2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Heuristic-based Incremental Probabilistic Roadmap for Efficient UAV Exploration in Dynamic Environments

Zhefan Xu*, Christopher Suzuki*, Xiaoyang Zhan, and Kenji Shimada

***Abstract*— Autonomous exploration in dynamic environments necessitates a planner that can proactively respond to changes and make efficient and safe decisions for robots. Although plenty of sampling-based works have shown success in exploring static environments, their inherent sampling randomness and limited utilization of previous samples often result in sub-optimal exploration efficiency. Additionally, most of these methods struggle with efficient replanning and collision avoidance in dynamic settings. To overcome these limitations, we propose the Heuristic-based Incremental Probabilistic Roadmap Exploration (HIRE) planner for UAVs exploring dynamic environments. The proposed planner adopts an incremental sampling strategy based on the probabilistic roadmap constructed by heuristic sampling toward the unexplored region next to the free space, defined as the heuristic frontier regions. The heuristic frontier regions are detected by applying a lightweight vision-based method to the different levels of the occupancy map. Moreover, our dynamic module ensures that the planner dynamically updates roadmap information based on the environment changes and avoids dynamic obstacles. Simulation and physical experiments prove that our planner can efficiently and safely explore dynamic environments. Our software[1] is available on GitHub with the experiment video[2].**

I. Introduction

Over the past decade, the integration of autonomous unmanned aerial vehicles (UAVs) into various industries for the rapid and reliable acquisition of information has become imperative. In scenarios where human workers co-exist with UAVs in dynamic and unknown environments, the need for an online informative path-planning algorithm becomes evident. Such an algorithm is crucial for efficiently exploring unknown areas while maintaining the safety of human workers. Consequently, developing an efficient and safe exploration path planning algorithm is indispensable for overcoming the challenges posed by dynamic environments.

The online exploration challenge involves identifying a sequence of informative sensor positions [1]. Early exploration strategies, pioneered with ground-based 2D robots, successfully gained insights into unknown regions by focusing on border regions (the frontiers) [2]. While subsequent efforts extended these concepts to aerial robots [3][4][5], the computational demands for identifying frontiers may escalate significantly with the expansion of the environment size. In contrast, sampling-based methods have gained favor in UAV exploration due to their computational efficiency and diverse information gain formulations [6][7][8][9]. Typically, these methods employ cost-utility functions to assess exploration potentials at sampling nodes and compute paths using single-query planners like the Rapidly Exploring Random Tree (RRT) [10] or its variations. However, the limitation of random sampling in a single iteration can result in incomplete coverage of mapped space nodes, hampering optimal viewpoint selection. Additionally, redundant computation may occur when evaluating previously sampled regions in subsequent iterations. Exploration being an iterative process, a multi-query planner like the Probabilistic Roadmap [11] is better suited with an efficient incremental sampling method to reduce computational costs and select viewpoints.

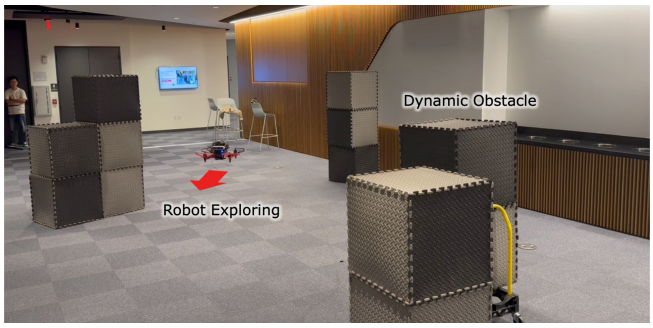


Fig. 1. The UAV exploring the unknown environment with the proposed exploration planner. In this dynamic environment, the robot is conducting unknown space exploration while actively avoiding dynamic obstacles.

To address these issues, we propose a novel Heuristic-based Incremental Probabilistic Roadmap Exploration (HIRE) planner for exploring dynamic environments. The proposed planner employs an incremental sampling strategy, utilizing a probabilistic roadmap constructed through heuristic sampling near unexplored regions adjacent to free space, defined as heuristic frontier regions. These frontier regions are identified using a lightweight vision-based technique applied to various levels of the occupancy map. Furthermore, our dynamic module ensures that the planner dynamically updates the roadmap information in response to environmental changes and effectively avoids dynamic obstacles. Simulation and physical experiments conclusively demonstrate that our planner can efficiently and safely explore dynamic environments. The main contributions of this work are:

- **Heuristic Incremental Roadmap Construction:** The proposed algorithm applies a lightweight vision-based

*The authors contributed equally.

Zhefan Xu, Christopher Suzuki, Xiaoyang Zhan, and Kenji Shimada are with the Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, 15213, USA. zhefanx@andrew.cmu.edu

[1]https://github.com/Zhefan-Xu/CERLAB-UAV-Autonomy

[2]https://youtu.be/fjVJCgDemjc?si=37-ot_i7Ycnir4Xt

method to identify heuristic frontier regions across various levels of the map and adopts our heuristic sampling strategy for incremental roadmap construction.

