2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Efficient UAV Exploration with Hybrid Global–Local Strategy and Adaptive Yaw Planning

Yangyang Xue, Xiaotao Liu, Shaojian Zhou, Jingtai Ruan, and Ting Huang

Abstract— **Autonomous exploration in complex environments is frequently hindered by inefficient back-and-forth movements and repetitive revisits to previously explored areas. To address these drawbacks, we propose a two-mode hybrid dynamic exploration strategy that detects isolated frontier clusters and adaptively switches between two modes: global exploration mode (GEM) and local clearance mode (LCM). The GEM generates sequences for frontier exploration access, while the LCM employs a flight-time greedy approach to select and clear isolated clusters, thereby avoiding redundant visits. In addition, to achieve adaptive yaw planning, the proposed exploration strategy generates a reference yaw sequence based on the frontiers near the path trajectory. The reference yaw sequence is then used to perform yaw optimization, with non-uniform B-spline time adjustments ensuring feasible yaw trajectories, fully leveraging the UAV's maneuverability and perception capabilities, and providing a plug-and-play solution for exploration research. Extensive simulations compared to state-of-the-art methods demonstrate that our approach significantly reduces both exploration time and distance, with real-world experiment confirming its practical effectiveness.**

I. Introduction

Autonomous exploration in unknown environments remains a core challenge in robotics research and is widely applied in tasks such as surveying [1], navigation [2], [3], rescue [4], and 3D reconstruction [5]. Unmanned aerial vehicles (UAVs), with their excellent field of view (FOV) [6] and high maneuverability [7], offer significant advantages for exploring complex and partially unknown environments. However, efficiently exploring such environments remains a critical challenge due to the constraints imposed by onboard size, weight, and power limitations.

Although extensive research has been conducted on autonomous exploration [8]–[10], many challenges remain unresolved. Frontier-based methods [11]–[13] and sampling-based methods [14], [15] typically rely on greedy strategies that prioritize local information gain while overlooking global efficiency, often leading to local optima or redundant exploration of already covered areas.

To address this issue, [16] proposed a hybrid strategy that combines local and global exploration to avoid incomplete exploration due to local optima. FUEL [17] introduces a frontier information structure (FIS) and utilizes a variant of the traveling salesman problem to formulate the exploration problem. This approach combines FIS with hierarchical planning to achieve rapid exploration. However, in the environments with dense obstacles, isolated frontier clusters may not be cleared in a timely manner due to occlusions, necessitating repeated visits to previously explored areas, which reduces exploration efficiency.

Furthermore, most methods separate exploration from planning, failing to leverage yaw planning to maximize exploration coverage. While FAEP [18] enhances exploration efficiency via a two-stage yaw planning strategy, it requires partitioning the trajectory into segments to reach an intermediate orientation, necessitating redundant optimization calls. By contrast, our method formulates yaw planning as a single, unified optimization problem by incorporating multiple reference angles as soft path point constraints, effectively maximizing coverage with higher computational efficiency.

To address these issues, we propose a two-mode exploration strategy and an adaptive yaw planning approach grounded on a reference yaw sequence. To enhance exploration efficiency and reduce redundant visits to previously explored areas, we detect isolated frontier clusters during incremental FISs updates and design a hybrid exploration strategy with mode switching, including GEM and LCM. We dynamically switch modes according to the presence of isolated clusters and the UAV's flight status. In GEM, we determine the optimal visiting order for frontier clusters by formulating an Asymmetric Traveling Salesman Problem (ATSP). In LCM, isolated frontier clusters are selected and cleared based on optimal flight time evaluation. The adaptive yaw planning method, developed based on a reference yaw sequence, requires only a single optimization to generate a B-spline yaw trajectory with time adjustment. This approach effectively maximizes environmental perception capability during exploration. Experimental results indicate that our method achieves complete exploration in various complex environments in a shorter time compared to state-of-the-art methods. The main contributions of this work are:

- A hybrid global–local exploration strategy is proposed, which effectively avoids redundant exploration by detecting isolated clusters and switching between GEM and LCM based on a dynamic exploration state machine, thereby improving overall exploration efficiency.
- An adaptive yaw planning method that is grounded on a reference yaw sequence and utilizes non-uniform B-spline time adjustment to ensure feasible yaw trajectories and maximize the coverage of the exploration area.
- Extensive simulations and real-world experiment validate the robustness and efficiency of the method.

The authors are with the Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China. This work is supported by the Natural Science Basic Research Program of Shaanxi (Program No. 2025JC-YBMS-735), and Guangzhou Key Research and Development Program 202206030003. (e-mail: {xue_yangyang, zhou_shaojian, jtruan}@stu.xidian.edu.cn; xtliu@xidian.edu.cn; gnauhgnith@gmail.com). Corresponding author: Xiaotao Liu.

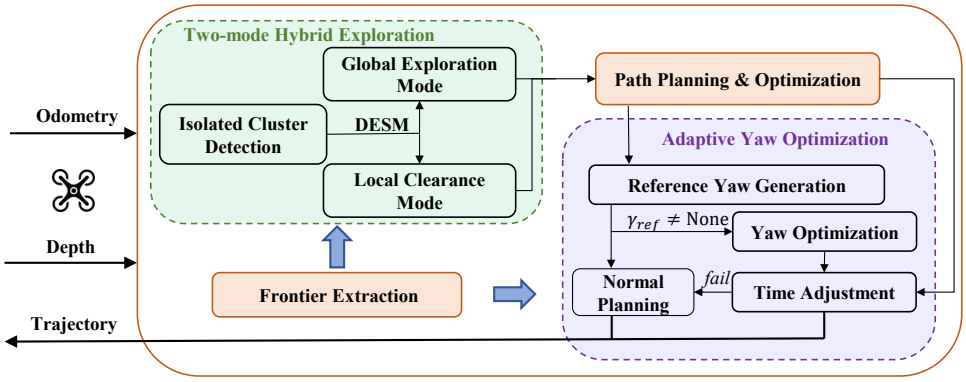


Fig. 1. The pipeline of our proposed method, with our main contribution represented in the dashed boxes.

