drones

MDPI

Article

Topological Map-Based Autonomous Exploration in Large-Scale Scenes for Unmanned Vehicles

Ziyu Cao [1,*], Zhihui Du [2] and Jianhua Yang [1]

[1] School of Automation, Northwestern Polytechnical University, Xi'an 710129, China; yangjianhua@nwpu.edu.cn
[2] Department of Precision Instrument, Tsinghua University, Beijing 100084, China; zhihuidu@mail.tsinghua.edu.cn
[*] Correspondence: caoziyu@mail.nwpu.edu.cn

Abstract: Robot autonomous exploration is a challenging and valuable research field that has attracted widespread research interest in recent years. However, existing methods often encounter problems such as incomplete exploration, repeated exploration paths, and low exploration efficiency when facing large-scale scenes. Considering that many indoor and outdoor scenes usually have a prior topological map, such as road navigation maps, satellite road network maps, indoor computer-aided design (CAD) maps, etc., this paper incorporated this information into the autonomous exploration framework and proposed an innovative topological map-based autonomous exploration method for large-scale scenes. The key idea of the proposed method is to plan exploration paths with long-term benefits by tightly merging the information between robot-collected and prior topological maps. The exploration path follows a global exploration strategy but prioritizes exploring scenes outside the prior information, thereby preventing the robot from revisiting explored areas and avoiding the duplication of any effort. Furthermore, to improve the stability of exploration efficiency, the exploration path is further refined by assessing the cost and reward of each candidate viewpoint through a fast method. Simulation experimental results demonstrated that the proposed method outperforms state-of-the-art autonomous exploration methods in efficiency and stability and is more suitable for exploration in large-scale scenes. Real-world experimentation has also proven the effectiveness of our proposed method.

Keywords: autonomous exploration; path planning; PCATSP; prior information; topological map

check for updates

Citation: Cao, Z.; Du, Z.; Yang, J. Topological Map-Based Autonomous Exploration in Large-Scale Scenes for Unmanned Vehicles. *Drones* **2024**, *8*, 124. https://doi.org/10.3390/drones8040124

Academic Editor: Diego González-Aguilera

Received: 29 February 2024
Revised: 23 March 2024
Accepted: 24 March 2024
Published: 27 March 2024

1. Introduction

Autonomous robot exploration technology requires robots to collect data within a given region and construct corresponding environmental maps. As a critical technology that reveals robotic autonomy, relevant research in robotics has garnered significant attention, driving widespread applications in geological exploration, 3D reconstruction, post-disaster rescue, and other fields.

Numerous autonomous exploration methods have been proposed in recent years and are divided into sampling-based and frontier-based categories. Sampling-based methods originated from the NBV (Next Best View) algorithm in the field of 3D reconstruction. RH-NBV (recurrent hybrid neural-based visual) first introduced the NBV algorithm into the autonomous exploration field [1], which consisted of the robot randomly sampling viewpoints in explored free space, constructing a rapidly exploring random tree (RRT), and evaluating the utility of each branch on the RRT. Finally, the robot focuses on the branch with the highest information reward and selects the first node of this branch as the local target. After that, numerous researchers have extended and improved the RH-NBV to meet the requirements of various application scenarios [2–6]. However, sampling-based autonomous exploration methods have lower exploration efficiency and lead to the robot being trapped. Ref. [7] first introduced the frontier-based exploration method, which groups free voxels adjacent to unknown voxels as frontier clusters and then drives the robot towards these frontier clusters to move to explore unknown areas. Since then, many frontier-based exploration methods have been proposed to meet various application requirements [8–10]. Ref. [11] proposed to select a viewpoint with minimal speed changes as the next goal within the Field Of View (FOV) of sensors, aiming to maintain the high movement speed of unmanned aerial vehicles (UAVs) and achieve efficient exploration. Fast UAV expLoration (FUEL) proposes the incremental frontier information structure (FIS) to address the problem of high computation of frontier extraction and low decision frequency of the path planner [12]. Based on FIS, UAVs can quickly and incrementally extract environmental information that the planner needs and promptly plan the exploration path.

However, most autonomous exploration methods tend to greedily guide the robot to exploration scenes with immediate rewards and neglect some targets with long-term rewards, resulting in lower efficiency in global exploration [12]. Although some methods plan paths from the global exploration standpoint, the robot inevitably overlooks some scenarios during exploration because of the limited perception range of sensors and the unpredictability of unknown environments [12,13]. To thoroughly explore a given region, the robot must revisit areas containing those missed scenarios, resulting in a waste of resources. Furthermore, when exploring large-scale scenes, the more information the robot collects with exploration, the more the path planner computes, which poses a significant challenge to onboard computers.

In order to solve the above problems, the work [14] proposed that supplying robots with prior information about a given region can aid them in making decisions that align with long-term benefits. Ref. [15] proposed a probabilistic information gain map as the prior knowledge to guide exploration. Ref. [16] introduced a general information theory framework to control multiple robots to search and rescue, wherein the prior knowledge of people is modeled to capture target positions and dynamics. Ref. [17] employs hand-drawn sketches as prior information, enabling the robot to explore even when the metric description of the environment is incomplete.

As the concision of a topological map, many researchers use them as prior information to guide robots in autonomous exploration. Ref. [18] proposed a novel autonomous exploration method based on a prior topometric graph, which verifies that prior information could aid the robot in swiftly completing the exploration of unknown environments. Ref. [19] proposed a path planning method based on topology information for 3D reconstruction, in which the multi-view stereo path planning is decomposed into a collection of overlapped viewing optimization problems that can be processed in parallel. In [14], the prior topometric map is employed to improve exploration efficiency and guide the robot to trigger a loop close, improving the localization accuracy of the Simultaneous Localization And Mapping (SLAM) system. Finally, the environmental information collected will be used to refine prior information.

Furthermore, some researchers focus on the generation of topological maps. Ref. [20] proposed a framework called "topomap" to provide robots with customized maps to simplify robot navigation tasks, transforming the sparse feature-based map from visual SLAM into a three-dimensional topological map. Ref. [21] proposed an efficient and flexible algorithm that generates a trajectory-independent 3D sparse topological skeleton graph captured from the spatial structure of free space.

Inspired by the abovementioned research, we select the topological map as the prior information to guide in robot exploration and employ the frontier-based exploration method suitable for exploring large-scale scenes to plan exploration paths. As a form of map representation, the topological map briefly provides relative position and connectivity between critical places in complex scenes, which could guide the robot in planning paths that follow long-term benefits [20]. In practical cases, many methods can easily acquire the skeleton structure of the environment as the topological map [20–23].

To fully take advantage of the guiding function of prior topological maps, we propose an autonomous exploration method based on topological maps. The proposed method employs a hierarchical path planning framework, integrates frontier information with prior topological maps, and plans the exploration path with long-term benefits. It first plans a global exploration path by solving the constructed Priority Constrained Asymmetric Traveling Salesman Problem (PCATSP). The global exploration path would follow optimal or customized global exploration strategies to guide the robot to cover frontiers but prioritizes exploring scenes outside the prior information, thus preventing the robot from revisiting previously visited areas. Then, the exploration path is refined from the global exploration path by quickly evaluating the rewards and costs of the candidate viewpoint for each frontier.

Because of the one-pass exploration process, our method will maintain the frontier at a small number, preventing excessive computational burden on the solver during exploration. The above properties make our method more suitable for autonomous exploration in large-scale scenes, and the contributions of this paper are as follows:

(1) An autonomous exploration method based on prior topological maps. The robot follows an optimal or customized strategy to explore a given region autonomously but prioritizes exploring scenes outside prior information, preventing the robot from revisiting the explored areas.

(2) A path planning method integrates information between frontiers and prior topological maps, which makes the topological map deeply involved in the path planning of robot exploration.

(3) A local path planning method, which quickly evaluates the rewards and costs of each candidate viewpoint to optimize the global exploration path, enhances the stability of exploration efficiency.

2. Design Objectives

Give the robot a region to autonomously explore, and provide it with a topological map of the region to be explored. The topological map should reflect the fundamental layout of the region but may not represent all of its spaces. The objectives we address are as follows:

Objective 1: The robot completes a comprehensive exploration of the given region. When there are no frontier clusters extractable within the given region, it indicates the completion of an information gathering task.

Objective 2: The robot utilizes real-time collected scene information and prior topological maps to plan exploration paths. When the robot encounters scenes that are not included in the priori information, the exploration path will guide the robot to prioritize exploring scenes beyond a priori information.

Objective 3: The exploration path enables the robot to complete an exploration of the visited area in a one-pass manner, preventing the robot from repeatedly visiting the areas that have been explored.