- **Dynamic Module for Node Update and Collision Avoidance:** Our dynamic module guarantees node information updates in dynamic environments while ensuring safe collision avoidance for dynamic obstacles.
- **High Exploration Efficiency:** We conducted comprehensive comparison experiments against state-of-the-art benchmark algorithms. The results demonstrate that our proposed planner is able to achieve higher exploration efficiency in different simulation environments.

II. Related Work

There are two major categories of autonomous exploration algorithms: the frontier-based method and the sampling-based method. Early frontier exploration has demonstrated high efficiency in 2D ground robots [2]. The idea of using frontiers to guide exploration is later extended into UAVs with fast speed [12] by selecting the goal frontier from its sensor field of view and minimizing the changes in its velocity. Meng et al.[3] adopt the idea of the frontier sampler for guiding the robot exploration. Lu et al. [4] apply a clustering method to filter the frontiers and pick the optimal frontier based on the information gain. In [5], the frontier information structure (FIS) is proposed, and their method utilizes three-step planning to achieve high exploration efficiency. While these works have demonstrated the success of frontier-based methods, it is important to note that the computational requirements for obtaining frontiers can grow exponentially with increasing environment size.

The sampling-based method has become more popular in recent years for UAV exploration. The Receding Horizon Next-Best-View (RH-NBV) planner provides a dependable solution for robot exploration [6]. It operates by expanding a tree from the robot's current position, selecting the best branch with the highest information gain, and having the robot execute the initial branch segment. Following the receding horizon manner, Papachristos et al. [13] minimize the localization and mapping uncertainties during the path selection. To enhance sampling efficiency and prevent the planner from becoming trapped in local minima, [7] stores a historical graph derived from previous samples to evaluate exploration potentials. In [14], a two-stage planner is employed to optimize saliency gain, taking into account the visual saliency of various objects in the environment. In [8], the RH-NBV is combined with the frontier-based algorithm to prevent early termination in local minima within their autonomous exploration planner (AEP). The frontiers are cached to determine the exploration goals, and these cached nodes aid in estimating the information gain using Gaussian processes. To efficiently reuse sampling information, a graph-based planner [15] was initially proposed and later extended to support team robot exploration, taking into account wireless communication [16]. With a similar incremental strategy in [17], their RRT*-based planner continuously expands and manages the tree with rewiring for path refinement, ensuring

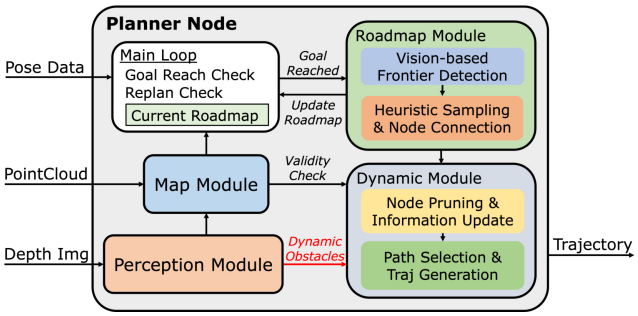


Fig. 2. Planner overview. The system input comprises the pose, pointcloud, and depth image data. At the beginning of each planning iteration, the roadmap module incrementally constructs the roadmap. Consequently, the dynamic module will update node information and generate trajectories for exploration and collision avoidance based on map and perception data.

that it does not discard the other nodes not part of the best branch. Similarly, the incremental probabilistic roadmap is employed in [18] to steer the exploration process. A group of methods [19][20][21][22] adopts a combined approach, maintaining both global and local planning horizons to generate exploration trajectories. Additionally, recent works have incorporated semantic information [23][24] and map prediction [25] for enhancing exploration strategies.

III. Problem Definition

A bounded environment, $V_{\text{env}} \in \mathbb{R}^3$, consists of free space V_{free} and occupied space V_{occ}. Without any occupancy information, the robot is required to explore the environment completely and build the voxel map $\mathcal{M}$ using its onboard sensor. At the beginning stage of the exploration, the entire environment is unknown except the robot's nearby region, the initial mapped space $\mathcal{M}_{\text{init}}$. With this initial map $\mathcal{M}_{\text{init}}$, the robot needs to iteratively generate collision-free trajectories to collect occupancy information using the onboard sensor. This process ends until the entire environment is completely explored as the environment map $\mathcal{M}_{\text{env}}$. Note that the final environment map should include all the free and occupied voxels except for some unreachable voxel (i.e., $\mathcal{M}_{\text{env}} = (V_{\text{free}} \cup V_{\text{occ}}) \backslash V_{\text{ur}}$). Due to the dynamic settings of the environment, there exist dynamic obstacles O_{d}. Some obstacles, such as moving tables, change locations infrequently, while some change positions quickly, such as pedestrians. As a result, the robot needs to update information based on these dynamic changes in the environment and avoid obstacles safely.

