

Article

Motion-Aware Autonomous Exploration Framework for AUVs in Complex Underwater Structures

Shihui Shen ¹, Kanghui Jiang ¹, Mingyang Dai ^{2,3} and Hongzhi Wang ^{1,*}

¹ PLA Naval Submarine Academy, Qingdao 266199, China; shihuishen2025@163.com (S.S.); 18153252388@163.com (K.J.)

² College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; mingyang.dai@hrbeu.edu.cn

³ Qingdao Innovation and Development Center, Harbin Engineering University, Qingdao 266404, China

* Correspondence: wanghz-315@163.com

Abstract

The autonomous exploration capability of Autonomous Underwater Vehicles (AUVs) in complex underwater structures is fundamentally constrained by the coupling between sensing and motion execution. Unlike ground and aerial robots, limited underwater communication makes it difficult for AUVs to rely on external computing resources for perception and motion computation, imposing higher requirements on algorithmic efficiency. Meanwhile, motion constraints require arc-shaped course adjustments rather than in-place turns, thereby increasing navigation costs. This paper focuses on fixed-depth two-dimensional exploration represented by an occupancy grid and assumes deterministic sonar observations in order to isolate the effect of sensing–motion coupling. This sensing–motion coupling makes conventional frontier-based exploration strategies inefficient for underwater environments. To address this issue, this paper proposes a motion-aware autonomous exploration framework for AUVs that jointly considers information acquisition and motion cost during exploration decision-making and path execution. The proposed framework constructs adaptive local viewpoint sampling regions directly from the real-time sonar detection coverage and introduces a weighted sampling strategy to improve viewpoint generation efficiency in structured underwater environments. A motion-aware viewpoint utility function is further designed by integrating frontier information gain, path cost, and heading deviation, enabling the exploration strategy to favor viewpoints with lower execution cost. To balance exploration efficiency and coverage completeness, a local-priority and global-backtracking target assignment mechanism is developed. In addition, motion-constrained path execution is achieved through Douglas–Peucker keypoint extraction, cubic Bezier path smoothing, and dynamic window approach (DWA)-based local trajectory tracking. Experiments on a high-fidelity Unreal Engine–ROS simulation platform show consistent reductions in exploration path length and mapping time relative to a classical frontier-based baseline, with average improvements of about 11–15% in the tested scenarios. In corridor scenarios, the adaptive viewpoint generation strategy improves local viewpoint generation efficiency by 58.5% compared with a sliding-window RRT baseline. The results demonstrate that the proposed framework can effectively improve exploration efficiency and motion consistency for AUV autonomous exploration in complex underwater structures.



Academic Editor: Giulio Reina

Received: 7 June 2026

Revised: 16 July 2026

Accepted: 17 July 2026

Published: 21 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: autonomous underwater vehicle; autonomous exploration; viewpoint generation; viewpoint evaluation; path planning; motion constraint

1. Introduction

Autonomous underwater vehicles (AUVs) play an important role in marine resource exploration, environmental monitoring, and submarine pipeline inspection [1–3]. In these tasks, an AUV is often required to autonomously explore unknown underwater environments while simultaneously constructing an environmental map without prior information. This capability, commonly referred to as autonomous exploration, is fundamental for improving operational autonomy and reducing dependence on teleoperation in complex marine environments [4–6].

Autonomous exploration has been extensively studied in mobile robotics. Frontier-based exploration methods represent one of the most widely used paradigms, where robots continuously move toward the boundary between known and unknown regions to gradually expand the explored map [7–10]. The existing exploration methods developed for ground or aerial robots are usually based on two implicit assumptions: Firstly, ground or aerial robots can often enhance the computational efficiency of various algorithms and data processing through external computing units, enabling them to have a higher tolerance for the complexity of autonomous exploration algorithms. Secondly, the motion cost resulting from course adjustment can be negligible, allowing the robots to be approximated as point masses during the exploration planning process. For most ground or aerial robots, these assumptions are largely valid.

However, when such exploration strategies are directly applied to AUVs, the above assumptions no longer hold. Unlike ground or aerial robots, AUVs often rely solely on their onboard computing units for algorithm execution due to the challenges of underwater communication. This results in limited computational resources, leading to poor performance when running complex algorithms. More importantly, due to fluid dynamics constraints and propulsion characteristics, AUVs cannot rotate in place; course adjustments require curved trajectories, which incur significant time and energy costs. Therefore, research on underwater autonomous exploration with AUVs necessitates careful design of exploration algorithms in terms of complexity, as well as planning and execution that account for the unique motion characteristics of AUVs.

The inconsistency between underwater autonomous exploration and traditional autonomous exploration assumptions spreads throughout the entire autonomous exploration process, including exploration viewpoint generation, viewpoint evaluation, and path execution. Traditional RRT-based viewpoint planners usually sample candidate viewpoints within fixed geometric windows, which are not aligned with the actual sonar-observable region and may generate many invalid samples in narrow passages or obstacle-dense environments. The existing viewpoint evaluation methods mainly focus on information gain and Euclidean distance while ignoring the turning cost required to align the AUV heading with the target viewpoint, potentially leading to exploration decisions with high execution cost. In addition, global paths generated by grid-based planners such as A* typically contain dense piecewise-linear turns that make it difficult to directly satisfy the continuous-curvature motion constraints of AUVs, thereby increasing local tracking difficulty and reducing execution smoothness.

To address the above issues, this paper proposes a motion-aware autonomous exploration framework for AUVs in complex underwater structures. The problem is formulated in a fixed-depth two-dimensional setting using a 360° scanning sonar model with deterministic range measurements. Sonar sensing constraints and AUV motion constraints are jointly incorporated throughout exploration decision-making and path execution. Specifically, an adaptive viewpoint generation method is designed based on the real-time sonar detection region to improve valid viewpoint generation efficiency in structured underwater environments. A motion-aware viewpoint utility function integrating frontier information

gain, path cost, and heading deviation is further developed to reduce unnecessary turning motions during exploration. To balance exploration efficiency and coverage completeness, a local-priority and global-backtracking target assignment mechanism is introduced. In addition, motion-constrained path execution is achieved through Douglas–Peucker keypoint extraction, cubic Bezier path smoothing, and DWA-based local trajectory tracking. The effectiveness of the proposed framework is systematically verified on a high-fidelity Unreal Engine–ROS simulation platform.

2. Related Work

Classical autonomous exploration methods are mainly based on frontier-driven exploration strategies, where robots continuously expand the explored region by navigating toward the boundary between known and unknown space [7–10]. Early studies mainly focused on improving frontier detection efficiency and frontier-point selection. Keidar et al. proposed efficient frontier detectors to reduce repeated full-map processing during exploration [11]. Sun et al. further improved frontier detection efficiency by combining submap alignment information and selectively updating frontier regions with large pose changes [8]. Li et al. combined frontier extraction with global topological graphs to generate safe and reachable frontier points [12]. Wang et al. introduced an incremental roadmap construction framework to improve exploration target selection and path planning efficiency [13]. Recent hierarchical exploration frameworks further improve global–local coordination and exploration efficiency in complex environments [9,10,14].

Although these methods perform well on ground or aerial robots, they are mostly based on two implicit assumptions: the carried computing units can run complex algorithms well and the robots have low-cost steering adjustment capabilities. For ground or aerial robots, these assumptions are generally valid; however, they no longer apply in the AUV exploration scenario. The boundary-based algorithms need to perform calculations across the entire exploration map each time they detect the boundary area, which involves a large amount of computation and is often difficult to achieve with AUVs due to limited computing resources. Additionally, due to the constraints of fluid mechanics and propulsion characteristics, the steering adjustment of AUVs requires arc-shaped movements rather than rotational movements on the spot, which will introduce significant time and energy costs during the exploration process. Therefore, the traditional exploration strategy based on boundary detection may not be effective for underwater autonomous exploration.

To reduce the separation between exploration decision-making and path execution, viewpoint-based exploration methods have been extensively studied. The next-best-view (NBV) strategy selects candidate viewpoints with high information value as exploration targets [15]. Since rapidly exploring random trees (RRTs) naturally tend to expand toward unexplored regions, they have gradually become widely used tools for viewpoint generation [16]. Bircher et al. proposed a receding-horizon NBV planner that evaluates accumulated information gain along RRT branches for online exploration decision-making [17]. Zhu et al. proposed the DSVP framework, which restricts viewpoint sampling within a fixed sliding window while combining global topological information to avoid missing potential viewpoints [18]. However, most existing viewpoint generation methods still rely on predefined geometric sampling regions that are decoupled from the actual sensor-observable space. In narrow passages or obstacle-dense environments, this inconsistency may generate a large number of invalid samples and reduce viewpoint generation efficiency.

In the field of underwater autonomous exploration, recent research has begun to incorporate the characteristics of AUVs into perspective planning. Palomeras et al. proposed probabilistic next-best-view planning methods for autonomous mapping of complex underwater structures [4,19]. Vidal et al. combined frontier detection and viewpoint gener-

ation for online underwater exploration in both two-dimensional and three-dimensional environments [5,20]. Zacchini et al. proposed an RRT-based viewpoint planning framework for forward-looking sonar and employed RRT* for path generation between the current position and the target viewpoint [6,21]. Although these methods have achieved important progress in underwater exploration, the coupling between exploration decision-making and motion-constrained execution is still insufficiently modeled. The existing methods mainly focus on viewpoint information gain while paying limited attention to the turning cost induced by heading adjustment. In addition, global paths are often represented as polylines or simple interpolations, which make it difficult to directly satisfy the continuous-curvature motion constraints of AUVs and may increase local tracking difficulty during execution.

Overall, the existing work still lacks unified modeling of sonar sensing constraints and AUV motion constraints throughout the exploration pipeline. Viewpoint generation is often inconsistent with the actual sonar-observable region, viewpoint evaluation usually ignores heading-related execution cost, and path planning is commonly decoupled from motion-constrained trajectory execution.

In contrast, the framework proposed in this paper focuses on the coupling among viewpoint generation, viewpoint evaluation, and path execution during AUV autonomous exploration. The local sampling region is constructed from real-time sonar detection coverage so that candidate viewpoints are generated in closer agreement with the currently observable space. The viewpoint utility function further combines information gain, Euclidean distance, and heading deviation, allowing the selected viewpoint to account for both exploration benefit and turning-related execution cost. For path execution, the grid-based path is simplified and converted into curvature-continuous Bezier trajectories for dynamic window tracking, improving the continuity between global planning and local motion execution. Through sonar-aware sampling, heading-aware viewpoint evaluation, and motion-constrained trajectory execution, the proposed framework provides an integrated treatment of sensing and motion coupling throughout autonomous exploration.