3. Methods

We define the topological map as follows. $G(S,E)$ consist of the global targets set $S = \{s_1, s_2, \dots, s_n\}$ and the undirected edges set $E = \{e_1, e_2, \dots, e_m\}$. s denotes a global target corresponding to a corner or intersection in the environment. $e_k = (s_i, s_j)$ is an undirected edge, connecting s_i and s_j, representing a straight-line scene, such as a road or corridor. With the support of a prior topological map, we can obtain a global exploration strategy $O = \{\overline{o_0}, \overline{o_1}, \overline{o_2}, \dots, \overline{o_h}\}$ for the given region by customizing or solving the Chinese Postman Problem (CPP) [24], which is a priority queue. $\overline{o_k} = (s_i, s_j)$ denotes a directed line segment from s_i to s_j, corresponding to the exploration guidance. Define f as the frontier cluster, and $F = \{f^1, f^2, \dots, f^n\}$ as a set of remaining frontier clusters in a scene. We adopt the incrementally frontier information structure (FIS) proposed by FUEL to update frontier clusters efficiently [12]. Viewpoint sequential queue $VP^k = \{vp_1^k, vp_2^k, \dots, vp_m^k\}$ of frontier cluster f^k is extracted by random sampling, where vp_1^k is a viewpoint with the largest reward of f^k and will replace f^k in constructing PCATSP.

Figure 1 shows an overview of the proposed method, which operates upon a voxel grid map. We employ a hierarchical architecture to plan the exploration path, which consists of global path planning (Section 3.1) and local path planning (Section 3.4). The global path planning module takes a prior topological map, global exploration strategy, and frontier clusters as input to plan the global exploration path based on the Priority Constrained Asymmetric Traveling Salesman Problem (PCATSP). Nodes with priority in PCATSP will be extracted (Section 3.2), and the movement cost of some frontier clusters will also be updated (Section 3.3). Then, the global exploration path is given to the local path planning module, which refines the input path based on rewards and costs of each viewpoint candidate to improve the stability of exploration efficiency. Finally, the exploration path will output to the trajectory generation module. The exploration task will be completed when no frontier clusters can be extracted from environment.

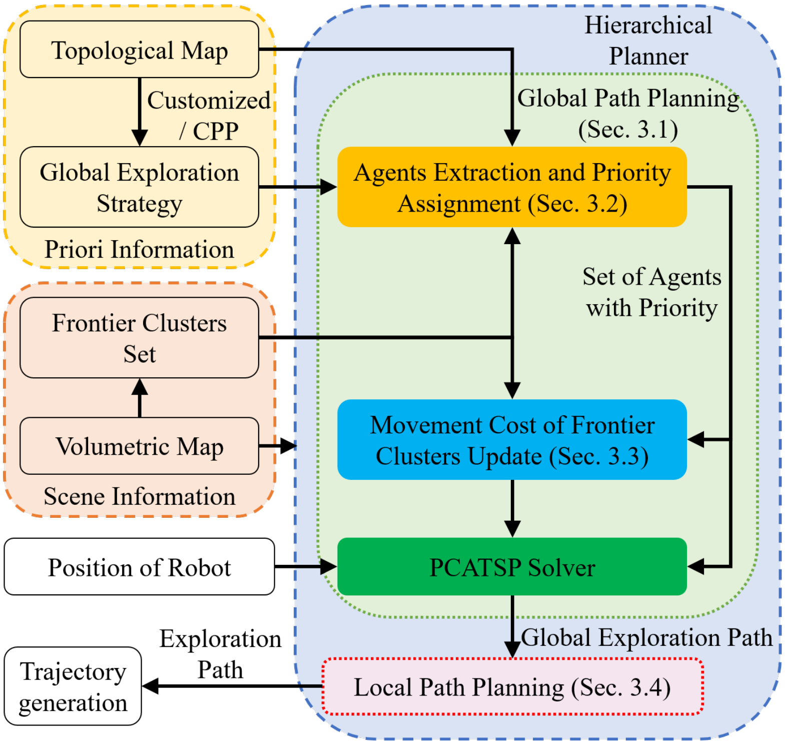

Topological Map
Customized / CPP
Global Exploration Strategy
Priori Information
Frontier Clusters Set
Volumetric Map
Scene Information
Position of Robot
Trajectory generation
Exploration Path
Hierarchical Planner
Global Path Planning (Sec. 3.1)
Agents Extraction and Priority Assignment (Sec. 3.2)
Set of Agents with Priority
Movement Cost of Frontier Clusters Update (Sec. 3.3)
PCATSP Solver
Global Exploration Path
Local Path Planning (Sec. 3.4)

Figure 1. The overview of the proposed exploration method.

3.1. PCATSP-Based Global Path Planning

PCATSP is a variation of the classic Traveling Salesman Problem (TSP), which aims to find a minimum-cost Hamiltonian circuit, with the constraint that some nodes must be visited before others. If splitting the start and end points of PCATSP into two nodes, PCATSP is equivalent to finding a path between the start and end points that satisfy priority constraints, which is also considered a Sequential Ordering Problem (SOP) [25]. To address Objective 3, our basic idea of global path planning is to solve PCATSP with frontiers and priority-constrained global targets. It is equivalent to inserting frontiers into a sequential queue of global targets, utilizing the global targets to influence the covered sequence of some frontier clusters.

However, the construction of PCATSP faces the following challenges: First, constrained by the perception range of sensor and obstacle obstruction, the robot cannot accurately calculate the movement distance between global targets in unknown space and viewpoints inside the free space. Second, for TSP, the farther the metric distance between frontier cluster and global target, the less influence the global target can exert. So, we need to enable global targets to influence specific frontier clusters.

To address the above challenges, we define the following: if a global target s_i is inside the free space, and the other global target s_j connected to s_i is in unknown space, then the shortest path P_{p_k,s_j} between any node p_k in free space and global target s_j in unknown space is given by

$$P_{p_k,s_j} = P_{p_k,a} + P_{a,s_j}, \tag{1}$$

where a is an intersection point of frontier and undirected edge e_k that connects s_i and s_j, as shown in Figure 2a. P_{a,s_j} is a portion of e_k in unknown space, and $P_{p_k,a}$ is a search path from p_k to a.

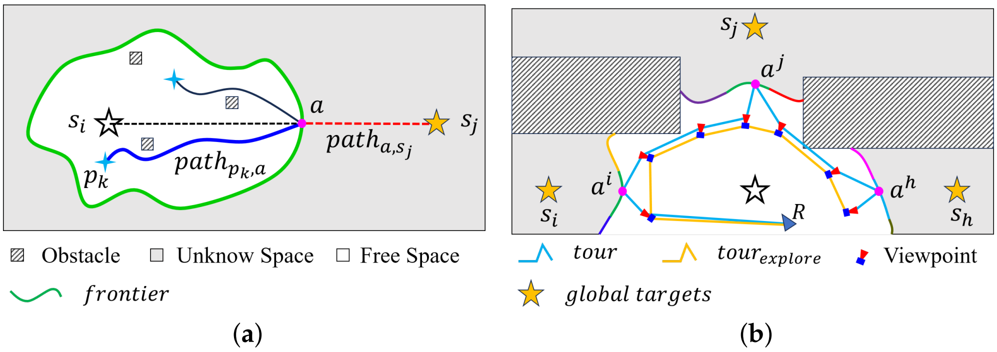


Figure 2. The basic scheme for autonomous exploration based on a prior topological map. (**a**) Extracting an agent for global target s_j that is in unknown space. (**b**) Obtain the exploration path $tour_{explore}$ from $tour$ provided by the PCATSP solver.

Based on the above definition and by incorporating the solving property of TSP, if all elements in a certain row or column of cost matrix D of TSP are subtracted by the same value, a new cost matrix D^* for the TSP will be obtained. However, D and D^* correspond to the same TSP solution [26]. Thus, we can subtract P_{a,s_j} from all P_{p,s_j}, and leave the TSP solution unchanged. It is equivalent to setting an agent point for the global target on the frontier. For constructing PCATSP, the intersection point a^j could be extracted on a frontier cluster as an agent for global target s_j and the priority can be set to a^j based on global exploration strategy O.

Hence, based on the above theory, we define agent a^j as the intersection point of a frontier cluster f^k and an undirected edge $e_k = (s_i, s_j)$, which is associated with a global target s_j in unknown space and possesses access priority. Then, the path between any point inside free space and an agent on frontier could be found by a path-searching algorithm, and then path length could also be accurately calculated. The method of agent extraction and priority assignment will be elaborated in Section 3.2.

