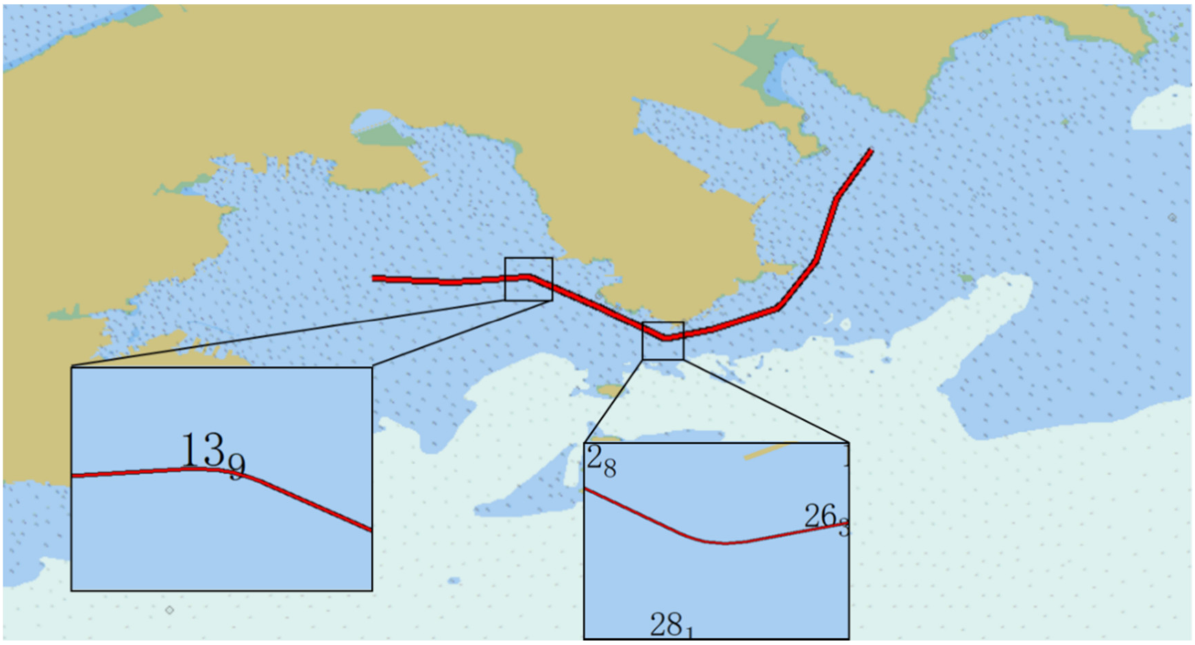


Figure 15. Overview of the planned route designed by the proposed algorithm on electric nautical charts.

In this section, we carry out two simulations validating the feasibility and advantages of the proposed algorithm. The result shows that the proposed algorithm meets the requirements of navigation and ensures safety.

5. Conclusions

This study introduces VK-RRT*, a novel algorithm for addressing SPR problems using ENC data. The algorithm uses Delaunay triangulation to organize ENC vector data, and searches for routes within it. Two cases are examined to demonstrate the effectiveness of VK-RRT*.

However, the study has some limitations that need to be addressed in the future. Firstly, this research only considers depth and land/reef areas; other traffic schemes, such as traffic separation, should also be considered. Secondly, local path planning, or collision avoidance, is not considered, which is also an important part for intelligent ships.

Moving forward, future research should focus on addressing these limitations to improve the effectiveness of the algorithm in practical applications. In addition, it is worth developing a route planning model that could be applied on the ECDIS platform. Finally, there should be a focus on local path planning considering dynamic vessels or obstacles if a better global route is to be obtained and smart navigation is to be achieved.