IV. Methodology

This section introduces the proposed planner for UAVs exploring dynamic environments. There are four main steps in our planner within two modules: the roadmap module and the dynamic module, as shown in Fig. 2. The system input comprises the pose, pointcloud, and depth image data. The main loop performs the goal reach check and replan check based on map and dynamic obstacle information from the map and perception modules. After the main loop initializes a new planning iteration, the roadmap module first applies a vision-based detector to find the heuristic frontier regions (Sec. IV-A) and then incrementally construct

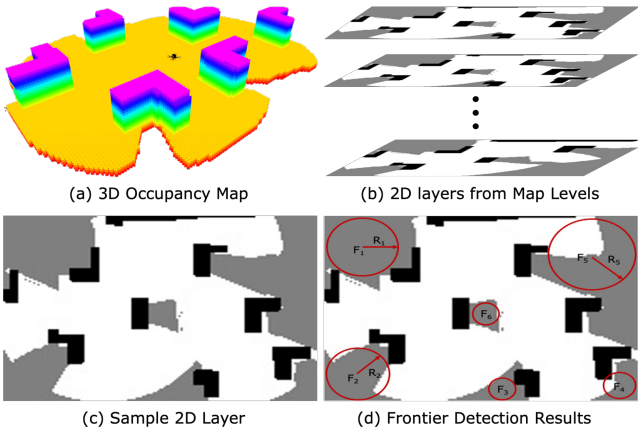


Fig. 3. Illustration of the vision-based heuristic frontier detection method. (a) The 3D occupancy voxel map. (b) The 2D map from different levels of the map. (c) One sample 2D map converted to an image. (d) The frontier detection results by applying a vision-based detector to the 2D map image.

the probabilistic roadmap (Sec. IV-B). Within the dynamic module, the roadmap updates node information gain and prunes invalid nodes based on the map validity check (Sec. IV-C). Finally, the best view path is selected to generate collision-free trajectories for safe exploration (Sec. IV-D).

A. Vision-based Frontier Detection

Heuristic Frontier Region: To avoid complete randomness of node sampling and improve sampling efficiency, we utilize the heuristic frontiers for guiding sampling directions. The classic frontiers are defined as the boundaries separating free and unknown areas. Finding the classic frontiers involves traversing the entire 3D voxel map, which demands high computation. Differently, we define our heuristic frontier $\mathcal{F}$ as a circle with center F and radius R at the level h where the circle should mainly contain the unexplored spaces as shown in Fig. 3d. The heuristic frontier regions $\mathcal{S}_f$ can be defined as a set containing all heuristic frontiers as follows:

$$\mathcal{S}_f = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}, \ \mathcal{F}_i = \mathbb{C}_i(F_i, R_i, h_i), \qquad (1)$$

where $\mathbb{C}$ represents a circle with its height h. In contrast to classic frontiers, our heuristic frontier regions serve as the direction guide for sampling and can be efficiently detected.

Frontier Region Detection: The process of detecting frontier regions is shown in Fig. 3. Given the robot's current 3D occupancy voxel map, we slice the voxel map into multiple 2D maps at different height levels (Fig. 3b). Then, based on the occupancy information from the 2D maps, we convert them into images (Fig. 3c), where grey pixels are the unexplored regions. Finally, we adopt the blob detection algorithm to the images to get the frontier regions. Note that we allow the heuristic frontier regions also to contain free and occupied spaces since they are primarily intended to guide the sampling process, as discussed in Section IV-B.

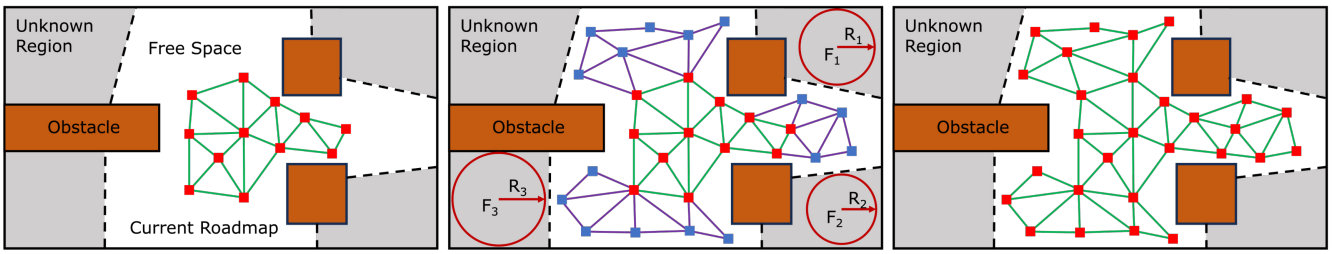
B. Heuristic Incremental Roadmap Construction

