

Article

Improved RRT Algorithm for AUV Target Search in Unknown 3D Environment

Juan Li , Chengyue Li, Tao Chen * and Yun Zhang

Institute of Ocean Installations and Control Technology, College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; lijuan041@hrbeu.edu.cn (J.L.); lichengyue@hrbeu.edu.cn (C.L.); 18341315841@163.com (Y.Z.)

* Correspondence: chentao409@hrbeu.edu.cn

Abstract: Due to the complexity of the marine environment, underwater target search and interception is one of the biggest problems faced by an autonomous underwater vehicle (AUV). At present, there is quite a lot of research in terms of the two-dimensional environment. This paper proposes an improved rapidly exploring random trees (RRT) algorithm to solve the problem of target search and interception in an unknown three-dimensional (3D) environment. The RRT algorithm is combined with rolling planning and node screening to realize path planning in an unknown environment, and then the improved RRT algorithm is applied to the search and interception process in a 3D environment. Combined with the search decision function and the three-point numerical differential prediction method, the RRT algorithm can search for and effectively intercept the target. Numerical simulations in various situations show the superior performance, in terms of time and accuracy, of the proposed approach.

Keywords: target search; AUV; path planning; RRT; interception



Citation: Li, J.; Li, C.; Chen, T.; Zhang, Y. Improved RRT Algorithm for AUV Target Search in Unknown 3D Environment. *J. Mar. Sci. Eng.* **2022**, *10*, 826. <https://doi.org/10.3390/jmse10060826>

Academic Editor: Jacopo Aguzzi

Received: 21 April 2022

Accepted: 14 June 2022

Published: 17 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

An autonomous underwater vehicle (AUV) is a powerful tool for exploring the unknown marine environment. With the development of science and society, artificial intelligence, electronic computers, and other high-tech performance technologies have made continuous progress. In most countries, AUVs have become the focus of marine technology research [1,2].

With the wide application range of AUVs, there are increasing restrictions and limitations regarding research using AUVs. Underwater unmanned technology is only a few decades old. The existence of fish and glaciers in the marine environment make it difficult for AUVs to carry out underwater unmanned operation, while an AUV's own energy consumption and communication restriction also limit the efficiency of underwater missions. Research on how to realize the target search and complete the corresponding tasks scientifically and efficiently in the complex 3D environment has been gaining popularity among scholars [3–5].

Juan [6] proposed a multi-AUV target search method based on dynamic prediction from two aspects, (i) improving underwater target search efficiency and (ii) positioning accuracy, and formulated three search strategies according to the existence of targets in the environment. However, the algorithm is simulated only in the two-dimensional environment and should be extended to the 3D environment before it is applied in practice, so as to meet the practical application scenarios. Jianjun [7] proposed a search algorithm based on improved DSA for AUV collaborative target search in a 3D environment. The algorithm improves the detection ability of the DSA algorithm by synthesizing the Levyflight algorithm based on fuzzy rules and is verified and compared with the general DSA algorithm and the PSO algorithm in the simulation. Although a search method for a 3D environment is proposed, the search time exceeds expectations. Ishida [8] proposed the moving target

search (MTS) algorithm for dynamic target search and introduced the heuristic idea into the process of dynamic target search. In the previous target search problems, the targets were static or knowable, but in this paper, the author assumed the target to be moving in an unknown environment, and the effectiveness of the MTS algorithm is verified in the simulation. Yug [9] applied the reinforcement learning technology to UAV search and rescue missions and used a reinforcement learning algorithm to predict the existence of survivors. Considering the possible movement of survivors, the YOLO algorithm is applied to the survivor tracking process. With the goal of target search and rescue in ocean rescue missions, Wang [10] and others proposed a multi-searcher, multi-target problem algorithm that decomposed the problem into two parts: search target assignment and search strategy selection. Three search strategies (spiral search, outer spiral search, and circumferential patrol) are selected according to the actual situation, and the real environment data are simulated in the simulation process. Ibenthal [11] proposed an estimation algorithm in which each UAV evaluates three different sets. One set is guaranteed to contain the detected target, one set may belong to the target that has not been detected, and one set is proved to contain no target. The algorithm is used to search and track a constantly changing dynamic target in a certain area where there are interference targets and obstacles.

The RRT algorithm is widely used in path planning because of its rapidity and extensibility. Yin [12] proposed a modified root node reselect RRT algorithm (RNR-RRT), which integrates the UAV dynamics constraints into the node expansion process by improving the root node selection strategy. The search tree will optimize the planned track, and the track smoothing method based on the B-spline curve is used to generate smooth traceable track. Wu [13] proposed a variable-probability-based bidirectional RRT algorithm (VPB-RRT). Using this algorithm to identify the probability by rasterizing the planning space and importing the concept of coverage rate has made UAV path planning satisfactory in terms of time and accuracy. Guo [14] proposed a flight-cost-based rapidly exploring random tree star (FC-RRT*) extending the standard rapidly exploring random tree star (RRT*) to deal with the safety requirements and flight constraints of UAVs in a complex 3D environment, which effectively overcomes the shortcomings of standard RRT* and has application value in UAV 3D global path planning. The main aspects of the aforementioned algorithms are summarized in Table 1.

Table 1. Main aspects of the aforementioned algorithms.

Algorithm	Optimization Methods	Merit	Deficiency
RNR-RRT	The selection strategy of the root node is improved.	B-spline curve trace smoothing Turn angle constraint Track distance constraint	2D environment
VPB-RRT	The offset of the new node is determined by the map coverage.	Rasterizing the planning space Import coverage rate Turn angle constraint	Tendency to create a local optimal solution
FC-RRT*	The flight cost function is used to inspire the expansion of new nodes and guide the update of the parent node.	Complex 3D environment Flight constraints Improved path safety	Excessive number of nodes

These improved RRT algorithms proposed for a UAV with a high degree of freedom and flexibility. While the AUV moves in the water, it cannot realize either high frequency turning or large-range turning. And the altitude angles, especially the pitch angle, cannot be maintained at a large angle continuously, unlike a UAV. These algorithms concentrate on the obstacles or restricted areas to improve the performance. However, in most underwater search cases, there are usually not too many large or complex obstacles, but the aim is to quickly traverse the region or search for all the targets. Therefore, this paper proposes an improved RRT algorithm for the target search task.

The main contribution of this paper is twofold. First, with the above background in mind, a search strategy for AUV target search is proposed, which takes the influence of environmental information, including obstacles, target probability, and regional ergodicity, into account so as to realize the complete search and confirmation of static targets and the interception of dynamic targets in an 3D unknown environment. Second, the improved RRT algorithm is combined with rolling planning, node screening, and secondary selection of the parent node to obtain a feasible path from the current position to the search decision point.

The rest of the paper is organized as follows. In Section 2, the problems studied in this paper are described and the AUV kinematic model, the sonar model, and the environment map model are constructed. In Section 3, a search decision function based on environmental information and an improved RRT algorithm in a 3D environment are proposed, which help reasonably in creating a calculation search decision point and path planning that conform to AUV kinematic characteristics. In Section 4, aiming at dynamic targets, a target interception algorithm based on a three-point numerical differential method is proposed. Section 5 provides the details of simulation setup and results, which show that the algorithm can safely and effectively realize static target confirmation and dynamic target interception with ideal navigation trajectories. Section 6 summarizes the conclusions and future work.

2. Problem Description and Modeling

2.1. Problem Description

The problem an AUV faces during target search can be described as follows: An AUV is required to locate as many targets as possible, while minimizing the search costs and maximizing the search efficiency. The 3D task search area is expanded into an $L_x \times L_y \times L_z$ cuboid. The target area is rasterized and divided into $M \times N \times K$ 3D actual grids of size $100 \text{ m} \times 100 \text{ m} \times 100 \text{ m}$. There are N_i static and N_j dynamic targets randomly distributed in the unknown mission area. The position of the static target must be calibrated after close confirmation when the AUV does not find a dynamic target. Otherwise, it is necessary to intercept the dynamic target and then continue to search the mission area. Obstacles need to be avoided during mission execution.

The simulations need to be carried out with certain assumptions regarding the search environment, which are as follows:

Assumption 1: Obstacles remain stationary and are expanded into spheres with different radii. Obstacles in the real environment are of different shapes and sizes, which may cause unnecessary computational burden. Thus, the obstacles are simplified as spheres according to the maximum edge to ensure the safety of AUV navigation, while reducing the computation as much as possible.

Assumption 2: The AUV and targets are all regarded as mass points, and it is assumed that the dynamic targets move at the same depth and perform a uniform linear motion, a uniform acceleration linear motion, or a uniform circular motion.

During target search, the search map is updated in real time according to the sonar, and the AUV decides the area to be searched in the next instance according to the decision function. The AUV will intercept the dynamic target once it is found and approach the static target for confirmation. The improved rolling RRT algorithm is used for path planning. The flow of the search process is shown in Figure 1.

The decision function is used to determine the search decision point, and the improved RRT algorithm is used to plan the path from the current position to the target position. These two parts will be elaborated in Section 3. The target interception algorithm will be explained in detail in Section 4.