II. Related Work

In recent years, extensive research has been conducted on autonomous exploration, with strategies primarily categorized into three types: frontier-based, sampling-based, and the hierarchical mixed approach.

Frontier-based exploration strategy was first introduced by [11]. Cieslewski et al. [19] improved efficiency by limiting frontier selection to the FOV and penalizing direction changes to ensure continuous flight. The Receding Horizon Next-Best-View (RH-NBV) planner [14], designed for UAVs, was proposed, which incrementally constructs RRT [20] for viewpoint sampling in 3D maps. Dharmadhikari et al. [21] achieved faster exploration by generating dynamically feasible motion primitives. However, the lack of global consideration in the above and similar methods may result in back-and-forth movements and unnecessarily revisiting explored regions.

Hierarchical mixed approaches also combine the advantages of both frontier-based and sampling-based methods. Selin et al. [16] combines RH-NBVP and frontier exploration, effectively avoiding the issue of RH-NBVP getting stuck locally. Yu et al. [22] generates candidate viewpoints through high-quality 2D Gaussian sampling and selects the best viewpoint using a heuristic function. UFOMap is employed [23] for hierarchical frontier extraction and clustering, while an approximate trajectory optimization strategy is incorporated to achieve efficient exploration. Cao et al. [24] initially plans a coarse path on a global scale and then generates high-quality feasible paths through local sampling of candidate viewpoints. To achieve rapid exploration at high frequencies, an incremental frontier information structure (FIS) is introduced [17], which generates global exploration paths by solving the Asymmetric Traveling Salesman Problem (ATSP) and samples viewpoints around the frontiers to create local path. Furthermore, the exploration strategy is refined by considering the spatial characteristics of frontiers relative to the exploration boundary and incorporating a two-stage yaw planning approach [18]. However, most existing methods treat yaw planning as a decoupled post-processing task or focus solely on the target orientation, failing to actively leverage the UAV's yaw maneuverability to maximize sensor coverage along the trajectory. This limitation leaves significant room for improving exploration efficiency through integrated yaw optimization.

The selection of the next goal is a critical factor influencing exploration efficiency. However, existing methods such as [16], [22], and [23] incorporate the time to reach the goal into their utility functions but assume UAVs always fly at maximum speed with constant velocity. This assumption does not account for dynamic constraints, leading to inefficiencies and frequent redundant exploration. In contrast, our approach not only incorporates flight time into the utility function but also ensures a more accurate calculation of flight time while adhering to dynamic constraints, resulting in a more precise and efficient assessment of exploration performance.

III. Methodology

An overview of the proposed exploration framework is depicted in Fig. 1. The system first updates the voxel grid map and frontiers based on the latest sensor measurements while detecting potential isolated frontier clusters. The UAV dynamically switches exploration modes according to the current frontier information and passes candidate targets to the path planning module. Then, the system performs trajectory planning and adaptive yaw planning, generating feasible trajectories that satisfy dynamic constraints and cover larger exploration areas than conventional methods that focus solely on the target orientation. The trajectories are ultimately executed by the flight control module.

A. Isolated Frontier Clusters Detection

Frontiers are defined as the boundary between known and unknown regions, consisting of known-free voxel cells directly adjacent to unknown voxel cells. These frontiers are grouped into clusters, referred to as frontier clusters, which represent exploration targets. However, in complex environments, the presence of obstacles or occlusions can result in the formation of isolated frontier clusters, as shown in Fig. 2. These clusters are characterized by having very little surrounding unknown space, often due to being surrounded by known areas or obstacles. If isolated clusters are not cleared in a timely manner, it may lead to redundant exploration of known regions. Identifying and distinguishing isolated frontier clusters is crucial for efficient exploration

planning, as it allows the robot to focus on more meaningful exploration targets.

In this work, we present a method for detecting isolated frontier clusters by analyzing the distribution and occupancy of the frontier clusters. An FIS is computed when a new frontier cluster is created. The axis-aligned bounding box (AABB) of the cluster is also computed. Each time the map is updated with sensor measurements, we perform a 2D analysis on each AABB. We select the average position of the cluster C_i on its corresponding plane B_i and expand this plane to generate an isolated detection region B_{iso}. We then evaluate the occupancy ratio of unknown voxels within the rectangular area defined between B_i and B_{iso}. If the occupancy ratio is below a threshold α, the cluster is considered isolated.

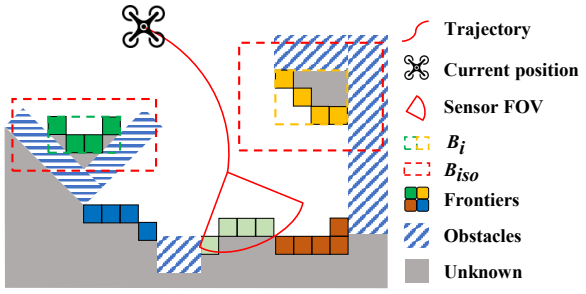


Fig. 2. Illustration of exploration process and detecting isolated clusters.

B. Two-Mode Exploration Based on Isolated Clusters

Based on the presence of isolated clusters and the UAV flight state, we design a dynamic exploration state machine. The UAV starts in GEM. When isolated regions are present within a clearance radius r_{clear}, and if the collision-free path to these isolated regions is shorter than $d_{\max}$, the UAV switches to LCM and continues to clear the isolated regions. After the clearing is completed, the UAV returns to GEM and resumes normal exploration.

Global Exploration Mode: The visiting order of all frontier clusters is determined by constructing an ATSP. We use the same method as in [18] to generate the ATSP cost matrix $\mathbf{C}_{\text{tsp}} \in \mathbb{R}^{(N_{\text{ftr}}+1)\times(N_{\text{ftr}}+1)}$, where N_{ftr} is the number of frontier clusters.