After detecting the heuristic frontier regions, we can use them to incrementally construct the probabilistic roadmap in each planning iteration. Our objective is to ensure the nodes in the roadmap can be evenly distributed within the free space and that the newly sampled nodes can make the roadmap grow toward the unexplored regions. The process of the incremental roadmap construction is shown in Fig. 4, and the proposed incremental roadmap construction algorithm is presented in Alg. 1. Within each planning iteration, the algorithm will take the robot's current roadmap, frontier regions, occupancy map, and its current pose as inputs (Lines 1-4). Note that we use the k-d tree data structure for the roadmap to achieve the efficient nearest neighbor search. Initially, we set a heuristic sampling failure number $\mathcal{N}^f_{\text{fail}}$ and draw samples until this value exceeds the threshold $\mathcal{N}_{\max}$ (Lines 5-6). For the roadmap node sampling, we begin by performing weighted sampling of the heuristic frontier n_f in the heuristic frontier regions $\mathcal{S}_f$ and then find its neighborhood N in the roadmap (Lines 7-8). Next, for each neighbor n_i in the neighborhood N, the candidate roadmap node $n_{i,ext}$ is obtained by extending the neighbor n_i toward the heuristic frontier n_f with a user-defined distance δ (Line 12). Finally, the validity check on the candidate node $n_{i,ext}$ will be applied to ensure the node is free and stays within the acceptable range to its nearest neighbor (Lines 11-16). Notably, this validity check can ensure the roadmap nodes are evenly distributed. After this sampling stage, the roadmap can efficiently grow toward the unknown regions.

Algorithm 1: Incremental Roadmap Construction

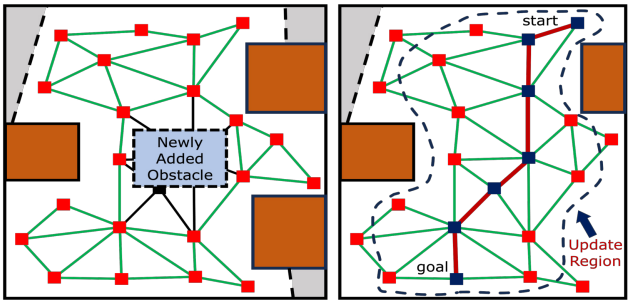
1 $\mathcal{R} \leftarrow$ roadmap ; ▷ robot's current roadmap
2 $\mathcal{S}_f \leftarrow \{\mathcal{F}_1, \mathcal{F}_2..., \mathcal{F}_n\}$; ▷ frontier regions
3 $\mathcal{M} \leftarrow$ occupancy map;
4 $\xi_r \leftarrow$ robot current pose;
5 $\mathcal{N}^f_{\text{fail}} \leftarrow 0$; ▷ sampling failure number
6 **while** $\mathcal{N}^f_{\text{fail}} < \mathcal{N}_{\max}$ **do**
7 $n_f \leftarrow$ **weightedSampleInFrontiers**$(\mathcal{S}_f)$;
8 $N \leftarrow \mathcal{R}$.**kNearestNeighbor**(n_f);
9 **for** n_i **in** N **do**
10 $n_{i,ext} \leftarrow$ **extendNodeToFrontier**(n_i, n_f);
11 **if** $\mathcal{M}$.**isNodeFree**$(n_{i,ext})$ **then**
12 $nn_{i,ext} \leftarrow \mathcal{R}$.**nearestNeighbor**$(n_{i,ext})$;
13 **if** $d_{\min} \leq ||n_{i,ext} - nn_{i,ext}||_2 \leq d_{\max}$ **then**
14 $\mathcal{R}$.**insert**$(n_{i,ext})$; ▷ add node
15 **else**
16 $\mathcal{N}^f_{\text{fail}} \leftarrow \mathcal{N}^f_{\text{fail}} + 1$;
17 $\mathcal{R} \leftarrow$ **localRegionSampling**$(\mathcal{R}, \mathcal{M}, \xi_r)$;
18 $\mathcal{R} \leftarrow$ **globalRegionSampling**$(\mathcal{R}, \mathcal{M}, \xi_r)$;
19 $\mathcal{R}$.**connectNode**$(\mathcal{M})$; ▷ node connection
20 **return** $\mathcal{R}$;

Although the previously mentioned heuristic sampling stage can guide the roadmap's growing direction, we also want to build the roadmap in the rest of the free space for generating high-quality exploration paths. As a result, besides the heuristic sampling stage, the local and global sampling stages are also added (Lines 17-18). For the local sampling stage, we randomly draw sample nodes in the predefined sample region and then perform the validity check



(a) Robot's current roadmap (b) Incremental roadmap construction (c) Robot's updated roadmap

Fig. 4. Heuristic incremental roadmap construction. With the robot's current roadmap shown in (a), the proposed algorithm first determines the heuristic frontier regions and then obtains sample nodes with edges toward the frontier regions depicted in (b). The updated roadmap is visualized in (c).