3. Motion-Constrained Autonomous Exploration Framework

3.1. Task Analysis and Framework Overview

The autonomous exploration task of the AUV consists of three tightly coupled components: real-time environment mapping, exploration decision-making, and motion-constrained navigation control, as shown in Figure 1. The mapping module continuously updates the local occupancy grid map around the AUV using sonar observations. Based on the updated map, the exploration module generates candidate viewpoints, evaluates their utility, and selects the next exploration target. The navigation module then plans and tracks a feasible path toward the assigned target. After the target is reached, the environmental map is updated and a new exploration cycle begins. The exploration process terminates when no valid frontier remains in the task region. The present formulation assumes a fixed-depth two-dimensional workspace, a regular occupancy grid with free, occupied, and unknown cells, and a scanning sonar model, represented by uniformly distributed azimuthal rays that provide deterministic range measurements over the full field of view.

Unlike conventional frontier-based exploration methods that approximately treat the robot as a point mass with omnidirectional perception, the proposed framework explicitly considers the coupling between sonar sensing and AUV motion execution throughout the exploration pipeline. Specifically, the framework addresses three key inconsistencies in underwater exploration:

- The mismatch between fixed geometric sampling regions and the actual sonar-observable space;
- The inconsistency between exploration utility evaluation and heading-related motion cost;

- The decoupling between global path planning and motion-constrained local trajectory execution.

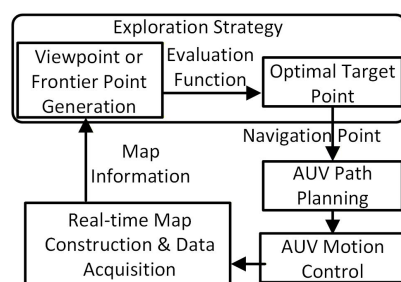


Figure 1. Framework of the AUV autonomous exploration task. The loop starts from sonar-based occupancy-grid update, proceeds through candidate viewpoint generation and evaluation, and terminates only when no valid local or global frontier viewpoint remains.

To address these issues, a motion-aware exploration framework is developed. The framework consists of sonar-aware viewpoint generation, motion-aware viewpoint evaluation, local–global exploration coordination, and motion-constrained path execution.

3.2. Sonar-Aware Viewpoint Generation

Conventional RRT-based exploration methods usually perform random sampling within global maps or fixed sliding windows [16,18]. Although restricting the sampling region can improve exploration efficiency, fixed geometric windows are independent of the actual sonar-observable region and therefore may generate a large number of invalid samples in narrow passages or obstacle-dense environments.

To improve viewpoint generation consistency with underwater sensing characteristics, the proposed method directly constructs the local sampling region from the real-time sonar detection coverage. Two RRT-based viewpoint generators are employed cooperatively: a local viewpoint generator for rapid nearby exploration and a global viewpoint generator for long-range exploration recovery.

The local viewpoint generator uses the current AUV position as the root node and performs random-tree expansion only within the sonar-observable region. Since the local sampling space is restricted to the actual detectable region, the generated viewpoints are more consistent with feasible sensing directions. The global viewpoint generator uses the initial exploration position as the root node and continuously grows over the explored map without reset, enabling the system to maintain exploration completeness when nearby valid viewpoints no longer exist.

Considering that sonar coverage changes dynamically due to obstacle occlusion, the actual sonar detection region is directly used as the local sampling space. The environment is represented by a regular occupancy grid; sonar rays update cells along each beam as free until the measured return or the maximum range is reached, while cells at obstacle returns are marked as occupied and unobserved cells remain unknown. Let the number of sonar beams be N and the maximum sonar detection range be $d_{\max}$. For beams with valid returns, the measured detection distance is stored directly. For beams without returns, the detection distance is set to $d_{\max}$. The detection distances in all beam directions are denoted as $d_1, d_2, \dots, d_N$.

A two-stage weighted sampling strategy is adopted to bias random-tree growth toward directions with larger observable space. In the first stage, beam directions are sampled according to the normalized detection distance

$$p_i = \frac{d_i}{\sum_{j=1}^N d_j}, \quad i = 1, 2, \dots, N \quad (1)$$

where p_i denotes the probability of selecting the i -th beam direction.

After a beam direction is selected, a distance $d_s \in [0, d_i]$ is uniformly sampled along the beam to generate the random sample

$$\mathbf{x}_{\text{rand}} = \mathbf{p}_{\text{auv}} + d_s \begin{bmatrix} \cos(\psi + \theta_i) \\ \sin(\psi + \theta_i) \end{bmatrix} \quad (2)$$

where $\mathbf{p}_{\text{auv}}$ denotes the current AUV position, ψ denotes the current heading angle, and θ_i denotes the relative angle of the selected sonar beam.

Compared with fixed-window random sampling, the proposed sampling strategy aligns the random-tree expansion direction with the actual sonar-observable space and therefore reduces invalid viewpoint generation in structured underwater environments.

After viewpoint sampling, standard RRT extension is performed to generate new candidate viewpoints. Collision-free nodes generated during random-tree growth are treated as candidate viewpoints, and the number of observable frontier cells within the corresponding detection region is calculated and stored for subsequent utility evaluation. The local viewpoint generation process is shown in Figure 2, and an example of local random-tree growth is shown in Figure 3.

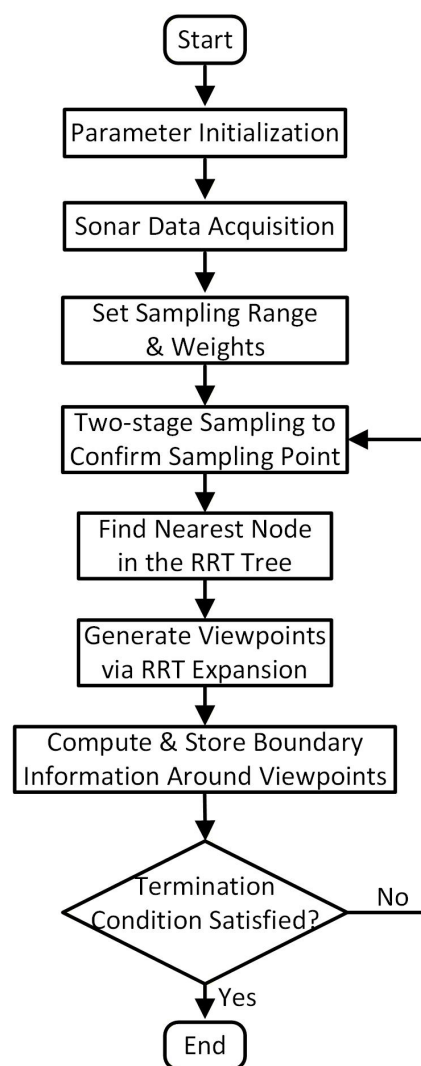


Figure 2. Flowchart of local viewpoint generation. First, the actual coverage area of the sonar detection is determined based on the double-layer sampling mechanism; then, collision-free viewpoints are generated through RRT expansion, and the viewpoints within the detection area that contain boundary units are stored as candidate viewpoints.

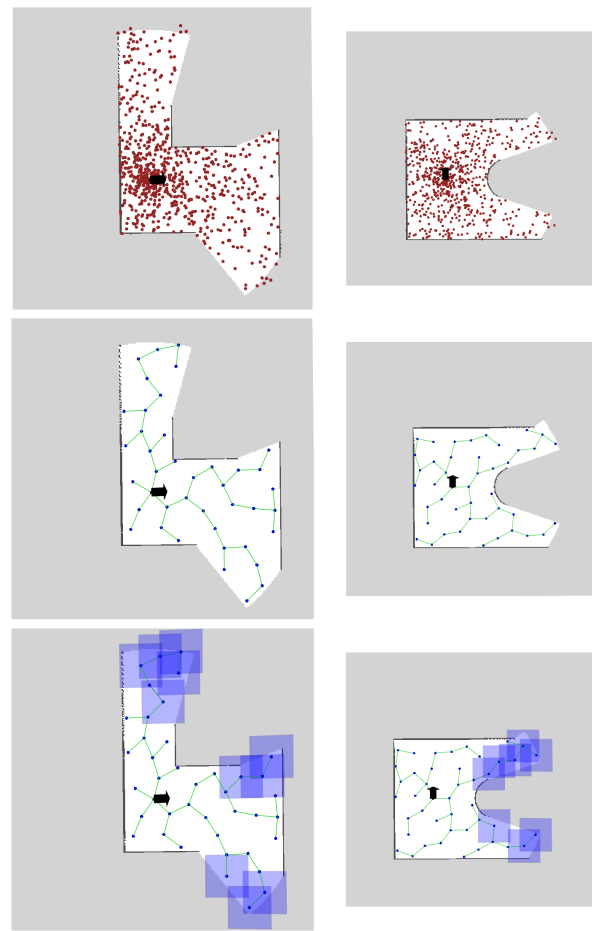


Figure 3. RRT growth of the local viewpoint generator. Red dots are random samples, blue dots are viewpoints, green lines are tree edges, blue squares are viewpoint boundary-detection regions, and the black arrow indicates the current AUV pose and heading.

The global viewpoint generator follows the same random-tree expansion mechanism but continuously maintains candidate viewpoints throughout the exploration process. When the local region no longer contains valid viewpoints, the global generator provides backtracking targets for exploration recovery. The growth result of the global viewpoint generator is shown in Figure 4.

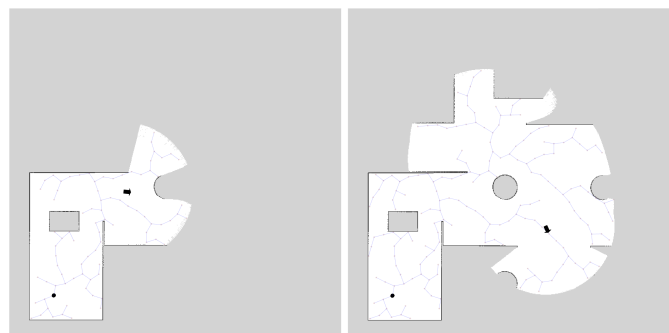


Figure 4. RRT growth of the global viewpoint generator. The global tree is maintained over the explored free space and provides reachable backtracking candidates when the local sonar-observable region no longer contains valid frontier viewpoints. The arrows indicate the AUV poses, the black dots indicate the starting points of the exploration task. The areas colored by grey are unexplored spaces.

3.3. Motion-Aware Viewpoint Evaluation

The proposed local–global viewpoint generators produce multiple candidate viewpoints with different exploration values and motion costs. Conventional viewpoint evaluation methods mainly focus on information gain and Euclidean distance while ignoring the additional execution cost caused by heading adjustment. However, for AUVs operating under hydrodynamic constraints, large heading deviations may introduce considerable turning cost and ineffective travel.

To jointly evaluate exploration benefit and motion cost, the proposed method evaluates candidate viewpoints from three aspects: information gain, path cost, and heading deviation.

Assume that the current AUV position is (x_{AUV}, y_{AUV}) and the current heading angle is ψ_{AUV} . For the i -th candidate viewpoint (x_i, y_i) , the path cost is approximated using the Euclidean distance:

$$d_i = \sqrt{(x_i - x_{AUV})^2 + (y_i - y_{AUV})^2} \quad (3)$$

The information gain of the viewpoint is represented by the number of observable frontier cells within the corresponding boundary detection region and is denoted by I_i .

