

GVD-TG: Topological Graph based on Fast Hierarchical GVD Sampling for Robot Exploration

Yanbin Li, Canran Xiao, Shenghai Yuan, Peilai Yu, Ziruo Li, Zhiguo Zhang, Wenzheng Chi* and Wei Zhang*

Abstract—Topological maps are more suitable than metric maps for robotic exploration tasks. However, real-time updating of accurate and detail-rich environmental topological maps remains a challenge. This paper presents a topological map updating method based on the Generalized Voronoi Diagram (GVD). First, the newly observed areas are denoised to avoid low-efficiency GVD nodes misleading the topological structure. Subsequently, a multi-granularity hierarchical GVD generation method is designed to control the sampling granularity at both global and local levels. This not only ensures the accuracy of the topological structure but also enhances the ability to capture detail features, reduces the probability of path backtracking, and ensures no overlap between GVDs through the maintenance of a coverage map, thereby improving GVD utilization efficiency. Second, a node clustering method with connectivity constraints and a connectivity method based on a switching mechanism are designed to avoid the generation of unreachable nodes and erroneous nodes caused by obstacle attraction. A special cache structure is used to store all connectivity information, thereby improving exploration efficiency. Finally, to address the issue of frontiers misjudgment caused by obstacles within the scope of GVD units, a frontiers extraction method based on morphological dilation is designed to effectively ensure the reachability of frontiers. On this basis, a lightweight cost function is used to assess and switch to the next viewpoint in real time. This allows the robot to quickly adjust its strategy when signs of path backtracking appear, thereby escaping the predicament and increasing exploration flexibility. And the performance of system for exploration task is verified through comparative tests with SOTA methods. The source code will be available upon acceptance at:

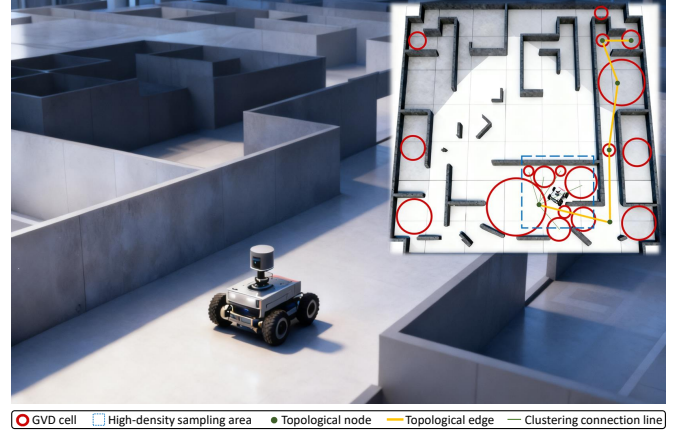


Fig. 1: To balance the detail of grid maps and the efficiency of topological maps, we propose an exploration framework based on dynamic hierarchical GVD, which avoids path backtracking and improves exploration efficiency by maintaining accurate topological details in real time.

https://github.com/littleBurgerrr/Hierarchical_GVD_Exploration.git.

Index Terms—Mobility and Navigation, Autonomous Navigation, Motion Planning, Path Planning.

I. INTRODUCTION

Autonomous exploration is essential for mobile robots to navigate unknown environments and has broad applications, including post-disaster rescue [1]–[6], agricultural monitoring [7], and warehousing logistics. In disaster zones, robots must rapidly map the environment with limited communication and no prior maps to aid rescue efforts. In dynamic warehouses, robots need to continuously update maps to adapt to changing layouts and ensure uninterrupted operations. These scenarios demand exploration strategies that are both highly efficient and capable of providing comprehensive environmental coverage within strict time and energy constraints.

Existing mainstream exploration frameworks generally use occupancy grids as the representation of the environment, which depict each cell in terms of the probability of being occupied by obstacles and thus can finely preserve the geometric details of the environment. However, in large-scale or long-term tasks, the number of grids grows linearly, causing

This project was supported by National Natural Science Foundation of China grant #62273246 awarded to Wenzheng Chi and by the Fund of State Key Laboratory of IPOC (BUPT) under Grant IPOC2025ZT07.

Yanbin Li, Wei Zhang and Zhiguo Zhang are with School of Electronics Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China (yanbinli@bupt.edu.cn; weizhang13@bupt.edu.cn; zhangzhiguo@bupt.edu.cn). Canran Xiao is with the School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China (xiaocanran999@gmail.com). Shenghai Yuan is with the School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798 (shyuan@ntu.edu.sg). Peilai Yu is with the Institute for Computer Science, Ludwig Maximilian University of Munich, Munich, Germany (peilai.yu@campus.lmu.de). Ziruo Li is with Key Lab of Smart Agricultural Systems, China Agricultural University, Beijing 100083, China (liziruo123@cau.edu.cn). Wenzheng Chi is with Robotics and Microsystems Center, School of Mechanical and Electric Engineering, Soochow University, Suzhou, 215021, Jiangsu, China (wzchi@suda.edu.cn).

* corresponding author

the global optimization to be solved in an extremely high-dimensional state space, with the computational complexity increasing exponentially. As a result, the system is forced to degrade to using local greedy strategies, which leads to the problem of path backtracking for robots. Meanwhile, the huge storage and search costs significantly increase the time consumption of replanning. Local obstacles that trigger replanning further exacerbate the risk of backtracking. In contrast, topological maps abstract the environment with a node-edge structure, which naturally fits the hierarchical architecture. That is, the global planner first quickly screens feasible regions at the topological level, and then the local planner refines the continuous trajectory. It is easier to identify and skip narrow branches in advance, thereby effectively suppressing backtracking. In addition, since the number of topological nodes is much smaller than that of grid cells, topological maps have higher real-time performance in large-scale or continuous exploration tasks and can quickly get out of trouble through lightweight node reconnection and edge weight updates.

The key challenge is how to maintain an accurate and detail-rich topological map in the long term. Despite the significant advantages of topological maps in suppressing backtracking and improving global efficiency, there are still problems in directly using them for autonomous exploration. Since topological nodes only retain the connectivity skeleton and lack environmental details, robots are prone to missing narrow passages or small objects, causing perceptual blind spots and thus limiting coverage. Secondly, the reliability of topological construction and maintenance is insufficient. A single misdetection may produce a broken edge, and existing methods lack an efficient error correction mechanism, causing the topological map to drift and even fail over time.

To address this issue, this paper proposes a real-time topological update and exploration framework based on the Generalized Voronoi Diagram (GVD). First, a hierarchical, non-overlapping adaptive GVD generation strategy is proposed to accurately extract environmental feature. By dividing the denoised incremental map into global and local regions and applying different sampling frequencies to them, the accuracy of topological nodes is ensured while enriching the capture of environmental details in the local region around the robot and suppressing the generation of redundant nodes to improve efficiency. Second, a mean shift clustering method with connectivity constraints is designed and accelerated by KD-tree to ensure that the generated topological nodes are always located in free space and avoid cross-obstacle attraction. The connectivity of topological nodes is ensured by a dual-layer switching connection mechanism, ensuring the completeness and efficiency of regional connectivity. Finally, by integrating the topological length, turning cost, and end deviation into a lightweight cost function, the real-time optimal selection of boundary viewpoints is realized. In addition, robots can dynamically reselect the next target according to the latest topological update during the movement, laying the foundation for timely escape and continuous efficient exploration. The primary contributions are:

- 1) Efficient environmental modeling integrating incremental noise reduction and hierarchical GVD sampling: An inno-

vative environmental feature extraction process is proposed. Firstly, local trusted neighborhood voting denoising is designed for the incremental map to eliminate perceived noise; Furthermore, a hierarchical coverage-aware GVD sampling strategy is adopted to achieve adaptive sampling of local refinement and global sparsity. This design ensures the accuracy of the topological structure and the efficiency of the calculation at the source.

- 2) Topology diagram construction method with connectivity constraints: A Mean Shift clustering algorithm combined with connectivity constraints and a double-layer switching connection mechanism are designed to ensure that all topological nodes are located in the reachability space, and the GVDs or bidirectional A* algorithm can be adaptively selected to guarantee the global connectivity and reliability of the topological diagram.
- 3) Morphological frontier detection and real-time viewpoint switching: Innovatively utilize morphological operations to quantify the connectivity between unknown regions and free space, effectively filtering out pseudo-boundary points. It is supplemented by a lightweight cost function, enabling the robot to dynamically select the optimal viewpoint based on the real-time updated topological diagram, significantly reducing ineffective exploration and path backtracking.