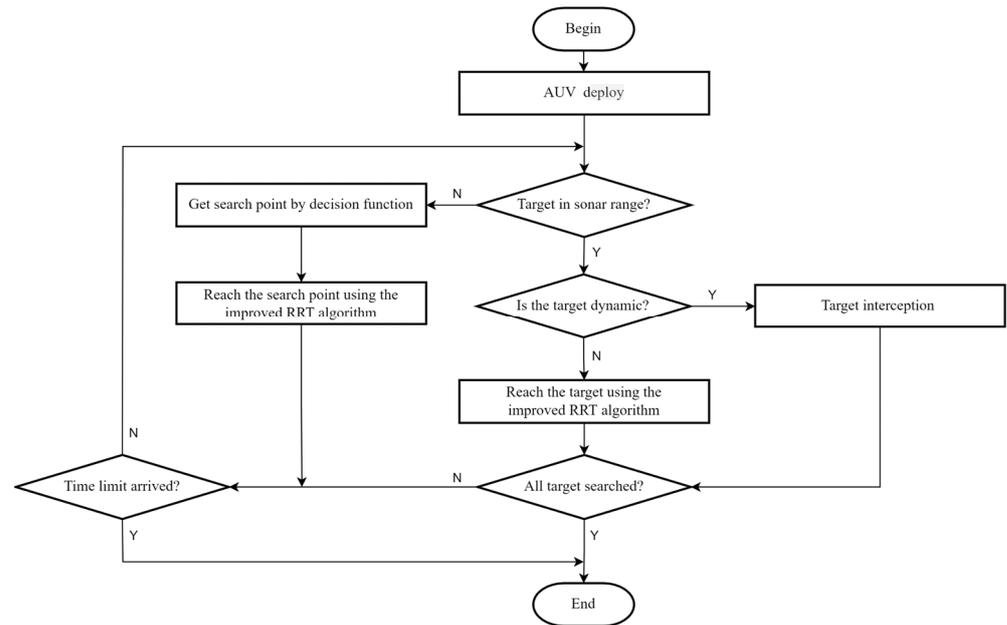


Figure 1. Search process flowchart.

2.2. AUV Kinematic Model

As mentioned above, the AUV can be simplified as a typical three-dimensional mass point model. $U_k = \{x_k, y_k, z_k, \varphi_k, \theta_k\}$ is defined as the AUV state at time k , where $\{x_k, y_k, z_k\}$ represents the position of the AUV at time k , φ_k represents the AUV yaw angle, and θ_k represents the AUV pitch angle [15]. The AUV kinematics equation can be simplified as:

$$U_{k+1} = U_k + f(U_k, v_k, \dot{\varphi}_k, \dot{\theta}_k) \tag{1}$$

$$f(U_k, v_k, \dot{\varphi}_k, \dot{\theta}_k) = \begin{bmatrix} v_k \cos(\theta_k + \dot{\theta}_k \cdot \Delta t) \cos(\varphi_k + \dot{\varphi}_k \cdot \Delta t) \cdot \Delta t \\ v_k \cos(\theta_k + \dot{\theta}_k \cdot \Delta t) \sin(\varphi_k + \dot{\varphi}_k \cdot \Delta t) \cdot \Delta t \\ v_k \sin(\theta_k + \dot{\theta}_k \cdot \Delta t) \cdot \Delta t \\ \varphi_k + \dot{\varphi}_k \cdot \Delta t \\ \theta_k + \dot{\theta}_k \cdot \Delta t \end{bmatrix} \tag{2}$$

where v_k represents the AUV moving speed, which is set to 3 m/s (about 6 kn). Then AUV should satisfy the following constraints while searching:

$$\begin{cases} -\pi/6 \leq \theta_k \leq \pi/6 \\ -\pi/6 \leq \dot{\theta}_k \leq \pi/6 \\ -\pi/3 \leq \dot{\varphi}_k \leq \pi/3 \end{cases} \tag{3}$$

2.3. Sonar Model

To realize omnidirectional detection centered on the AUV, 360-degree scanning sonar is selected to obtain spherical scanning data [16]. To reduce the difficulty of research, the sonar detection range is idealized, which means targets within a range of 100 m can be perceived. Assuming that the coordinates of the AUV are (x_0, y_0, z_0) and the target coordinates are (x, y, z) , the sonar can detect the target if the coordinates meet the following conditions:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 \leq r^2 \tag{4}$$

The sonar sensor model is shown in Figure 2. It is assumed that the sonar can obtain environmental information within the spherical detection range.

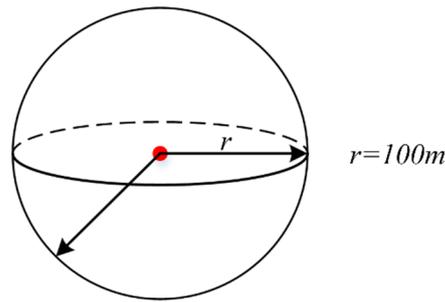


Figure 2. Sonar sensor model.

2.4. Real-Time Perception Map Model

2.4.1. Target Probability Map

The target probability map represents the probability of the existence of a target in a 3D environment [17]. The target probability map of the 3D environment is defined in Equation (5).

$$M_{TP}(k) = \{P_a(k) | a \in \Omega\} \tag{5}$$

where $P_a(k)$ represents the target probability in grid a at time k . If $P_a(k) = 1$, it means that the AUV considers that there is a target in grid a at moment k . The opposite is true for $P_a(k) = 0$. Ω indicates the global environment. Since the environment is unknown, the possibility of the existence of targets in grid a is the same before the AUV performs the task, $P_a(0) = 0.5$.

The target probability update equation of grid a at time k is:

$$P_{a,k+1} = \frac{P_d \cdot P_{a,k}}{P_d \cdot P_{a,k} + (1 - P_f) \cdot (1 - P_{a,k})} \tag{6}$$

where P_d is the detection probability, which represents the probability of finding the target in the grid where the target exists. P_f is the false alarm probability, indicating the probability of detecting a target in a grid where there is no target.

For the convenience of calculation, the target probability formula is transformed into a linear form by nonlinear transformation [11]:

$$Q_{a,k} = \ln\left(\frac{1}{P_a(k)} - 1\right) \tag{7}$$

2.4.2. Uncertainty Map

An uncertainty map [11] reflects the AUV’s understanding of a certain grid in the environment. As the AUV visits the grid, the target probability in the grid increases or decreases. Either way, the AUV’s judgment of the grid becomes more accurate. The calculation equation of uncertainty is:

$$\mu_{a,k} = e^{-l \cdot |Q_{a,k}|} \tag{8}$$

where $\mu_{a,k}$ represents the uncertainty of grid a at time k and l is the attenuation parameter, which measures the rate at which the uncertainty decays with the change in the target probability.

2.4.3. Regional Ergodicity Map

Region ergodicity indicates the degree to which the area is traversed. The region ergodicity of each grid in the environment is fluctuant at different times, which is updated over time. $B_a(k)$ is used to represent the region ergodicity of grid a at time k . Considering

that all grids in the environment at the initial time have not been traversed, $Ba(0) = 0, a \in \Omega$ is set.

$$B_a(k) = \begin{cases} 0, & 0 < \frac{V_s}{V} \leq 0.35 \\ 0.5, & 0.35 < \frac{V_s}{V} \leq 0.75 \\ 1, & 0.75 < \frac{V_s}{V} \leq 1 \end{cases} \quad (9)$$

where V_s represents the volume that has been searched in grid a , and V represents the volume of grid a .

3. Three-Dimensional Environment Target Search Algorithm

3.1. Search Decision Function

Reducing the uncertainty and improving the regional ergodicity of the entire environment can effectively improve the efficiency of the search [18]. To improve the AUV's target search capability, we need to:

- Improve the AUV's awareness of environmental information;
- Avoid the rudder loss caused by frequent steering;
- Maximize the number of confirmed targets within a certain period of time.

To meet the above requirements, a search decision function is established by comprehensively considering three aspects: uncertainty benefit, search task benefit, and regional ergodicity benefit. The AUV obtains the best search decision point by solving the search decision function, which represents the area the AUV needs to search at the next instance. Then the improved RRT algorithm is used to plan the search path.

3.1.1. Uncertainty Benefit

The uncertainty benefit $I_A(a, k)$ represents the sum of uncertainty of all grids in the detection area, with grid a as the center and the sensor detection range as the radius at time k [11].

$$I_A(a, k) = \sum_{b \in \phi(a)} \mu_{b,k} \quad (10)$$

Net $\phi(a)$ represents all grids within the detection range of the AUV with a as the center.

3.1.2. Search Task Benefit

The search task benefit $I_B(a, k)$ represents the sum of the path length cost and the steering angle cost of the selectable search decision points at time k , with grid a as the center. The higher the cost of the search decision point, the lower the search task benefit, and vice versa.

$$I_B(a, k) = k_1 \cdot dis(Si, k) + k_2 \cdot turn(Si, k) \quad (11)$$

In the equation, $dis(Si, k)$ represents the path length cost from the current position to the candidate search decision point Si at time k , and $turn(Si, k)$ represents the steering angle cost from the current position to Si at time k .

Assuming that the current coordinates of the AUV are (x_a, y_a, z_a) and the coordinates of the candidate search point within the detection range are (x_1, y_1, z_1) , then the path length cost $dis(Si, k)$ of the AUV at time k is:

$$d_1 = \sqrt{(x_1 - x_a)^2 + (y_1 - y_a)^2 + (z_1 - z_a)^2} \quad (12)$$

$$dis(k) = \tau \cdot d_1 \quad (13)$$

As mentioned above, $turn(Si, k)$ is the steering angle cost of the AUV at the current position Si at time k . In some studies, scholars directly discard the points with a large steering angle, ignoring the situation where the possible target is around. Considering the wide range of underwater operations, if the target is near the AUV but the AUV chooses to sail along the given course due to the excessive steering angle, the opportunity to discover the target may be lost. To avoid this problem, an inertia weight coefficient is introduced.