(a) Update for dynamic obstacles (b) Update for information gain

Fig. 5. Illustration of the node information update. (a) Pruning invalid nodes and edges, represented in black, due to newly added obstacles. (b) Updating nodes' information gain after completing the exploration path.

as Lines 11-16. If the sampled node passes the check, we add the node to the roadmap. Otherwise, we increment the failure count and exit the sampling stage when the failure count exceeds a threshold. For the global sampling stage, we follow a similar manner as the local counterpart but only change the sampling range. After getting all new samples, we perform the node connection operation (Line 19), which builds the collision-free edges to each node's neighborhood.

C. Node Pruning & Information Update

Since the exploration environments can be dynamic, some previously valid nodes and edges from the incrementally built roadmap can become invalid due to the changing position of obstacles. For example, new obstacles can be added to the environment, as depicted in Fig. 5a. As a result, after the incremental roadmap construction process discussed in the previous section, we will go through nodes and edges within the roadmap to check their occupancy information and prune them if they become invalid due to environmental changes.

Different from the traditional probabilistic roadmap, in which each node only contains position and edge information, our roadmap's nodes also contain the information gain for the exploration guide. The information gain function for the ith node in the roadmap in the view direction angle ϕ is:

$$\mathbf{IG}(n_i, \phi) = |\{v | v \in V_{\text{unk}}, v \in \textbf{sensorRange}(n_i, \phi)\}|, \quad (2)$$

where the sensor range function returns the onboard camera's range positioned at node n_i facing at direction angle ϕ, and the information gain counts the unknown voxel number in the sensor range for the node at the view angles. For

Algorithm 2: Path Selection and Traj. Generation

1 $\mathcal{R} \leftarrow$ roadmap ; ▷ `robot's current roadmap`
2 $\mathbb{PQ}_{\text{ig}} \leftarrow$ information gain priority queue;
3 $\mathbf{IG}_{\text{curr}} \leftarrow 0$, $\mathbf{IG}_{\text{max}} \leftarrow \textbf{highestInfoGain}(\mathbb{PQ}_{ig})$;
4 $S_{\sigma,\text{max}} \leftarrow 0, \sigma_{\text{wp}}^{\text{best}} \leftarrow \emptyset$;
5 **while** $\mathbf{IG}_{\text{curr}} > \alpha \cdot \mathbf{IG}_{\text{max}}$ **and** $\mathbb{PQ}_{\text{ig}}.\textbf{size}() \neq 0$ **do**
6 $n_{\text{gc}}, \mathbf{IG}_{\text{curr}} \leftarrow \mathbb{PQ}_{\text{ig}}.\textbf{top}()$;
7 $\sigma_{\text{wp}} \leftarrow \textbf{shortestPathSearch}(n_{\text{gc}}, \mathcal{R})$;
8 $S_\sigma \leftarrow \textbf{evaluatePathScore}(\sigma)$;
9 **if** $S_\sigma > S_{\sigma,\text{max}}$ **then**
10 $S_{\sigma,\text{max}} \leftarrow S_\sigma$;
11 $\sigma_{\text{wp}}^{\text{best}} \leftarrow \sigma_{\text{wp}}$; ▷ `best waypoint path`
12 $\mathbb{PQ}_{\text{ig}}.\textbf{pop}()$;
13 $\sigma_{\text{traj}} \leftarrow \textbf{optimizeTrajectory}(\sigma_{\text{wp}}^{\text{best}})$;
14 **return** σ_{traj};

reducing computation, we discretize the view angles to store the information gain and use interpolation to find the information gain at any angle. With the process of environment exploration, the node information gain needs to be updated. So, in each planning iteration, we collect newly sampled nodes and the nodes that are closed to the robot path to perform information gain evaluation and update as shown in Fig. 5b. After information gain updating, the updated nodes will be inserted into a priority queue $\mathbb{PQ}_{\text{ig}}$ with the higher information gain (at the highest information gain angle) node on the top for path selection. With node pruning and updating, we can ensure the roadmap always has collision-free nodes with updated information gain data.

D. Path Selection & Trajectory Generation

This section introduces the path selection and trajectory generation method (Alg. 2) based on our probabilistic roadmap. After constructing the roadmap with node information update, we can obtain the node order based on its information gain value from the priority queue $\mathbb{PQ}_{\text{ig}}$ (Line 2), with the highest on the top. Then, we traverse through all nodes from the top node to the node value with information gain value less than α times the highest information gain $\mathbf{IG}_{\text{max}}$, where α is a user-defined threshold (Line 5). For each node, we search the shortest path based on the roadmap, then evaluate the path score and find the best waypoint path $\sigma_{\text{wp}}^{\text{best}}$ with the highest score $S_{\sigma,\text{max}}$ (Lines 6-11). The path score

TABLE I

ROBOT PARAMETERS AND SETTINGS.