Finally, we can naturally combine the prior topological map with real-time updated scene information in path planning based on PCATSP. We can then solve the PCATSP with the extracted agent set $A = \{a^i, a^j, \ldots, a^l\}$ and frontier cluster set F, where agents with priority will influence the visited order of nearby frontier clusters.

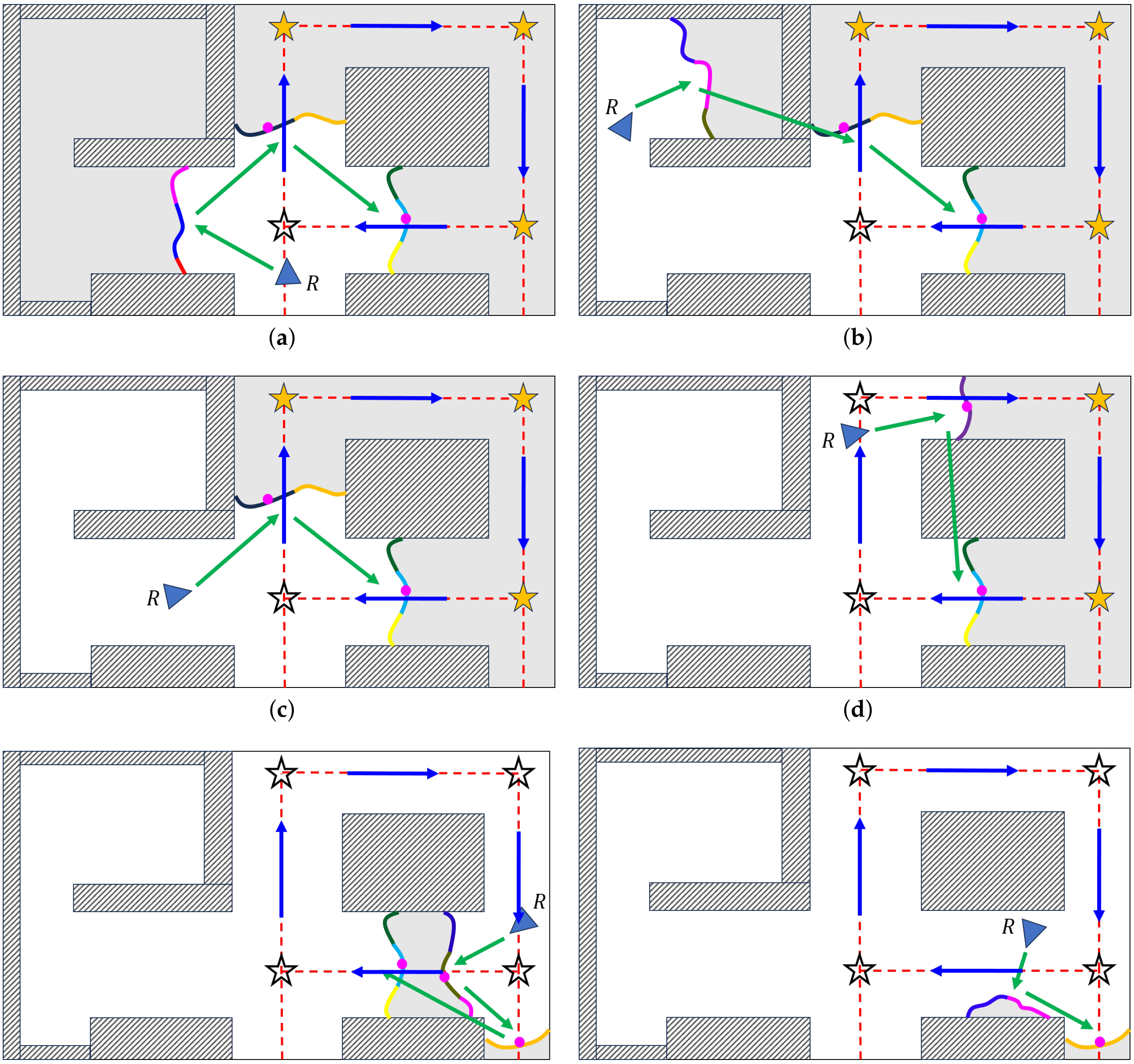
In this section, we suppose that the movement cost $c(p_i, p_j)$ between any two nodes p_i and p_j is calculated as follows:

$$c(p_i, p_j) = \frac{L\left(P_{p_i,p_j}\right)}{v_{\max}}, \tag{2}$$

where P_{p_i,p_j} is the shortest search path between nodes p_i and p_j; $L()$ denotes the length of search path; and v_{max} is the maximum velocity of the robot.

PCATSP solver provides a path $tour$ composed of input nodes. By removing A from $tour$, we obtain a global exploration path $tour_{explore} = \{vp_1^i, vp_1^j, \ldots, vp_1^l\}$ that satisfies global exploration strategy O, as shown in Figure 2b [27]. However, $tour_{explore}$ cannot guarantee that the robot with the priority will explore areas outside prior information, i.e., Objective 2. To address this objective, frontier clusters that guide the robot towards global targets will be recognized, and movement cost between these frontier clusters and the robot will be increased. Details are discussed in Section 3.3.

Figure 3 is a schematic diagram of the robot exploration process. The blue arrows represent the global exploration strategy, and green arrows are basic exert programs for global exploration paths. Based on our method, the robot explores the given region according to the global exploration strategy but prioritizes exploring scenarios outside prior information. Finally, following the global exploration strategy, the robot actively loops close and completes the exploration.



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

(**c**) (**d**)

(**e**) (**f**)

Figure 3. The schematic diagram of the exploration process. (**a**,**b**) Robot prioritizes exploring scenarios outside prior information. (**c**,**d**) After exploring scenarios outside prior information, the robot explores the unknown environment according to a global exploration strategy. (**e**,**f**) Robot actively conducts loop exploration according to the global exploration strategy and finally covers the frontier clusters near the lowest-priority agent.

3.2. Agent Extraction and Priority Assignment

When the position of a global target s_i is explored by the robot, other global targets s_j connected to s_i will be searched on a topological map. Then, the directed line segment $\overline{e_k} = (s_i, s_j)$ is constructed based on $e_k = (s_i, s_j)$, indicating from s_i to s_j. During implementation, we set up some satellite points for each global target to prevent the robot from missing them. The satellite points are evenly distributed at a set distance around the global target point.

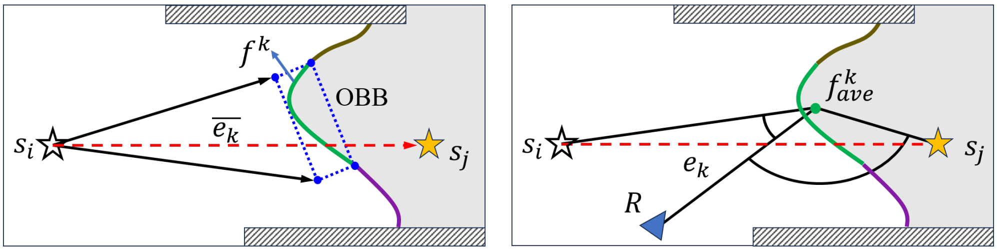
The Oriented Bounding Box (OBB) of each new frontier cluster is extracted by principal component analysis (PCA), which is used to extract agents. As shown in Figure 4a, if f^k is crossed by $\overline{e_k}$, OBB of f^k must be crossed, and the following condition is met:

$$\begin{gathered}\left(\left(s_j - s_i\right) \times \left(obb_p - s_i\right)\right)_z \left(\left(s_j - s_i\right) \times \left(obb_q - s_i\right)\right)_z < 0, \\ p, q \in \{0, 1, 2, 3\},\ p \neq q,\end{gathered} \tag{3}$$

where *obb* are vertices of OBB. For the new frontier clusters that meet the condition, proceed with the following secondary evaluation to identify which global target the forthcoming agent will belong to:

$$\begin{gathered}\pi > \cos^{-1}\left(\left(R - f^k_{ave}\right) \cdot \left(s_j - f^k_{ave}\right)\right) \geq \chi, \\ 0 < \cos^{-1}\left(\left(R - f^k_{ave}\right) \cdot \left(s_i - f^k_{ave}\right)\right) \leq \psi,\end{gathered} \tag{4}$$

where R is the position of the robot, and f^k_{ave} is the average position of f^k, as shown in Figure 4b.



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

Figure 4. Method of agent extraction. (**a**) Crossing determination. (**b**) Angle determination.

If the above conditions are met with the newly extracted frontier cluster f^k, the cell closest to f^k_{ave} in f^k will be defined as an agent a^j for global target s_j. An agent of the global target is independent of the frontier cluster but is associated with it; if a frontier cluster is deleted or updated, the corresponding agent will be deleted.