Combined with the actual situation, the steering angle boundary value is set for the AUV. Only the inertia weight coefficient of the search decision point whose required steering angle exceeds the boundary is not 0 and follows the law that the greater the excess of steering angle over the boundary, the greater the coefficient [19]. The equation of the inertia weight coefficient is defined as:

$$w = \begin{cases} 0 & \beta \leq \alpha \\ \frac{k}{1+e^{-\frac{5\beta}{\alpha}}} & \beta > \alpha \end{cases} \tag{14}$$

Therefore, the steering angle cost is defined as:

$$turn(k) = w \cdot |\alpha - \beta| \tag{15}$$

where β is the steering angle of the AUV, α is the boundary value, and k is a constant.

3.1.3. Regional Ergodicity Benefit

Regional ergodicity benefit $I_D(a, k)$ represents the ergodic situation in the region with grid a as the center and the AUV detection range as the radius at time k .

$$I_D(a, k) = \sum_{b \in \phi(a)} B_b(k) \tag{16}$$

Regional ergodicity indicates the degree to which the region is searched and traversed, and it will be updated over time.

Combining the uncertainty benefit, the search task benefit, and the regional ergodicity benefit, the search decision function is defined as $f(a, k)$, which represents the search value of grid a at time k . The specific search decision function is defined as:

$$f(a, k) = \omega_a \cdot I_A(a, k) + \omega_b \cdot I_B(a, k) + \omega_c \cdot I_D(a, k) \tag{17}$$

where ω_a , ω_b , and ω_c are the weight coefficients of uncertainty benefit, search task benefit, and regional ergodicity benefit, respectively, which can be adjusted according to changes in the environment.

3.2. Path Planning Based on the Improved RRT Algorithm

The improved RRT algorithm is used for path planning while searching so as to reach the search decision point from the current position.

3.2.1. RRT Algorithm

The rapidly exploring random tree (RRT) is a sampling-based path planning algorithm [20]. It is designed to search non-convex and high-dimensional spaces efficiently. By randomly generating samples in a continuous space, it inherently fills the space toward to unexplored areas uniformly.

First, the algorithm randomly samples in the motion space to get a random node p_{rand} . It then locates the node p_{near} with the smallest Euclidean distance from p_{rand} on the random tree, expands in the direction of the random node p_{rand} with a step size r , and acquires a new node p_{new} . If the connection between p_{near} and p_{new} passes the collision test, which means there is no obstacle between the two points, then the new node p_{new} and the connection will be added to the random tree. Otherwise, the random node will be reselected for expansion and detection. The abovementioned steps of random sampling and node expansion are repeated until the distance between the random tree and the target point is less than the threshold or the number of random sampling times exceeds the limitation. The planned path can be obtained by connecting the nodes in the random tree from the root node to the target point. The RRT algorithm flow is shown in Figure 3. On the basis of

the two-dimensional expansion, the expansion in the 3D environment increases the search dimension and computational complexity. The expansion process is shown in Figure 4.

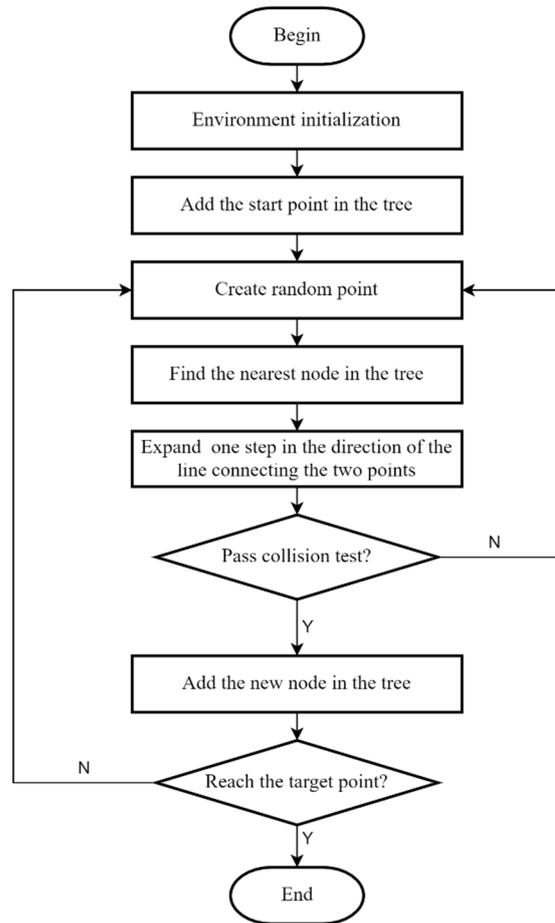


Figure 3. RRT algorithm flow.

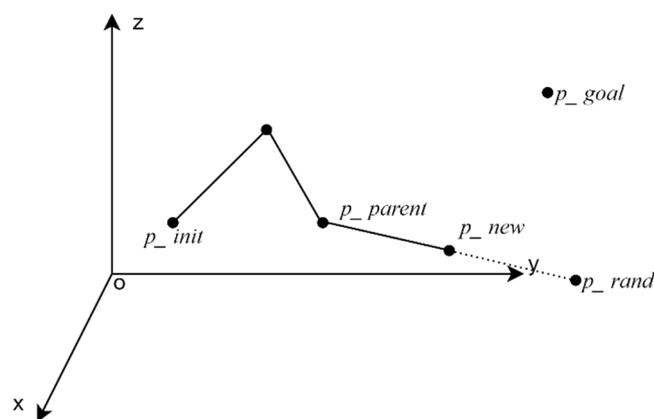


Figure 4. Tree expansion of the RRT in a 3D environment.

3.2.2. Improved RRT Algorithm

Generally, the quality of the path planned by the RRT algorithm is unsatisfactory with obvious corners, low smoothness, superabundant nodes, poor reproducibility, and other defects. The RRT algorithm can be improved by (i) introducing a rolling window in the planning process to enhance path quality and solving the problem of obstacle avoidance in an unknown environment and (ii) reducing the useless nodes generated in the random

tree expansion by introducing the node screening strategy and the secondary selection of parent nodes.

1. Rolling Planning

The search environment of the AUV is usually complex and changeable, and it is often difficult to obtain complete global information, which means it is necessary to ensure that the AUV perceives the environment and plans its path in real time. In an unknown environment, the acquisition of environmental information depends on the sensors carried by the vehicle. To plan the path in an unknown environment, it is necessary to repeatedly perform local planning, via rolling planning, using the obtained environmental information, so as to replace the result of global planning [21].

In the rolling process, on the basis of the detected environmental information, a local environment model centered on the AUV is established and the optimal sub-target point is obtained according to the sub-target point selection strategy. When approaching the sub-target point, the local path is supplemented and corrected according to the new environmental information [22], and a new sub-target point is established. In this paper, the rolling window is defined on the basis of the sonar model.

Rolling planning is mainly divided into three steps: (1) A local environment model centered on the position of the vehicle is established in each rolling according to the environmental information detected by the vehicle. (2) The local path is planned. According to the sub-target point selection strategy, the next optimal sub-target point is obtained, and the path planning is carried out toward the sub-target point in the local environment model. (3) Correction made considering feedback information is introduced for the next rolling step. When the AUV heads toward the sub-target point, the algorithm establishes a new local environment model, and the local path planned according to the local environment at the previous moment is corrected.

2. Sub-target Point Selection

In the complex underwater environment, there may be a variety of situations and, accordingly, different sub-target selection strategies. If the final target point p_goal is within the window, the final target is marked as the sub-target point (Figure 5a). If p_goal is not within the window and the intersection of the line between the current position point p_init and p_goal with the rolling window is not in the obstacle, the intersection is selected as the sub-target point (Figure 5b). If the intersection is within the obstacle, the sub-target point moves up and down alternately on the window until it is not within the obstacle (Figure 5c).

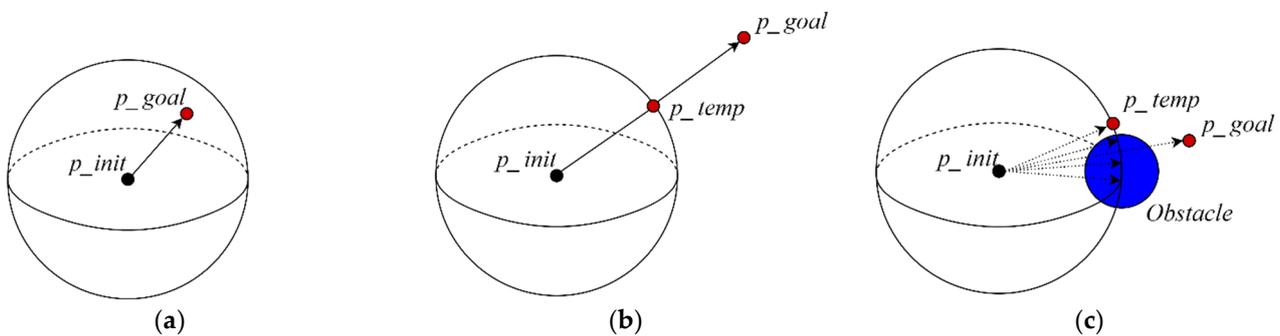


Figure 5. Sub-target point selection strategy: (a) target point within the window; (b) the intersection not in the obstacle; (c) the intersection in the obstacle.

According to the above three situations, the equations used to determine the sub-target points, respectively, are:

$$(a) = \begin{cases} d(p_init, p_goal) \leq R \\ p_temp = p_goal \end{cases} \tag{18}$$

$$(b) = \begin{cases} d(p_init, p_temp) = R \\ p_temp = \min(d(p_init, p_goal)) \end{cases} \quad (19)$$

$$(c) = \begin{cases} d(p_init, p_temp) = R \\ Collisiontest(p_temp) = true \end{cases} \quad (20)$$

where $d(p_init, p_goal)$ represents the distance between p_init and p_goal and $Collisiontest(p_temp)$ is a function to detect whether p_temp is in the obstacle. If $Collisiontest(p_temp) = true$, it means that p_temp is not in the obstacle. Otherwise, it means that p_temp is in the obstacle.