The main part consists of the block matrix $N_{\text{ftr}} \times N_{\text{ftr}}$, which is a symmetric matrix recording the connection costs between frontier clusters. The calculation of these costs is given by

$$\begin{aligned} C_{\text{tsp}}(k_1,k_2) &= C_{\text{tsp}}(k_2,k_1) \\ &= t_{lb}(V_{k_1},V_{k_2}) \\ & k_1,k_2 \in \{1,2,\ldots,N_{\text{ftr}}\} \end{aligned} \tag{1}$$

where $t_{lb}(V_{k_1},V_{k_2})$ represents the time cost from viewpoint V_{k_1} to V_{k_2}, and is calculated as:

$$t_{lb}(V_{k_1},V_{k_2}) = \max\left\{\frac{\text{length}(\mathbf{p}_{k_1},\mathbf{p}_{k_2})}{v_{\max}}, \frac{\min\left(|\xi_{k_1}-\xi_{k_2}|, 2\pi-|\xi_{k_1}-\xi_{k_2}|\right)}{\dot{\xi}_{\max}}\right\}. \tag{2}$$

The viewpoint V_{k_1} contains position $\mathbf{p}_{k_1}$ and yaw ξ_{k_1}. $\text{length}(\mathbf{p}_{k_1},\mathbf{p}_{k_2})$ represents the collision-free path length between the two points, and $v_{\max}$ and $\dot{\xi}_{\max}$ are the maximum linear and angular velocities, respectively.

The connection cost from the current viewpoint $V_0 = (\mathbf{p}_0,\xi_0)$ to the N_{ftr} frontier clusters is

$$C_{\text{tsp}}(0,k) = t_{lb}(V_0,V_k) + w_c\cdot c_c(V_k) + w_b\cdot c_b(k) - w_f\cdot c_s(k) \tag{3}$$

where $c_c(V_k)$ is the direction consistency cost, $c_b(k)$ is the boundary cost, $c_s(k)$ is the probability of the isolated small region, and w_c, w_b, w_f are the corresponding weights.

Local Clearance Mode: A greedy strategy is used to select the next best viewpoint, with the UAV focusing on clearing isolated frontier clusters left during exploration and avoiding unnecessary revisits, thereby enhancing overall exploration efficiency. We define the heuristic utility function $H(V_k)$ for isolated frontier clusters. The UAV evaluates candidate isolated clusters when there is a collision-free path and the distance is less than d_{max}, and selects the isolated clusters with the maximum heuristic utility value as the next target.

We incorporate flight time to more accurately assess the utility function by solving the optimal boundary value problem (OBVP) to compute the optimal flight time. This approach avoids the simplified assumption that the UAV always flies at maximum speed. The utility function is defined as

$$H(V_k) = Cov(V_k)\cdot e^{\lambda T(V_0,V_k)} \tag{4}$$

where $Cov(V_k)$ represents the coverage of the isolated frontier cluster corresponding to viewpoint V_k, and is evaluated as the number of frontier cells that comply with the sensor model and are not occluded by occupied voxels. $T(V_0,V_k)$ represents the time cost for the UAV to travel from the current state V_0 to the isolated frontier cluster viewpoint V_k, and λ is a negative parameter.

Given the search path $\mathcal{P} = \{\mathbf{p}_0,\mathbf{p}_1,\ldots,\mathbf{p}_n\}$ from the current UAV position to the isolated frontier cluster, the velocity vector for each segment is computed as follows. First, the displacement vector $\Delta\mathbf{p}_i = \mathbf{p}_i - \mathbf{p}_{i-1}$ is determined, and its magnitude is given by $\Delta s_i = \|\Delta\mathbf{p}_i\|$. The unit direction vector is then defined as $\hat{\mathbf{d}}_i = \Delta\mathbf{p}_i/\Delta s_i$. Finally, given a constant desired speed v_m, the velocity vector for the UAV along the segment is expressed as $\mathbf{v}_i = v_m\hat{\mathbf{d}}_i$, where $\mathbf{v}_0$ denotes the initial velocity.

For each segment $\mathbf{p}_i\mathbf{p}_{i+1}$ of the path, the optimal trajectory in terms of time and control cost is computed using OBVP, yielding the optimal flight time T_i. The cost of the trajectory is defined as

$$\mathcal{J}(T) = \int_0^T \|\mathbf{u}(t)\|^2 dt + \rho T. \tag{5}$$

The UAV state $\mathbf{x}_i$ consists of position $\mathbf{p}_i$ and velocity $\mathbf{v}_i$. By applying Pontryagin's Minimum Principle [25], the optimal trajectory that minimizes the cost function $\mathcal{J}(T)$

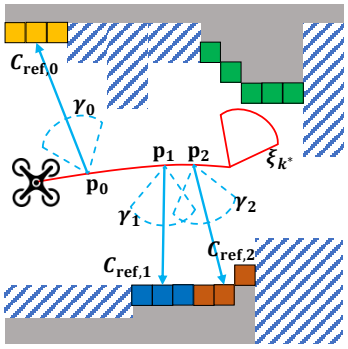


Fig. 3. Illustration of the adaptive yaw planning method using the reference yaw sequence (blue FOV), optimizing the generated yaw trajectory to sequentially approach the reference yaw angles γ_i and explore the visible frontier clusters C_{ref}.

from the current state $\mathbf{x}_i$ to the target state $\mathbf{x}_{i+1}$ is derived as follows:

$$p_\mu^*(t) = \frac{\alpha_\mu}{6}t^3 + \frac{\beta_\mu}{2}t^2 + v_{\mu c}t + p_{\mu c} \tag{6}$$

where $p_\mu^*(t)$ represents the trajectory of the UAV along the μ-th coordinate (with $\mu \in \{x, y, z\}$) over time t, and α_μ and β_μ are the coefficients to be determined.