II. RELATED WORK

A. Metric Map-based Exploration

Liu *et al.* [8] biased sampling within the free voxels of the occupancy grid via heuristically weighted random points to accelerate indoor robot exploration. Batinović *et al.* [9] identified frontiers in a multi-resolution 3-D occupancy octree, refining voxel clusters progressively to balance accuracy and computation for aerial exploration. Zhou *et al.* [10] presented FUEL, it maintained an incremental frontier information structure (FIS) using a voxel grid map and employed hierarchical planning to achieve efficient exploration. Zhang *et al.* [11] clustered frontier cells from a 3-D occupancy grid into macro-frontiers and selected the next target via an information-gain-over-cost metric to accelerate UAV mapping. Zhao *et al.* [12] presented an autonomous exploration method for UAVs with limited FOV sensors. They generated and evaluated spiral trajectories in a rolling 2-D occupancy grid based on frontier visibility and path cost, selecting the best trajectory to enhance exploration efficiency. The aforementioned exploration methods based on gridded or voxelized metric maps face substantial burdens in terms of storage, computation, and memory in high-dimensional or large-scale environments, with the magnitude of these issues increasing exponentially as resolution improves. To maintain real-time performance, algorithms often employ coarse resolutions, which in turn increase localization errors and collision risks. Moreover, the high computational complexity, low update frequency, and lack of global structural information in metric maps lead to low path planning efficiency and a propensity for path backtracking, particularly in complex environments.

B. Heuristic Methods based on Topological Information

Some methods leverage the topological information of the environment to heuristically guide the robot's path planning, enhancing exploration efficiency by extracting and utilizing the structured information of the environment. Chen *et al.* [13] extracted a topological skeleton from a 2-D occupancy grid and used it as a homotopy-heuristic guide within an RRT* variant to focus sampling only where refinement is necessary. Liu *et al.* [8] biased sampling within the free voxels of the occupancy grid via heuristically weighted random points to accelerate indoor robot exploration. Cao *et al.* [14] presented a hierarchical exploration framework that employs a topological approach by partitioning the 3D environment into subspaces for global planning and uses local RRT* within selected subspaces to efficiently explore complex 3D environments. Soroya *et al.* [15] proposed an active SLAM method that constructs a connectivity graph from a probabilistic occupancy map. The graph captures the topological structure of the environment, and its edge weights guide the robot to maximize loop-closure probability during exploration. Yang *et al.* [16] proposed a method for multi-agent exploration using topological graphs derived from occupancy grids. Graph neural networks are trained to predict next-best nodes for agents, refining the joint map during decentralized exploration. Zhang *et al.* [17] developed a multi-robot exploration technique that builds local topological maps within each robot's occupancy grid and merges them into a global topology during intermittent communication, facilitating exploration under bandwidth limitations. Huang *et al.* [18] presented a fast large-scale ground-robot exploration method. It detects frontiers in a 2-D occupancy grid, clusters them into a topological graph, and solves a TSP-like optimization to select the next best viewpoint. However, these heuristic methods based on topological information heavily rely on the accuracy of the heuristic module. Moreover, these methods often require prior knowledge to extract and utilize topological information, such as predefined environmental features like rooms and corridors. These features are difficult to obtain in completely unknown environments, thereby limiting their applicability. Meanwhile, they typically assume that the environment has structural characteristics, which restricts their application in unstructured environments.

C. Topological Map-based Exploration

Topological maps address the aforementioned issues by extracting the key structures of the environment and retaining only the essential nodes and connections. This approach significantly reduces the consumption of storage and computational resources and enables the rapid generation of navigation paths, making it suitable for real-time applications. Moreover, topological maps provide global layout information of the environment, assisting robots in understanding environmental connectivity and achieving efficient long-range planning, thereby avoiding path backtracking problems. Zhao *et al.* [19] extracted a skeleton graph from a global occupancy grid and applied graph partitioning plus shortest-path queries to build a consistent topological map for multi-robot coordination.

Saroya *et al.* [20] developed a topology-informed Growing Neural Gas algorithm to generate sparse navigation roadmaps in uncertain environments. The method uses persistent homology to identify key topological features, guiding the roadmap generation to focus on challenging regions. Wang *et al.* [21] built an incremental roadmap directly inside a 2-D occupancy grid, inserting nodes and edges as new free space was discovered and selecting the next best view via graph search to guide autonomous exploration. However, topological maps primarily focus on the connectivity of the environment and key nodes, and they do not provide precise descriptions of the specific geometric shapes and the exact locations of obstacles. This limitation makes it difficult to explore the details of the environment, thereby reducing the completeness of the exploration task. Moreover, when generating coverage paths, methods based on topological maps do not sufficiently consider the safe distance between the path and obstacles, which may lead to collision risks in practical applications.

D. GVD

Relying solely on the limited information provided by topological maps, robots struggle to accurately avoid obstacles and determine their next moves. In contrast, the Generalized Voronoi Diagram (GVD) enriches the environmental information available to robots by extracting key geometric structures from the environment, thereby facilitating a better understanding of the layout and aiding in decision-making. The inherent properties of the GVD ensure a safe distance between the robot and obstacles, as the medial axis generated by the GVD naturally provides a collision-avoidance path for the robot. Chi *et al.* [22] reused a GVD-based feature tree extracted from the occupancy grid to provide long-range homotopy classes, guiding an RRT* variant to rapidly escape trapped spaces while retaining completeness. Chi *et al.* [23] built a lightweight GVD skeleton on the 2-D occupancy map and used the resulting feature matrix to bias RRT* sampling toward wide passages, cutting planning time in cluttered mobile-robot scenarios. Vachhani *et al.* [24] computed the GVD directly from range data via a hardware-efficient control-law implementation, producing a sparse medial-axis roadmap for real-time mobile robot navigation without storing the full metric grid. Chen *et al.* [25] extracted a fast GVD from the occupancy grid and fused its medial-axis segments with frontier clusters to create a multi-strategy target assignment, enabling efficient autonomous exploration while leveraging the GVD's natural connectivity prior. The existing exploration methods based on GVD mainly have several key drawbacks: Firstly, they adopt a uniform sampling strategy, which makes it difficult to balance global efficiency and local details, resulting in omissions at key structures such as narrow channels. Secondly, the lack of an effective redundant control mechanism will result in a large number of repetitive and ineffective GVD nodes in the explored areas, wasting computing resources. Thirdly, the generated topological nodes are vulnerable to interference from obstacles, which may occur in inaccessible areas or cause incorrect cross-obstacle connections, seriously affecting the reliability of the map. Fourth, the determination

of the frontier is rough, often misjudging unknown areas that are inaccessible due to obstacles as targets, resulting in low exploration efficiency.

In response to these deficiencies, this paper proposes a systematic improvement plan: by means of hierarchical adaptive sampling and overlay map maintenance, the balance problem between sampling density and redundancy is solved; The clustering method using connectivity constraints ensures the reachability and accuracy of topological nodes. A frontier detection mechanism based on morphology was designed to effectively screen out truly reachable exploration targets, thereby achieving efficient and reliable autonomous exploration.

III. PRELIMINARY

This section aims to provide a theoretical basis for the geometric safety and topological completeness of the exploration method based on GVD.

In the continuous decision-making process of autonomous exploration, the safety of path planning depends on whether the maximum passable margin can be maintained continuously.

GVD is an ideal structure that balances topological integrity and collision risk control. As shown in (1), the GVD sampling point set V is defined as the set of all points that maintain equal distance to at least two obstacles. Among them, o_i and o_j are two different obstacle coordinates in the obstacle set O , and v is the sampling point coordinate of a certain GVD cell, d denotes the minimum Euclidean distance.

$$V = \{v \in Z^2 \mid \exists o_i, o_j \in O, i \neq j, d(v, o_i) = d(v, o_j)\} \quad (1)$$

This ensures that every GVD sampling points lies on the medial axis of at least two obstacles, thereby naturally avoiding proximity to any single obstacle boundary.

To simplify the calculation process, we define the shape of GVD cell as a circle with v as the center and r as the radius. Furthermore, the maximum inscribed circle radius $r(v)$ corresponding to each GVD sampling point v is given by (2), which quantifies the maximum safe area available at that point. Since $r(v)$ is explicitly maximized, when a robot traverses along the V , its body contour remains entirely within a free disk of radius $r(v)$, guaranteeing a minimum safety distance of at least $r(v)$ in all directions. This property not only keeps the path away from obstacles but also provides a quantifiable safety buffer to accommodate sensor noise, localization errors, and dynamic obstacles. It can thus be concluded that, from both geometric and topological perspectives, the GVD demonstrates clear applicability and superiority in robotic exploration tasks. It offers the robot quantifiable and extensible safe passages.

$$r(v) = \max \{r \mid B(v, r) \cap o = \emptyset, \forall o \in O\} \quad (2)$$

Here, $B(v, r)$ is a circle with v as the center and r as the radius.