The movement cost between an agent a^j and other nodes of PCATSP are calculated as follows:

$$\begin{gathered}c\left(R, a^j\right) = C, \\ c\left(a^j, R\right) = 0, \\ c\left(a^j, x\right) = c\left(x, a^j\right) = \frac{L\left(P_{a^j,x}\right)}{v_{\max}},\ x = vp^k_1, a^i,\end{gathered} \tag{5}$$

where C is a large value, ensuring that the robot prioritizes exploring scenes outside prior information.

The priority pry_{a^j} of agent a^j is assigned as follows:

$$pry_{a^j} = \begin{cases} u,\ \overline{e_k} = \overline{o_u} \\ size(O) + 1,\ others \end{cases}, \tag{6}$$

which is determined by the sequential position of its $\overline{e_k}$ in O. $\overline{o_u}$ is a directed line segment in O, where u represents its ordinal position in O. The higher the sequential position of $\overline{e_k}$, the lower the priority of the agent extracted from $\overline{e_k}$. If $\overline{e_k}$ is not found in O, the priority of the agent equals $size(O) + 1$, and these agents with the lowest priority will not be actively explored, as shown in Figure 3. Additionally, during exploration, not only will one agent be generated by a directed line segment, but all agents will point towards the same global target. In such cases, we define the agent that is further away from the global target to have a higher priority, ensuring the robot does not miss the scene during exploration.

3.3. Update of Movement Cost of Frontier Clusters

As mentioned earlier, to ensure that the robot prioritizes exploring scenes outside prior information, the system needs to recognize frontier clusters that would guide the robot towards a global target and set a higher cost between them and the robot. To achieve this, we classify frontier clusters into three categories:

The first class includes frontier clusters with agents and other frontier clusters that are adjacent to these frontier clusters. They will guide the robot towards global targets.

The second class includes frontier clusters adjacent to the first class frontier clusters. These frontier clusters may guide the robot towards global targets. We utilize density-based spatial clustering of applications with noise (DBSCAN) algorithm to recognize these frontier clusters and rely on the following methods to identify whether they could guide the robot to global targets [28]:

$$\begin{gathered} e^{-r_{s_j,f^k}/r_{s_j,s_i}} e^{\cos\alpha^k - 1} \cos\beta^k \geq \varepsilon, \\ r_{s_j,f^k} < r_{s_j,R} \leq r_{s_j,s_i} + \gamma, \end{gathered} \tag{7}$$

as shown in Figure 5a, α^k is the angle between vector $\overrightarrow{f^k_{ave}s_j}$ and $\overrightarrow{s_i s_j}$; β^k is the angle between vector $\overrightarrow{f^k_{ave}s_j}$ and $\overrightarrow{Rf^k_{ave}}$; and r is the Euclidean distance between two points.

However, as the robot explores, the relative position between frontier clusters and robot changes, and frontier clusters cannot always satisfy Equation (7), as shown in Figure 5b,c. Thus, to keep the consistency of determination for these frontier clusters during exploration, we employ the Dynamic Time Warping (DTW) algorithm to evaluate similarities between path P_{R,a^j} and all paths of $P_{R,F} = \{P_{R,f^1}, P_{R,f^2}, \ldots, P_{R,f^n}\}$, as shown in Figure 5d [29]. If the similarity ranking of f^k satisfies the following condition, f^k is still believed to guide the robot towards global target s_j:

$$rank\Big(dtw\Big(P_{R,a^j}, P_{R,f^k}\Big)\Big) \leq \varphi. \tag{8}$$

The third class consists of the remaining frontier clusters. They will guide the robot to explore scenes outside prior information.

For the frontier clusters f^{k^*} that guide the robot towards global targets, the movement cost from the robot to them is set as follows:

$$c\Big(R, vp_1^{k^*}\Big) = C. \tag{9}$$

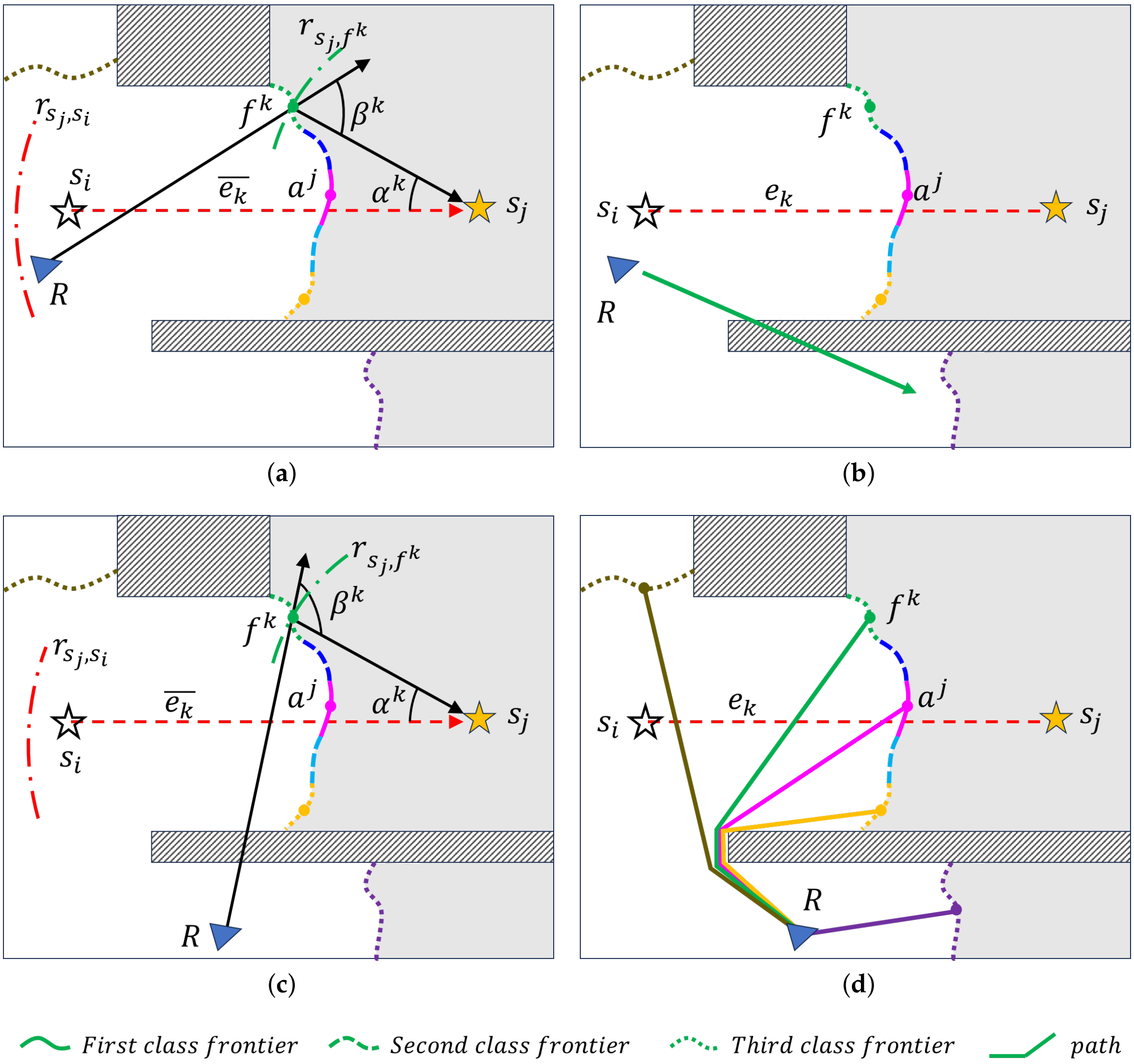
The movement cost from robot to other frontier clusters f^k is computed as follows:

$$c\Big(R, vp_1^k\Big) = \max\left\{\frac{L\Big(P_{R,vp_1^k}\Big)}{v_{\max}}, \frac{\min\big(|\xi_1^k - \xi_R|, 2\pi - |\xi_1^k - \xi_R|\big)}{\omega_{\max}}\right\} + \lambda\cos^{-1}\frac{(vp_1^k - R)\cdot v_R}{\|vp_1^k - R\|\,\|v_R\|}\ , \tag{10}$$

which considers the path length, yaw change, and motion consistency, where p_1^k and ξ_1^k are coordinates, and the yaw angle of viewpoint vp_1^k, ω_{max} is the maximum angular change rate; ξ_R is the yaw angle of the robot; and v_R is the current velocity.