The coefficients α_μ and β_μ are calculated from the boundary conditions using the following system:

$$\begin{bmatrix}\alpha_\mu \\ \beta_\mu\end{bmatrix} = \frac{1}{T^3}\begin{bmatrix}-12 & 6T \\ 6T & -2T^2\end{bmatrix}\begin{bmatrix}p_{\mu g} - p_{\mu c} - v_{\mu c}T \\ v_{\mu g} - v_{\mu c}\end{bmatrix} \tag{7}$$

where $p_{\mu c}, v_{\mu c}, p_{\mu g}, v_{\mu g}$ represent the current and target positions and velocities in each dimension μ.

The total cost function $\mathcal{J}^*(T)$ is then given by:

$$\mathcal{J}^*(T) = \sum_{\mu\in\{x,y,z\}}\left(\frac{1}{3}\alpha_\mu^2 T^3 + \alpha_\mu\beta_\mu T^2 + \beta_\mu^2 T\right) \tag{8}$$

which represents the sum of the costs over each coordinate dimension.

To determine the optimal traversal time T_i for each segment, we solve for the time that minimizes $\mathcal{J}(T)$ by setting the derivative $\frac{\partial \mathcal{J}^*(T)}{\partial T} = 0$, yielding the optimal time T_i.

Therefore, the total time to traverse the path from $\mathbf{p}_0$ to $\mathbf{p}_n$ is the sum of individual segment times:

$$T(V_0, V_k) = \sum_{i=0}^{n-1} T_i \tag{9}$$

where T_i is the optimal time for the i-th trajectory segment.

C. Reference-Guided Adaptive Yaw Planning

We propose an adaptive yaw planning method based on reference yaw sequences. By constructing soft constraint conditions for reference yaw sequence, our method maximizes the drone's environmental perception capability with a single yaw optimization, as shown in Fig. 3. This approach reduces the number of optimizations and significantly improves the coverage of unexplored surrounding areas, thereby enhancing overall exploration efficiency.

As shown in Alg. 1, initially, we compute all visible frontier clusters C_{ref} within the sensor range on the path

Algorithm 1: Reference-Guided Adaptive Yaw Planning

$C_{\text{ref}} \leftarrow$ **SearchClusters**(C)
$\mathcal{P}_{\text{ref}} \leftarrow$ **NearTrajPos**(C_{ref})
$\gamma_{\text{ref}} \leftarrow$ **CalcRefYaw**$(\mathcal{P}_{\text{ref}}, \text{Centroid}(C_{\text{ref}}))$
if $\gamma_{ref} \neq$ None **then**
  $\mathcal{T}_{\text{ref}} \leftarrow$ **TimeAtReferenceYaw**$(\mathcal{P}_{\text{ref}}, \gamma_{\text{ref}})$
  $\Psi_{\text{ref}} \leftarrow$ **RefCtrlPts**$(\mathcal{T}_{\text{ref}}, \Psi)$
  $Y \leftarrow$ **YawTrajOpt**$(\xi_0, \xi_{k^*}, \Psi_{\text{ref}}, T_{\text{path}})$
  if YawRateLimit*(Y)* **then**
    $T_{\text{adj}} \leftarrow$ **AdjustTime**(T_{path})
    if $\frac{T_{\text{adj}}}{T_{\text{ref}}} \geq r$ **then**
      $Y \leftarrow$ **GeneralYawTrajOpt**$(\xi_0, \xi_{k^*}, T_{\text{path}})$
    return Y
$Y \leftarrow$ **GeneralYawTrajOpt**$(\xi_0, \xi_{k^*}, T_{\text{path}})$
return Y

to the next target viewpoint using **SearchClusters()**. Next, we calculate the corresponding observation positions $\mathcal{P}_{\text{ref}} = [\mathbf{p}_0, \ldots, \mathbf{p}_N]$ of C_{ref} on the trajectory using **NearTrajPos()**, and then determine the reference yaw sequence $\gamma_{\text{ref}} = [\gamma_0, \ldots, \gamma_M]$ by calculating rays from these observation positions to the centroid of C_{ref} using **CalcRefYaw()**.

The yaw trajectory $\psi(t)$ is parameterized as a uniform B-spline curve with control points $\Psi = [\psi_0, \ldots, \psi_N]$ and knot span Δt. If a reference yaw sequence exists, we perform reference yaw planning. Otherwise, we use a standard soft path point constraint without reference yaw. Using **TimeAtReferenceYaw()**, γ_{ref} is mapped to time points $\mathcal{T}_{\text{ref}}$ on the B-spline trajectory. **RefCtrlPts()** matches $\mathcal{T}_{\text{ref}}$ with the original control points Ψ to extract the closest control points, forming the reference control point sequence $\Psi_{\text{ref}} = [\psi_{r,0}, \ldots, \psi_{r,M}]$. **YawTrajOpt()** uses Ψ_{ref} as soft path point constraints to ensure that the B-spline optimized yaw trajectory approximates the target yaw angles γ_{ref}. The objective function of the optimization problem is defined as:

$$J = \lambda_1 J_s + \lambda_2 J_c + \lambda_3(\psi(0) - \xi_0) + \lambda_4(\psi(T) - \xi_{k^*}) + \lambda_5(J_v + J_a) \tag{10}$$

where J_c is the soft path point constraint for reference yaw sequence, and is defined as

$$J_c = \sum_{i=0}^{M}(\psi_{r,i} - \gamma_i)^2 \tag{11}$$

where J_s represents the smoothness of the trajectory. The third and fourth terms are the start and end yaw soft path point constraints. J_v and J_a are used to penalize infeasible velocity and acceleration, respectively. The weight parameters λ_1, λ_2, λ_3, λ_4, and λ_5 correspond to the respective weights of each constraint term. The computation methods for J_s, J_v, and J_a follow the approaches detailed in [26].