3. Node Screening

To ensure that the planning result is an executable path for the AUV, it is necessary to screen the nodes, which consists of two parts: (1) The angle is projected to the plane xoy and the plane $yozy$, respectively, to acquire angles μ_1 and μ_2 . In addition, μ_1 and μ_2 are limited to being acute angles or the expansion is given up to avoid selecting nodes in the opposite direction to the target. The projection process is shown in Figure 6. (2) The projection angles α_1 and α_2 of the angles on plane xoz and plane $yozy$ should not exceed 30 degrees, and the projection angle β on plane xoy should not exceed 60 degrees. Meanwhile, considering the limitation of the AUV's attitude in water, the pitch angle θ formed by (p_parent, p_new) and plane xoy should not exceed 30 degrees so that it meets the kinematic characteristics of the AUV. The projection process is shown in Figure 7.

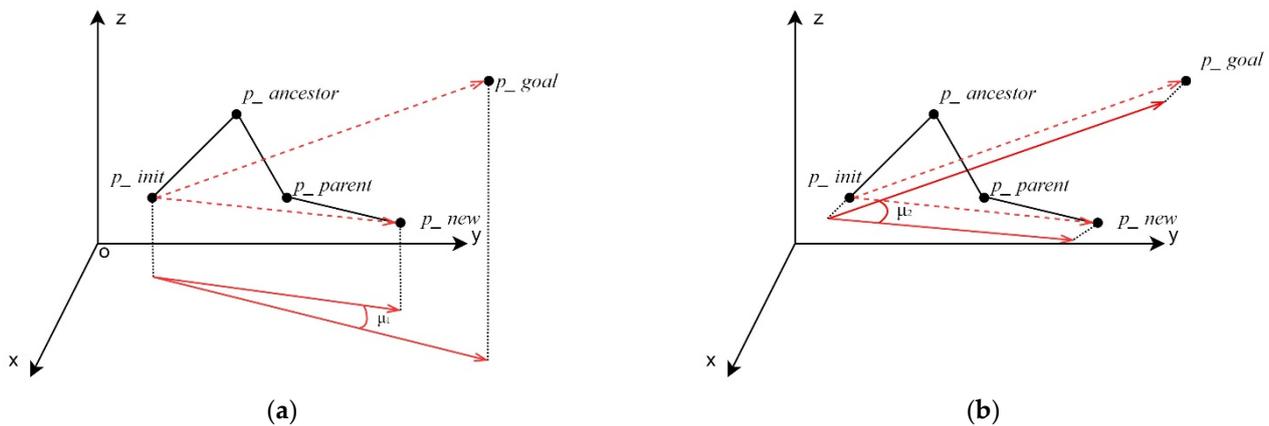


Figure 6. The projection of the angle between the vector (p_init, p_new) and the vector (p_init, p_goal) on different planes: (a) projection on plane xoy and (b) projection on plane $yozy$.

Flag to judge whether p_new can be added to the random tree is defined as follows:

$$flag1 = \begin{cases} true & \mu_1 < \pi/2 \ \& \ \mu_2 < \pi/2 \\ false & else \end{cases} \quad (21)$$

$$flag2 = \begin{cases} true & \alpha_1 < \pi/6 \ \& \ \alpha_2 < \pi/6 \\ false & else \end{cases} \quad (22)$$

$$flag3 = \begin{cases} true & \theta < \pi/6 \ \& \ \theta > -\pi/6 \\ false & else \end{cases} \quad (23)$$

$$flag = flag1 \cap flag2 \cap flag3 \quad (24)$$

4. Secondary Selection of the Parent Node

To reduce unnecessary nodes in the random tree, the secondary selection of the parent node is introduced (Figure 8). If the connection between $p_ancestor$, the parent node of p_parent , and p_new passes the collision test and meets the node screening strategy, the parent node of p_new to $p_ancestor$ is reset.

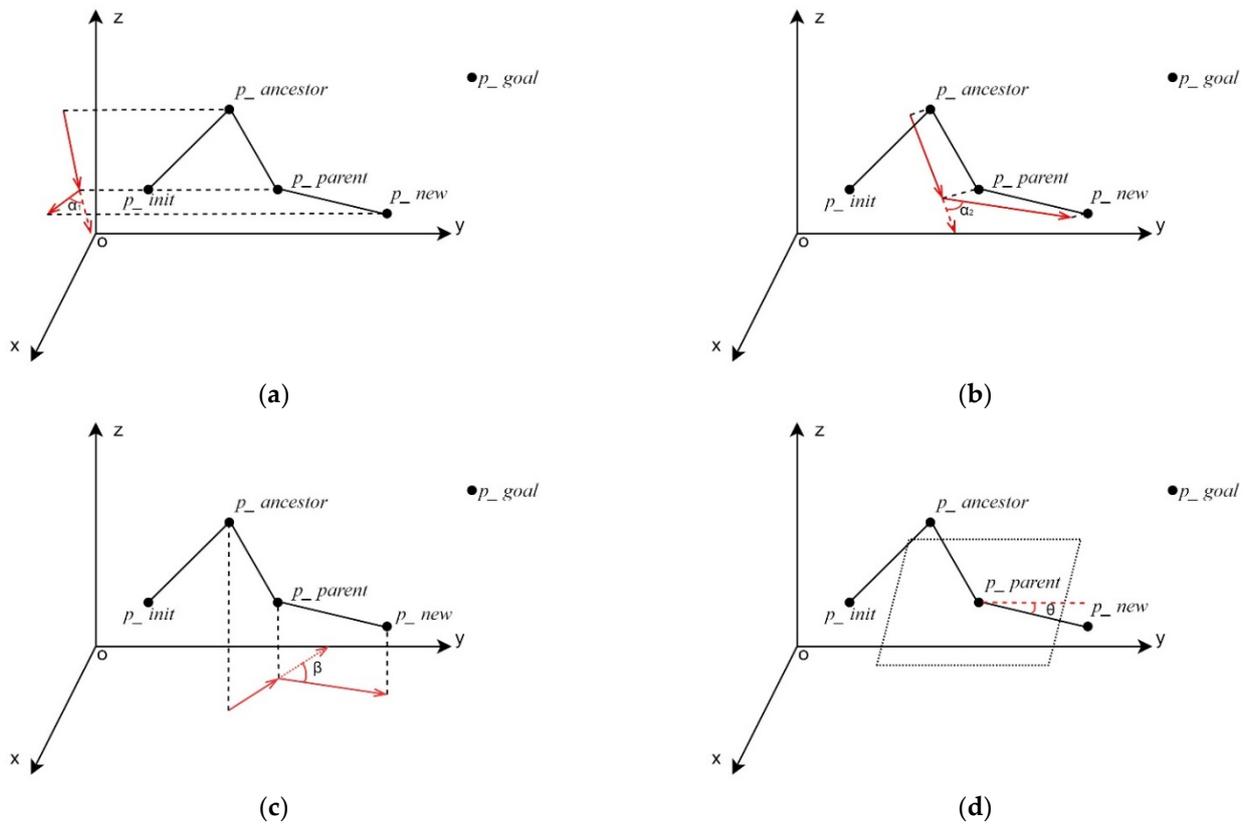


Figure 7. The pitch angle and projection of the angle between the vector (p_parent, p_new) and the vector $(p_ancestor, p_parent)$ on different planes: (a) projection on plane xoz ; (b) projection on plane yoz ; (c) projection on plane xoy ; and (d) pitch angle.

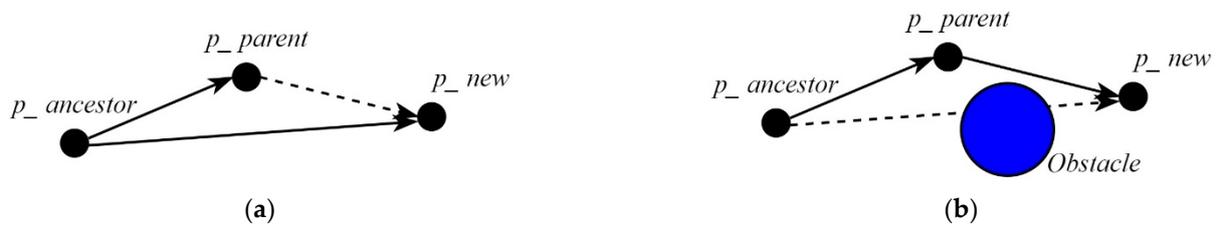


Figure 8. Secondary selection of the parent node: (a) reset the parent node and (b) abort resetting the parent node.