To model turning-related execution cost, the heading deviation between the current AUV heading and the direction toward the target viewpoint is defined as

$$\Delta\psi_i = |\text{atan2}(y_i - y_{AUV}, x_i - x_{AUV}) - \psi_{AUV}| \quad (4)$$

For n candidate viewpoints, the three evaluation terms are normalized as

$$I'_i = \frac{I_i}{\sum_{j=1}^n I_j}, \quad d'_i = \frac{d_i}{\sum_{j=1}^n d_j}, \quad \Delta\psi'_i = \frac{\Delta\psi_i}{\sum_{j=1}^n \Delta\psi_j} \quad (5)$$

The final viewpoint utility is defined as

$$J_i = \alpha I'_i - \beta d'_i - \gamma \Delta\psi'_i \quad (6)$$

where α , β , and γ are weighting coefficients.

In Equation (6), α controls the preference for frontier information gain, β penalizes travel distance, and γ penalizes heading deviation. The three terms are normalized before weighting, so the coefficients are selected according to their relative priority; in this study, information gain is kept dominant while the distance and heading penalties are tuned to suppress unnecessary long-distance motion and large turns. The weight coefficients were selected according to a clear priority principle. Since all three terms in Equation (6) are normalized to the same scale, the coefficients directly reflect their relative importance rather than absolute magnitudes. In our implementation, α (information gain) is set as the dominant term to ensure exploration progress, while β (distance penalty) and γ (heading penalty) are assigned smaller values to suppress unnecessary long-distance travel and excessive turning motions. For DWA evaluation in Equation (15), β (obstacle clearance) is prioritized for safety, followed by α (heading alignment) and γ (forward velocity) for smooth tracking. The same coefficient set is used across all experiments without scenario-specific tuning, which demonstrates that the framework is not overly sensitive to fine-grained weight variations as long as the priority order is maintained.

The optimal viewpoint is selected according to

$$\mathbf{v}^* = \arg \max_{\mathbf{v}_i \in V} J_i \quad (7)$$

Compared with conventional information-driven viewpoint evaluation, the proposed utility function explicitly incorporates heading-related motion cost into exploration

decision-making and therefore improves the consistency between viewpoint selection and actual motion execution.

3.4. Local–Global Exploration Coordination

To balance exploration efficiency and exploration completeness, a local-priority and global-backtracking exploration strategy is adopted.

After each local viewpoint-generation cycle, all local candidate viewpoints are evaluated using (6). If valid local viewpoints exist, the viewpoint with the maximum utility is selected as the next exploration target. This local-priority mechanism encourages the AUV to continuously explore nearby unknown regions with lower execution cost.

When no valid local viewpoint remains, the framework switches to the global viewpoint generator. Stored global viewpoints are updated and re-evaluated, and the best reachable global viewpoint is selected as the next navigation target. After the target is reached, the system returns to local exploration around the updated AUV position. When neither local nor global valid viewpoints exist, the exploration task is considered complete. Thus, the decision sequence in Figure 5 should be interpreted as follows: first evaluate all local candidates and then select the local maximum-utility viewpoint if at least one valid frontier-containing candidate exists; otherwise, re-evaluate reachable global candidates for backtracking; finally, stop only when both candidate sets are empty. The overall target assignment mechanism is shown in Figure 5.

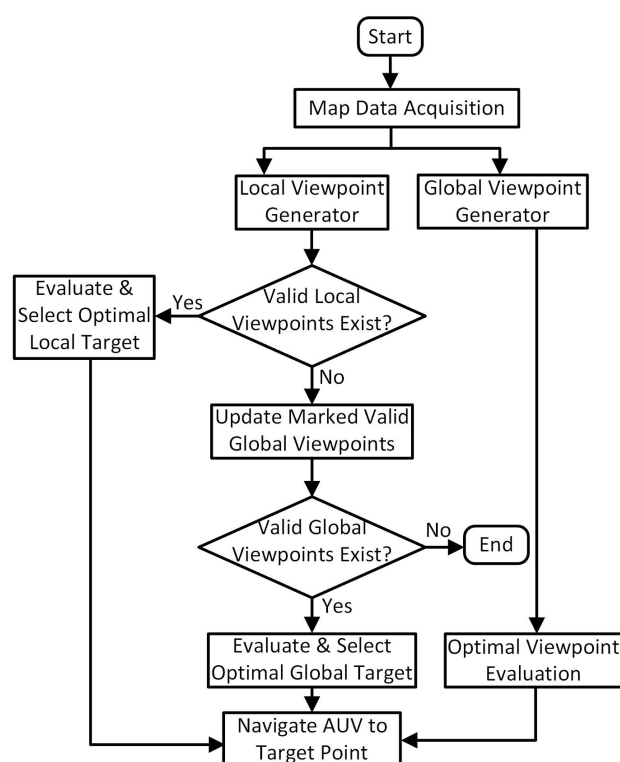


Figure 5. Target assignment mechanism.

3.5. Motion-Constrained Path Planning and Tracking

After a target viewpoint is selected, a safe and smooth path is generated on the current occupancy grid map. Since raw grid-based paths usually contain dense piecewise-linear turns that make it difficult to directly satisfy AUV continuous-curvature motion constraints, a motion-constrained path execution framework is developed by combining global path planning, path simplification, smooth trajectory generation, and local trajectory tracking.

An A*-based global planner is first employed to generate a collision-free path

$$f(n) = g(n) + h(n) \quad (8)$$

where $g(n)$ denotes the accumulated path cost and $h(n)$ denotes the heuristic distance to the target node. In the grid map used here, $h(n)$ is the Euclidean distance from node n to the target node, which provides an admissible heuristic for the eight-connected occupancy-grid search.

The raw A* path usually contains redundant intermediate points and dense turning segments. Therefore, the Douglas–Peucker algorithm is employed to extract representative keypoints while preserving the overall path geometry [22]. The corresponding process is illustrated in Figure 6.

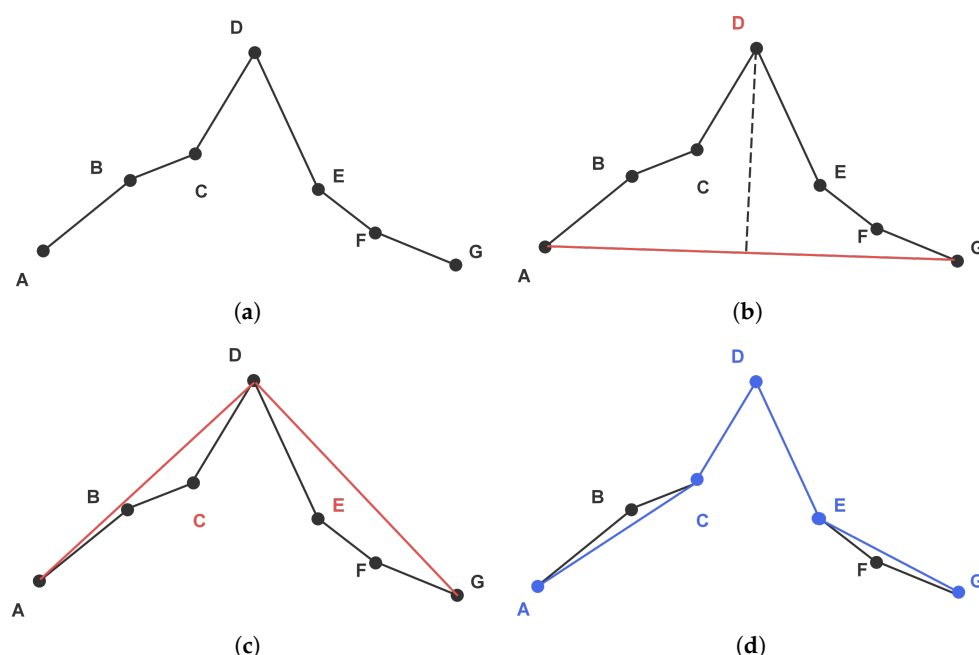


Figure 6. Illustration of Douglas–Peucker keypoint extraction. The black lines represent the original paths, capital letters represent the path points, the red lines are the connections between the first and last path points of the paths, the dashed line represents the distance from the farthest path point to the first and last path points, and the blue lines represent the paths formed by the path points obtained after the down-sampling process.

After keypoint extraction, cubic Bezier curves are employed to generate curvature-continuous trajectories suitable for AUV motion execution. The general form of the Bezier curve is

$$\mathbf{B}(t) = \sum_{i=0}^n P_i B_{i,n}(t), \quad B_{i,n}(t) = C_n^i t^i (1-t)^{n-i} \quad (9)$$

To balance smoothness and numerical stability, cubic Bezier curves are adopted:

$$\mathbf{B}(t) = (1-t)^3 \mathbf{P}_0 + 3(1-t)^2 t \mathbf{P}_1 + 3(1-t) t^2 \mathbf{P}_2 + t^3 \mathbf{P}_3, \quad t \in [0, 1] \quad (10)$$

For adjacent keypoints $P_1(x_1, y_1, \psi_1)$ and $P_2(x_2, y_2, \psi_2)$, additional control points are constructed as

$$\begin{aligned} x_{C_1} &= x_1 + \eta_{P_1 P_2} \cos \psi_1, & y_{C_1} &= y_1 + \eta_{P_1 P_2} \sin \psi_1, \\ x_{C_2} &= x_2 - \eta_{P_1 P_2} \cos \psi_2, & y_{C_2} &= y_2 - \eta_{P_1 P_2} \sin \psi_2 \end{aligned} \quad (11)$$

where

$$\eta_{P_1P_2} = \gamma d_{P_1P_2}, \quad d_{P_1P_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (12)$$

Here, the γ in (12) denotes the Bezier control-point offset coefficient, which is different from the heading-deviation weight in (6) and the velocity weight in (15). A larger value increases the control-point distance and produces smoother but more outward curves.

The generated Bezier trajectory is subsequently tracked using the dynamic window approach (DWA). Considering actuator limits, acceleration constraints, and obstacle safety constraints, the admissible dynamic window is defined as

$$V_r = V_m \cap V_a \cap V_d \quad (13)$$

For each sampled velocity pair (u, r) , the predicted AUV trajectory is updated according to

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \psi_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \psi_t \end{bmatrix} + \begin{bmatrix} u_t \cos \psi_t \\ u_t \sin \psi_t \\ r_t \end{bmatrix} \tau \quad (14)$$

The trajectory evaluation function jointly considers heading alignment, obstacle clearance, and motion efficiency

$$G(u, r) = \alpha \overline{heading}(u, r) + \beta \overline{obsdist}(u, r) + \gamma \overline{velocity}(u, r) \quad (15)$$

where α , β , and γ denote the weighting coefficients. In (15), these coefficients, respectively, weight heading alignment, obstacle clearance, and forward velocity after normalization. The values are chosen to prioritize safe path following while maintaining smooth forward motion.