IV. METHODOLOGY

In this section, we will introduce the proposed methods in detail. Fig. 2 illustrates the pipeline of our system. After receiving sensor data, the robot build environmental map and

obtain position using SLAM (Gmapping [26]). At the next time frame, the system computes the map variation between consecutive frames to generate incremental map M_{inc} , which is then denoised using a trusted neighborhood voting based approach to eliminate noise-induced erroneous GVD sampling. And then, a hierarchical GVD sampling strategy is applied to adaptively regulate sampling density, while the GVD coverage matrix M_c is utilized to prevent repeated sampling in overlapping areas, thereby yielding the GVD. Subsequently, all GVD nodes are clustered via a improved mean shift clustering algorithm based on connectivity constraints to derive topological nodes of the environment. Both GVD nodes and topological nodes are updated frame by frame. After that, a switching mechanism-based topological connection method is employed to generate a complete topological graph. Following topological graph construction, frontier are extracted from all GVD and topological nodes using a morphology-aware detection approach. These frontier are then evaluated by a specially designed cost function to select the next viewpoints.

A. Environmental Feature Extraction Module based on GVD

1) *Local Trusted Neighborhood-based Denoising*: Map noise arises from the combined effects of LiDAR frequency and environmental structures. As shown in cases 1 and 2 of Fig. 3, GVD nodes erroneously generated by such noise typically exhibit small coverage areas and information scarcity, which can degrade both the real-time performance of exploration and the accuracy of the topological structure. Therefore, it is necessary to rapidly eliminate these sparse noise points.

To minimize computational time, denoising is applied only to the incremental portion of the map. The specific procedure is outlined in Algorithm 1. The difference between the map of the previous time frame M_{old} and the current time frame M_{new} is computed to obtain an incremental map M_{inc} that contains only newly observed data. Subsequently, the minimum bounding rectangle (x, y, w, h) of the newly added region is calculated, and a cropped local map M_{crop} is extracted accordingly. For each incremental grid cell $p \in M_{crop}$ that remains marked as unknown, the number n of grids observed as free within the 5×5 square neighborhood $N(p)$ centered at p is counted. If $n > 3$, the state of p is updated to free according to a trusted neighborhood voting strategy; otherwise, its original occupancy state is retained. The final output is a denoised map $M_{denoised}$ that applies noise reduction only to the incremental region.

As illustrated in Fig. 3, this approach fully preserves the structure of the prior map while avoiding redundant computation and error propagation caused by noise, enabling the exploration task to extract environmental structures more accurately with fewer GVD nodes.

2) *hierarchical GVD Sampling with Coverage-Aware Prioritization*: The sampling density of the GVD directly determines both the accuracy of the topological graph and the computational efficiency of the entire system. As shown in Fig. 4(a) and Fig. 4(b), the sampling density critically influences how well the GVD captures key structural features such as

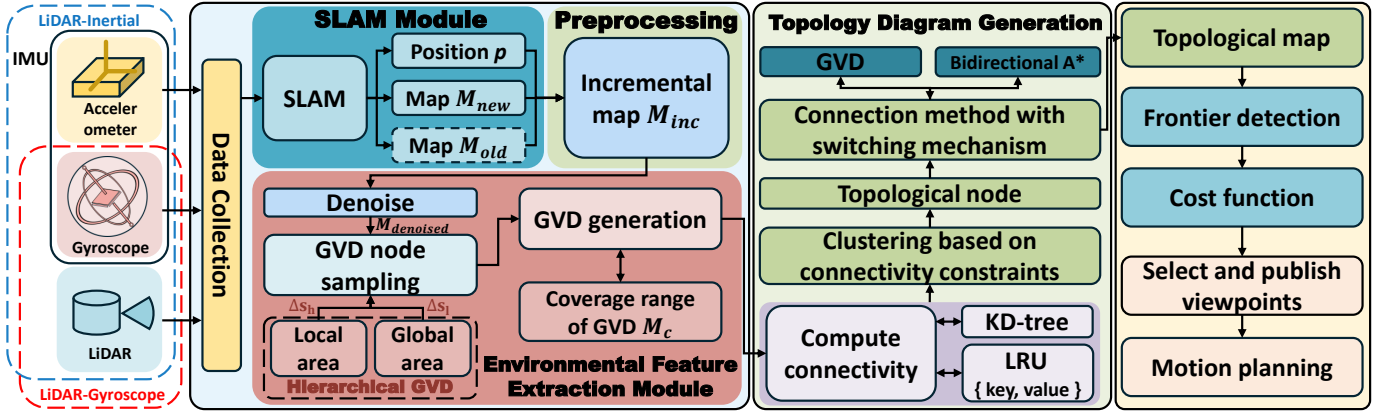


Fig. 2: Pipeline.

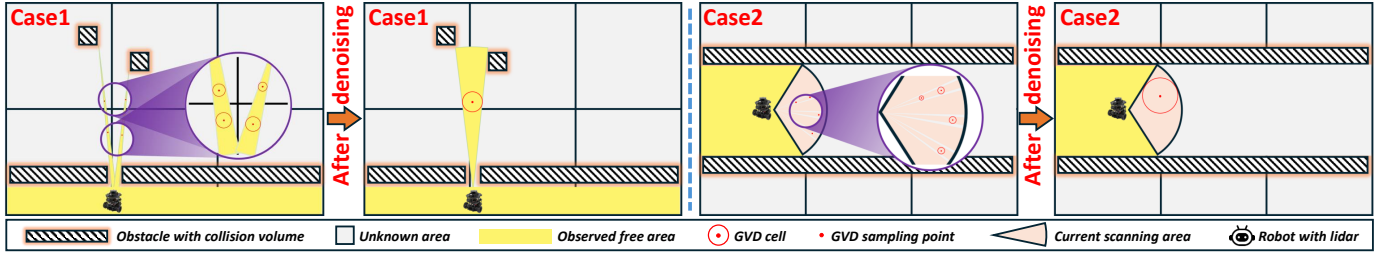


Fig. 3: In case 1, LiDAR beams passing through narrow structures create isolated free regions (yellow) separated by large unknown areas (gray). In case 2, limited scanning frequency causes gaps between consecutive frames to be misclassified as unknown space. Both cases produce undersized GVD cells that reduce feature extraction efficiency. With map denoising, fewer GVD nodes are needed to capture environmental topology, significantly improving task performance.

Algorithm 1: Incremental Map Denoising

Input: New map $M_{\text{new}} \subseteq \mathbb{Z}^2 = \{p = (x, y) | x, y \in \mathbb{Z}\}$,
old map $M_{\text{old}} \subseteq \mathbb{Z}^2 = \{p = (x, y) | x, y \in \mathbb{Z}\}$

Output: Denoised map M_{denoised}

- 1 Incremental map: $M_{\text{inc}} = M_{\text{new}} - M_{\text{old}}$
- 2 Bounding box:
 $(x, y, w, h) \leftarrow \text{MinimumBoundingRectangle}(M_{\text{inc}})$
- 3 Cropped map: $M_{\text{crop}} = \{p = (u, v) \in M_{\text{new}} | x \leq u < x + w, y \leq v < y + h\}$
- 4 **for** $p = (x, y) \in M_{\text{inc}}$ **do**
- 5 **if** $M_{\text{crop}}(p) = \text{unknown}$ **then**
- 6 $\mathcal{N}(p) \leftarrow \{q \in \mathbb{Z}^2, \|q - p\|_{\infty} \leq 2\}$
- 7 $n \leftarrow |\{q \in \mathcal{N}(p), M_{\text{crop}}(q) \neq \text{unknown}\}|$
- 8 **if** $n > 3$ **then**
- 9 $M_{\text{denoised}}(p) \leftarrow \text{free}$
- 10 **return** M_{denoised}

narrow passages, corners, and bifurcations. Existing GVD-based methods typically use uniform sampling strategies, which face a fundamental trade-off: as illustrated in Fig. 4(c), while high-density sampling preserves environmental details, it introduces significant computational overhead and node redundancy; conversely, low-density sampling tends to omit critical narrow passages, thereby reducing exploration coverage. This limitation becomes particularly pronounced in

long-duration exploration tasks, where inefficient sampling can rapidly deplete computational resources.