The derivation that this method always reduces the path length is given below [23]. Equation (25) defines the path length between any two nodes p_i and p_j in the random tree.

$$D(p_i, p_j) = \sqrt{(p_i.x - p_j.x)^2 + (p_i.y - p_j.y)^2} \tag{25}$$

Here, p_i refers to the i -th inserted arbitrary node and takes the x and y coordinate values of the node as an element. Function ξ receives an arbitrary node as a variable and returns the parent node of this node. The ξ function to the power of n ($n > 0$) can be represented as $\xi^n(p_i) := \left(\overbrace{\xi \circ \xi \circ \dots \circ \xi}^n \right) (p_i)$; when n is 0, $\xi^0(p_i) := p_i$ holds. The distance between p_i and its parent can be expressed as follows in Equation (26).

$$d_1(p_i) = D(\xi^0(p_i), \xi^1(p_i)) \tag{26}$$

With further induction and summary, $d_n(p_i)$ can be defined by:

$$d_n(p_i) = D(\zeta^n(p_i), \zeta^{n+1}(p_i)) \tag{27}$$

Equations (28) and (29) show the path length D_R from the start position p_{start} to the goal position p_{goal} by the RRT algorithm:

$$\zeta^{\delta+1}(p_{goal}) := p_{start} \tag{28}$$

$$D_R = \sum_{n=0}^{\delta} d_n(p_{goal}) \tag{29}$$

If the secondary selection of the parent node is performed k times in the planning process, the path length D'_R is expressed as:

$$D'_R = \sum_{n=0}^{\delta-k} d_n(p_{goal}) \tag{30}$$

If the distances between the edges connecting each node are the α between p_{new} and p_{parent} , the β between p_{parent} and $p_{ancestor}$, and the γ between p_{new} and $p_{ancestor}$, this can be represented as Equation (31) using the principle of triangular inequality.

$$\alpha + \beta \geq \gamma \tag{31}$$

Equation (32) shows the distance relationship between the ancestor nodes of p_{new} .

$$D(\zeta^0(p_{new}), \zeta^1(p_{new})) + D(\zeta^1(p_{new}), \zeta^2(p_{new})) \geq D(\zeta^0(p_{new}), \zeta^2(p_{new})) \tag{32}$$

Therefore, it can be deduced by analogy that the length of the path introducing the secondary selection of the parent node D'_R is always less than or equal to the original length of the path D_R .

The pseudo code of the improved RRT algorithm (Algorithm 1) is shown below.

Algorithm 1 Improved RRT Algorithm

```

pos = p_init
while ||p_new - p_goal|| ≤ d_min
  V = { pos }
  p_subtarget = Choose_subtarget(pos, p_goal)
  for i = 1 to I do
    p_rand = Random()
    p_near = Nearest(V, p_rand)
    p_new = Extend(p_near, r, p_rand)
    if Collision_free(p_new, p_near) && Node_screen(p_new, p_near) then
      Grow_tree(V, p_new)
      Parent_node_selection(V, p_new)
    end if
    if ||p_new - p_subtarget|| ≤ d_min then
      pos = V(2)
      break
    end if
  end for
end while

```

4. Target Interception Strategy

When a dynamic target is found, the AUV should predict the target position and intercept it to prevent other intentions. Taking into account the problems involved in an

interception mission, a predictive interception method based on the three-point numerical differential method is proposed. The higher the accuracy of prediction, the higher the success rate of interception. The interception process is shown in the Figure 9.

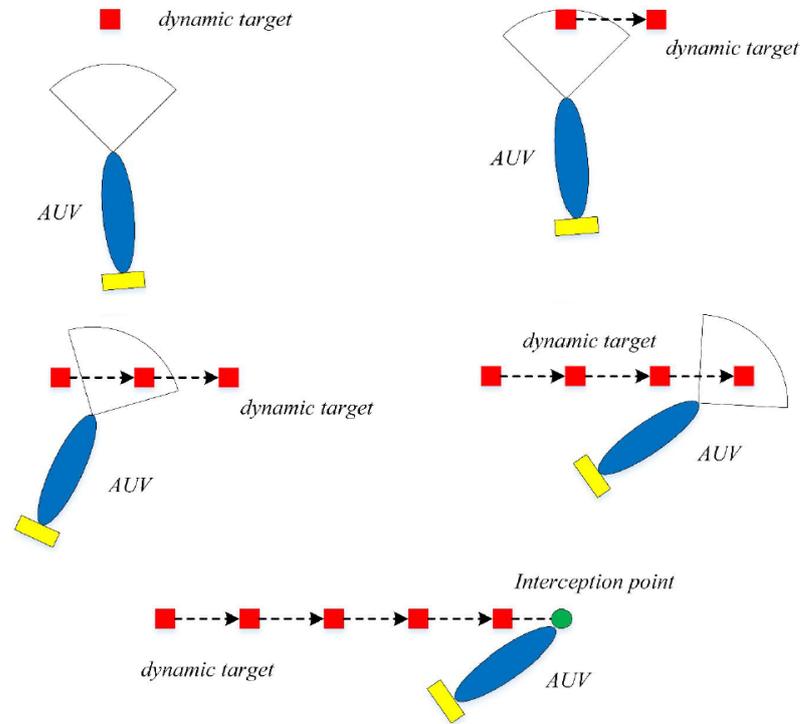


Figure 9. The interception process.

Due to the instability of higher-order interpolation, two-point, three-point, and five-point interpolation differentiation formulas are mostly used in the practical application of the numerical differential method. Because two-point differentiation is usually less accurate, while five-point differentiation significantly increases computational loads and is always difficult to implement, the three-point numerical differential method is used to solve the velocity of the dynamic target at the next instance [24].

Let (x_k, y_k) be the dynamic target position detected by the sonar. The three-point numerical differentiation method will be used to acquire the coordinates of the interception point. The three-point numerical differential equation of the first derivative is given below. Let $T(x)$ be the function defined on interval $[a, b]$, which refers to the abscissa movement range of the dynamic target. The function value of $T(x)$ at the node x_k is $y_k = T(x_k)$, where $k = 0, 1, 2, a \leq x_0 < x_1 < x_2 \leq b; x_0, x_1, x_2$ is the equidistant node, which means $x_2 - x_1 = x_1 - x_0 = h$. Lagrange interpolation was performed twice for $T(x)$ on interval $[a, b]$, as follows:

$$T(x) \approx L_2(x) = \frac{(x - x_1) \cdot (x - x_2)}{(x_0 - x_1) \cdot (x_0 - x_2)} \cdot T(x_0) + \frac{(x - x_0) \cdot (x - x_2)}{(x_1 - x_0) \cdot (x_1 - x_2)} \cdot T(x_1) + \frac{(x - x_0) \cdot (x - x_1)}{(x_2 - x_0) \cdot (x_2 - x_1)} \cdot T(x_2) \quad (33)$$

where $L_2(x)$ is the quadratic Lagrange interpolation. Let $x = x_0 + th, t \in (0, 1, 2)$. Then Equation (33) can be expressed as:

$$L_2(x_0 + th) = \frac{1}{2} \cdot (t - 1) \cdot (t - 2) \cdot T(x_0) - t \cdot (t - 2) \cdot T(x_1) + \frac{1}{2} \cdot t \cdot (t - 1) \cdot T(x_2) \quad (34)$$

Taking the derivation of both ends of the above equation with respect to t (equivalent to the derivation of x), we get:

$$L'_2(x_0 + th) = \frac{1}{2h}[(2t - 3)T(x_0) - (4t - 4)T(x_1) + (2t - 1)T(x_2)] \quad (35)$$

Taking $t = 0, 1, 2$, the three-point differential equation of the first derivative of node x_k is obtained:

$$T'(x_0) \approx L'_2(x_0) = \frac{1}{2h}[-3T(x_0) + 4T(x_1) - T(x_2)] \quad (36)$$

$$T'(x_1) \approx L'_2(x_1) = \frac{1}{2h}[-T(x_0) + T(x_2)] \quad (37)$$

$$T'(x_2) \approx L'_2(x_2) = \frac{1}{2h}[T(x_0) - 4T(x_1) + 3T(x_2)] \quad (38)$$

In the actual prediction, $T(x)$ is the position of the three known points, and x_k is the refresh time interval of the dynamic target. Using the three-point differential equation, the moving speed can be calculated, and the next dynamic target can be generated according to the time interval. The predicted position formula is calculated as follows:

$$x_{k+1} = x_k + v_{x,k}(t_{k+1} - t_k) \quad (39)$$

$$y_{k+1} = y_k + v_{y,k}(t_{k+1} - t_k) \quad (40)$$

where (x_{k+1}, y_{k+1}) is the next prediction point, and $v_{x,k}, v_{y,k}$ is the velocity component of the target at time k .

5. Simulation Results

5.1. Improved RRT Algorithm Verification

To prove the effectiveness and rapidity of the improved RRT algorithm, a comparative simulation experiment is conducted using the conventional RRT algorithm and the improved RRT algorithm. Assume that the simulation environment is a 3D space of 800 m × 800 m × 400 m, the detection radius of the AUV is 100 m, the coordinates of the starting point are (0, 0, 0), and the coordinates of the target point are (750, 750, 350).

Figure 10 shows the path planning of the conventional RRT algorithm, Figure 11 shows the path planning of the improved RRT algorithm, and Figure 12 shows the comparison of all nodes generated by the improved RRT algorithm and the conventional RRT algorithm.