Max. Linear Vel.	1.0 m/s	Collision Box	[0.5, 0.5, 0.3]m^3
Max Angular Vel.	0.8 rad/s	Camera Range	5 m
Map Resolution	0.1 m	Camera FOV	[86, 57]°

is the information gain rate of the path consisting of each waypoint node at the moving angle to its next waypoint:

$$S_\sigma = \frac{1}{t}\sum_{i=0}^{N} \mathbf{IG}(n_i, \mathbf{atan2}(\frac{n_{i+1}.y - n_i.y}{n_{i+1}.x - n_i.x})),\ n_i \in \sigma_{\text{wp}}, \quad (3)$$

where t is the estimated path execution time, and note that the last node's angle is its highest information gain angle. After getting the best exploration path, we apply the vision-aided B-spline trajectory [26] to navigate waypoints of the paths and avoid dynamic obstacles (Line 13). In some cases, the dynamic obstacles might occupy the path waypoints, resulting in the failure of the trajectory optimization. In this case, the proposed planner will re-select a path for exploration. Since the nodes and edges of the roadmap are guaranteed to be collision-free, our path search does not need to perform the collision checking operation, making the best path replanning extremely fast for collision avoidance.

V. RESULT AND DISCUSSION

To assess the performance of the proposed planner, we conduct a series of experiments, including simulations and physical flight tests in various environments. The algorithm is implemented using C++ and ROS. For the simulation experiments, we utilize Gazebo/ROS and run the algorithm on Intel I7-12700k@3.8GHz. The physical flight tests are conducted using our customized quadcopter, equipped with an Intel RealSense D435i camera and an NVIDIA Orin NX onboard computer. We apply the visual-inertial odometry (VIO) algorithm [27] for robot state estimation and the dynamic obstacle detector [28] for collision avoidance. The parameters and settings for the robot in the simulation experiments are detailed in Table I. For the physical flight tests, we lower the maximum linear velocity limits to 0.5m/s and the maximum angular velocity to 0.5rad/s, respectively.

A. Simulation Experiments

To evaluate the exploration efficiency, we perform the benchmarking experiments against two state-of-the-art exploration planners [16][20] in simulation environments. We prepare three simulation environments of different scales: corridor (small), office (medium), and maze (large), as shown in Fig. 6. From the figure, one can see that our planner is able to fully explore environments of different scales. To compare the exploration performance, we recorded the exploration rate, the trajectory length, and the exploration time of three planners by running each algorithm 10 times in each environment. The comparison of the exploration rates with one-standard-deviation ranges between planners is visualized in Fig. 7. It's evident that our planner consistently maintains the highest exploration rate throughout most of the exploration time across all three environments. Moreover, as the size of the environments increases, our planner's performance surpasses that of the other two planners even more noticeably. Based on our experimental observations, this performance improvement can be attributed to the extensive coverage of the probabilistic roadmap, particularly in the frontier areas due to the heuristic sampling. This comprehensive coverage results in more efficient exploration trajectories. Among the benchmarking algorithms, we noticed a higher occurrence of back-and-forth trajectories during exploration in the larger-sized environments. In addition to the exploration rate, we provide a comparison of trajectory length and exploration time for the three planners required to fully explore the three environments in Table II. As demonstrated in the table, our proposed planner exhibits the shortest trajectory length and the fastest exploration time across all three environments.

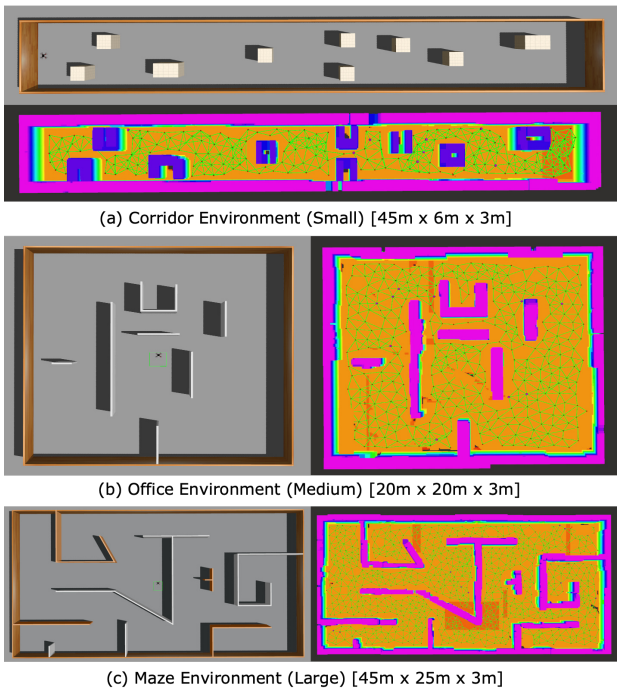


Fig. 6. Visualization of the simulation environments (Corridor, Office, and Maze) with their fully explored map built by the proposed algorithm.

TABLE II

EACH PLANNER'S AVERAGE TRAJECTORY LENGTH AND THE EXPLORATION TIME WITH STANDARD DEVIATIONS IN SIMULATION.