To address this challenge, we propose hierarchical GVD sampling with coverage-aware prioritization, a method that integrates three key innovations to enable intelligent allocation of sampling resources. First, we design a coverage-aware sampling exclusion strategy. By maintaining a global GVD coverage matrix M_c , the system dynamically tracks areas already covered by existing GVD cells. As outlined in Algorithm ??, the process begins by taking the denoised map M_{denoised} as input and extracting the set of all free grid cells p_f , whose traversal order is randomly shuffled to produce an unordered set p_s . Each candidate grid cell (x, y) is then checked against M_c to determine whether it has already been recorded. A GVD node is generated only if it lies in an uncovered area, ensuring that each new node provides unique spatial information and maximizes the information gain per sample.

Second, we establish a dual-grained spatial prioritization mechanism. Based on the robot's real-time position and following (3), the environment is dynamically divided into a local high-frequency sampling region M_{high} and a global low-frequency sampling region M_{low} (all areas outside M_{high}). Within M_{high} , a fine step size Δs_h is used to capture detailed structural features necessary for decision-making, whereas in M_{low} , a coarse step size Δs_l is applied to reduce computa-

tional load while maintaining topological consistency.

$$M_{high} = \left\{ (x, y) \left| \begin{array}{l} x \in \left[x_r - \frac{L}{2\rho}, x_r + \frac{L}{2\rho} \right], \\ y \in \left[y_r - \frac{L}{2\rho}, y_r + \frac{L}{2\rho} \right] \end{array} \right. \right\} \quad (3)$$

Here, L denotes the search range size and ρ represents the map resolution. Their values are adjusted such that the local region corresponds to a 5×5 grid area. The robot's position is denoted as $p_r = (x_r, y_r)$.

Third, we introduce an efficiency-oriented GVD generation criterion. To determine the effective range of each GVD cell, the search radius r is incrementally expanded from initial value r_{min} with a step size Δr . The Manhattan distance d_m defined in (4) is used to rapidly estimate obstacle locations until an obstacle is first detected within scanning region A .

$$d_m(p_s, p_o) = |x - x_o| + |y - y_o| \quad (4)$$

Among them, $p_o = (x_o, y_o)$ is the coordinate of the obstacle.

At this point, the Euclidean distance $d(p_s, p_o)$ shown in (5) to the nearest obstacle is recorded, and all equidistant obstacles O are collected as shown in (6).

$$d(p_s, p_o) = \sqrt{(x - x_o)^2 + (y - y_o)^2} \quad (5)$$

$$O = \{o_i \mid d_M(p, o_i) < d_{min} + \delta\} \quad (6)$$

If $|O| > 1$ and the Euclidean distance d_o between obstacles exceeds a predefined threshold, the multi-obstacle bisection criterion is applied to compute the GVD node coordinates (x_g, y_g) and its maximum inscribed circle radius r_g , ensuring that the node lies in the safe medial region between multiple obstacles. Nodes (x_g, y_g) satisfying $r_g > r_{min}$ and their corresponding radii r_g are recorded into $\mathcal{V}_{GVD}$ and $\mathcal{R}_{GVD}$, respectively, to facilitate rapid retrieval during subsequent topological node generation. Finally, following (7), coverage regions are marked in M_c according to radius priority, and the updated GVD is published in real time, providing a zero-redundancy topological partition with practical navigational value.

$$M_c = \left\{ (x, y) \mid \sqrt{(x - x_c)^2 + (y - y_c)^2} \leq \lceil r \rceil + 1 \right\} \quad (7)$$

The refined sampling mechanism described above enables high-frequency scanning around the robot to capture environmental details, while low-frequency scanning is applied in other areas to reduce computational load. Combined with the incremental GVD coverage matrix and its labeled access mechanism, the system ensures that each node contributes maximally to exploration gains under limited sampling budgets. As shown in Fig. 4c and Fig. 4d, this strategy achieves a dual optimization of both accuracy and efficiency, significantly reducing redundant nodes and path backtracking. During incremental robotic exploration, the GVD adaptively supplements nodes as new areas are discovered, thereby continuously enriching and correcting the environmental topology.

Algorithm 2: GVD

Input: Denoised map $M_{denoised} \subseteq \mathbb{Z}^2$
Output: GVD node set $\mathcal{V}_{GVD}$, node radius set $\mathcal{R}_{GVD}$

```

1 Initialize coverage map  $M_c$ 
2 while system is running do
3   Find free grids:
4      $p_f = \{(x, y) \in \mathbb{Z}^2 \mid M_{denoised}(x, y) = \text{free}\}$ 
5   Shuffle sampling order:  $p_s \leftarrow \text{Shuffle}(p_f)$ 
6   Get robot position:  $p_r \leftarrow \text{GetPosition}()$ 
7   Local map around robot with high sampling
   density:  $M_{high} \leftarrow \text{LocalRange}(p_r)$ 
8   for  $(x, y)$  in  $p_s$  with step  $\Delta s_h$  do
9     if  $(x, y)$  in  $M_c$  then
10       continue
11     if  $((x, y)$  not in  $M_{high})$  and (not global
       sampling step  $\Delta s_l$ ) then
12       continue
13     Search radius  $r \leftarrow r_{min}$ 
14     while no obstacle found do
15       Build search matrix  $A$  centered at  $(x, y)$ 
16       with radius  $r$ 
17       if obstacle exists in  $A$  then
18         break
19        $r = r + \Delta r$ 
20     Compute distance to nearest obstacles  $d$ 
21     Select nearest obstacle set  $O$ 
22     if  $|O| > 1$  then
23       for each obstacle  $o_j \in O$  do
24         if  $d_o > \text{threshold}$  then
25           Compute GVD node  $(x_g, y_g)$ 
26           Compute radius  $r_g$ 
27           if  $r_g > r_{min}$  then
28              $\mathcal{V}_{GVD} \leftarrow \{(x_g, y_g)\}$ 
29              $\mathcal{R}_{GVD} \leftarrow \{r_g\}$ 
30           break
31     Update  $M_c$  with node with the largest radius
32 return  $\mathcal{V}_{GVD}$  and  $\mathcal{R}_{GVD}$ 

```

B. Topology Diagram Generation and Maintenance Module

1) *Connectivity-Constrained Mean Shift Clustering with KD-Tree Acceleration:* The traditional mean shift clustering algorithm exhibits three significant limitations in the context of topology construction, as illustrated in case 1 and case 2 of Fig. 5. First, cluster centers may fall into obstacle regions, rendering the generated topological nodes unreachable in the physical environment. Second, GVD nodes separated by obstacles may be incorrectly merged due to density attraction, leading to loss of local topological nodes and coverage gaps. Finally, as the environmental scale increases, the number of GVD nodes grows rapidly. If a linear traversal method is still used for neighborhood search, the computational complexity

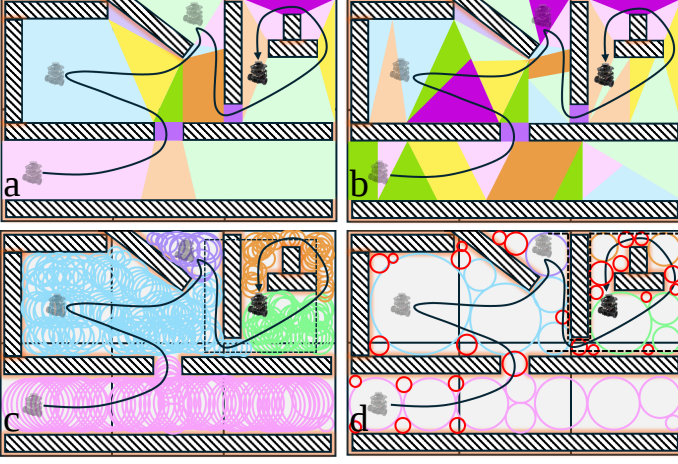


Fig. 4: (a) and (b) illustrate how the sampling density of conventional Voronoi diagrams affects the partitioning of functional regions, while (c) is the regional division effect under high-density GVD sampling. In contrast, (d) shows that our hierarchical sampling strategy achieves broader area coverage with significantly fewer GVD nodes, thereby improving topological accuracy while minimizing computational time.

increases sharply, significantly degrading real-time performance and constraining online deployment capability.