As $c(vp_1^{k^*}, R)$ and $c(vp_1^k, R)$ do not impact the solution results of open-loop path planning, we set them to 0. The movement cost between all frontier clusters are calculated as follows:

$$c\left(vp_1^i, vp_1^j\right) = c\left(vp_1^j, vp_1^i\right) = \max\left\{ \frac{L\left(P_{vp_1^i, vp_1^j}\right)}{v_{\max}}, \frac{\min\left(\left|\xi_1^i - \xi_1^j\right|, 2\pi - \left|\xi_1^i - \xi_1^j\right|\right)}{\omega_{\max}} \right\}. \quad (11)$$



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

Figure 5. Determination of the second category of frontier clusters. (**a**) Determine whether the second-category frontier f^k guides the robot towards the global target based on angle and distance conditions. (**b**) Robot moves towards other scenes following the generated exploration path. (**c**) As the robot moves, f^k no longer meets the initial judgment criteria. (**d**) To maintain the consistency of the previous judgment to f^k, perform a secondary judgment of f^k by DTW.

3.4. Local Path Planning

The global path planning module aims to assist the robot in making decisions at a global standpoint for efficient exploration. The local path planning module aims to find the best viewpoints to make the robot to follow. Many previous works refine a path by evaluating the cost and reward for efficient exploration, but they consume significant computational resources in information evaluation [1,30]. Thus, we define the potential reward of a candidate viewpoint as a volume of unknown space within its a Field of View

(FoV) and propose a simple and fast reward evaluation method based on incremental frontier information structure (FIS). Finally, the local path is refined by synthesizing the reward and movement cost of each viewpoint candidate.

Cells of a frontier cluster are stored in FIS [12]. We use them to evaluate the volume of unknown space in FOV. As shown in Figure 6, each truncated pyramid is constructed based on cells that the candidate viewpoint could cover, and its volume is calculated as follows:

$$V = \left(h_{rw} \left(\frac{h_{rw} e_{cell}}{h_{cell}} \right)^2 - h_{cell} (e_{cell})^2 \right) / 3, \tag{12}$$

where e_{cell} is width of cell, h_{cell} is distance between cell and candidate viewpoint. h_{rw} is effective distance to calculate reward and computed by:

$$h_{rw} = \min(h_{\max}, h_{cell} + \delta), \tag{13}$$

h_{max} is maximum range of FOV, and δ is used to control the depth of truncated pyramid to balance movement cost and expected rewards.

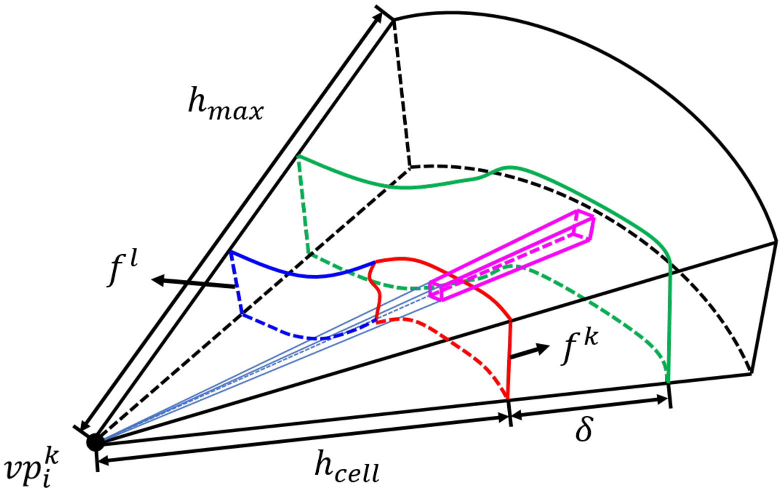


Figure 6. The schematic diagram of calculating the reward of a viewpoint.

The expected reward rw_i^k of a candidate viewpoint vp_i^k is evaluated by accumulating V:

$$rw_i^k = \sum_{x=1}^{n} V_x^k + \eta \sum_{y=1}^{m} V_y^l, \tag{14}$$

where V_x^k is taken from current frontier cluster f^k, and V_y^l is taken from next adjacent frontier cluster f^l to be visited. m and n, respectively, represent the number of cells that vp_i^k could cover. η is a weight coefficient.

We formulate local path planning as a graph search problem and refine an optimal path from the global exploration path by balancing expected reward and movement cost, where viewpoints of each frontier cluster serve as candidate points. Suppose that the optimal exploration path $path = \{vp_i^1, vp_k^2, \ldots, vp_j^{N_{rf}}, vp_1^{N_{rf}+1}\}$ provided by the Dijkstra algorithm will minimize the cost/reward ratio:

$$c\left(P_{R, vp_1^{N_{rf}+1}}\right) = \frac{c\left(R, vp_i^1\right)}{W_i^1} + \frac{c\left(vp_j^{N_{rf}}, vp_1^{N_{rf}+1}\right)}{W_1^{N_{rf}+1}} + \textstyle\sum_{n=1}^{N_{rf}-1} \frac{c\left(vp_k^n, vp_k^{n+1}\right)}{W_k^{n+1}} \;, \tag{15}$$

where $N_{rf} + 1$ is the size of the frontier clusters to be optimized, and

$$W_k^n = \frac{rw_k^n}{V_{FOV}}, \tag{16}$$

V_{FOV} is the volume of FOV.

Finally, the exploration path *path* is output to the trajectory generation module.

4. Experiments

4.1. Implementation Details

We set $\psi = \pi/3$, $\chi = 2\pi/3$, and $C = 500$ in Section 3.2, $\varepsilon = 0.4$, $\gamma = 3$ m, and $\varphi = 6$ in Section 3.3, and $\delta = 1.5$ m and $\eta = 1.25$ in Section 3.4. Additionally, we employ the SOP solver from LKH-3.0 to solve PCATSP and implement the Chinese Postman Problem (CPP) solver ourselves for the global exploration strategy [27,31]. All simulation experiments are conducted in ROS-Kinetic Gazebo platform on Ubuntu 18.04 computer system, running on an CPU. For real-world experiments, the unmanned ground vehicle shown in Figure 7 was utilized to explore a given region. We equipped it with a depth camera, an inertial measurement unit, and an onboard computer with Ubuntu 18.04 computer system.

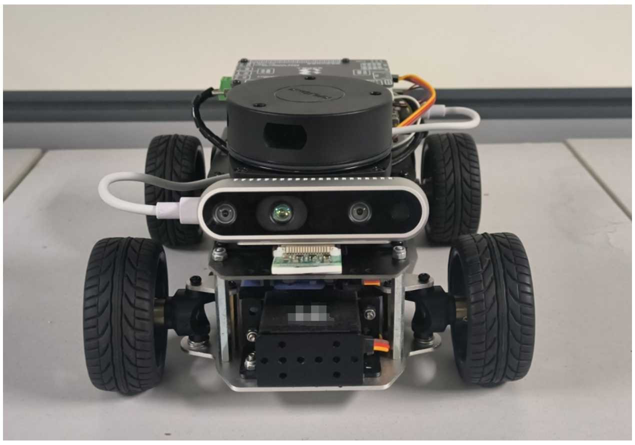


Figure 7. Real-world experiment vehicle platform.

4.2. Benchmark Comparisons

In this section, we conduct benchmark comparisons using simulation experiments to verify the effectiveness and exploration efficiency of the proposed method. Robot exploration in maze scenes is the most effective method to verify the efficiency of autonomous exploration [13]. Thus, we manually constructed two large-scale mazes, Maze-1 ($48 \times 63 \times 2$ m^3) and Maze-2 ($66 \times 62 \times 2$ m^3), in Gazebo simulation platform. The cross-sectional length of the road in Maze-1 is 4∼6 m, and in Maze-2, 8∼10 m. The topological maps of these mazes are generated by manually placing global targets on corners and intersections, and then connecting them according to the topology of mazes to simulate maze information [18]. But, we leave some space for the robot to explore autonomously without prior information. The mazes and their prior topological maps are shown in Figure 8.

We employ the optimal exploration strategy provided by the CPP solver to guide robot exploration and compare it with FUEL and FAEP [12,13]. They are state-of-the-art frontier-based methods which have been proposed in recent years, and which exhibit high exploration efficiency and have open-sourced their code to serve community. In all experiments, we utilize UAV as the exploration robot, with $v_{max} = 0.6$ m/s, $\omega_{max} = 0.9$ rad/s, and the maximum acceleration is 0.6 m/s^2. UAV equips depth camera to collect environmental information. FOV of depth camera is configured as $[80 \times 60]$ deg, h_{max} is 4.5 m. The grid map of the local update range is $5 \times 5 \times 2$ m^3. In all scenarios, three methods are run more than six times with the same configuration.