Through the combination of global path smoothing and motion-constrained local trajectory optimization, the proposed framework improves the consistency between exploration planning and feasible AUV motion execution.

4. Experiments and Analysis

4.1. Experimental Platform and Evaluation Setup

The proposed method is evaluated on a high-fidelity AUV autonomous exploration simulation platform jointly developed using Unreal Engine (version 4.27, Epic Games, Inc., Cary, NC, USA) and Robot Operating System (ROS, version 1 Melodic, Open Robotics, Mountain View, CA, USA), as shown in Figure 7. The simulation platform integrates underwater environment rendering, scanning sonar simulation, AUV motion control, mapping, exploration planning, and path tracking. The IMU and scanning sonar in Unreal Engine provide vehicle state information and environmental observations. The sonar point cloud and AUV pose are transmitted to ROS topics, and GMapping is used to construct a two-dimensional occupancy grid map. Based on the constructed map, the proposed exploration framework performs local-global viewpoint generation, viewpoint evaluation, target assignment, and motion-constrained path execution. The platform is an in-house integration built from standard Unreal Engine visualization, ROS communication, GMapping-based mapping, and the exploration and control modules; the key interfaces and experimental parameters needed to reproduce the reported pipeline are summarized in Figure 7 and Table 1.

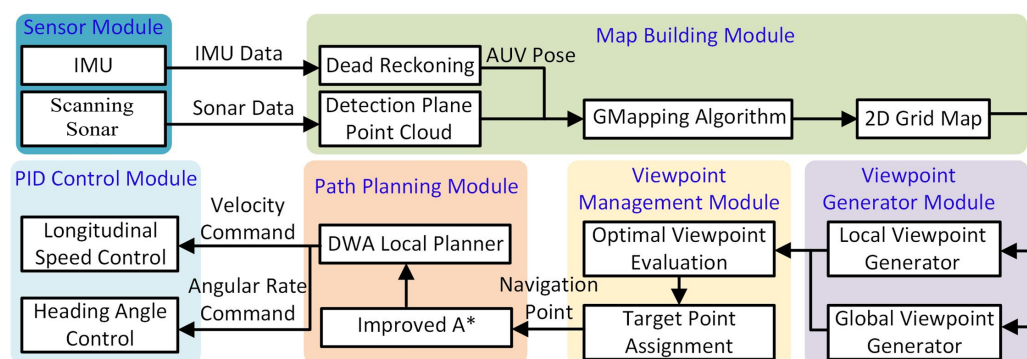


Figure 7. Simulation framework for AUV autonomous exploration in an unknown underwater environment.

Recent underwater simulation environments and datasets have shown that high-fidelity simulation platforms are effective tools for evaluating underwater navigation, inspection, and autonomous decision-making algorithms [23–26].

Two representative underwater exploration scenarios are constructed to evaluate the proposed method. Scenario 1 is an open environment with relatively sparse obstacles, while Scenario 2 contains surrounding obstacles and corridor-like structures with stronger geometric constraints. Both scenarios have dimensions of $60\text{ m} \times 60\text{ m}$, as shown in Figure 8. In the top-view experiments, the scanning sonar field of view is centered at the current AUV position; therefore, the observable region used by the local viewpoint generator changes with both vehicle pose and obstacle occlusion.

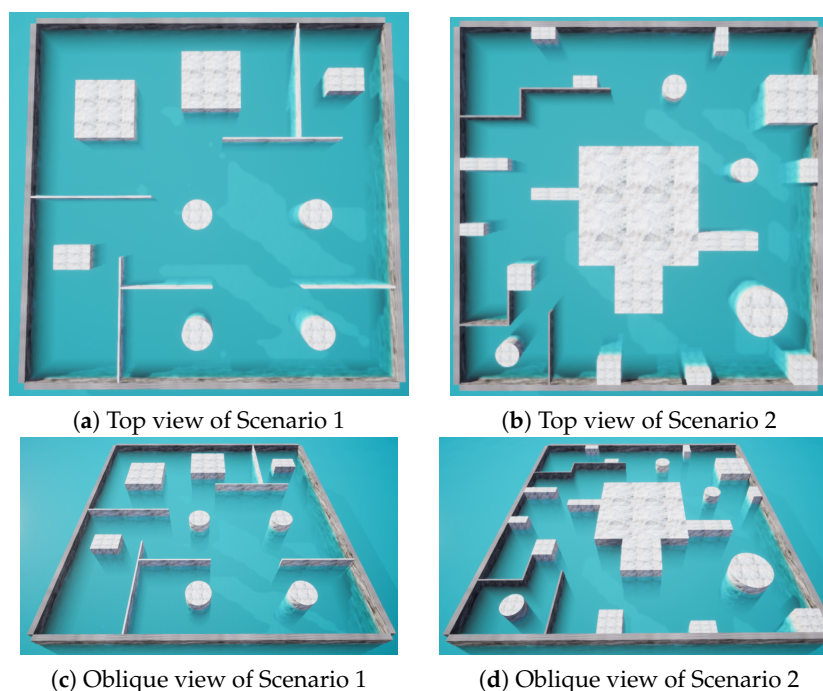


Figure 8. Experimental scenarios based on the AUV simulation platform. Panels (a,b) show the top-view layouts, while panels (c,d) show the corresponding three-dimensional visual scenes.

The exploration task is considered complete when the unknown region inside the task area has been explored and neither the local nor the global viewpoint generator produces a valid viewpoint containing frontier information. Exploration path length and mapping time are used as the primary evaluation metrics.

All the simulations are conducted on a computer equipped with an Intel Core i9-13980HX CPU and 32 GB RAM. Unreal Engine and Simulink (version R2023a, MathWorks,

Inc., Natick, MA, USA) run on Windows 11, while ROS runs on Ubuntu 18.04 under WSL2. The common experimental parameters are listed in Table 1. The maximum surge command of 0.5 m/s corresponds to a conservative low-speed inspection setting near underwater structures, where sensing stability and collision avoidance are prioritized over long-range transit speed.

Table 1. Experimental parameter configuration.

Parameter	Value
Max surge command	0.5 m/s
Max yaw-rate command	1 rad/s
Operation depth	10 m
RRT step length	2 m
Sonar beams	720
Max sonar range	15 m
Sonar sector angle	360°
Sonar update rate	20 Hz

4.2. Validation of Sonar-Aware Local Viewpoint Generation

To verify the effectiveness of the proposed sonar-aware local viewpoint generation strategy, experiments are conducted in a corridor environment. The corridor size is 30 m × 5 m, the AUV is initialized at the center, the maximum sonar range is 15 m, the RRT step length is 1 m, and the iteration number is set to 200 for both methods.

The proposed adaptive local viewpoint generator is compared with a conventional sliding-window RRT viewpoint generator. The comparison results are shown in Figure 9 and Table 2.

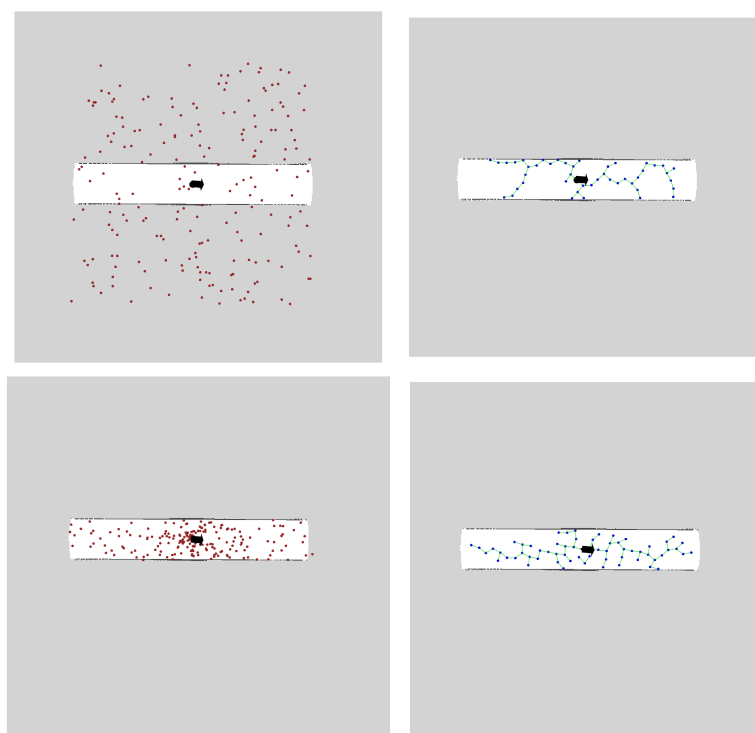


Figure 9. Comparison of local viewpoint generation efficiency. The black arrows indicate the current poses of the AUV, red dots indicate random sampling points, the blue dots are the nodes on the generated random tree, which can also be called viewpoints, the green line segments are the connections of nodes on the random tree.

Table 2. Experimental results of local viewpoint generation efficiency.

Algorithm	Trial	Viewpoints	Type-I Invalid	Type-II Invalid
Proposed	1	63	4	133
Proposed	2	68	11	121
Proposed	3	59	7	134
Proposed	4	66	12	122
Proposed	5	71	13	116
Proposed	Average	65	-	-
Sliding window	1	46	139	15
Sliding window	2	39	152	9
Sliding window	3	37	147	16
Sliding window	4	43	136	21
Sliding window	5	42	140	18
Sliding window	Average	41	-	-

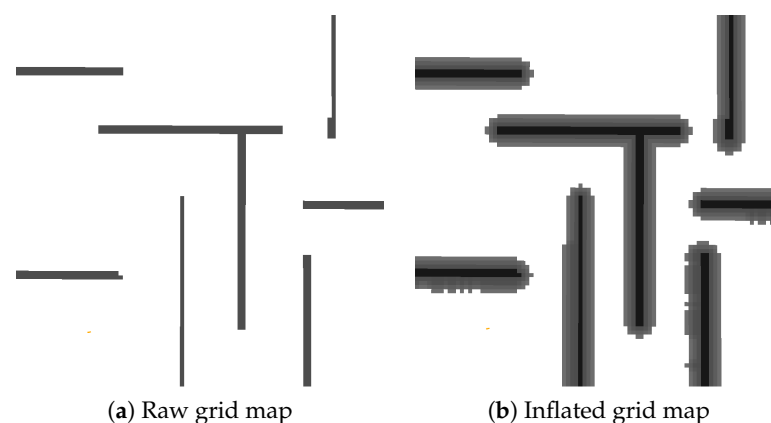
The proposed method generates significantly more valid viewpoints in the corridor environment. The average number of generated viewpoints increases from 41 to 65, corresponding to a 58.5% improvement over the sliding-window method. The sliding-window strategy produces a large number of Type-I invalid samples because the rectangular sampling region is inconsistent with the actual sonar-detectable area. In contrast, the proposed method aligns the sampling space with the real sonar coverage and therefore guides the RRT expansion more effectively toward reachable exploration regions.