To address the aforementioned issues, we propose an improved mean shift framework based on connectivity constraints, which systematically integrates map reachability information into the entire iterative process of mean shift. Furthermore, for the first time in the mean shift algorithm, we introduce KD-tree as the core spatial indexing structure to alleviate the computational burden caused by large-scale GVD nodes. The specific procedure is as follows: taking the set of GVD sample points as input, recursive spatial partitioning is performed on the two-dimensional plane. The root node is split based on the median of the x-coordinates, the next level switches to the y-coordinates, and the subsequent level returns to the x-coordinates, alternating sequentially until the number of points contained in a leaf node falls below a preset threshold. This strategy ensures that the height of the KD-tree remains at $O(\log N)$, reducing the average complexity of any radius-based neighborhood query from $O(N)$ for linear traversal to $O(\log N)$, thereby ensuring real-time responsiveness for subsequent iterations.

To prevent the generation of unreachable cluster centers and suppress cross-obstacle attraction, we deeply integrate connectivity constraints into the mean shift iterative process. The algorithm precomputes connectivity between any two nodes by performing collision-free line segment detection on the occupancy grid map along the straight line connecting them. If the entire segment lies within free space, the two nodes are considered connected; otherwise, they are deemed disconnected. All detection results are stored as key-value pairs in an LRU cache, establishing a multi-resolution, shareable connectivity caching system that avoids redundant computations. Subsequently, the bandwidth parameter r_m is adaptively determined according to (8): the Euclidean distances between all point pairs in

the input dataset are computed, and a threshold from the distance distribution is selected based on a specified quantile to serve as the final bandwidth parameter. The distance value corresponding to this quantile is used as the radius of the neighborhood in the mean shift clustering process.

$$r_m = \text{Quantile}_q(\{||\mathcal{V}_i - \mathcal{V}_j|| \mid 1 \leq i < j \leq n\}) \quad (8)$$

where q denotes the quantile parameter and is set to $q = 0.3$, n represents the total number of GVD nodes, and $\mathcal{V}_i$ and $\mathcal{V}_j$ refer to the i -th and j -th nodes in $\mathcal{V}_{GVD}$, respectively.

Subsequently, all sampling points are used as initial seeds and undergo iterative mean shift. During the iteration, the algorithm first employs the KD-tree to rapidly retrieve the set of neighborhood points $\mathcal{N}_{p_m}$ within the radius r_m that are connected to the current center point $p_m = (x, y)$, as shown in (9).

$$\mathcal{N}_{p_m} = \{p_m \in \mathcal{V}_{GVD} \mid ||p_m - p'_m|| < h, \text{ connected}(p_m, p'_m)\} \quad (9)$$

where p_m denotes the current center point, and p'_m represents the other points within the neighborhood $\mathcal{N}_{p_m}$ of radius r_m .

By integrating the map and the connectivity cache, only points that are unobstructed and reachable are included in the mean calculation and used to update the center point, as expressed in (10).

$$p_m = \frac{1}{|\mathcal{N}_{p_m}|} \sum_{p'_m \in \mathcal{N}_{p_m}} p'_m \quad (10)$$

If the new and old center points are unreachable, or if the new center point falls within an obstacle area, the iteration is terminated prematurely to ensure that the cluster center always remains in free space and is reachable. Convergent centers that are too close to each other are then merged to obtain a set of cluster centers c that are reasonably distributed in space and non-overlapping. Finally, in the label assignment phase, the algorithm adopts the nearest neighbor principle, as defined in (11), to assign each sampling point to the nearest and connected cluster center. For each sample $\mathcal{V}_i$, the assigned label corresponds to the closest and connected cluster center.

$$\ell_i = \arg \min_k \{||\mathcal{V}_i - c_k|| \mid \text{connected}(\mathcal{V}_i, c_k), c_k \in c\} \quad (11)$$

As illustrated in the case 1 and case 2 of Fig. 5, compared to the traditional mean shift algorithm, the proposed method not only prevents cluster centers from being located on obstacles but also effectively suppresses the cross-structural node "attraction" phenomenon caused by obstacle barriers. This enhancement improves both the spatial coverage completeness and usability of the topological nodes. The overall process fully integrates spatial density with environmental reachability information, providing an accurate topological foundation for subsequent exploration tasks. Furthermore, the introduction of the KD-tree and LRU cache mitigates the computational burden associated with large-scale GVD nodes.

2) *Connection Of Topological Map Based On Switch Mechanism*: For the topological nodes obtained via mean-shift clustering, we propose a dual-layer switching connection strategy to construct a globally consistent and accurate environmental topological graph. First, both the clustered topological nodes

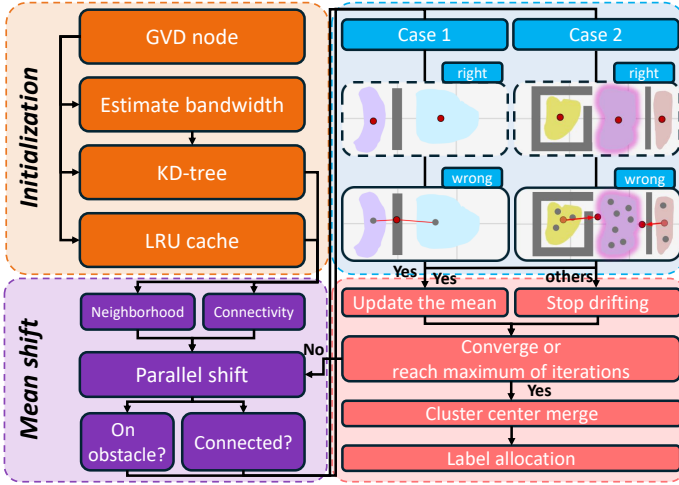


Fig. 5: Mean shift based on connectivity constraints. In case 1 and case 2, the red dots are topological nodes and the gray dots are GVD nodes. The correct situation should be that the GVD nodes in each color area are clustered into one topological node. Error situation in case1 using traditional mean shift: The topological node generated by the clustering of two GVD nodes is generated on an obstacle. Error situation in case2 using traditional mean shift: The original correct topological nodes in the GVD areas of yellow and skin color were attracted to the pink area by crossing boundaries.

and all GVD nodes are embedded into the topological graph. Based on the traversability of the GVD nodes, collision-free connectivity information between nodes is precomputed and stored using an LRU cache. Whenever multiple connected components appear in the topological graph, the smallest connected subgraph is identified, and the shortest GVD path between this subgraph and other nodes is queried. If such a path exists, an edge is directly added to achieve efficient GVD-based completion. If no feasible GVD path is available, a fallback mechanism is triggered, employing a bidirectional A* algorithm to compute the shortest collision-free trajectory in the occupancy grid using nearest-neighbor heuristics. This trajectory is then converted into a topological edge. The resulting topological graph represents nodes as major passages or regions in the environment, with edges denoting feasible paths between these regions, forming an environmental topological representation based on spatial connectivity constraints. By leveraging the switching mechanism between GVD and A*, the approach preserves the efficiency of GVD connectivity priors in large open spaces while utilizing the completeness of A* to ensure feasible connections in narrow or obstacle-dense areas. This achieves a balance among global topological integrity, real-time performance, and computational cost.

C. Viewpoints Selection Module

1) *Frontier Detection Based on Morphology*: Unlike traditional frontier-based exploration methods, our method utilizes topological nodes and GVD nodes as navigation viewpoints, making the identification of frontier a critical challenge. Conventional frontier extraction methods determine frontier solely

by detecting whether unknown grids exist in the GVD cell. As shown in Fig. 6, when an unknown region is separated from the observed free area by obstacles and the node itself lies within the observed free region, designating such a node as the next navigation target will not yield new environmental information due to the obstruction, leading to repeated exploration. Therefore, additional verification is required to determine whether a valid passage for LiDAR beam exists between the unknown and known regions.

Specifically, if a GVD cell contains only unknown and free regions without any obstacles, the corresponding GVD node can be directly regarded as valid frontier. If obstacles are present, the reachability of the passage must be further evaluated. To this end, we propose the following frontier detection algorithm: first, a local submap is cropped from the global map using the center and radius of the candidate GVD node. Then, morphological dilation is applied separately to the unknown, free, and obstacle regions using an 8-neighborhood expansion to generate their respective neighborhood masks. Next, the number of overlapping pixels between the unknown and free regions is calculated via pixel-wise multiplication to extract the contact frontiers between the two regions. As shown in case 1 of Fig. 6, if the number of overlapping pixels is minimal (< 5), it indicates limited contact between the unknown and observed free regions, and the node is classified as a non-frontier point.