Env. Name	Planner	Traj. Length (m)	Expl. Time (s)
Corridor	GBP2 [16]	265.2 ± 20.2	301.2 ± 14.9
	TARE[20]	233.8 ± 17.8	289.6 ± 23.7
	HIRE	$\mathbf{192.8 \pm 16.5}$	$\mathbf{244.8 \pm 28.5}$
Office	GBP2 [16]	366.7 ± 30.6	486.3 ± 34.6
	TARE [20]	312.3 ± 29.9	406.1 ± 38.6
	HIRE	$\mathbf{231.9 \pm 24.9}$	$\mathbf{310.8 \pm 32.6}$
Maze	GBP2 [16]	1132.2 ± 50.5	1426.8 ± 70.3
	TARE [20]	701.3 ± 35.5	912.8 ± 52.3
	HIRE	$\mathbf{427.4 \pm 40.2}$	$\mathbf{712.3 \pm 46.3}$

B. Physical Flight Test

To test the robustness of our planner, we perform physical flight tests in different dynamic environments, as shown in

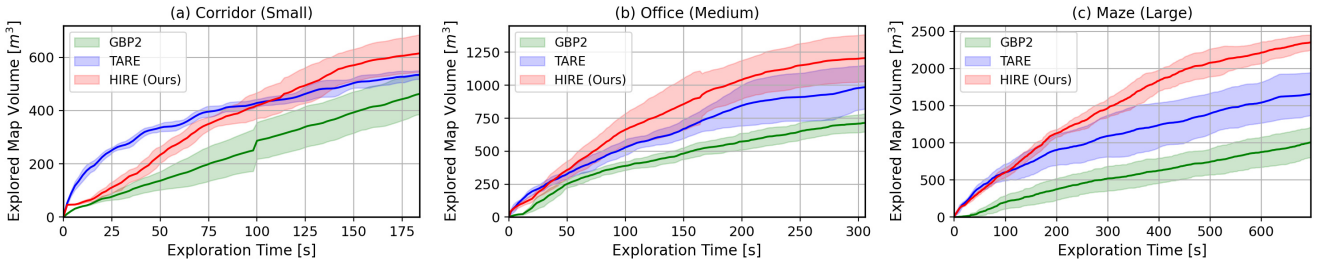


Fig. 7. Comparison of the average exploration rates in simulation environments: Corridor (Small), Office (Medium), and Maze (Large). The shaded areas show the one-standard-deviation ranges of the exploration rates. Our planner achieves the highest exploration rate across three environments.

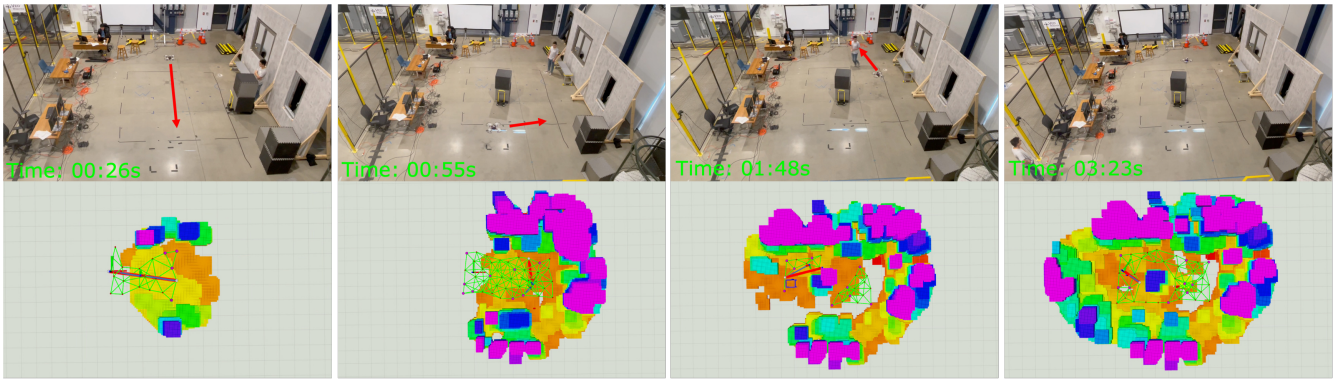


Fig. 8. Illustration of a physical exploration flight experiment in a dynamic environment. The top and bottom figures display the environment and the Rviz visualization of the current mapped area using the incremental probabilistic roadmap, respectively. The environment includes walking pedestrians, and we also modify static obstacle positions while the robot is exploring. The robot can fully explore the environment while safely avoiding obstacles.

TABLE III

THE RUNTIME OF EACH COMPONENT OF THE PROPOSED PLANNER MEASURED USING THE NVIDIA ORIN NX ONBOARD COMPUTER.