The results indicate that the proposed sonar-aware sampling strategy improves the consistency between viewpoint generation and actual underwater sensing coverage, particularly in geometrically constrained environments.

4.3. Validation of Motion-Constrained Path Execution

To verify the effectiveness of the proposed motion-constrained path execution framework, path planning and local tracking experiments are conducted in RVIZ and the underwater simulation environment.

A known occupancy grid map with dimensions of 10 m $\times$ 10 m and a resolution of 0.1 m/cell is first constructed, as shown in Figure 10. White cells represent free space and black cells represent occupied obstacles. To improve navigation safety, obstacle inflation is performed before path planning.

**Figure 10.** Experimental map for path planning.

Two path planning experiments with different obstacle layouts are conducted, as shown in Figures 11 and 12. The start position is (−3.0, 3.0) with an initial heading of 0°.

In Experiment 1, the target pose is $(2.5, -3.5, -30^\circ)$, while, in Experiment 2, the target pose is $(3.5, 1.0, 0^\circ)$.

The Douglas–Peucker error threshold is set to 0.25 m, the maximum distance between adjacent keypoints is 2 m, the Bezier interpolation step is 0.2 m, and the control-point offset coefficient is 0.3. These values are implementation parameters of the path-processing and DWA modules; they were selected to keep the smoothed path close to the A* route while avoiding overly dense waypoints. In the figures, the black curve denotes the raw A* path, the purple nodes denote the extracted keypoints, and the green curve denotes the final Bezier-smoothed path.

The results show that the raw A* path contains dense turning points and discontinuous heading variations that are not suitable for direct AUV tracking. After Douglas–Peucker simplification, the extracted keypoints preserve the global path trend while reducing redundant intermediate points. Cubic Bezier smoothing further improves trajectory continuity and generates motion-consistent paths that are suitable for AUV execution.

The smoothed global paths are subsequently tracked using the DWA local planner in the underwater simulation environment. The global path generated by the improved planner is shown in Figure 13a, while the actual AUV tracking trajectory generated by DWA is shown in Figure 13b.

The trajectory, heading angle, surge velocity, and yaw rate during autonomous navigation are shown in Figure 14. The smooth heading evolution and stable motion-state variations indicate that the proposed planning-and-tracking framework can satisfy the continuous-motion constraints of the AUV while maintaining stable local trajectory tracking.

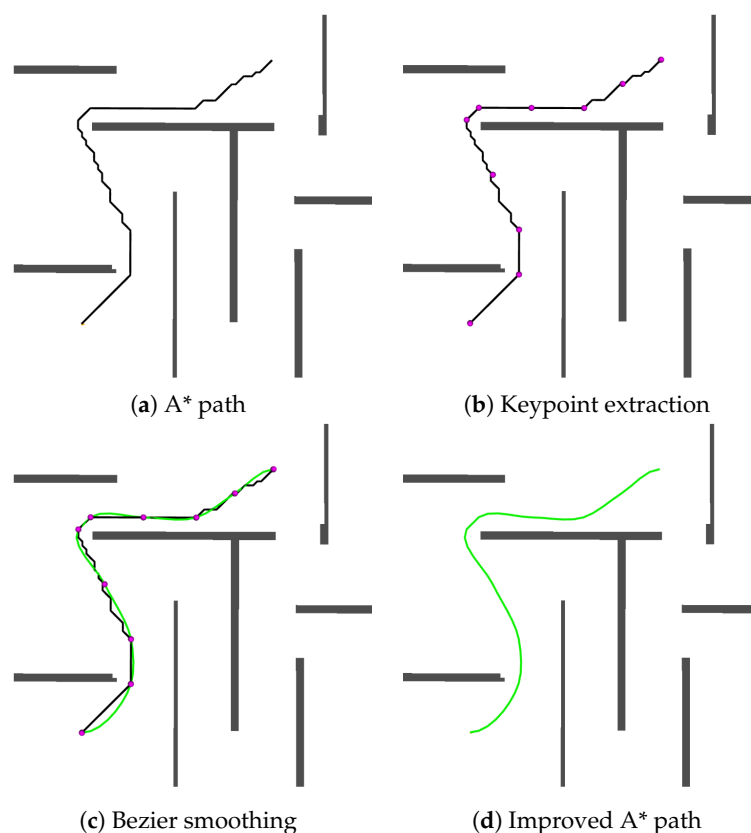


Figure 11. Path planning results in Experiment 1. The black paths represent the paths planned by the original A* algorithm, purple points are path key points extracted by Douglas Peucker algorithm, and green curves represent the path obtained after smoothing by the third-order Bezier curve.

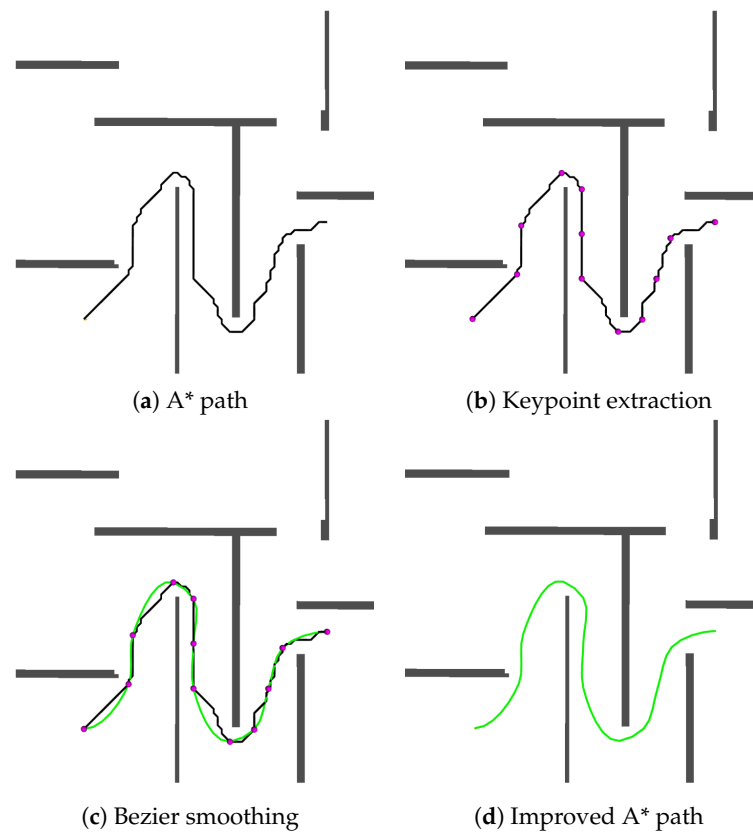


Figure 12. Path planning results in Experiment 2. The black paths represent the paths planned by the original A* algorithm, purple points are path key points extracted by Douglas Peucker algorithm, and green curves represent the path obtained after smoothing by the third-order Bezier curve.

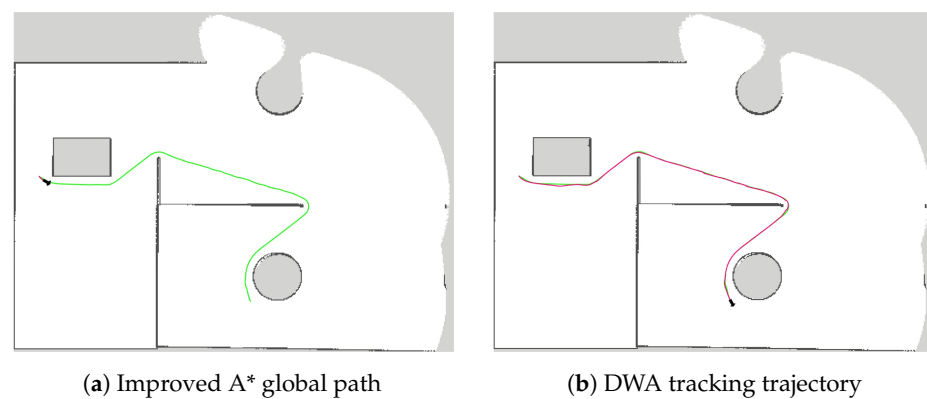


Figure 13. Experimental result of the DWA local planner. The black arrows indicate the real-time pose of the AUV, green paths are the paths planned by the improved A* algorithm, and the red paths indicate the actual routes of AUV. Panel (a) shows the smoothed global reference path, and panel (b) shows the actual trajectory tracked by the AUV under DWA velocity commands.

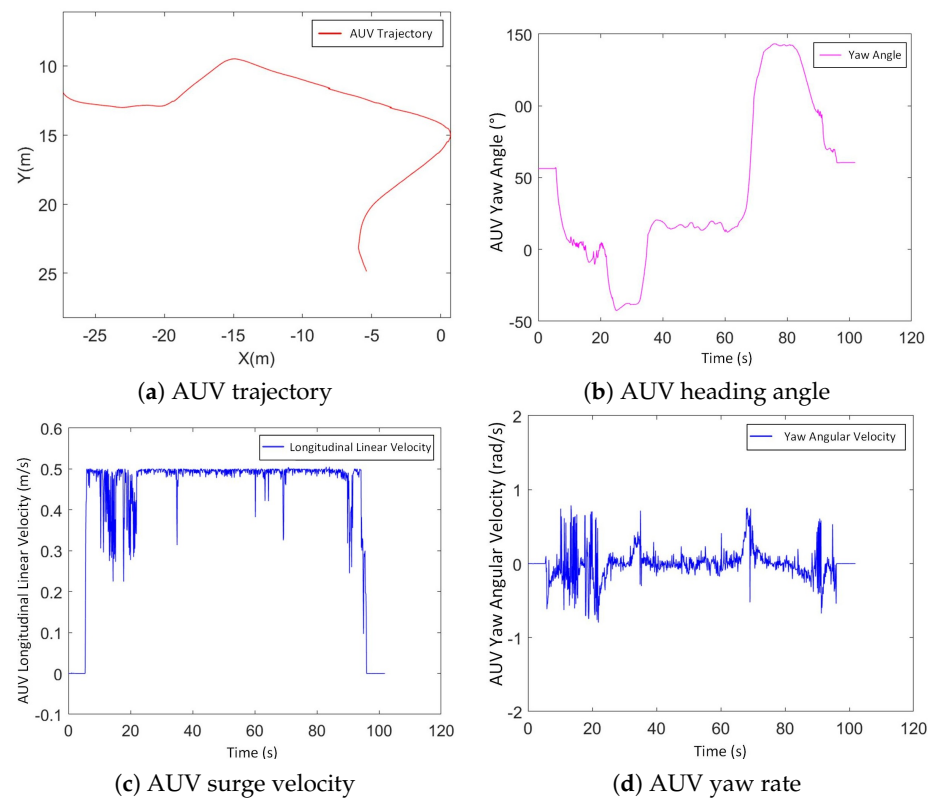


Figure 14. AUV motion-state data during autonomous exploration. Panels (a–d), respectively, report the AUV trajectory, heading angle, surge velocity, and yaw rate, which are used to verify that the smoothed path can be tracked with continuous heading and bounded velocity commands.