Although constraints on velocity and acceleration are imposed in the yaw optimization, the introduction of the optimal reference yaw may expand the range of yaw changes. Consequently, the quadrotor could perform larger rotations

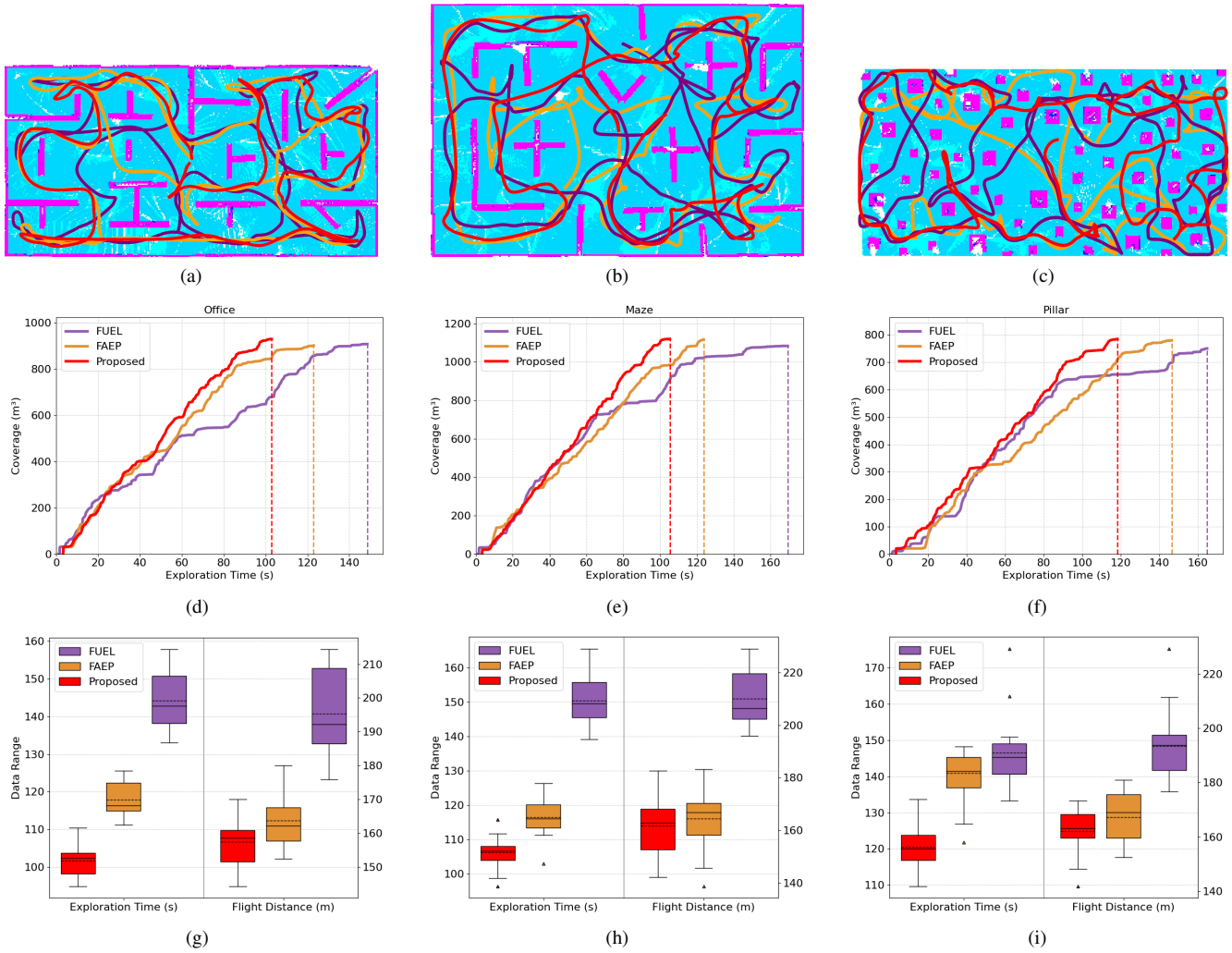


Fig. 4. Exploration performance in various scenarios. (a)–(c) show exploration trajectories for Office, Maze, and Pillar scenarios, where red, orange, and purple paths represent the proposed method, FAEP, and FUEL, respectively. (d)–(f) illustrate exploration progress. (g)–(i) present the distribution of flight time and distance across different scenarios.

within the same time frame, potentially leading to infeasible yaw planning, especially when approaching physical limits. To address this issue, we propose a yaw rate adjustment method based on non-uniform B-splines, dynamically adjusting the time allocation between control points to ensure trajectory feasibility.

Specifically, the first derivative control points $\dot{\Psi}'$ of the non-uniform B-spline represent angular velocity, which can be computed using

$$\dot{\psi}_i' = \frac{p_b(\psi_{i+1} - \psi_i)}{t_{i+p_b+1} - t_{i+1}}. \tag{12}$$

Let $\dot{\psi}_i'$ be an infeasible angular velocity control point with infeasibility satisfying $|\dot{\psi}_i'| = \dot{\xi}_m$. To adjust the infeasible control points, we change the time span to

$$\hat{t}_{i+p_b+1} - \hat{t}_{i+1} = \mu(t_{i+p_b+1} - t_{i+1}). \tag{13}$$

The adjusted angular velocity control point is

$$\hat{\dot{\psi}}_i = \frac{1}{\mu}\dot{\psi}_i'. \tag{14}$$

By setting $\mu = \frac{\dot{\xi}_m}{\dot{\xi}_{\max}}$, the adjusted angular velocity satisfies the feasibility condition

$$|\hat{\dot{\psi}}_i| = \dot{\xi}_{\max} \in [-\dot{\xi}_{\max}, \dot{\xi}_{\max}]. \tag{15}$$

The same method can be applied to adjust angular acceleration.

By iteratively identifying infeasible control points using **YawRateLimit()** and adjusting the corresponding time spans with **AdjustTime()**, we address issues in the yaw planning process. If, after adjustment, $\frac{T_{\text{adj}}}{T_{\text{ref}}} \geq r$ results in the exploration process proceeding at an extremely slow speed, we will revert to using normal yaw planning.