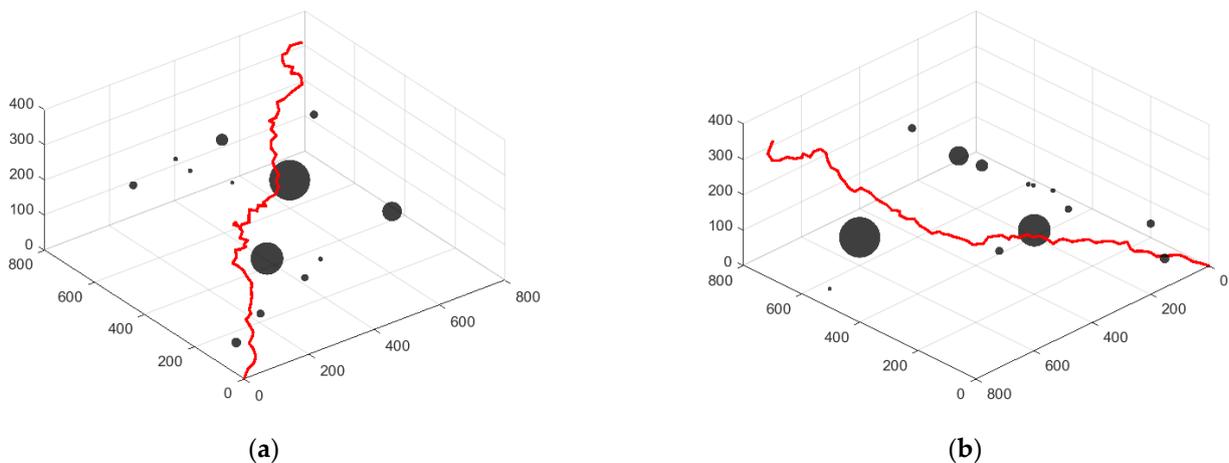


Figure 10. Different view of the path planning of the conventional RRT algorithm: (a) view 1; (b) view 2.

By comparing the path planning results of the two algorithms, it can be concluded that the improved RRT algorithm greatly reduces the number of redundant nodes by limiting

the selection of nodes, reduces the exploration of useless space, and greatly improves the efficiency of path planning.

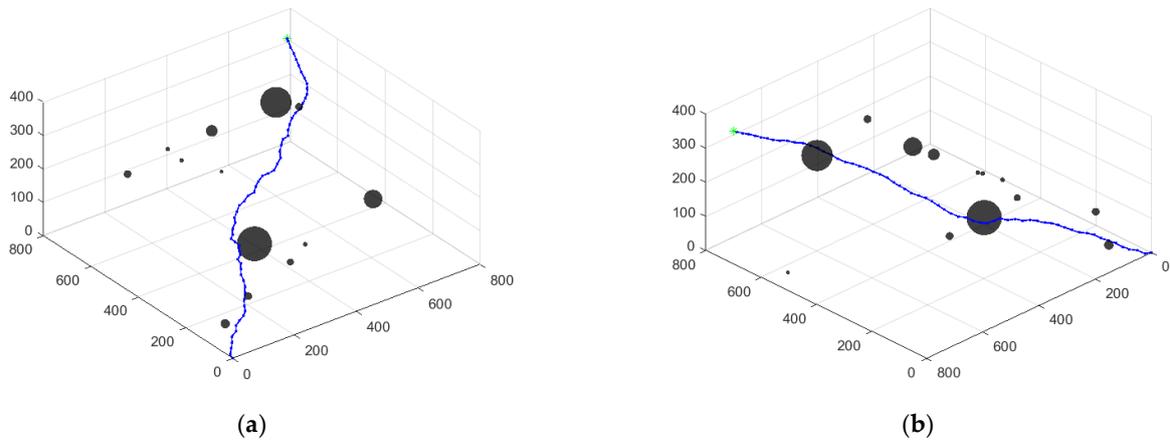


Figure 11. Different view of the path planning of the improved RRT algorithm: (a) view 1; (b) view 2.

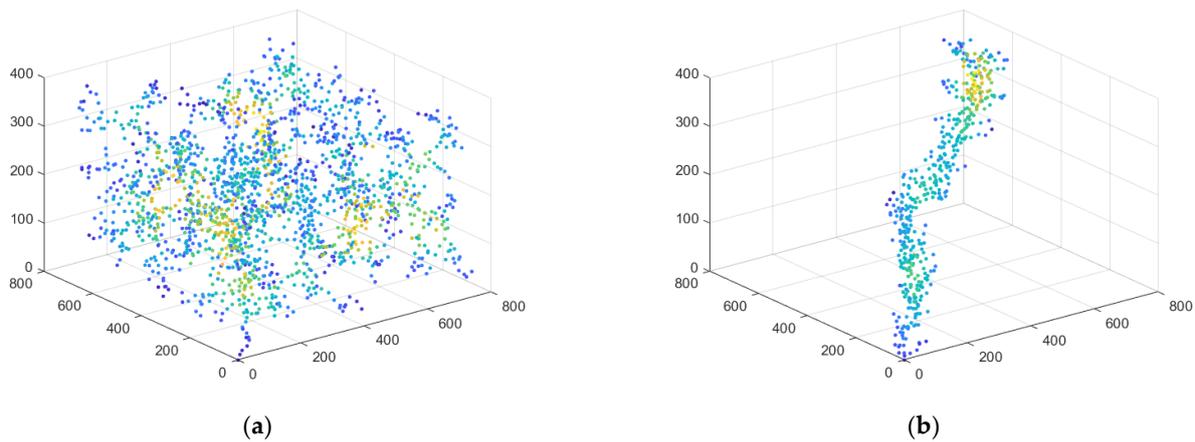


Figure 12. A comparison of all nodes generated by the improved RRT algorithm and the conventional RRT algorithm: (a) nodes generated by the conventional RRT algorithm; (b) nodes generated by the improved RRT algorithm. The brighter the color in the picture, the denser the nodes.

In all, 20 groups of simulation verifications were carried out on the RRT algorithm before and after the improvement. Table 2 records the total number of nodes, the path length, and the computation time of the conventional RRT algorithm, and Table 3 records the same data for the improved RRT algorithm. They are compared in Figures 13–15.

Table 2. Number of nodes, path length, and time cost of the conventional RRT algorithm.

Number of Nodes	Path Length	Time Cost	Number of Nodes	Path Length	Time Cost
1209	1644	4.97	1209	1518	4.87
1609	1522	7.15	1564	1359	6.68
958	1571	3.22	1353	1630	7.23
2492	1649	12.54	1553	1608	6.98
1301	1646	5.71	1888	1462	9.08
2809	1549	16.94	1268	1709	7.23
844	1486	2.87	1315	1629	7.06
2155	1505	11.53	1256	1588	5.22
1234	1508	5.03	974	1462	4.88
2434	1653	14.08	1065	1474	4.89

Table 3. Number of nodes and path length of the improved RRT algorithm.

Number of Nodes	Path Length	Time Cost	Number of Nodes	Path Length	Time Cost
735	1217	2.94	633	1307	2.41
602	1210	2.28	658	1186	2.34
802	1459	3.34	728	1225	2.59
598	1252	2.21	669	1254	2.69
652	1224	2.34	663	1195	2.39
677	1230	2.31	599	1335	2.33
619	1218	2.77	657	1335	2.40
648	1362	2.86	752	1450	3.18
675	1243	2.57	662	1198	2.25
701	1232	3.24	608	1231	2.19

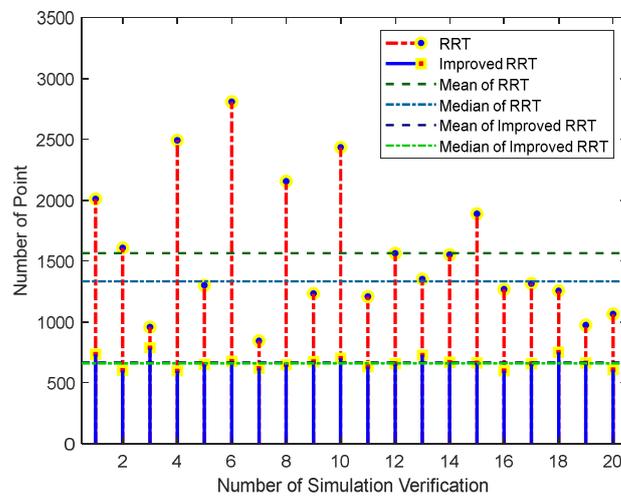


Figure 13. Comparison of the number of nodes produced by the two algorithms and their mean and median.

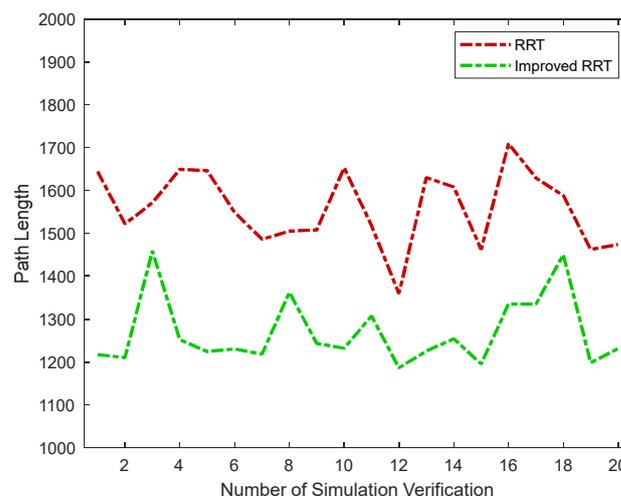


Figure 14. Comparison of the path lengths of the two algorithms.

The data show that the average number of nodes reduced by 52% and the median by 50%. The average path length was reduced by 19%. The average computation time was reduced by 65%. The figures clearly show that the improved RRT algorithm is significantly better than the conventional RRT algorithm in terms of the number of nodes, the length

of path, and the computation time. The indexes are not only lower but also more stable, which proves the effectiveness of the improved algorithm.