We evaluate the performance of the above methods based on exploration time, flight distance, and coverage efficiency. FAEP also utilizes FIS to update frontier cluster information and based TSP to plan global exploration paths. Therefore, we set the frontier cluster length limit of all methods to 2 m and counted the remaining number of frontier clusters during exploration. The number could directly reflect the computational burden of the path planner: the more frontier clusters remain in scene, the more enormous the solver computation.

Figure 8 displays exploration trajectories of different methods in two mazes. Our method prevents the robot from unnecessarily revisiting explored areas but ensures that the robot does not miss scenes outside prior information, even in scenes with large cross-sectional road lengths like Maze-2. In contrast, other methods re-explore already visited areas during exploration, leading to lower efficiency and inevitable resource wastage.

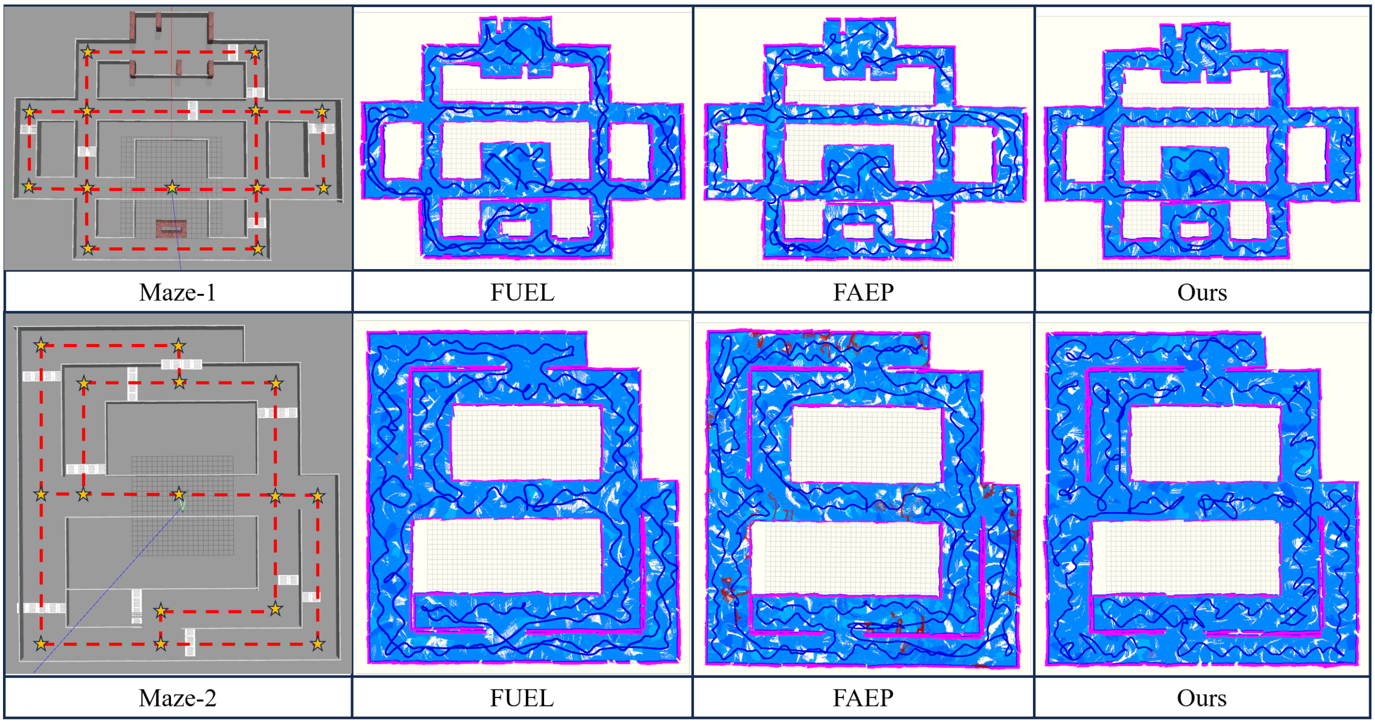


Figure 8. Benchmark comparison of exploration trajectories of the proposed method, FUEL, and FAEP, in two mazes.

The detailed exploration process of our method in Maze-1 is illustrated in Figure 9. The robot prioritizes exploring areas outside prior information in the given region. Subsequently, it follows the global exploration strategy to explore other areas, disregarding the direction of other frontier clusters during exploration. Finally, the robot actively loops close to cover the frontier clusters near the agent with the lowest priority. The exploration process aligns with the concept depicted in Figure 3.

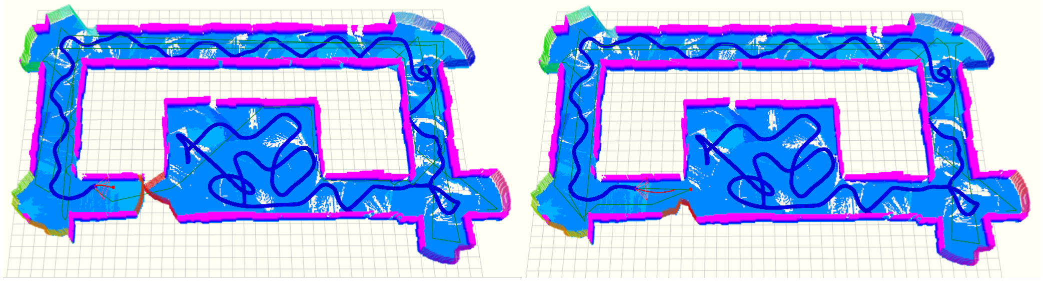


Figure 9. Partial exploration trajectories generated by our method in Maze-1.

Figure 10 displays the process of volume coverage for all methods in two mazes. Our method demonstrates higher efficiency and nearly linear performance in conducting exploration. In contrast, other methods exhibit coverage stagnation or slower growth during exploration, meaning the robot moved towards a previously missed or visited space.

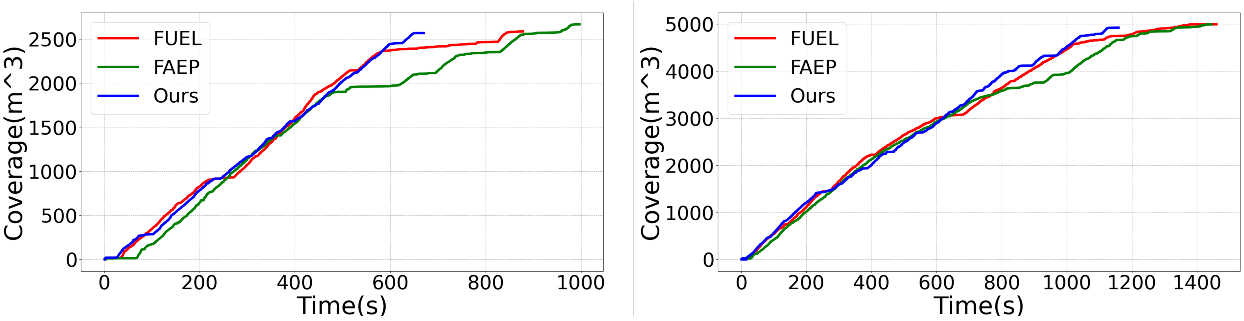


Figure 10. The exploration progress of three methods in Maze-1 (**left**) and Maze-2 (**right**).

The number of the remaining frontier clusters during exploration for all methods is shown in Figure 11. Compared with other methods, our method ensures a lower count of remaining frontier clusters. It proves that our method can avoid much computational burden for an onboard computer and is suitable for exploring large-scale scenes.

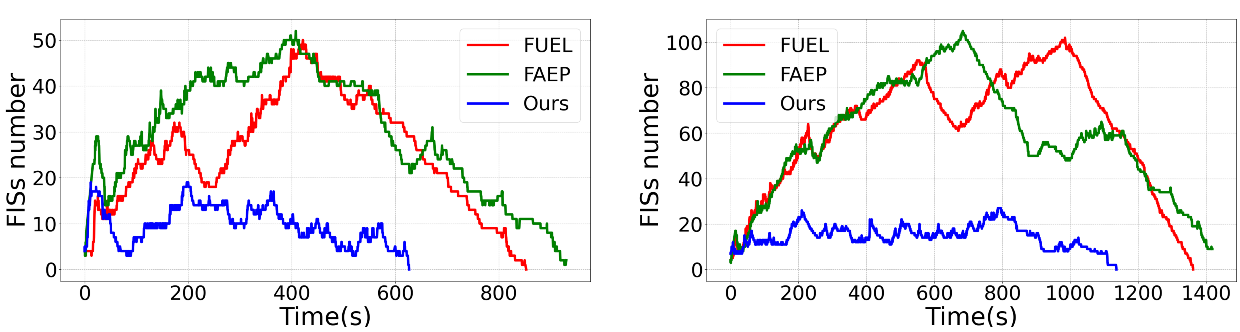


Figure 11. The number of remaining frontier clusters during exploration of the three methods in two mazes.