Furthermore, the obstacle neighborhood mask is multiplied with the overlapping region of the unknown and free areas, and the proportion of overlapping pixels relative to the total unknown-free overlap is calculated. As illustrated in case 2 and case 3 of Fig. 6, when this proportion exceeds 0.8, it suggests that the unknown region is primarily adjacent to obstacle edges, representing an unreachable or pseudo-frontier with no practical exploration value. Such nodes should also be identified as explored non-frontier points. Only in cases outside the above conditions—where the unknown region sufficiently contacts the free region and is not concentrated near obstacles—is the node determined to be valid frontier.

This method effectively integrates spatial topology, morphological operations, and quantitative threshold criteria, successfully avoiding misjudgments caused by map noise or obstacle occlusion. It enhances the robustness and accuracy of frontier detection, providing a reliable basis for autonomous exploration tasks.

2) *Viewpoint Selection based on Designed Cost Function*: Upon completion of the topological graph construction and frontiers generation, the robot must select the optimal exploration target in real time from multiple candidate frontiers within the environment. To this end, we propose a lightweight cost evaluation framework that integrates topological, geometric, and kinematic information, balancing decision-making accuracy with computational efficiency.

Specifically, as shown in Fig. 7, the framework first maps all candidate frontiers to their corresponding topological nodes in the constructed graph. Based on connectivity, it quickly determines whether an obstacle-free straight-line path exists between the robot's current position and the target point, as expressed in (12). If such a path exists, the Euclidean

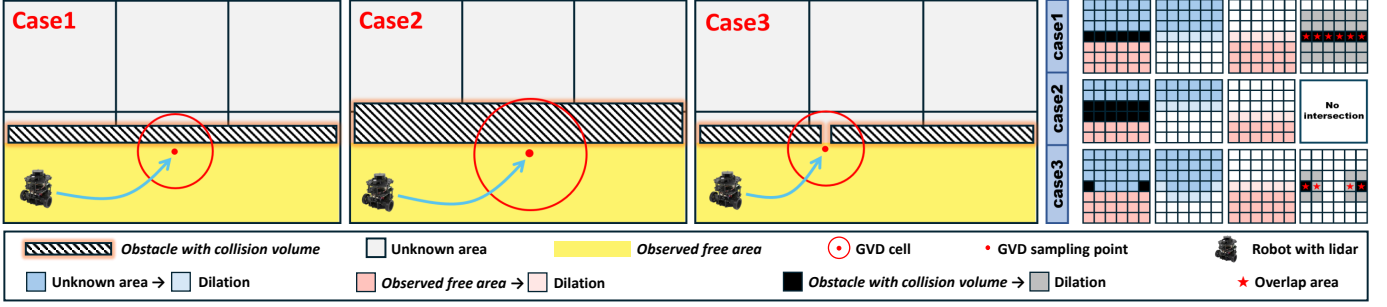


Fig. 6: Frontier detection. The last figure shows the morphological dilation of unknown area, observed free area and obstacle respectively, and the calculation of the size of the overlapping area to determine whether it is frontier.

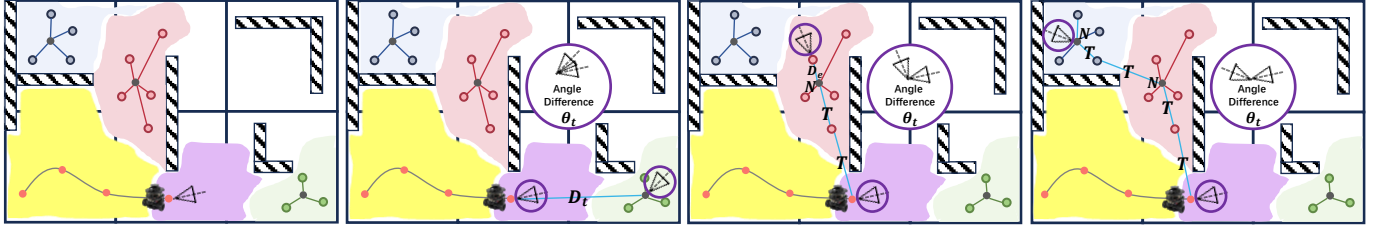


Fig. 7: Frontier selection. The gray dots are topological nodes obtained by clustering GVD nodes, while the remaining dots are the original GVD nodes. When calculating cost, D_t is the Euclidean distance between two nodes (when there is a straight path), while T , N and D_e respectively represent the total length of topological path to be detour, the number of topological nodes passed through and the distance to the end node. The last three subfigure respectively represent the cost for robot to evaluate the travel to three candidate target points in different color areas from the current position, and the one with the lowest cost will be selected as the next viewpoint.

distance D_t is directly adopted as the base cost, with an additional weighting correction based on the heading angle θ_t to reduce motion losses caused by sharp turns. If no direct path exists, the topological path length T is used as the main cost component, supplemented by a weighted combination of the terminal distance D_e , the number of topological nodes N , and the relative heading angle θ_t , forming a comprehensive cost metric.

Here, the topological path length T consists of two parts: the actual path length from the robot to the nearest topological node and the terminal distance D_e from the topological endpoint to the target point. This design effectively captures the influence of obstacles on the actual path while avoiding excessive computational overhead from point-by-point path planning. The inclusion of the terminal distance D_e compensates for spatial deviations between topological nodes and actual frontiers, ensuring accurate path estimation. The weighted node count N encourages the robot to prioritize structurally simpler paths, thereby reducing the frequency of turns and replanning. The angle term θ_t , leveraging the robot's current orientation, further minimizes unnecessary rotational movements.

By integrating these factors, the proposed cost function maintains high accuracy in path length estimation while significantly reducing the time required for candidate point evaluation. This enhances the robot's real-time exploration

efficiency in complex and unknown environments.

$$\begin{cases} C = D_t + \alpha \cdot \theta_t \cdot D_t, & \text{if } R \subseteq M_{free} \\ C = T + \alpha \cdot N \cdot T + \beta \cdot \theta_t \cdot D_e, & \text{if } R \not\subseteq M_{free} \end{cases} \quad (12)$$

V. EXPERIMENTS

We verify the exploration efficiency of the system respectively through simulations and real-world experiments. In simulations, we adopt a computer with 13th Gen Intel Core i7-13700K CPU and NVIDIA GeForce RTX2060 super GPU. The robot used in simulations is turtlebot3-burger. In real-world experiments, we use a laptop with Intel i7-7700HQ CPU and NVIDIA GeForce GTX1060 GPU to remotely maneuver an turtlebot3-burger for validation. Turtlebot3-burger employs a Raspberry Pi 3 for high-level computation and an STM32F7-based OpenCR board for real-time motor control, complemented by dual dynamixel servomotors XL430-W250, a 360 laser distance sensor LDS-01, and an on-board IMU. The environments of simulations are shown as S1-S4 in Fig. 8 and the environments of real-world experiments are shown as R1-R2 in Fig. 8. In addition, experimental setup is shown in Table. I, and experimental equipment is shown in Fig. 8g.

A. Simulations

To evaluate the exploration efficiency of the system, simulation experiments are conducted in four different simulation environments. As shown in Table. II, by comparing two

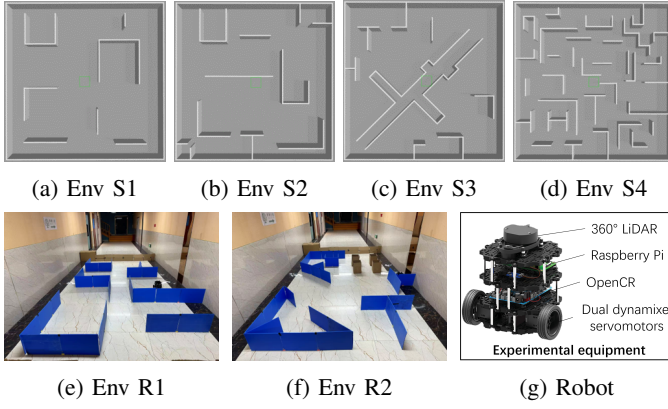


Fig. 8: Experimental environments and equipment.

TABLE I: Experimental setup.