Our method effectively generates trajectories that meet dynamic constraints. Compared to traditional two-stage methods, our approach reduces the number of optimizations and improves coverage efficiency of unexplored areas.

IV. Result and Discussion

To evaluate the performance of the proposed planner, we conducted a series of experiments, including various simu-

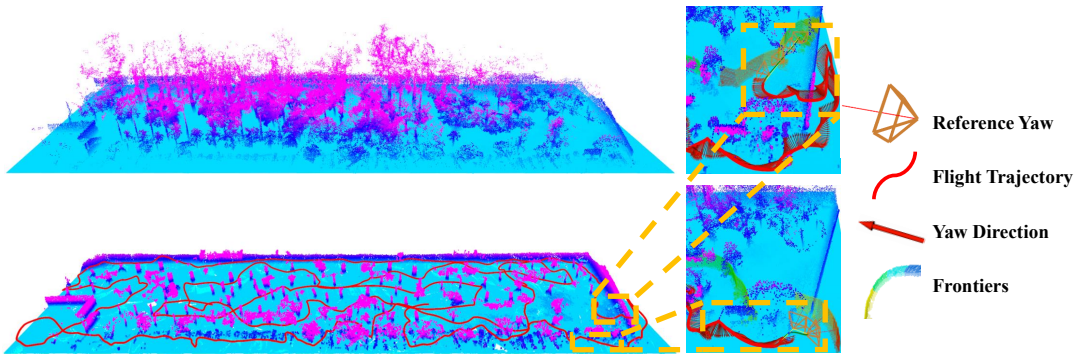


Fig. 5. Our method's exploration of the forest scene is shown. The top-left image presents the forest point cloud map, the bottom-left shows the exploration results with the online-generated volumetric map and flight trajectory, and the right illustrates the Reference-Guided Adaptive Yaw Planning during the process.

lation environments and real-world exploration experiment. Our algorithm was implemented in C++ and ROS and ran on Ubuntu 20.04. For simulation experiments, we use the Lin-Kernighan heuristic (LKH) [27] to solve ATSP in the global planning. For trajectory optimization, a general nonlinear optimization solver NLopt is used to solve the problem. The simulation experiments were performed on a computer equipped with a Core i7-10700 at 2.90GHz, 16GB memory.

TABLE I

PARAMETERS FOR SIMULATION EXPERIMENTS

Max Linear Vel.	2.0 m/s	Max Linear Accel.	3.0 m/s^2
Max Angular Vel.	1.57 rad/s	Max Angular Accel.	1.57 rad/s^2
Sensor FoV	$[80°, 60°]$	Sensing Depth	5 m

A. Simulation Experiments

In the simulation experiments, we benchmarked our method against two state-of-the-art approaches, FAEP [18] and FUEL [17], across three different environments. The exploration is considered complete when all unexplored areas have been covered and no new frontiers are updated. For all methods, we utilized their open-source implementations. In each scenario, we tested each method from five distinct initial positions, with each position spaced more than 5 m apart. Each method was evaluated three times per position, resulting in a total of 15 runs for statistical analysis. We assessed the efficiency of autonomous exploration using metrics such as time consumption and path length. The same configuration was applied across all experiments, as detailed in Table I. We considered three challenging simulation environments with varying complexity and scale:

1) **Office**: As shown in Fig. 4(a), a small-scale indoor scene of $30 \times 16 \times 2$ m^3.
2) **Maze**: As shown in Fig. 4(b), a complex maze scene with dimensions of $30 \times 20 \times 2$ m^3.
3) **Pillar**: As shown in Fig. 4(c), a hand-crafted forest with dimensions of $28 \times 14 \times 2$ m^3.

Experimental results in Fig. 4 and Table II demonstrate that the proposed method is significantly faster than state-of-the-art benchmarks. The superior performance mainly arises from effective isolated cluster detection and clearance, along with the integration of exploration and yaw planning that fully exploits yaw angle dynamics. Across three scenarios, FUEL consistently shows the longest exploration time and flight distance with unstable efficiency. Compared to FAEP, our method reduces exploration time by 13.8%, 8.8%, and 14.5%. In terms of flight distance, our method shortens the distance by 20% compared to FUEL, though the gains over FAEP are marginal. This is mainly because FAEP prioritizes boundary exploration. Once the outermost rings are explored, isolated clusters (even if present) are closer to the center of the map. Consequently, they have minimal impact on the exploration distance, resulting in an exploration length that is slightly longer than ours.

TABLE II

EXPLORATION STATISTICS IN THREE SCENARIOS

Scene	Method	Exploration time (s)				Flight distance (m)			
		Avg	Std	Max	Min	Avg	Std	Max	Min
Office	FUEL [17]	144.2	7.5	157.7	132.9	195.3	12.7	214.3	175.9
	FAEP [18]	117.8	4.7	125.5	111.1	163.7	7.9	180.0	152.3
	Proposed	**101.6**	**4.1**	**110.4**	**94.8**	**157.4**	**6.9**	**170.0**	**144.3**
Maze	FUEL [17]	150.3	7.4	165.4	139.1	210.0	10.7	229.0	195.9
	FAEP [18]	116.4	5.7	126.3	102.9	164.3	11.6	183.2	**138.6**
	Proposed	**106.1**	**4.6**	**115.7**	**96.3**	**161.5**	**10.0**	**182.6**	142.1
Pillar	FUEL [17]	146.5	10.3	175.3	133.3	193.4	13.1	229.4	176.7
	FAEP [18]	140.9	9.9	166.5	121.7	167.1	9.2	180.9	152.3
	Proposed	**120.4**	**6.5**	**133.6**	**109.6**	**162.0**	**8.9**	**173.2**	**141.7**

In the Maze scenario, the proposed method achieves an exploration time that is 8.8% shorter than that of FAEP. The Maze scenario features a structured environment with long, straight corridors on the outer ring and open spaces in the center. In such an environment, the benefits of isolated cluster detection and reference yaw sequence are less pronounced, which results in a slight reduction in the performance of our method. Conversely, in the Pillar scenario, our method demonstrates a significant performance improvement due to the high-density obstacles and numerous occlusion-induced isolated clusters, achieving the shortest exploration times. The adaptive yaw planning consistently maintained a high yaw rate throughout the experiments, effectively leveraging the UAV's maneuverability and perception capabilities, and

providing a robust, plug-and-play solution for future exploration research.