4.4. Overall Autonomous Exploration Performance

The proposed exploration framework is compared with the classical frontier-based exploration method in both experimental scenarios to evaluate the overall exploration efficiency under different underwater structural conditions. The frontier-based baseline is selected as a controlled reference because it directly corresponds to target selection without heading-dependent sensing. Therefore, the experimental comparison focuses on the benefits of sonar-aware sampling and motion-aware execution under the same AUV simulation, mapping, and control settings.

Figures 15 and 16 show the exploration process and the resulting exploration trajectory of the proposed method in Scenario 1.

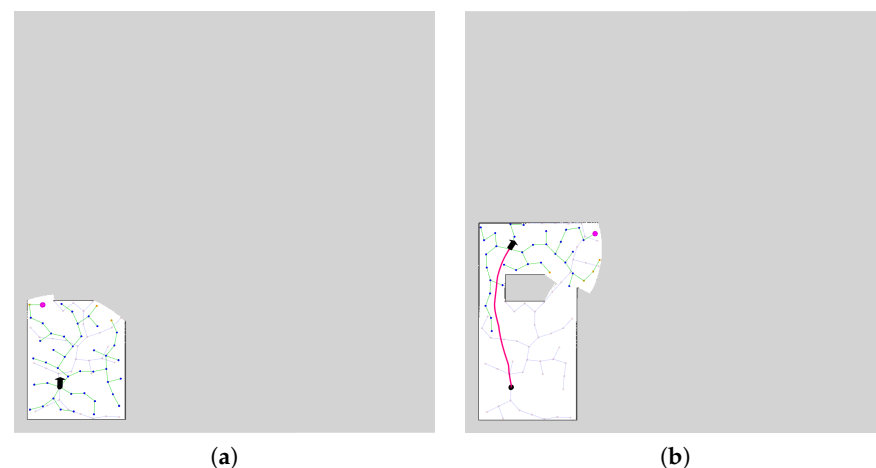


Figure 15. Cont.



Figure 15. AUV autonomous exploration process of the proposed method in Scenario 1. The red paths mark actual navigation trajectories of AUV. Its starting position, current target point, and real-time pose are indicated by the black dot, purple dot, and black arrows, respectively. For exploration planning, the blue dots and green lines represent the viewpoints and edges of the local random tree, while the brown dots and dark blue lines denote those of the global random tree.

During exploration, the local viewpoint generator continuously produces candidate viewpoints near the current AUV position, allowing the system to preferentially explore nearby unknown regions. When the local generator no longer produces valid viewpoints containing frontier information, the target assignment module switches to the global viewpoint generator and selects the globally optimal reachable viewpoint for backtracking exploration, as shown in Figure 15d. This local–global coordination mechanism enables

the AUV to maintain exploration continuity while avoiding unnecessary long-distance traversal during local exploration.

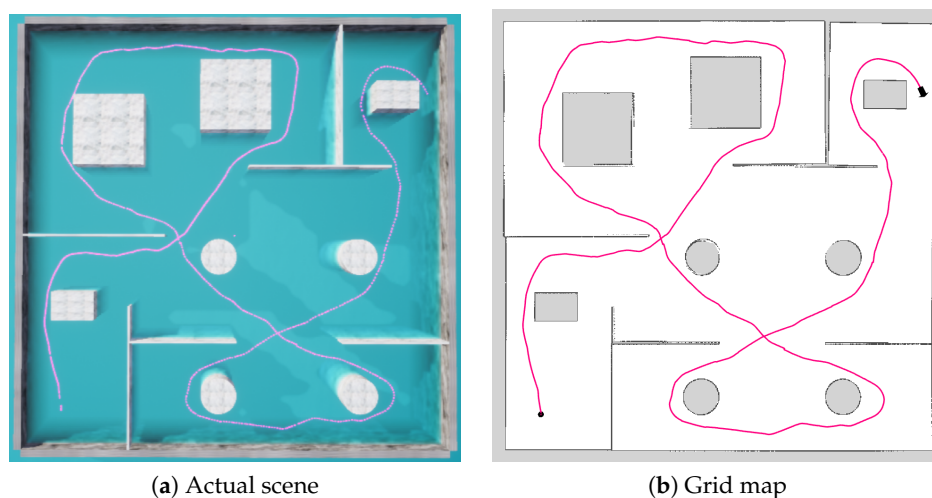


Figure 16. Path comparison of the proposed method in the actual scene and grid map in Scenario 1. The purple path represents the route of AUV in the simulation environment, and the red path represents its route in the built 2D grid map.

The corresponding exploration result of the frontier-based method in Scenario 1 is shown in Figure 17.

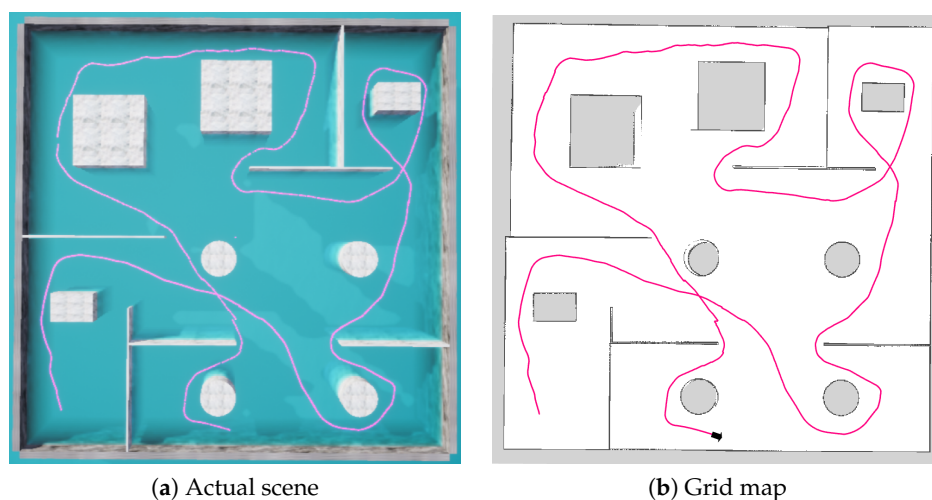


Figure 17. Path comparison of the frontier-based algorithm in the actual scene and grid map in Scenario 1. The purple path represents the route of AUV in the simulation environment, and the red path represents its route in the built 2D grid map.

Because the frontier-based strategy selects frontier targets mainly according to global frontier information without explicitly considering heading-dependent sensing and turning cost, the resulting exploration trajectory contains additional long-distance transitions and repeated backtracking motions. In contrast, the proposed method incorporates sensing coverage and motion cost into viewpoint generation and evaluation, thereby improving exploration efficiency.

To further contextualize the performance of the proposed method, we provide a qualitative comparison against two representative state-of-the-art exploration frameworks: the DSVP [18] and TARE [10]. The DSVP restricts viewpoint sampling within a fixed sliding window that is decoupled from the actual sensor field of view. The actual detection range of the sonar will dynamically change according to the surrounding environment, resulting in

some areas within the sliding window being inaccessible. This fixed-window strategy tends to generate a large number of invalid samples outside the observable region, which is the same issue we quantitatively analyzed in Section 4.2. TARE employs a hierarchical global–local planning structure; however, its global planner does not explicitly account for the AUV’s continuous-curvature constraints, and its local viewpoint generation still relies on a predefined geometric space rather than real-time sonar coverage. In contrast, our method constructs the sampling region directly from the instantaneous sonar detection sector and incorporates heading deviation into the utility function, thereby avoiding these two limitations. A full quantitative comparison against these methods on a unified simulation platform is left as immediate future work. To quantitatively evaluate the exploration performance, five repeated experiments are conducted in Scenario 1. The experimental results are listed in Table 3.

As shown in Table 3, the proposed method achieves shorter exploration paths and lower mapping time in all the trials. Compared with the frontier-based method, the average exploration path length is reduced by 15.12%, while the average mapping time is reduced by 13.83%. The results indicate that the proposed sensing–motion-aware exploration framework can improve exploration efficiency in open underwater environments.

Table 3. Repeated experimental results in Scenario 1.

Trial	Algorithm	Path Length (m)	Mapping Time (s)
1	Proposed	285.36	608.43
	Frontier-Based	328.32	692.96
2	Proposed	290.14	614.97
	Frontier-Based	331.47	696.33
3	Proposed	282.63	605.78
	Frontier-Based	342.78	714.12
4	Proposed	279.05	593.72
	Frontier-Based	325.89	690.17
5	Proposed	276.94	591.75
	Frontier-Based	337.59	704.78
Average	Proposed	282.82	602.93
	Frontier-Based	333.21	699.67

To further evaluate the exploration performance under stronger geometric constraints, additional experiments are conducted in Scenario 2, which contains surrounding obstacles and corridor-like structures. The exploration process and trajectory comparison are shown in Figures 18 and 19.

Because the initial exploration area in Scenario 2 contains corridor-like structures, the proposed sonar-aware local viewpoint generator tends to produce viewpoints distributed along reachable sensing regions. This characteristic allows the AUV to gradually move toward the corridor exit while maintaining efficient local exploration. Furthermore, the viewpoint evaluation function explicitly incorporates heading deviation cost, which suppresses excessive turning motions during exploration. As a result, the AUV trajectory remains smoother during surrounding exploration around obstacle-dense regions.

When the local viewpoint generator no longer contains valid frontier viewpoints, the target assignment module switches to the global viewpoint generator and selects the best reachable global viewpoint for exploration continuation, as shown in Figure 18f. This mechanism improves exploration completeness while reducing blind backtracking behavior.

The exploration result of the frontier-based method in Scenario 2 is shown in Figure 20.

Because the frontier-based method mainly relies on global frontier information without explicitly modeling motion constraints and heading-dependent sensing, the resulting exploration trajectory still contains repeated surrounding motions and additional long-distance transitions.

Five repeated experiments are conducted in Scenario 2, and the quantitative comparison results are listed in Table 4.

As shown in Table 4, the proposed method consistently outperforms the frontier-based method in both exploration path length and mapping time. Compared with the frontier-based method, the average exploration path length is reduced by 12.31%, while the average mapping time is reduced by 11.41%.

Overall, the experimental results demonstrate that the proposed framework achieves stable improvements in viewpoint generation efficiency, motion-consistent path execution, and overall exploration performance. By coupling sonar-aware viewpoint generation, motion-aware viewpoint evaluation, local–global target assignment, and motion-constrained path execution into a unified framework, the proposed method improves exploration efficiency in both open and geometrically constrained underwater environments.

Table 4. Repeated experimental results in Scenario 2.