Planner Components	Run Time (ms)	Portion (%)
Heuristic Frontier Detection	3.25	4.22%
Incremental Roadmap Construction	25.50	33.12%
Node Pruning & Info. Update	10.50	13.63%
Path Selection & Traj. Generation	37.75	49.03%
Planner Total Time	**77.00**	**100.00%**

Fig. 1 and Fig. 8. One of the test field measures 10m in length, 6m in width, and 3m in height and an illustration of one such physical flight test is presented in Fig. 8. In the figure, it is evident that the explored area, as depicted in Fig. 8 bottom, steadily expands over time, allowing the robot to thoroughly explore the entire environment within 3.5 minutes. During our experiments, we introduced two categories of dynamic obstacles into the environments. The first type involved continuously moving dynamic obstacles, where a human pedestrian walks through the environment, prompting the robot to detect and dynamically replan a collision-free trajectory for collision avoidance, as illustrated in Fig. 8c. The second type of dynamic obstacles entailed alterations to the static environment by adding or removing static obstacles. This adjustment triggered updates and the initiation of node pruning within the explored region. As depicted in Fig. 8a-b and Fig. 8d, we introduced a new obstacle into the center of the environment. The experiment demonstrates the successful execution of both replanning and node pruning in real-world scenarios with dynamic obstacles. The final exploration trajectory length is 32.33m. The average recorded runtime of the proposed planner for the robot onboard computer is presented in Table III. The overall planner runtime is 77ms, operating at a frequency of 10Hz to guarantee real-time performance. Noteworthy is the fact that heuristic frontier detection consumes only 3.25ms, constituting 4.22% of the total planner runtime.

VI. CONCLUSION AND FUTURE WORK

This paper introduces the Heuristic-based Incremental Probabilistic Roadmap Exploration (HIRE) planner designed for UAV exploration in dynamic environments. The proposed algorithm first adopts a computationally efficient vision-based algorithm to detect the heuristic frontier regions from different map layers. Then, it applies the incremental sampling strategy to construct the roadmap for exploration. Additionally, our dynamic module ensures timely updates to roadmap information to accommodate dynamic environments, facilitating the efficiency of exploration and safety of dynamic obstacle avoidance. The simulation experiments show that the proposed method outperforms the state-of-the-art benchmark algorithms' exploration efficiency. Furthermore, our physical flight test demonstrates that our algorithm can make UAVs explore safely in dynamic environments.

VII. ACKNOWLEDGEMENT

The authors would like to thank YKK AP Inc. for their financial support in this work.

REFERENCES

[1] C. Connolly, "The determination of next best views," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 432–435.

[2] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[3] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang, "A two-stage optimized next-view planning framework for 3-d unknown environment exploration, and structural reconstruction," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1680–1687, 2017.

[4] L. Lu, C. Redondo, and P. Campoy, "Optimal frontier-based autonomous exploration in unconstructed environment using rgb-d sensor," *Sensors*, vol. 20, no. 22, p. 6507, 2020.

[5] 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, no. 2, pp. 779–786, 2021.

[6] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[7] C. Witting, M. Fehr, R. Bähnemann, H. Oleynikova, and R. Siegwart, "History-aware autonomous exploration in confined environments using mavs," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[8] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3-d environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[9] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.

[10] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[11] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[12] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2135–2142.

[13] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4568–4575.

[14] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2526–2533.

[15] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3105–3112.

[16] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, R. Siegwart, M. Hutter, and K. Alexis, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3306–3313.

[17] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.

[18] Z. Xu, D. Deng, and K. Shimada, "Autonomous uav exploration of dynamic environments via incremental sampling and probabilistic roadmap," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2729–2736, 2021.

[19] L. Schmid, V. Reijgwart, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, "A unified approach for autonomous volumetric exploration of large scale environments under severe odometry drift," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4504–4511, 2021.

[20] 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, 2021.

[21] H. Zhu, C. Cao, Y. Xia, S. Scherer, J. Zhang, and W. Wang, "Dsvp: Dual-stage viewpoint planner for rapid exploration by dynamic expansion," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7623–7630.

[22] D. Duberg and P. Jensfelt, "Ufoexplorer: Fast and scalable sampling-based exploration with a graph-based planning structure," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2487–2494, 2022.

[23] R. P. de Figueiredo, J. le Fevre Sejersen, J. G. Hansen, M. Brandão, and E. Kayacan, "Real-time volumetric-semantic exploration and mapping: An uncertainty-aware approach," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9064–9070.

[24] M. Dharmadhikari and K. Alexis, "Semantics-aware exploration and inspection path planning," *arXiv preprint arXiv:2303.07236*, 2023.

[25] C. Feng, H. Li, F. Gao, B. Zhou, and S. Shen, "Predrecon: A prediction-boosted planning framework for fast and high-quality autonomous aerial reconstruction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1207–1213.

[26] Z. Xu, Y. Xiu, X. Zhan, B. Chen, and K. Shimada, "Vision-aided uav navigation and dynamic obstacle avoidance using gradient-based b-spline trajectory optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1214–1220.

[27] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[28] Z. Xu, X. Zhan, B. Chen, Y. Xiu, C. Yang, and K. Shimada, "A real-time dynamic obstacle tracking and mapping system for uav navigation and collision avoidance with an rgb-d camera," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 645–10 651.