Author Contributions: Conceptualization, Q.H. and X.Z.; methodology, Q.H.; software, Z.H.; validation, Q.H. and Z.H., writing—original draft preparation, Q.H., writing—review and editing, X.Z.; visualization, Z.H.; supervision, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
2. Deng, Y.; Chen, Y.X.; Zhang, Y.J.; Mahadevan, S. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Appl. Soft Comput.* **2012**, *12*, 1231–1237. [CrossRef]
3. Noto, M.; Sato, H. A method for the shortest path search by extended Dijkstra algorithm. In Proceedings of the 2000 IEEE International Conference on Systems, Man and Cybernetics, Nashville, TN, USA, 8–11 October 2000; pp. 2316–2320.
4. Broumi, S.; Bakal, A.; Talea, M.; Smarandache, F.; Vladareanu, L. Applying Dijkstra algorithm for solving neutrosophic shortest path problem. In Proceedings of the 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), Melbourne, VIC, Australia, 30 November–3 December 2016; pp. 412–416.
5. Wang, H.L.; Mao, W.G.; Eriksson, L. A Three-Dimensional Dijkstra's algorithm for multi-objective ship voyage optimization. *Ocean. Eng.* **2019**, *186*, 106131. [CrossRef]
6. Zhu, X.; Wang, H.; Shen, Z.; Lv, H. Ship weather routing based on modified Dijkstra algorithm. In Proceedings of the 2016 6th International Conference on Machinery, Materials, Environment, Biotechnology and Computer, Tianjin, China, 11–16 June 2016; Atlantis Press: Amsterdam, The Netherlands, 2016.
7. Daniel, K.; Nash, A.; Koenig, S. Theta*: Angle angle path finding on grids. *J. Artif. Intell. Res.* **2010**, *39*, 533–579. [CrossRef]
8. Guruji, A.K.; Agarwal, H.; Parsediya, D.K. Time-efficient A* Algorithm for Robot Path Planning. *Procedia Technol.* **2016**, *23*, 144–149. [CrossRef]
9. Liu, C.; Mao, Q.; Chu, X.; Xie, S. An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning. *Appl. Sci.* **2019**, *9*, 1057. [CrossRef]
10. Xie, W.; Fang, X.; Wu, S. 2.5 D Navigation Graph and Improved A-Star Algorithm for Path Planning in Ship inside Virtual Environment. In Proceedings of the 2020 Prognostics and Health Management Conference (PHM-Besançon), Besancon, France, 4–7 May 2020; pp. 295–299.
11. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; University of Illinois Urbana Champaign: Champaign, IL, USA, 1998.
12. Braun, J.; Brito, T.; Lima, J.; Costa, P.; Costa, P.; Nakano, A. A Comparison of A* and RRT* algorithms with dynamic and real time constraint scenarios for mobile robots. In Proceedings of the 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Prague, Czech Republic, 29–31 July 2019; pp. 398–405.
13. Xiao, S.; Bergmann, N.; Postula, A. Parallel RRT$*$ architecture design for motion planning. In Proceedings of the 2017 27th International Conference on Field Programmable Logic and Applications (FPL), Ghent, Belgium, 4–8 September 2017; pp. 1–4.
14. Jeong, I.-B.; Lee, S.-J.; Kim, J.-H. Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. *Expert Syst. Appl.* **2019**, *123*, 82–90. [CrossRef]
15. Li, Y.; Wei, W.; Gao, Y.; Wang, D.; Fan, Z. PQ-RRT*: An improved path planning algorithm for mobile robots. *Expert Syst. Appl.* **2020**, *152*, 113425. [CrossRef]