Trial	Algorithm	Path Length (m)	Mapping Time (s)
1	Proposed	270.19	574.92
	Frontier-Based	305.84	638.31
2	Proposed	264.47	564.83
	Frontier-Based	313.93	665.09
3	Proposed	267.02	566.64
	Frontier-Based	310.62	653.49
4	Proposed	272.86	577.76
	Frontier-Based	298.35	625.57
5	Proposed	268.71	571.42
	Frontier-Based	303.15	640.74
Average	Proposed	268.65	571.11
	Frontier-Based	306.38	644.64

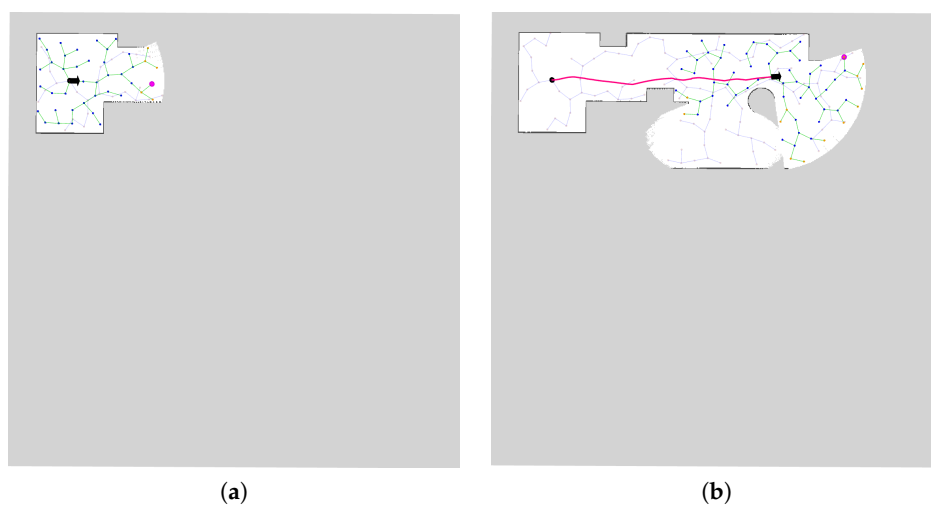


Figure 18. *Cont.*

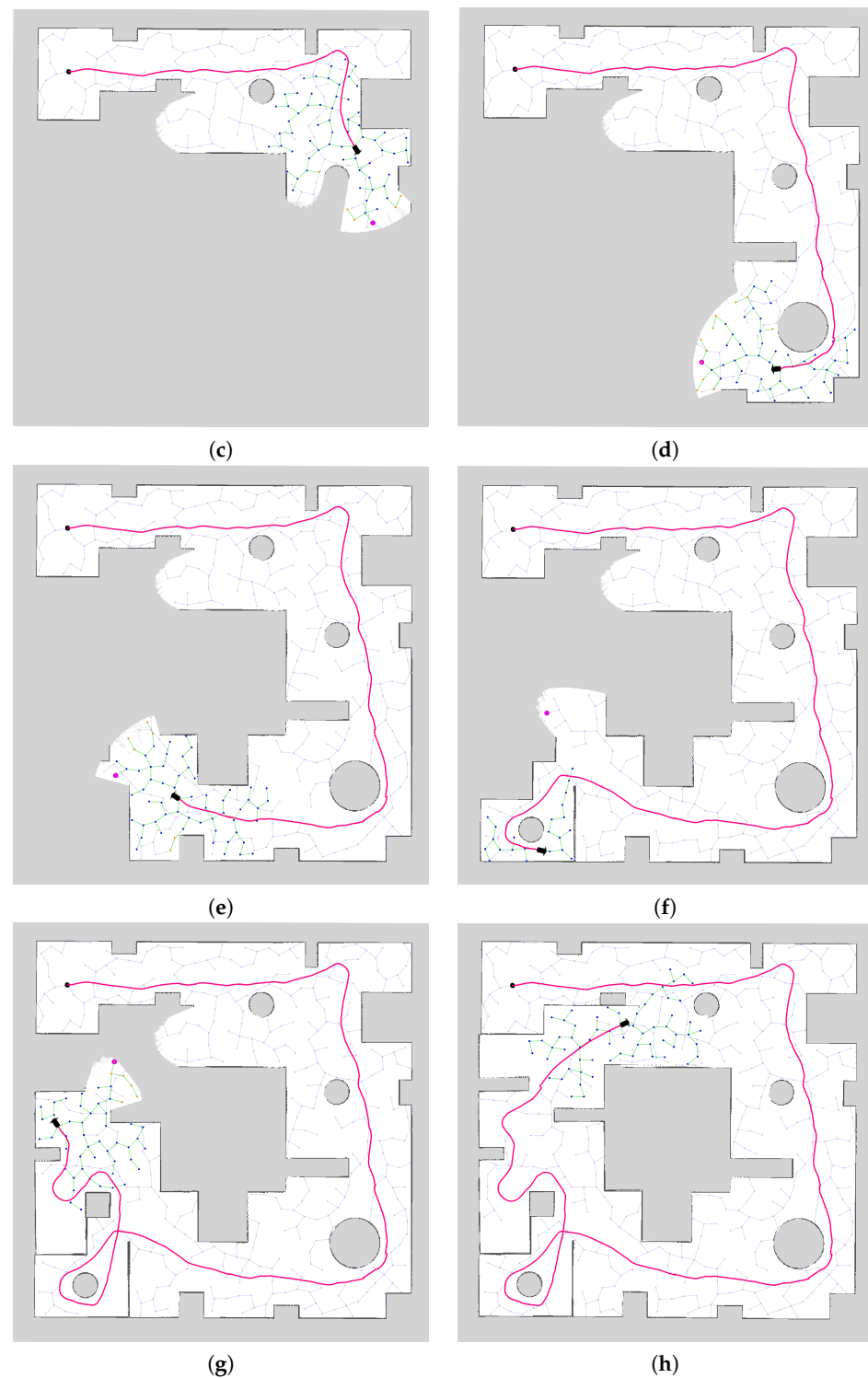


Figure 18. AUV autonomous exploration process of the proposed method in Scenario 2. The red paths mark actual navigation trajectories of AUV. Its starting position, current target point, and real-time pose are indicated by the black dot, purple dot, and black arrows, respectively. For exploration planning, the blue dots and green lines represent the viewpoints and edges of the local random tree, while the brown dots and dark blue lines denote those of the global random tree.

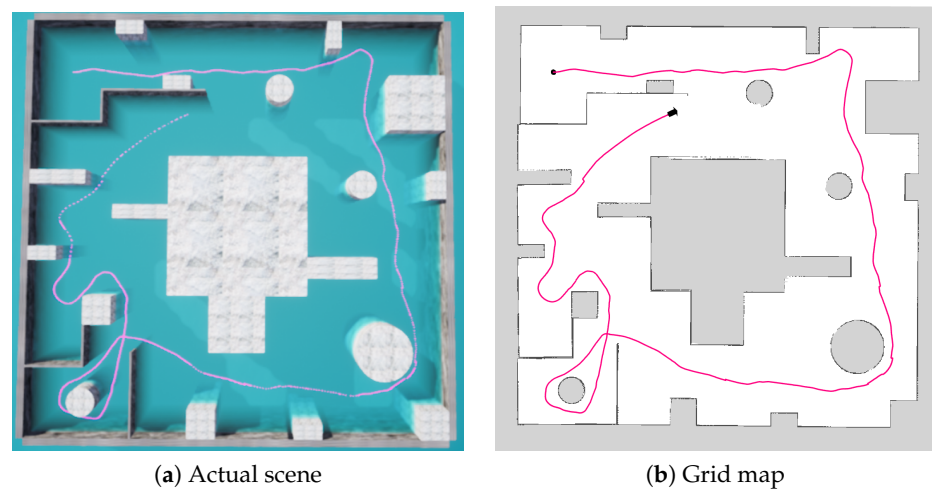


Figure 19. Path comparison of the proposed method in the actual scene and grid map in Scenario 2. The purple path represents the route of AUV in the simulation environment, and the red path represents its route in the built 2D grid map.

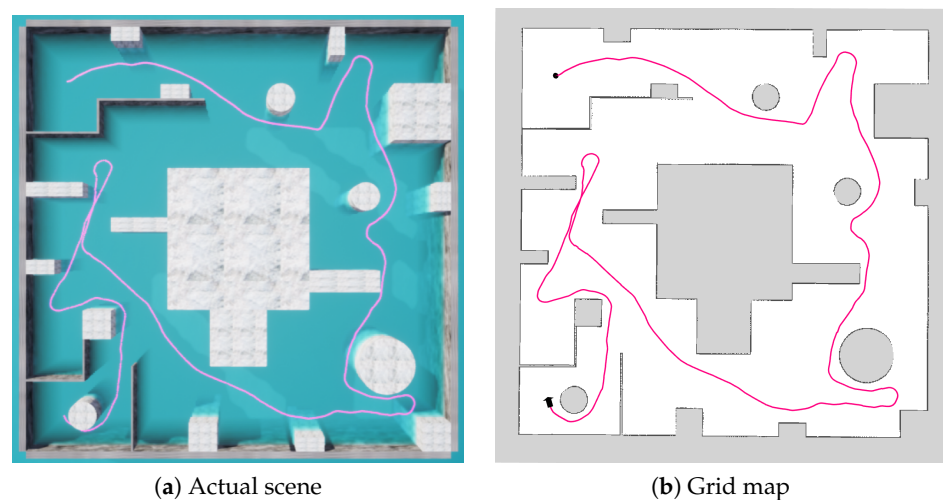


Figure 20. Path comparison of the frontier-based algorithm in the actual scene and grid map in Scenario 2. The purple path represents the route of AUV in the simulation environment, and the red path represents its route in the built 2D grid map.

It should be noted that the current study is conducted in a fixed-depth two-dimensional exploration setting, and the sonar model is simplified relative to real underwater acoustic sensing conditions. Furthermore, the current validation is limited to simulation experiments. Future work will focus on extending the proposed framework to three-dimensional underwater environments, integrating higher-fidelity sonar sensing models, and conducting real-world AUV experiments in marine environments.

4.5. Discussion

The experimental results indicate that the main challenge in AUV autonomous exploration lies not merely in boundary search or path planning itself but rather in designing exploration planning algorithms that are suitable for the AUV's motion perception characteristics and have appropriate complexity under its limited computational resources. When designing such algorithms, it is essential to minimize computational complexity, reduce unnecessary computational overhead, and incorporate the cost of AUV heading adjustments into the evaluation criteria for various decision-making processes.

From this perspective, the effectiveness of the proposed framework mainly comes from maintaining consistency between sensing coverage and motion execution during

exploration. In local viewpoint generation, the sampling region is directly constrained by the actual sonar detection coverage instead of a predefined geometric window. This mechanism changes the spatial distribution of random-tree expansion and reduces invalid growth toward unreachable or weakly observable regions. In viewpoint evaluation, introducing heading deviation into the utility function suppresses exploration decisions that require excessive turning. The experiments in surrounding-obstacle environments indicate that this mechanism can reduce repeated heading adjustments and unnecessary backtracking motions during exploration. In path execution, the combination of path smoothing and motion-constrained local tracking further improves the continuity between global planning and AUV motion behavior, reducing the inconsistency between planned paths and executable trajectories.