Type	Scene	Max Speed(m/s)	Max Angle(°/s)	Lidar range(°)	Termination condition
Simulation	S1	0.3	17.1	270	95%
	S2	0.3	17.1	270	95%
	S3	0.3	17.1	270	95%
	S4	0.3	17.1	270	95%
Real-World	R1	0.3	18.9	360	95%
	R2	0.3	18.9	360	95%

metrics (exploration time and movement distance) with three methods (TARE [14], FAEL [18], GVD [25]), each method is executed ten times at the same location in each environment. In Fig. 9, we also present the trajectory routes of each method to compare the path backtracking frequencies.

In S1 (20m $\times$ 20m), we deploy a small number of obstacles. Given the simplicity of this scenario, all methods achieve 100% success rate. Compared to other methods, our method reduces exploration time by up to 36.9%. Moreover, as shown in the Fig. 9, the trajectory of our method exhibits no overlapping points and avoids path backtracking. In S2 (20m $\times$ 20m), we further complicate the environmental structure. As illustrated in the Fig. 9, GVD [25] fails to enter the narrow passage at the red marked location and returns to the same point during the final phase of the task, resulting in path backtracking. In contrast, our method accurately captures narrow passages through hierarchical GVD sampling, thereby maximizing exploration efficiency. Compared to other methods, our method reduces exploration time by up to 23.9%. In S3 (20m $\times$ 20m), we add a bidirectional passage to partition the area. As seen in the Fig. 9, GVD [25] again misses the narrow passage at the red marked location, leading to repeated path backtracking. Our method reduces exploration time by up to 27.9% compared to GVD [25]. In S4 (20m $\times$ 20m), we configure an extremely complex obstacle structure, making the environment highly challenging for exploration tasks. In this environment, our method achieves 100% success rate, while other methods only achieve success rates between 40% - 60%. The trajectory of our method generally follows a trend of circumventing the environment, unlike other methods that frequently traverse the environment arbitrarily, resulting in frequent path backtracking. Compared to other methods, our method reduces exploration time by up to 25.8%.

Our method demonstrates significant advantages in explo-

TABLE II: Result of simulations.

Scene	Method	Exploration Time(s)		Movement Distance(m)		Average Speed(m/s)	Time-limited Number(s)	Success Number
		avg	std	avg	std			
S1	Ours	287.3	10.7	86.19	16.9	0.3	500	10
	TARE [14]	450.6	21.6	112.65	18.1	0.25	500	10
	FAEL [18]	455.1	18.4	127.4	19.2	0.28	500	10
	GVD [25]	414.1	25.2	111.8	20.5	0.27	500	10
S2	Ours	372.3	14.4	111.6	16.8	0.3	550	10
	TARE [14]	464.5	23.7	130.1	22.8	0.28	550	10
	FAEL [18]	446.5	17.2	120.6	18.5	0.27	550	10
	GVD [25]	489.5	35.1	122.4	25.3	0.25	550	10
S3	Ours	401.3	16.5	120.4	18.2	0.3	600	10
	TARE [14]	416.3	33.1	108.2	22.1	0.26	600	10
	FAEL [18]	454.3	27.2	122.6	21.8	0.27	600	10
	GVD [25]	556.3	29.6	144.6	25.7	0.26	600	8
S4	Ours	523.3	20.2	151.8	21.2	0.29	750	10
	TARE [14]	639.3	35.8	159.9	28.9	0.25	750	4
	FAEL [18]	705.2	32.4	183.4	27.8	0.26	750	6
	GVD [25]	665	41.7	172.9	33.5	0.26	750	4

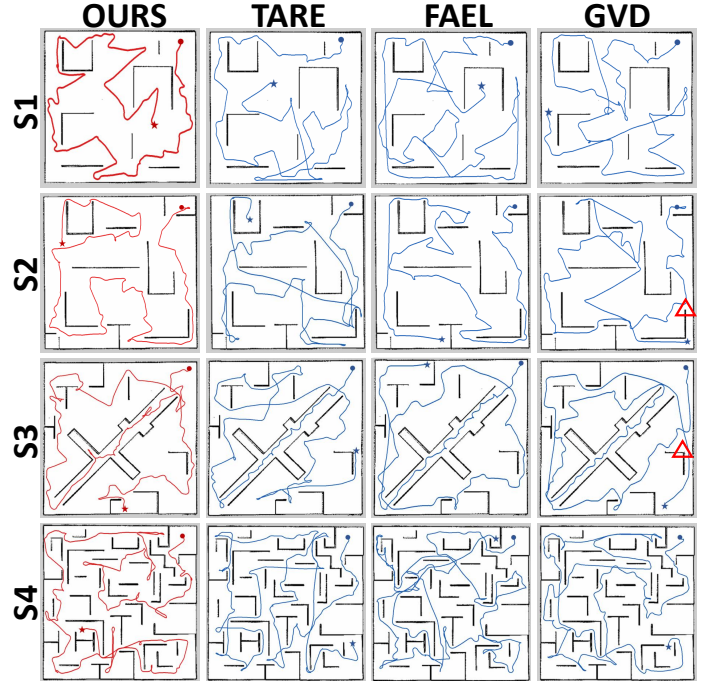


Fig. 9: Trajectory of simulations. The dots represents starting point, and star represents ending point.

ration efficiency and path planning quality across various environments. Furthermore, our method maintains perfect success rates in challenging scenarios where other methods deteriorate to 40-60% success rates, proving its superior robustness in complex obstacle configurations.

B. Real-World Experiments

We also conduct real-world experiments in two different real-world environments. As shown in Table. III, by comparing two metrics (exploration time and movement distance) with three methods (TARE [14], FAEL [18], GVD [25]), each method is executed five times at the same location in each environment. In Fig. 10, we also present the trajectory routes of each method to compare the path backtracking frequencies.

In R1 (7m $\times$ 3m), all methods achieved 100% success rate. Our method consumed the shortest time. Compared with other methods, the exploration time of ours is reduced by up to

TABLE III: Result of real-world experiments.

Scene	Method	Exploration Time(s)		Movement Distance(m)		Average Speed(m/s)	Time-limited Number(s)	Success Number
		avg	std	avg	std			
R1	Ours	35.1	10.3	10.5	3.7	0.3	150	5
	TARE [14]	49.1	16.1	14.2	5.9	0.29	150	5
	FAEL [18]	52.6	12.3	15.8	5.2	0.3	150	5
	GVD [25]	123.9	14.5	18.6	3.2	0.15	150	5
R2	Ours	51.3	12.4	15.4	5.1	0.3	150	5
	TARE [14]	67.3	23.7	15.5	5.2	0.23	150	4
	FAEL [18]	70.5	20.5	19.7	6.9	0.28	150	5
	GVD [25]	95.7	15.4	25.9	7.8	0.27	150	3

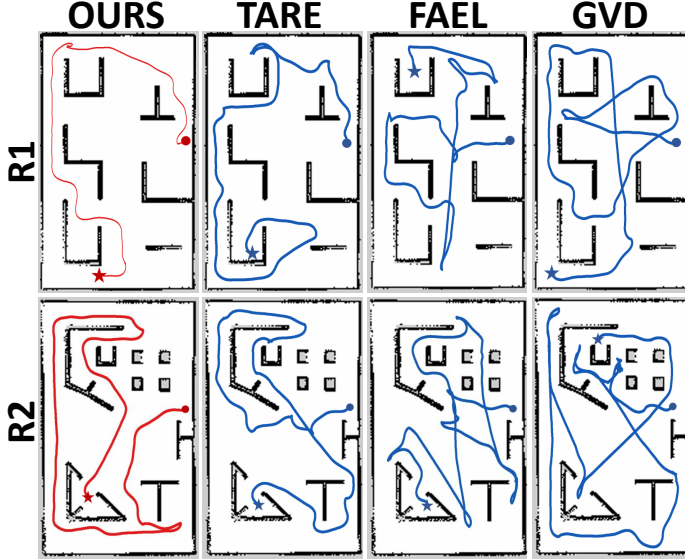


Fig. 10: Trajectory of real-world experiments. The dots represents starting point, and star represents ending point.

71.7%. As can be seen from Fig. 10, both FAEL [18] and GVD [25] experience varying degrees of path backtracking. This leads to a relatively low exploration efficiency in actual scenarios. In R2 (7m $\times$ 3m), the success rate of GVD [25] is only 60%, while our method could still achieve 100% exploration success rate. Compared with other method, the exploration time of ours is reduced by up to 46.4%.

In actual scenario tests, our method has demonstrated significant advantages, with the exploration efficiency increasing by up to 71.7% and maintaining 100% task success rate consistently. It has shown stronger practicality and reliability of our method in real and complex environments.