16. Kuffner, J.J.; LaValle, S.M. RRT-connect: An efficient approach to single-query path planning. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; pp. 995–1001.
17. Klemm, S.; Oberländer, J.; Hermann, A.; Roennau, A.; Schamm, T.; Zollner, J.M.; Dillmann, R. RRT∗-Connect: Faster, asymptotically optimal motion planning. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 1670–1677.
18. Xinyu, W.; Xiaojuan, L.; Yong, G.; Jiadong, S.; Rui, W. Bidirectional potential guided RRT* for motion planning. *IEEE Access* **2019**, *7*, 95046–95057. [CrossRef]
19. Mashayekhi, R.; Idris, M.Y.I.; Anisi, M.H.; Ahmedy, I.; Ali, I. Informed RRT*-connect: An asymptotically optimal single-query path planning method. *IEEE Access* **2020**, *8*, 19842–19852. [CrossRef]
20. Lau, C.; Byl, K. Smooth RRT-connect: An extension of RRT-connect for practical use in robots. In Proceedings of the 2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, USA, 11–12 May 2015; pp. 1–7.
21. Enevoldsen, T.T.; Reinartz, C.; Galeazzi, R. COLREGs-Informed RRT* for Collision Avoidance of Marine Crafts. *arXiv* **2021**, arXiv:2103.14426.
22. 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]
23. Xiong, C.; Zhou, H.; Lu, D.; Zeng, Z.; Lian, L.; Yu, C. Rapidly-Exploring Adaptive Sampling Tree*: A Sample-Based Path-Planning Algorithm for Unmanned Marine Vehicles Information Gathering in Variable Ocean Environments. *Sensors* **2020**, *20*, 2515. [CrossRef] [PubMed]
24. Chiang, H.-T.L.; Tapia, L. COLREG-RRT: An RRT-based COLREGS-compliant motion planner for surface vehicle navigation. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2024–2031. [CrossRef]
25. Zilu, O.; Hongdong, W.; Jianyao, W.; Hong, Y. Automatic collision avoidance algorithm for unmanned surface vessel based on improved Bi-RRT algorithm. *Chin. J. Ship Res.* **2019**, *14*, 9.
26. Orozco-Rosas, U.; Picos, K.; Montiel, O. Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots. *IEEE Access.* **2019**, *7*, 156787–156803. [CrossRef]
27. Orozco-Rosas, U.; Montiel, O.; Sepúlveda, R. Mobile robot path planning using membrane evolutionary artificial potential field. *Appl. Soft Comput.* **2019**, *77*, 236–251. [CrossRef]
28. Orozco-Rosas, U.; Picos, K.; Pantrigo, J.J.; Montemayor, A.S.; Cuesta-Infante, A. Mobile robot path planning using a QAPF learning algorithm for known and unknown environments. *IEEE Access* **2022**, *10*, 84648–84663. [CrossRef]
29. Shen, H.; Hashimoto, H.; Matsuda, A.; Taniguchi, Y.; Terada, D.; Guo, C. Automatic collision avoidance of multiple ships based on deep Q-learning. *Appl. Ocean. Res.* **2019**, *86*, 268–288. [CrossRef]
30. 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. [CrossRef]
31. Zhao, B.; Zhang, X.; Liang, C. A novel path-following control algorithm for surface vessels based on global course constraint and nonlinear feedback technology. *Appl. Ocean. Res.* **2021**, *111*, 102635. [CrossRef]
32. IHO. *IHO Transfer Standard for Digital Hydrographc Data*, 3.1 ed.; Special Publication No. 57; IHO: Monaco, France, 2000.
33. Zhang, Q.; Zhang, X.-K.; Im, N.-K. Ship nonlinear-feedback course keeping algorithm based on MMG model driven by bipolar sigmoid function for berthing. *Int. J. Nav. Archit. Ocean. Eng.* **2017**, *9*, 525–536. [CrossRef]
34. Nguyen, C.M.; Rhodes, P.J. Delaunay triangulation of large-scale datasets using two-level parallelism. *Parallel Comput.* **2020**, *98*, 102672. [CrossRef]
35. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
36. Liu, H.; Zhang, X.; Wen, J.; Wang, R.; Chen, X. Goal-biased bidirectional RRT based on curve-smoothing. *IFAC-Pap.* **2019**, *52*, 255–260. [CrossRef]
37. Urmson, C.; Simmons, R. Approaches for heuristically biasing RRT growth. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, NV, USA, 27–31 October 2003; Volume 2, pp. 1178–1183.
38. Jin, X.; Yan, Z.; Yang, H.; Wang, Q.; Yin, G. A goal-biased RRT path planning approach for autonomous ground vehicle. In Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 18–20 December 2020; pp. 743–746.
39. Ayawli, B.B.; Mei, X.; Shen, M.; Appiah, A.Y.; Kyeremeh, F. Optimized RRT-A* path planning method for mobile robots in partially known environment. *Inf. Technol. Control* **2019**, *48*, 179–194. [CrossRef]
40. Liang, C.; Zhang, X.; Han, X. Route planning and track keeping control for ships based on the leader-vertex ant colony and nonlinear feedback algorithms. *Appl. Ocean. Res.* **2020**, *101*, 102239. [CrossRef]

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.