The comparison with the frontier-based method also reflects the influence of sensing-motion coupling on exploration behavior. Frontier-based exploration tends to prioritize regions with higher frontier information gain from a global perspective. However, when directional sensing and turning costs are considered, the globally information-rich region may not correspond to the motion-efficient exploration target. This mismatch becomes more significant in corridor-like and surrounding-obstacle environments, where large heading adjustments and repeated navigation transitions can substantially increase exploration cost. The proposed framework alleviates this issue by incorporating sensing constraints and motion characteristics into viewpoint generation, evaluation, and execution simultaneously rather than treating them as isolated modules.

Several limitations still exist in the current study. The proposed framework is developed under a fixed-depth two-dimensional exploration assumption and therefore does not fully capture the complexity of three-dimensional underwater motion and perception. This 2D setting is appropriate for applications such as fixed-depth inspection along pipelines, walls, and planar structural sections, where the main decision difficulty is horizontal viewpoint selection under directional sensing. In addition, the sonar sensing model used in simulation is simplified relative to real underwater acoustic propagation and does not explicitly model uncertainty caused by acoustic noise, multipath effects, water-current disturbances, or dynamic environmental interference. Ocean currents may change effective turning cost, energy consumption, and tracking error; therefore, current-aware cost terms and disturbance observers should be incorporated before deployment in strong-flow environments. Furthermore, the current experiments are conducted in relatively static environments, while real underwater exploration tasks may involve moving obstacles and time-varying environmental conditions. These factors may influence viewpoint evaluation, trajectory execution, and long-term exploration stability in practical deployments.

5. Conclusions

This paper proposed a motion-aware autonomous exploration framework for AUVs operating in complex underwater environments. Unlike conventional exploration methods that implicitly support high-complexity algorithms and low-cost turning, the proposed framework has been designed with an effective sampling mechanism based on the actual detection coverage area of the sonar for viewpoint generation and has also been designed with corresponding planning and motion execution methods in combination with the movement characteristics of the AUV. Meanwhile, the framework integrates sonar-aware local viewpoint generation, motion-aware viewpoint evaluation, local-global target assignment, and motion-constrained path execution into a unified autonomous exploration pipeline.

In the viewpoint-generation stage, the actual sonar detection coverage is used directly as the local sampling region, ensuring that the random-tree expansion process remains consistent with the AUV's effective sensing region. A two-stage weighted sampling strategy

further improves valid viewpoint generation efficiency in geometrically constrained environments. In the viewpoint-evaluation stage, heading deviation cost is incorporated into the utility function to suppress excessive turning motions during exploration. A local-priority and global-backtracking target assignment strategy is then introduced to balance local exploration efficiency and global exploration completeness. Finally, a motion-constrained path execution framework that combines Douglas–Peucker keypoint extraction, cubic Bezier smoothing, and DWA-based local tracking is designed to generate trajectories that are consistent with AUV motion characteristics.

Extensive experiments are conducted on a high-fidelity underwater simulation platform developed using Unreal Engine and ROS. The experimental results demonstrate that, in corridor environments, the proposed adaptive sampling strategy increases valid viewpoint generation by 58.5%. In open and obstacle-surrounded scenarios, it reduces average exploration path length by 15.12% and 12.31% and mapping time by 13.83% and 11.41%, respectively. The results further verify that explicitly considering the coupling between sensing coverage and motion cost can improve the overall efficiency and motion consistency of AUV autonomous exploration.

The current study is conducted in a fixed-depth two-dimensional exploration setting with a simplified sonar sensing model. Although the proposed framework demonstrates stable performance improvements in simulation environments, several challenges remain for practical underwater deployment. Future work will focus on extending the framework to fully three-dimensional underwater exploration, integrating higher-fidelity acoustic sensing and uncertainty modeling, and validating the proposed method on physical AUV platforms in real marine environments. In addition, future research will investigate co-operative multi-AUV exploration and learning-based adaptive exploration strategies for large-scale unknown underwater environments.

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

Funding: This research received no external funding.

Data Availability Statement: All data supporting the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Panda, M.; Das, B.; Subudhi, B.; Pati, B.B. A Comprehensive Review of Path Planning Algorithms for Autonomous Underwater Vehicles. *Int. J. Autom. Comput.* **2020**, *17*, 321–352. [\[CrossRef\]](#)
2. Cheng, C.; Sha, Q.; He, B.; Li, G. Path Planning and Obstacle Avoidance for AUV: A Review. *Ocean Eng.* **2021**, *235*, 109355. [\[CrossRef\]](#)
3. Ling, Y.; Li, Y.; Ma, T.; Cong, Z.; Xu, S.; Li, Z. Active Bathymetric SLAM for Autonomous Underwater Exploration. *Appl. Ocean Res.* **2023**, *130*, 103439. [\[CrossRef\]](#)
4. Palomeras, N.; Hurtos, N.; Vidal, E.; Carreras, M. Autonomous Exploration of Complex Underwater Environments Using a Probabilistic Next-Best-View Planner. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1619–1625. [\[CrossRef\]](#)
5. Vidal, E.; Palomeras, N.; Istenic, K.; Gracias, N.; Carreras, M. Multisensor Online 3D View Planning for Autonomous Underwater Exploration. *J. Field Robot.* **2020**, *37*, 1123–1147. [\[CrossRef\]](#)
6. Zacchini, L.; Franchi, M.; Ridolfi, A. Sensor-Driven Autonomous Underwater Inspections: A Receding-Horizon RRT-Based View Planning Solution for AUVs. *J. Field Robot.* **2022**, *39*, 499–527. [\[CrossRef\]](#)
7. Yamauchi, B. A Frontier-Based Approach for Autonomous Exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*; IEEE: New York, NY, USA, 1997; pp. 146–151.

8. Sun, Z.; Wu, B.; Xu, C.Z.; Sarma, S.E.; Yang, J.; Kong, H. Frontier Detection and Reachability Analysis for Efficient 2D Graph-SLAM Based Active Exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: New York, NY, USA, 2020; pp. 2051–2058. [\[CrossRef\]](#)
9. Zhou, B.; Zhang, Y.; Chen, X.; Shen, S. FUEL: Fast UAV Exploration Using Incremental Frontier Structure and Hierarchical Planning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 779–786. [\[CrossRef\]](#)
10. Cao, C.; Zhu, H.; Choset, H.; Zhang, J. TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments. In *Proceedings of the Robotics: Science and Systems, Virtual*, 12–16 July 2021. [\[CrossRef\]](#)
11. Keidar, M.; Kaminka, G.A. Efficient Frontier Detection for Robot Exploration. *Int. J. Robot. Res.* **2014**, *33*, 215–236. [\[CrossRef\]](#)
12. Li, X.; Qiu, H.; Jia, S.; Gong, Y. Dynamic algorithm for safe and reachable frontier point generation for robot exploration. In *Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation*; IEEE: New York, NY, USA, 2016; pp. 2088–2093. [\[CrossRef\]](#)
13. Wang, C.; Chi, G.; Sun, Y.; Meng, M.Q.H. Autonomous Robotic Exploration by Incremental Road Map Construction. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1720–1731. [\[CrossRef\]](#)
14. Zhou, B.; Xu, H.; Shen, S. RACER: Rapid Collaborative Exploration With a Decentralized Multi-UAV System. *IEEE Trans. Robot.* **2023**, *39*, 1816–1835. [\[CrossRef\]](#)
15. Gonzalez-Banos, H.H.; Latombe, J.C. Navigation Strategies for Exploring Indoor Environments. *Int. J. Robot. Res.* **2002**, *21*, 829–848. [\[CrossRef\]](#)
16. LaValle, S.M.; Kuffner, J.J. Rapidly-Exploring Random Trees: Progress and Prospects. In *Algorithmic and Computational Robotics: New Directions*; CRC Press: Boca Raton, FL, USA, 2001; pp. 293–308.
17. Bircher, A.; Kamel, M.; Alexis, K.; Oleynikova, H.; Siegwart, R. Receding Horizon “Next-Best-View” Planner for 3D Exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*; IEEE: New York, NY, USA, 2016; pp. 1462–1468. [\[CrossRef\]](#)
18. Zhu, H.; Cao, C.; Xia, Y.; Scherer, S.; Zhang, J.; Choset, H. DSVP: Dual-Stage Viewpoint Planner for Rapid Exploration by Dynamic Expansion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: New York, NY, USA, 2021; pp. 7623–7630. [\[CrossRef\]](#)
19. Palomeras, N.; Hurtos, N.; Carreras, M.; Ridao, P. Autonomous Mapping of Underwater 3-D Structures: From View Planning to Execution. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1965–1971. [\[CrossRef\]](#)
20. Vidal, E.; Palomeras, N.; Istenic, K.; Hernandez, J.D.; Carreras, M. Two-Dimensional Frontier-Based Viewpoint Generation for Exploring and Mapping Underwater Environments. *Sensors* **2019**, *19*, 1460. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Karaman, S.; Frazzoli, E. Sampling-Based Algorithms for Optimal Motion Planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
22. Douglas, D.H.; Peucker, T.K. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *Can. Cartogr.* **1973**, *10*, 112–122. [\[CrossRef\]](#)
23. Zhang, M.M.; Choi, W.S.; Herman, J.; Davis, D.; Vogt, K.; McCarrin, M.; Vijay, K.; Dutia, D.; Lew, T.H.; Peters, J.; et al. DAVE Aquatic Virtual Environment: Toward a General Underwater Robotics Simulator. In *Proceedings of the IEEE/OES Autonomous Underwater Vehicles Symposium*; IEEE: New York, NY, USA, 2022; pp. 1–8. [\[CrossRef\]](#)
24. Potokar, E.; Ashford, S.; Kaess, M.; Mangelson, J.G. HoloOcean: An Underwater Robotics Simulator. In *Proceedings of the IEEE International Conference on Robotics and Automation*; IEEE: New York, NY, USA, 2022; pp. 3040–3046. [\[CrossRef\]](#)
25. Alvarez-Tunon, O.; Kanner, H.; Marnet, L.R.; Pham, H.X.; Sejerssen, J.I.F.; Brodskiy, Y.; Kayacan, E. MIMIR-UW: A Multipurpose Synthetic Dataset for Underwater Navigation and Inspection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: New York, NY, USA, 2023; pp. 6141–6148. [\[CrossRef\]](#)
26. Liu, W.; Bai, K.; He, X.; Song, S.; Zheng, C.; Liu, X. FishGym: A High-Performance Physics-Based Simulation Framework for Underwater Robot Learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*; IEEE: New York, NY, USA, 2022; pp. 6268–6275. [\[CrossRef\]](#)

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.