B. Ablation Studies

To evaluate the impact of the proposed modules on exploration performance, we conducted a series of ablation studies for quantitative assessment. The experiments utilize complex large-scale forest point cloud map provided by MARSIM [28], as illustrated in Fig. 5. The exploration area is defined as $95\ \text{m} \times 27\ \text{m} \times 2\ \text{m}$, and each baseline variant is tested in five exploration tasks. When the next target viewpoint is relatively close to the UAV's current position, a large-range yaw adjustment can cover more frontier areas. However, a smaller angular velocity leads to excessively long yaw adjustment time, significantly reducing exploration efficiency. To fully exploit the potential of yaw planning, we adjust the angular velocity limit to 2.35 rad/s, while keeping the other parameters in the experiment consistent with the configuration in Table I.

TABLE III
ABLATION STUDY OF TWO-MODE EXPLORATION

Scene	Method	Exploration Time (s)	Distance (m)	Computation Time (ms)
Forest	NoClearance	445.1 ± 8.2	671.4 ± 14.6	155.4 ± 16.1
	Full	$\mathbf{410.5 \pm 11.1}$	$\mathbf{634.5 \pm 8.3}$	$\mathbf{123.8 \pm 6.2}$

1) Two-Mode Exploration: To verify the effectiveness of the Two-mode hybrid exploration strategy, we conduct a comparative experiment between our complete approach (Full) and a variant, NoClearance, that removes the LCM component described in Section III-B. As shown in Fig. 5 and Table III, the advantage of Two-mode hybrid exploration strategy is significant: the UAV is able to explore the space following a more reasonable frontier exploration sequence, which significantly reduces both exploration time and flight distance. Our strategy avoids redundant visits to already explored areas by considering the remaining isolated frontier clusters during exploration, and it also demonstrates the benefits of dynamically switching between local and global exploration strategies.

In complex large-scale environments, the number of frontier clusters increases as exploration progresses, resulting in a corresponding growth in computational time. The computation time for a single iteration includes frontier detection, isolated frontier detection, viewpoint generation and cost calculation, generation of the global exploration sequence (in GEM), selection of the local best viewpoint (in LCM), and path planning time. Compared to the generation of the global exploration sequence, the selection of the local best viewpoint requires less time. The hybrid strategy significantly reduces the computation time for a single planning iteration while improving exploration efficiency. Furthermore, our method demonstrates robustness and efficiency even in challenging large-scale complex environments.

2) Adaptive Yaw Planning: To validate the effectiveness and contribution of our proposed Reference-Guided Adaptive Yaw Planning in overall exploration efficiency, we compare our complete approach (Full) with two variants. The first variant, Normal, adopts the yaw planning method in FUEL, generating a smooth trajectory from the current yaw to the target yaw. The second variant, Two-Stage, employs the yaw planning method from FAEP to generate an intermediate yaw angle, and then performs a broader yaw adjustment through two stages of the Normal planning method, thereby covering a larger number of frontiers. Except for the yaw planning module, all other modules use our method.

As shown in the Table IV, our proposed method outperforms all other variants across all metrics, demonstrating the effectiveness of our yaw planning approach. Normal does not consider exploration during yaw planning , which limits its performance in complex environments. Two-Stage offers higher exploration efficiency compared to Normal, but it fails to fully exploit the potential of yaw planning. As illustrated in Fig. 5, our method explicitly considers frontier clusters along the planned path and generates the reference yaw sequence. Throughout the exploration process, our approach maintains a high average angular velocity, achieving a good balance between covering more frontiers and improving the exploration speed.

TABLE IV
ABLATION STUDY OF ADAPTIVE YAW PLANNING

Scene	Method	Exploration Time (s)	Distance (m)	Angular Vel. (rad/s)
Forest	Normal	450.1 ± 10.5	695.6 ± 16.2	0.643 ± 0.026
	Two-Stage	439.4 ± 4.2	689.3 ± 10.6	0.764 ± 0.042
	Full	$\mathbf{410.5 \pm 11.1}$	$\mathbf{634.5 \pm 8.3}$	$\mathbf{0.894 \pm 0.028}$

C. Real-World Exploration Experiment

To validate the performance of the proposed method in a real-world scenario, we used our custom-built quadcopter equipped with an Intel RealSense D435i camera, an NVIDIA Orin 16GB onboard computer, and a Pixhawk 6C autopilot. The dynamics limitations of the quadcopter are set to $v_m = 0.5$ m/s, $a_m = 1.0$ m/s^2, $\dot{\xi}_m = 1.05$ rad/s, and $\ddot{\xi}_m = 1.05$ rad/s^2. We applied the visual-inertial state estimator VINS-Fusion [29] for robot state estimation.

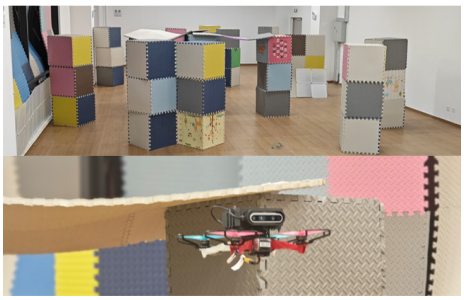
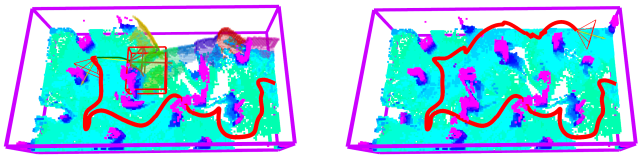


Fig. 6. Real-world exploration experiment scene and our custom UAV.