Table 1 presents a quantitative performance of three methods in two mazes. Compared with FUEL and FAEP, the average exploration time of our method is reduced by 18.76% and 20.87% in Maze-1, and 18.05% and 21.77% in Maze-2. Moreover, the average flight distance of our method is reduced by 18.93% and 17.19% in Maze-1, and 15.91% and 6.79% in Maze-2. In contrast, our proposed method outperforms other methods in exploration time, flight distance, and stability.

The average computation time of each module is shown in Table 2. The proposed method conducts a one-path planning with approximately 130 ms, meeting the frequency requirements for most robots. Furthermore, only 12 ms is consumed by local path planning, additionally proving the efficiency of the proposed reward evaluation method.

We deconstruct the proposed method to analyze the global and local path planning module performances. Ours$^{\text{Local}}$ and Ours$^{\text{Global}}$ correspond to implementing local and global path planning, both built upon FUEL framework. The exploration data of Ours$^{\text{Local}}$, Ours$^{\text{Global}}$ in two mazes are list in Table 1.

Table 1. Exploration statistics in Maze-1 and Maze-2.

Scene	Method	Exploration Time (s)				Flight Distance (m)			
		Avg	Std	Max	Min	Avg	Std	Max	Min
Maze-1	FUEL	826.599	25.275	849.384	777.630	618.773	16.821	635.879	589.081
	FAEP	848.614	32.779	906.516	813.997	605.419	28.813	655.931	561.852
	Ours	671.101	13.880	694.328	663.532	501.523	12.490	520.579	486.109
	Ours$^{\text{Local}}$	826.881	10.485	855.695	814.620	616.217	8.390	635.330	605.435
	Ours$^{\text{Global}}$	670.742	24.009	693.757	626.820	497.563	16.467	517.316	469.384
Maze-2	FUEL	1407.048	46.729	1503.542	1356.909	1012.588	18.818	1036.984	984.151
	FAEP	1474.686	67.125	1569.142	1418.180	913.552	33.502	960.631	885.397
	Ours	1153.677	29.733	1200.470	1104.911	851.646	22.824	885.435	819.141
	Ours$^{\text{Local}}$	1396.996	27.841	1439.908	1339.974	1002.997	18.952	1037.262	980.147
	Ours$^{\text{Global}}$	1210.028	52.735	1277.481	1138.502	886.096	41.552	940.927	831.196

Table 2. Average computation time of each module.

Scene	Average Computation Time (ms)		
	Global Planning	Local Planning	Total Planning
Maze-1	141.79	17.39	159.19
Maze-2	127.45	12.35	130.85

Comparing the simulation results of Ours$^{\text{Global}}$ and FUEL in two mazes, the standard deviations of Ours$^{\text{Global}}$ in exploration time and flight distance are identical to FUEL. However, the average exploration time of Ours$^{\text{Global}}$ is reduced by 18.88% and 14.01%, respectively, in Maze-1 and Maze-2. The average flight distance of Ours$^{\text{Global}}$ is reduced by 19.57% and 12.45%, respectively, in Maze-1 and Maze-2. The average performance of Ours$^{\text{Global}}$ outperforms FUEL, indicating that the primary contribution of global path planning lies in improving exploration efficiency. The same inference can also be drawn from comparing Ours and Ours$^{\text{Local}}$.

The average exploration time and flight distance of Ours$^{\text{Local}}$ and FUEL in both mazes are approximate. But, Ours$^{\text{Local}}$ maintains a low standard deviation, indicating more stability. Similar performance is observed between Ours$^{\text{Global}}$ and Ours. It is demonstrated that the proposed local path planning method can effectively enhance the stability of exploration efficiency.

Finally, we can conclude that the exploration efficiency is primarily attributed to global path planning, while local path planning enhances the stability of exploration efficiency.

4.3. Real-World Experiment

In order to further validate the effectiveness of our proposed method, we conduct a real-world experiment with a ground vehicle. Based on the prior topological map, the ground vehicle will explore an indoor corridor of size $55 \times 15 \times 2$ m^3, and a cross-sectional length of road that is 2 m, as shown in Figure 12. The prior information we provided is a rectangular topological map that outlines the basic structure of the indoor corridor. In the indoor corridor, the open spaces and nooks served as regions beyond prior information, testing the effectiveness of our method.

In the experiment, we run VINS-Fusion on GPU to provide the positional state, while the proposed method runs on CPU to plan an exploration path [32]. We set $v_{max} = 0.5$ m/s, $\omega_{max} = 0.8$ rad/s, the maximum acceleration as 0.5 m/s^2, and the grid map of local update range as $4 \times 4 \times 2$ m^3. The FOV of the depth camera is set as $[80 \times 60]$ deg, and the maximum range h_{max} is 3.5 m.

Figure 13 shows the exploration trajectory of our method with the indoor environment, where the small labeled images represent the key nodes during exploration: (a) presents the robot exploring in the direction following a global exploration strategy; (b) and (c) present the instances where the vehicle prioritizes exploration directions beyond the prior information. The exploration time of the whole process is 738 s, and the movement distance is 167 m. It can be seen from the trajectory that the vehicle did not revisit the explored areas during exploration, which proves the effectiveness of our proposed method.

Figure 14, respectively, shows the process of volume coverage and the number of remaining frontier clusters during exploration. It reveals a stable and linear exploration process while the number of frontier clusters is maintained at a small level, demonstrating the efficiency of our proposed method.

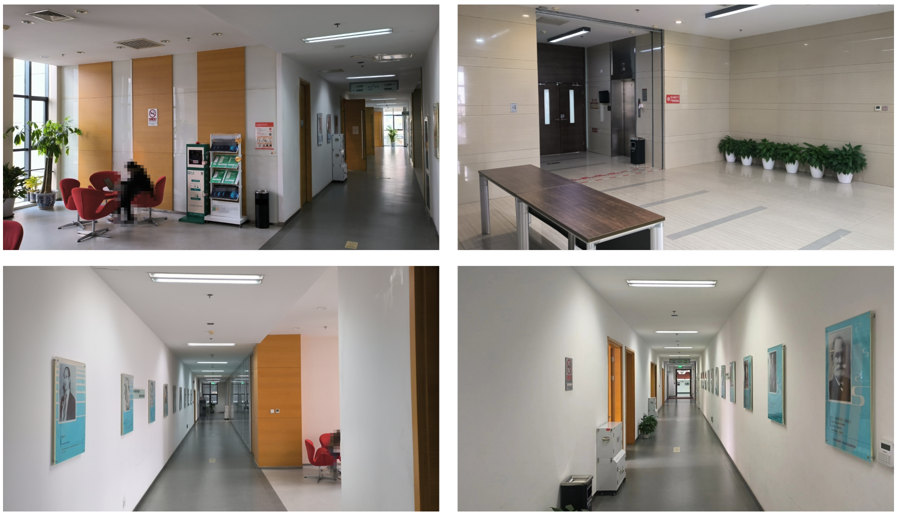


Figure 12. The experimental scene for real-world experiment.

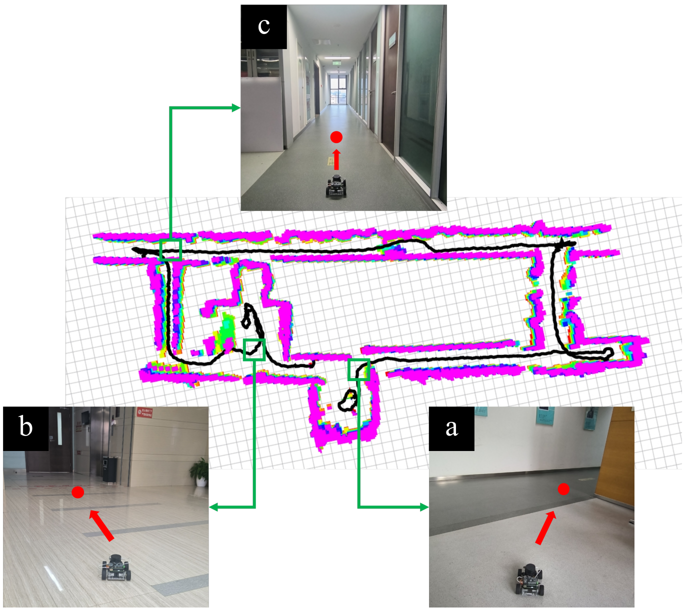


Figure 13. The exploration trajectory of the proposed method in real-world experiment. (**a**) shows the robot exploring in the direction following a global exploration strategy; (**b**,**c**) show the instances where the vehicle prioritizes exploration directions beyond the prior information.