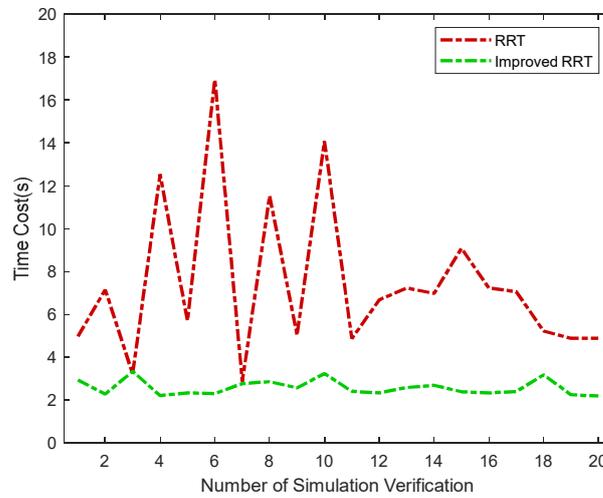


Figure 15. Comparison of the time costs of the two algorithms.

5.2. Simulation of Target Search Algorithm in an Open Environment

In this simulation, the conventional RRT algorithm and the improved RRT algorithm are applied in turn to perform target search on the basis of the search decision function. Assume that the initial coordinates of the AUV are (0, 0, 0), and 10 static targets, displayed in Table 4, are set in the environment. The targets in the figure are represented by squares, and the AUV’s real-time search path is represented by a red line. The simulation verification is shown in Figure 15.

Table 4. Target coordinates.

Target Code	X/m	Y/m	Z/m
1	104	40	10
2	310	170	190
3	230	178	234
4	130	477	476
5	630	250	500
6	200	530	490
7	118	120	63
8	430	490	600
9	600	160	200
10	200	590	130

As can be seen from Figure 16, the search paths of the conventional RRT target search algorithm and the improved RRT search algorithm are roughly the same due to the search decision function. However, according to Table 5, the search time of the improved RRT algorithm is 28% lower than that of the conventional RRT algorithm, indicating that the search speed has been greatly improved. The path length of the improved RRT algorithm is also reduced by 28%.

Table 5. Comparison of the search time and path lengths of the two algorithms.

Algorithm	Time/s	Path Length/m
Conventional RRT	30,681	61,367
Improved RRT	21,987	43,969

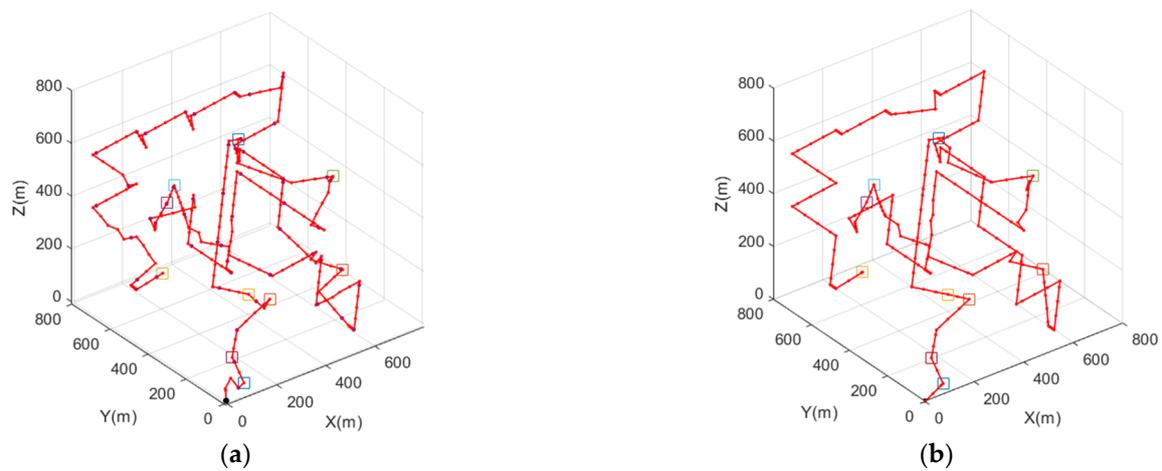


Figure 16. The comparison of the search paths generated by the two algorithms: (a) search path generated by the conventional RRT algorithm and (b) search path generated by the improved RRT algorithm.

5.3. Simulation of Target Search Algorithm in an Obstacle Environment

The simulation environment is the same as above. Since the search path in the 3D environment is relatively chaotic, to avoid the difficulty of observing, six static targets and 16 obstacles are set to verify the effectiveness of the algorithm in the obstacle environment. The target coordinate information is displayed in Table 6, and the center coordinate information of the obstacle is displayed in Table 7. The simulation verification is shown in Figure 17.

Table 6. Target coordinates.

Target Code	X/m	Y/m	Z/m
1	310	170	190
2	350	383	328
3	680	345	555
4	118	120	63
5	84	160	147
6	200	590	130

Table 7. Obstacle center coordinates.

Target Code	X/m	Y/m	Z/m
1	30	108	104
2	315	168	101
3	545	758	20
4	168	435	334
5	52	100	40
6	115	85	107
7	150	470	358
8	365	171	135
9	300	480	355
10	564	456	345
11	324	675	456
12	98	125	654
13	630	230	150
14	630	110	650
15	120	600	234
16	210	610	420

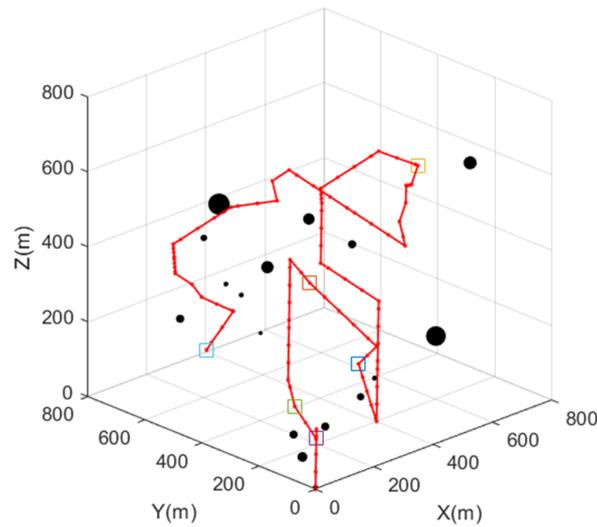
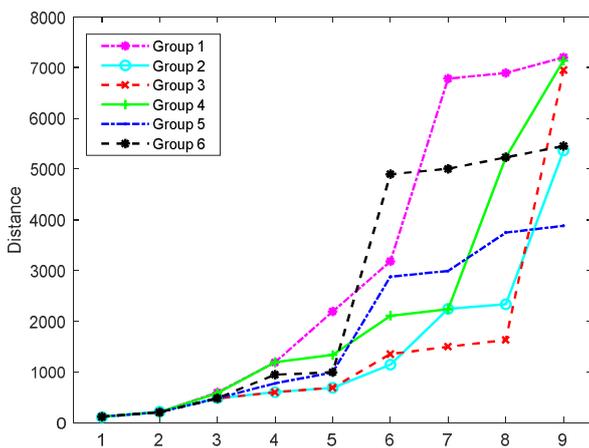
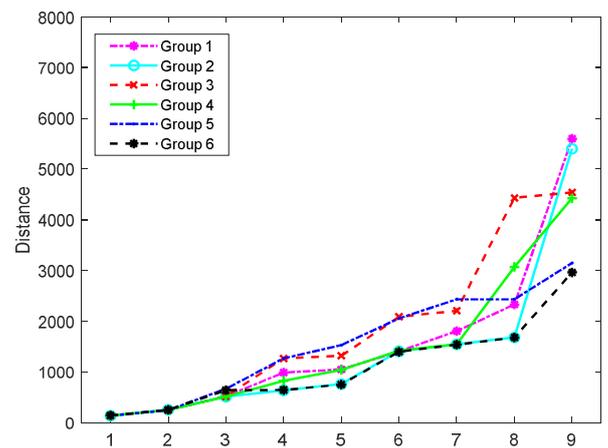


Figure 17. Target search in an obstacle environment.

To test the efficiency of the search algorithm proposed in this paper, the same number of targets are set in the same environment but at different positions, and the algorithm is verified by comparing the distances the AUV covered in the process of searching. Figure 18 shows the sailing distances required by the conventional RRT algorithm and the improved RRT algorithm to complete the target search under six groups of different target positions. Figure 19 fits the six curves and compares their mean absolute difference (MAD). As can be seen from the figure, the improved RRT algorithm has better target search efficiency than the conventional RRT algorithm after the increase in the number of targets. Meanwhile, the improved RRT algorithm reduces the MAD of the fitting curve, which represents a lower degree of data dispersion. This indicates that the improved RRT algorithm has stable search efficiency when the target position changes.



(a)



(b)

Figure 18. The length of the path required to complete the target search: (a) search using the conventional RRT algorithm and (b) search using the improved RRT algorithm.

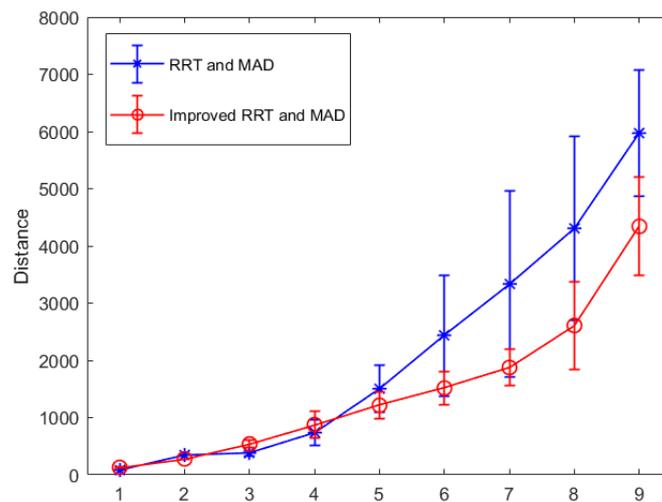


Figure 19. Curve fitting and MAD comparison.