We conducted a fully autonomous exploration experiment in an indoor environment. Within the experimental area, we randomly deployed obstacles to create a cluttered environment. The scene is an office-like environment with dimensions $15 \times 8 \times 2.5$ m^3, as shown in Fig. 6. The exploration



(a) (b)

Fig. 7. Illustration of real-world exploration experiment. (a) Generation of reference yaw (yellow FOV) and detection of isolated cluster (red box) during the exploration process; (b) Exploration of the entire environment, with the red line representing the exploration trajectory.

task was completed in 82.4 seconds, with a trajectory length of 36.7 meters, as demonstrated in Fig. 7. The proposed planner successfully enabled fast autonomous exploration with the quadcopter, despite the limited sensor FOV. Notably, our exploration planner was capable of promptly detecting and clearing isolated frontier clusters. The yaw trajectory effectively enhanced perception during the exploration process, leading to efficient exploratory motion with minimal duration and flight distance. The executed trajectory required almost no back-and-forth movement or revisits to previously explored areas. Overall, the real-world experimental result demonstrates the effectiveness and robustness of our method in a complex real-world scenario.

V. Conclusion

In this paper, we propose a two-mode exploration strategy that detects isolated frontier clusters during incremental frontier updates and adaptively switches between two modes: GEM and LCM. Our designed adaptive yaw planning method, based on a reference yaw sequence, requires only a single optimization to generate a B-spline yaw trajectory with time adjustment. This approach maximizes yaw variation to cover a broader range of frontiers. In complex environments, where isolated clusters are formed due to occlusions by obstacles, our method significantly improves exploration efficiency. Extensive simulations and real-world experiment demonstrate the effectiveness of our approach.

References

[1] H. Yao and X. Liang, "Autonomous exploration under canopy for forest investigation using lidar and quadrotor," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–19, 2024.

[2] X. Tao, N. Lang, H. Li, and D. Xu, "Path planning in uncertain environment with moving obstacles using warm start cross entropy," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 2, pp. 800–810, 2021.

[3] R. Wang, H. Li, B. Liang, Y. Shi, and D. Xu, "Policy learning for nonlinear model predictive control with application to USVs," *IEEE Trans. Ind. Electron.*, vol. 71, no. 4, pp. 4089–4097, 2024.

[4] M. Wang, B. Xin, M. Jing, and Y. Qu, "A priority-based multi-robot search algorithm for indoor source searching," *IEEE Trans. Autom. Sci. Eng.*, 2025.

[5] C. Feng, H. Li, M. Zhang, X. Chen, B. Zhou, and S. Shen, "FC-Planner: A skeleton-guided planning framework for fast aerial coverage of complex 3D scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2024, pp. 8686–8692.

[6] Q. Yang and H. Li, "RMPC-based visual servoing for trajectory tracking of quadrotor UAVs with visibility constraints," *IEEE/CAA J. Automat. Sinica*, vol. 11, no. 9, pp. 2027–2029, 2024.

[7] H. Rao, L. Xie, J. Yang, Y. Xu, W. Lv, Z. Zheng, Y. Deng, and H. Guo, "Puffin platform: A morphable unmanned aerial/underwater vehicle with eight propellers," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 7621–7630, 2024.

[8] B. Lindqvist, A. Patel, K. Löfgren, and G. Nikolakopoulos, "A tree-based next-best-trajectory method for 3D UAV exploration," *IEEE Trans. Robot.*, vol. 40, pp. 3496–3513, 2024.

[9] X. Liu, M. Cao, G. Lu, Y. Xue, and J. Liu, "A UAV autonomous exploration method based on high-quality viewpoints and infilled guidance," *IEEE/ASME Trans. Mechatron.*, 2025.

[10] X. Zhang, Y. Chu, Y. Liu, X. Zhang, and Y. Zhuang, "A novel informative autonomous exploration strategy with uniform sampling for quadrotors," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13 131–13 140, 2022.

[11] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 1997, pp. 146–151.

[12] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, "A multi-resolution frontier-based planner for autonomous 3D exploration," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4528–4535, 2021.

[13] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an MAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9570–9576.

[14] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "Next-Best-View" planner for 3D exploration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1462–1468.

[15] V. M. Respall, D. Devitt, R. Fedorenko, and A. Klimchik, "Fast sampling-based next-best-view exploration algorithm for a MAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 89–95.

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

[17] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 779–786, 2021.

[18] 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 Trans. Ind. Electron.*, vol. 71, no. 5, pp. 4933–4943, 2024.

[19] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2135–2142.

[20] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[21] M. Dharmadhikari, T. Dang, L. Solanka, J. Loje, H. Nguyen, N. Khedekar, and K. Alexis, "Motion primitives-based path planning for fast and agile exploration using aerial robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 179–185.

[22] J. Yu, H. Shen, J. Xu, and T. Zhang, "ECHO: An efficient heuristic viewpoint determination method on frontier-based autonomous exploration for quadrotors," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 5047–5054, 2023.

[23] H. Zhang, S. Wang, Y. Liu, P. Ji, R. Yu, and T. Chao, "EFP: Efficient frontier-based autonomous UAV exploration strategy for unknown environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2941–2948, 2024.

[24] C. Cao, H. Zhu, H. Choset, and J. Zhang, "TARE: A hierarchical framework for efficiently exploring complex 3D environments." in *Proc. Robot. Sci. Syst.*, vol. 5, 2021.

[25] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1294–1310, 2015.

[26] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3529–3536, 2019.

[27] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *Eur. J. Oper. Res.*, vol. 126, no. 1, pp. 106–130, 2000.

[28] F. Kong, X. Liu, B. Tang, J. Lin, Y. Ren, Y. Cai, F. Zhu, N. Chen, and F. Zhang, "MARSIM: A light-weight point-realistic simulator for LiDAR-Based UAVs," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2954–2961, 2023.

[29] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv:1901.03642*, 2019.