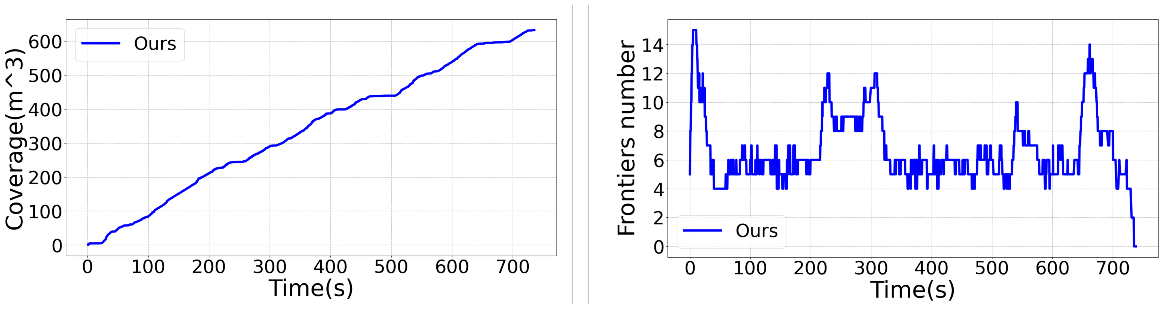


Figure 14. The process of volume coverage (**left**) and the number of remaining frontier clusters (**right**) during exploration in real-world scenario.

5. Conclusions

This paper introduces a novel autonomous exploration method based on a prior topological map, in which a robot explores the given large-scale region, following a global exploration strategy, but prioritizes exploring scenes outside prior information. The proposed method employs a hierarchical framework to plan exploration paths. Based on PCATSP, the global path planning module merges the prior topological map with real-time scene information, planning efficient global exploration paths. Subsequently, the local path planning module rapidly evaluates the rewards and movement costs for each candidate viewpoint to refine the input global exploration paths. Finally, the output exploration path is used to generate local trajectories. Simulation results prove that the proposed method enables the robot to efficiently and rapidly explore a given region and is suitable for operation in large-scale scenes. The ablation study also demonstrates that our proposed local path planning method could enhance the stability of exploration efficiency. The experiment conducted in the real world further validates the effectiveness of our method.

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

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Bircher, A.; Kamel, M.; Alexis, K.; Oleynikova, H.; Siegwart, R. Receding horizon "next-best-view" planner for 3d exploration. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1462–1468.
2. Duberg, D.; Jensfelt, P. Ufoexplorer: Fast and scalable sampling-based exploration with a graph-based planning structure. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2487–2494. [CrossRef]
3. Selin, M.; Tiger, M.; Duberg, D.; Heintz, F.; Jensfelt, P. Efficient autonomous exploration planning of large-scale 3-d environments. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1699–1706. [CrossRef]
4. Xu, Z.; Deng, D.; Shimada, K. Autonomous UAV exploration of dynamic environments via incremental sampling and probabilistic roadmap. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2729–2736. [CrossRef]
5. Respall, V.M.; Devitt, D.; Fedorenko, R.; Klimchik, A. Fast sampling-based next-best-view exploration algorithm for a MAV. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 89–95.
6. Zhu, H.; Cao, C.; Xia, Y.; Scherer, S.; Zhang, J.; Wang, W. DSVP: Dual-stage viewpoint planner for rapid exploration by dynamic expansion. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September– 1 October 2021; pp. 7623–7630.
7. Yamauchi, B. A frontier-based approach for autonomous exploration. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', Monterey, CA, USA, 10–11 July 1997; pp. 146–151.

8. Dornhege, C.; Kleiner, A. A frontier-void-based approach for autonomous exploration in 3d. *Adv. Robot.* **2013**, *27*, 459–468. [CrossRef]
9. Zhong, P.; Chen, B.; Lu, S.; Meng, X.; Liang, Y. Information-driven fast marching autonomous exploration with aerial robots. *IEEE Robot. Autom. Lett.* **2021**, *7*, 810–817. [CrossRef]
10. Deng, D.; Duan, R.; Liu, J.; Sheng, K.; Shimada, K. Robotic exploration of unknown 2d environment using a frontier-based automatic-differentiable information gain measure. In Proceedings of the 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Virtual, 6–10 July 2020; pp. 1497–1503.
11. Cieslewski, T.; Kaufmann, E.; Scaramuzza, D. Rapid exploration with multi-rotors: A frontier selection method for high speed flight. In Proceedings of the RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2135–2142.
12. 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]
13. Zhao, Y.; Yan, L.; Xie, H.; Dai, J.; Wei, P. Autonomous Exploration Method for Fast Unknown Environment Mapping by Using UAV Equipped with Limited FOV Sensor. *IEEE Trans. Ind. Electron.* **2023**, *5*, 4933–4943. [CrossRef]
14. Soragna, A.; Baldini, M.; Joho, D.; Kümmerle, R.; Grisetti, G. Active SLAM using connectivity graphs as priors. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 340–346.
15. Zhang, Y.; McCalmon, J.; Peake, A.; Alqahtani, S.; Pauca, P. A Symbolic-AI Approach for UAV Exploration Tasks. In Proceedings of the 2021 7th International Conference on Automation, Robotics and Applications (ICARA), Prague, Czech Republic, 4–6 February 2021; pp. 101–105.
16. Krzysiak, R.; Butail, S. Information-based control of robots in search-and-rescue missions with human prior knowledge. *IEEE Trans.-Hum.-Mach. Syst.* **2021**, *52*, 52–63. [CrossRef]
17. Boniardi, F.; Valada, A.; Burgard, W.; Tipaldi, G.D. Autonomous indoor robot navigation using a sketch interface for drawing maps and routes. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2896–2901.
18. Oßwald, S.; Bennewitz, M.; Burgard, W.; Stachniss, C. Speeding-up robot exploration by exploiting background information. *IEEE Robot. Autom. Lett.* **2016**, *1*, 716–723.
19. Shang, Z.; Shen, Z. Topology-based UAV path planning for multi-view stereo 3D reconstruction of complex structures. *Complex Intell. Syst.* **2023**, *9*, 909–926. [CrossRef]
20. Blochliger, F.; Fehr, M.; Dymczyk, M.; Schneider, T.; Siegwart, R. Topomap: Topological mapping and navigation based on visual slam maps. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3818–3825.
21. Chen, X.; Zhou, B.; Lin, J.; Zhang, Y.; Zhang, F.; Shen, S. Fast 3D Sparse Topological Skeleton Graph Generation for Mobile Robot Global Planning. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 10283–10289.
22. Mei, J.; Li, R.J.; Gao, W.; Cheng, M.M. CoANet: Connectivity attention network for road extraction from satellite imagery. *IEEE Trans. Image Process.* **2021**, *30*, 8540–8552. [CrossRef]
23. Tan, Y.Q.; Gao, S.H.; Li, X.Y.; Cheng, M.M.; Ren, B. Vecroad: Point-based iterative graph exploration for road graphs extraction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8910–8918.
24. Sokmen, O.C.; Emec, S.; Yilmaz, M.; Akkaya, G. An overview of Chinese postman problem. In Proceedings of the 3rd International Conference on Advanced Engineering Technologies, Bayburt, Turkey, 19–21 September 2019; Volume 10, pp. 1175–1184.
25. Gouveia, L.; Pesneau, P. On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Netw. Int. J.* **2006**, *48*, 77–89. [CrossRef]
26. Morrison, D.R.; Jacobson, S.H.; Sauppe, J.J.; Sewell, E.C. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discret. Optim.* **2016**, *19*, 79–102. [CrossRef]
27. Helsgaun, K. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Roskilde Rosk. Univ.* **2017**, *12*, 966–980.
28. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Kdd, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
29. Salvador, S.; Chan, P. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.* **2007**, *11*, 561–580. [CrossRef]
30. Tao, Y.; Wu, Y.; Li, B.; Cladera, F.; Zhou, A.; Thakur, D.; Kumar, V. SEER: Safe efficient exploration for aerial robots using learning to predict information gain. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 1235–1241.

31. Helsgaun, K. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *Eur. J. Oper. Res.* **2000**, *126*, 106–130. [CrossRef]
32. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [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.