5.4. Target Interception Algorithm Simulation

The initial position of the AUV is (0, 0, 0), and nine static targets and one dynamic target are set in the environment. The static target position is shown in Table 8, and the initial coordinates of the dynamic target are (250, 400, 400). The squares in Figure 20 are the static targets, the red sphere is the dynamic target, and the black spheres are the obstacles.

Table 8. Target coordinates.

Target Code	X/m	Y/m	Z/m
1	310	170	190
2	230	178	234
3	434	349	238
4	130	477	476
5	240	620	230
6	680	345	555
7	118	120	63
8	84	160	147
9	200	590	130

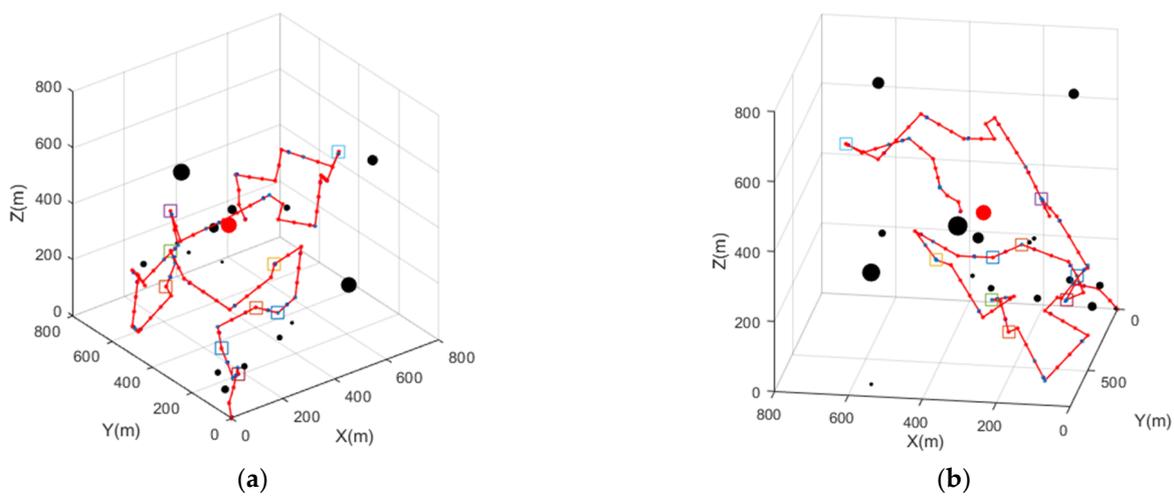


Figure 20. Different view of target interception: (a) view 1; (b) view 2.

In the process of dynamic target interception simulation, since there are still static targets in the environment, the AUV is also responsible for the search task of the static target, so when the speed of the AUV and the dynamic target are consistent, the dynamic target will not be intercepted. Therefore, the dynamic target movement speed in this paper is set to 1/5~1/3 of the AUV speed.

6. Conclusions

An autonomous search algorithm has been developed to improve the exploration capability of the AUV in multiple-target search in an unknown environment. The proposed algorithm is able to achieve close-range confirmation of static targets and interception of dynamic targets. The search decision point can be conducted through the search decision function taking into consideration the uncertainty benefit, the search task benefit, and the regional ergodicity benefit. From the perspective of search efficiency, path planning in the search mission and interception task is implemented using the improved RRT algorithm. By combining with rolling planning and other methods, the improved RRT algorithm can effectively reduce the number of redundant nodes and shorten the path length. Numerical simulations are conducted in an open or an obstacle-ridden environment, which proves the effectiveness of searching and interception.

There are several areas to be explored to improve the performance of the proposed method and make it more robust in reality. First, the interception speed of the AUV must be faster than the target speed. Otherwise, the AUV cannot intercept effectively. However, this is often difficult to guarantee in real situations. Second, although the randomness and blindness of the RRT algorithm have been improved, the problem of low search efficiency due to unstable planning persists. Third, a single AUV has limited capabilities in terms of target search and interception. In the future, the algorithm should be extended to group work. Lastly, future work should focus on experiments in real situations.

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

Funding: This search work was funded by the National Natural Science Foundation of China, grants no. 5217110503 and no. 51809060; by the Research Fund from Science and Technology on Underwater Vehicle Technology, grant no. JCKYS2021SXJQR-09; and by Heilongjiang Provincial Natural Science Foundation of China, grant no. E2018023.

Data Availability Statement: Not applicable.

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

References

1. Glaviano, F.; Esposito, R. Management and sustainable exploitation of marine environments through smart monitoring and automation. *J. Mar. Sci. Eng.* **2022**, *10*, 297. [[CrossRef](#)]
2. Ru, J.; Yu, S.; Wu, H.; Li, Y.; Wu, C.; Jia, Z.; Xu, H. A multi-AUV path planning system based on the omni-directional sensing ability. *J. Mar. Sci. Eng.* **2021**, *9*, 806. [[CrossRef](#)]
3. Liu, H.; Xu, B.; Liu, B. An automatic search and energy-saving continuous tracking algorithm for underwater targets based on prediction and neural network. *J. Mar. Sci. Eng.* **2022**, *10*, 283. [[CrossRef](#)]
4. Chen, T.; Qu, X.; Zhang, Z.; Liang, X. Region-searching of multiple autonomous underwater vehicles: A distributed cooperative path-maneuvering control approach. *J. Mar. Sci. Eng.* **2021**, *9*, 355. [[CrossRef](#)]
5. Mao, Y.; Gao, F.; Zhang, Q.; Yang, Z. An AUV Target-Tracking Method Combining Imitation Learning and Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 383. [[CrossRef](#)]
6. Li, J.; Zhang, J. Target search of multiple autonomous underwater vehicles in an unknown environment. *J. Harbin Eng. Univ.* **2019**, *40*, 1951–1957, 1972.
7. Ni, J.; Yang, L. An improved DSA-based approach for multi-AUV cooperative search. *Comput. Intell. Neurosci.* **2018**, *2018*, 2186574. [[CrossRef](#)] [[PubMed](#)]

8. Ishida, T.; Korf, R.E. Moving-target search: A real-time search for changing goals. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *15*, 609–619. [[CrossRef](#)]
9. Ajmera, Y.; Singh, S.P. Autonomous UAV-based target search, tracking and following using reinforcement learning and YOLOFlow. In Proceedings of the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Abu Dhabi, United Arab Emirates, 4–6 November 2020.
10. Wang, P.; Meghjani, M. Lost at sea: Multi-searcher multi-target search. In Proceedings of the Global Oceans 2020: Singapore—U.S. Gulf Coast, Biloxi, MS, USA, 5–30 October 2020.
11. Ibenthal, J.; Meyer, L.; Piet-Lahanier, H. Target search and tracking using a fleet of UAVs in presence of decoys and obstacles. In Proceedings of the 59th IEEE Conference on Decision and Control (CDC), Jeju, Korea, 14–18 December 2020.
12. Yin, G.; Zhou, S.; Wu, Q. An improved RRT algorithm for UAV path planning. *Acta Electron. Sin.* **2017**, *45*, 1764.
13. Wu, X.G.; Guo, C.; Li, Y.B. Variable probability based bidirectional RRT algorithm for UAV path planning. In Proceedings of the 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, China, 31 May–2 June 2014.
14. Guo, Y.; Liu, X.; Liu, X.; Yang, Y.; Zhang, W. FC-RRT*: An improved path planning algorithm for UAV in 3D complex environment. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 112. [[CrossRef](#)]
15. Li, J.; Zhang, Y.X. Formation control of a multi-autonomous underwater vehicle event-triggered mechanism based on the hungarian algorithm. *Machines* **2022**, *9*, 346. [[CrossRef](#)]
16. Li, J.; Zhai, X.L. Target search algorithm for AUV based on real-time perception maps in unknown environment. *Machines* **2021**, *9*, 147. [[CrossRef](#)]
17. Hu, J.W.; Xie, L.; Xu, J.; Xu, Z. Multi-Agent Cooperative Target Search. *Sensors* **2014**, *14*, 9408–9428. [[CrossRef](#)] [[PubMed](#)]
18. Song, D.L.; Yao, P. Search for static target in nonwide area by AUV: A prior data-driven strategy. *IEEE Syst. J.* **2021**, *15*, 3185–3188. [[CrossRef](#)]
19. Zhu, J.; Zhao, S.; Zhao, R. Path planning for autonomous underwater vehicle based on artificial potential field and modified RRT. In Proceedings of the 2021 International Conference on Computer, Control and Robotics (ICCCR), Shanghai, China, 8–10 January 2021.
20. Cho, Y.; Kim, E. Path planning of a robot manipulator using retrieval RRT strategy. *Int. J. Fuzzy Logic Intell. Syst.* **2007**, *7*, 138–142.
21. Wang, R.P.; Xi, W.; Guo, X. Path following for snake robot using crawler gait based on path integral reinforcement learning. In Proceedings of the 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM), Chongqing, China, 3–5 July 2021.
22. Guo, H.; Qin, J.L. Rolling path planning of mobile robot based on automatic diffidence ant algorithm. In Proceedings of the 12th International Conference on Graphics and Image Processing (ICGIP), Xi'an, China, 13–15 November 2020.
23. Kang, J.G.; Lim, D.W. Improved RRT-connect algorithm based on triangular inequality for robot path planning. *Sensors* **2021**, *21*, 333. [[CrossRef](#)] [[PubMed](#)]
24. Meng, X.Q.; Sun, B. Harbour protection: Moving invasion target interception for multi-AUV based on prediction planning interception method. *Ocean Eng.* **2021**, *219*, 108268. [[CrossRef](#)]