VI. CONCLUSION

This paper proposed a topological map construction method based on hierarchical GVD sampling, which effectively solved the key problems such as frequent path backtracking, missed detection of narrow channels, and unreachable topological nodes when robots conducted autonomous exploration in unknown and complex environments. Firstly, a hierarchical GVD sampling strategy with coverage awareness was proposed. By conducting global-local dual-granularity sampling and coverage map maintenance for the denoised map, efficient capture of environmental features and effective suppression of redundant nodes were achieved. Secondly, a connection-constrained mean shift clustering algorithm was designed, combined with

KD-tree acceleration and a double-layer switching connection mechanism, to ensure the reachability of topological nodes and the connectivity of the map. Finally, a frontier detection method based on morphological dilation was introduced, combined with a lightweight cost function, to achieve precise screening of real exploration targets and selection of the optimal viewpoint. The simulation and real-world experiment results showed that this method can maintain 100% exploration success rate in various complex scenarios, with the exploration time reduced by up to 71.7%, and there was very little backtracking phenomenon. It was significantly superior to the comparison methods, fully verifying the superiority of our method in improving exploration efficiency and robustness.

REFERENCES

- [1] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.
- [2] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.
- [3] H. Balta, J. Bedkowski, S. Govindaraj, K. Majek, P. Musialik, D. Serano, K. Alexis, R. Siegwart, and G. De Cubber, "Integrated data management for a fleet of search-and-rescue robots," *Journal of Field Robotics*, vol. 34, no. 3, pp. 539–582, 2017.
- [4] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 3, pp. 367–385, 2003.
- [5] K. Wu, W. S. Lee, and D. Hsu, "Pomdp to the rescue: Boosting performance for robocup rescue," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5294–5299. IEEE, 2015.
- [6] S. Kohlbrecher, A. Romay, A. Stumpf, A. Gupta, O. Von Stryk, F. Bacim, D. A. Bowman, A. Goins, R. Balasubramanian, and D. C. Conner, "Human-robot teaming for rescue missions: Team vigir's approach to the 2013 darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 3, pp. 352–377, 2015.
- [7] M. P. Christiansen, M. S. Laursen, R. N. Jørgensen, S. Skovsen, and R. Gislum, "Designing and testing a uav mapping system for agricultural field surveying," *Sensors*, vol. 17, no. 12, p. 2703, 2017.
- [8] J. Liu, Y. Lv, Y. Yuan, W. Chi, G. Chen, and L. Sun, "An efficient robot exploration method based on heuristics biased sampling," *IEEE Transactions on Industrial Electronics*, vol. 70, DOI 10.1109/TIE.2022.3203762, no. 7, pp. 7102–7112, 2023.
- [9] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, "A multi-resolution frontier-based planner for autonomous 3d exploration," *IEEE Robotics and Automation Letters*, vol. 6, DOI 10.1109/LRA.2021.3068923, no. 3, pp. 4528–4535, 2021.
- [10] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, DOI 10.1109/LRA.2021.3051563, no. 2, pp. 779–786, 2021.
- [11] H. Zhang, S. Wang, Y. Liu, P. Ji, R. Yu, and T. Chao, "Efpr: Efficient frontier-based autonomous uav exploration strategy for unknown environments," *IEEE Robotics and Automation Letters*, vol. 9, DOI 10.1109/LRA.2024.3363531, no. 3, pp. 2941–2948, 2024.
- [12] Y. Zhao, L. Yan, H. Xie, J. Dai, and P. Wei, "Autonomous exploration method for fast unknown environment mapping by using uav equipped with limited fov sensor," *IEEE Transactions on Industrial Electronics*, vol. 71, DOI 10.1109/TIE.2023.3285921, no. 5, pp. 4933–4943, 2024.
- [13] Y. Chen, W. Jiang, Z. Zhang, L. Zhang, G. Zhu, H. Zhang, and Y. Wang, "Sampling-focused marching tree: Optimal planning based on minimized topological refinement and homotopy-heuristic exploration," *IEEE/ASME Transactions on Mechatronics*, DOI 10.1109/TMECH.2024.3462506, pp. 1–12, 2024.
- [14] C. Cao, H. Zhu, H. Choset, and J. Zhang, "Tare: A hierarchical framework for efficiently exploring complex 3d environments," in *Robotics: Science and Systems*, vol. 5, p. 2, 2021.

- [15] A. Soragna, M. Baldini, D. Joho, R. Kümmerle, and G. Grisetti, "Active slam using connectivity graphs as priors," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, DOI [10.1109/IROS40897.2019.89688613](https://doi.org/10.1109/IROS40897.2019.89688613), pp. 340–346, 2019.
- [16] X. Yang, Y. Yang, C. Yu, J. Chen, J. Yu, H. Ren, H. Yang, and Y. Wang, "Active neural topological mapping for multi-agent exploration," *IEEE Robotics and Automation Letters*, vol. 9, DOI [10.1109/LRA.2023.3331892](https://doi.org/10.1109/LRA.2023.3331892), no. 1, pp. 303–310, 2024.
- [17] Z. Zhang, J. Yu, J. Tang, Y. Xu, and Y. Wang, "Mr-topomap: Multi-robot exploration based on topological map in communication restricted environment," *IEEE Robotics and Automation Letters*, vol. 7, DOI [10.1109/LRA.2022.3192765](https://doi.org/10.1109/LRA.2022.3192765), no. 4, pp. 10 794–10 801, 2022.
- [18] J. Huang, B. Zhou, Z. Fan, Y. Zhu, Y. Jie, L. Li, and H. Cheng, "Fael: Fast autonomous exploration for large-scale environments with a mobile robot," *IEEE Robotics and Automation Letters*, vol. 8, DOI [10.1109/LRA.2023.3236573](https://doi.org/10.1109/LRA.2023.3236573), no. 3, pp. 1667–1674, 2023.
- [19] W. Zhao, R. Lin, S. Dong, and Y. Cheng, "A study of the global topological map construction algorithm based on grid map representation for multirobot," *IEEE Transactions on Automation Science and Engineering*, vol. 20, DOI [10.1109/TASE.2022.3198801](https://doi.org/10.1109/TASE.2022.3198801), no. 4, pp. 2822–2835, 2023.
- [20] M. Saroya, G. Best, and G. A. Hollinger, "Roadmap learning for probabilistic occupancy maps with topology-informed growing neural gas," *IEEE Robotics and Automation Letters*, vol. 6, DOI [10.1109/LRA.2021.3068886](https://doi.org/10.1109/LRA.2021.3068886), no. 3, pp. 4805–4812, 2021.
- [21] C. Wang, W. Chi, Y. Sun, and M. Q.-H. Meng, "Autonomous robotic exploration by incremental road map construction," *IEEE Transactions on Automation Science and Engineering*, vol. 16, DOI [10.1109/TASE.2019.2894748](https://doi.org/10.1109/TASE.2019.2894748), no. 4, pp. 1720–1731, 2019.
- [22] W. Chi, J. Wang, Z. Ding, G. Chen, and L. Sun, "A reusable generalized voronoi diagram-based feature tree for fast robot motion planning in trapped environments," *IEEE Sensors Journal*, vol. 22, DOI [10.1109/JSEN.2021.3054888](https://doi.org/10.1109/JSEN.2021.3054888), no. 18, pp. 17 615–17 624, 2022.
- [23] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized voronoi diagram-based efficient heuristic path planning method for rrts in mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 69, DOI [10.1109/TIE.2021.3078390](https://doi.org/10.1109/TIE.2021.3078390), no. 5, pp. 4926–4937, 2022.
- [24] L. Vachhani, A. D. Mahindrakar, and K. Sridharan, "Mobile robot navigation through a hardware-efficient implementation for control-law-based construction of generalized voronoi diagram," *IEEE/ASME Transactions on Mechatronics*, vol. 16, DOI [10.1109/TMECH.2010.2076825](https://doi.org/10.1109/TMECH.2010.2076825), no. 6, pp. 1083–1095, 2011.
- [25] D. Chen, X. Gong, A. Xiao, A. Luo, L. Sun, Y. Lv, J. Wang, and W. Chi, "Gvd-exploration: An efficient autonomous robot exploration framework based on fast generalized voronoi diagram extraction," *IEEE Transactions on Intelligent Vehicles*, DOI [10.1109/TIV.2024.3503763](https://doi.org/10.1109/TIV.2024.3503763), pp. 1–13, 2024.
- [26] G. Grisetti, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.