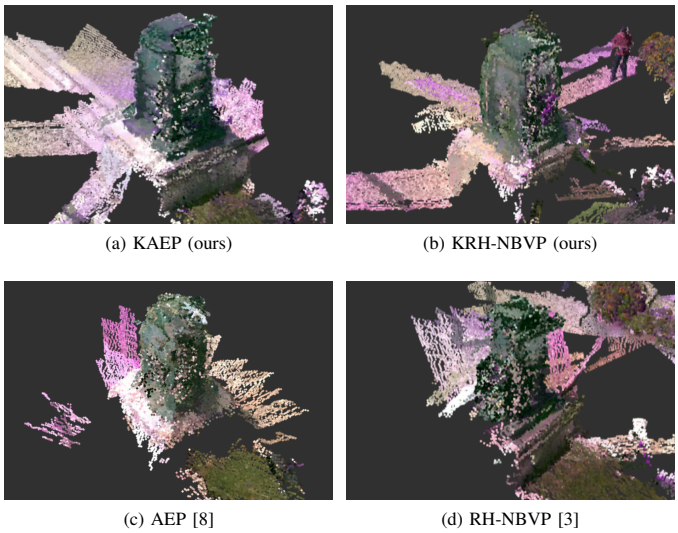
IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Kinodynamic Trajectory Planning for Efficient UAV Exploration and Reconstruction of Unknown Environments

João Félix Mendes, Meysam Basiri, and Rodrigo Ventura

***Abstract*—Autonomous exploration of unknown 3D environments requires motion planners that can efficiently identify informative regions to explore while continuously adapting to the evolving map of the environment. While existing sampling-based methods have demonstrated strong real-time performance, they often ignore the robot's kinodynamic model and constraints. Consequently, they generate only target positions, neglecting kinodynamic considerations in the next-best-view decision process. This results in frequent slowdowns and abrupt maneuvers, reducing coverage speed and exploration efficiency. In this work, we propose a kinodynamic motion planning framework designed for fast and efficient exploration of unknown environments. By incorporating the robot's kinodynamic model and constraints into a kinodynamic RRT, our approach bridges the gap between dynamically feasible motion and effective viewpoint selection, producing smoother and faster trajectories that improve exploration performance. Additionally, we present an Iterative Minimum Gain (IMG) approach to improve global coverage, and a novel informed yaw optimization method that accelerates optimal yaw selection, capable of achieving up to more than twice the speed of state-of-the-art methods. We validate our framework through extensive simulation and real-world experiments, demonstrating improved exploration rates, higher average velocities, and better global coverage over existing methods.**

***Index Terms*—Motion and Path Planning, Autonomous Agents, Aerial Systems: Perception and Autonomy**



(a) KAEP (ours) (b) KRH-NBVP (ours)

(c) AEP [8] (d) RH-NBVP [3]

Fig. 1. Reconstruction of Lamp environment after 40 seconds of running the KAEP (a), KRH-NBVP (b), AEP (c) and RH-NBVP (d).

I. Introduction

OVER the past few years, robots have become increasingly autonomous, showcasing their utility in a variety of industries. Particularly, the ability to autonomously explore unknown environments is critical to a robot's autonomy. This capability has drawn interest in applications such as assessment of dangerous environments like narrow caves [1], search and rescue missions [2], and urban planning [3].

Achieving real-time autonomous exploration in unknown environments poses multiple challenges. Robots operate under constraints such as limited operational time, narrow sensor fields of view, kinodynamic limits, and restricted onboard computation, requiring efficient motion planners. Robots must also continuously update their map and adapt their paths based on incoming sensor data. At the core of this process is the Next-Best-View (NBV) selection problem, which determines where the robot should move next given the current map. By continuously guiding exploration toward unexplored regions, NBV enables efficient path planning as the map is updated.

To address these challenges and solve the NBV problem, many methods rely on sampling-based planners, particularly those built using Rapidly-Exploring Random Trees (RRTs). Following a NBV strategy, they iteratively expand a tree, evaluate each candidate viewpoint using an objective function, and execute the initial segment of the highest-scoring branch. The objective function specifies how candidate viewpoints are evaluated to achieve the optimization goal: maximize expected information gain about unexplored areas while minimizing travel cost. Such strategies have proven effective for exploration [4]–[6], simultaneous mapping and localization [7], and reconstruction of large-scale environments [3], [8], [9].

However, many state-of-the-art sampling-based approaches are prone to local minima, choosing only local solutions due to their limited planning horizon. This can result in suboptimal paths and leave the planner stuck in uninformative regions, failing to achieve global coverage. Also, the approaches that do achieve global coverage ignore the robot's kinodynamic constraints in NBV selection, which can lead them to favor informative viewpoints that are inefficient to reach when considering the robot's kinodynamics, thus slowing exploration.

To overcome these issues, we propose an exploration framework that integrates the robot's kinodynamics directly into planning using Kinodynamic Rapidly-Exploring Random Trees (KRRTs) [10]. Unlike traditional methods, this framework accounts for the robot's velocity and acceleration dur-

Manuscript received: July 31, 2025; Revised October 15, 2025; Accepted November 12, 2025.

This paper was recommended for publication by Editor Giuseppe Loianno upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Aero.Next project and LARSyS FCT funding.

Authors are with Institute for Systems and Robotics, Department of Electrical and Computer Engineering, Instituto Superior Técnico, Portugal joao.felix.mendes@tecnico.ulisboa.pt.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

ing trajectory generation, producing smooth and dynamically feasible trajectories. The NBV selection is formulated as an optimization problem within KRRT: candidate trajectories are evaluated by an objective function that maximizes information gain while minimizing trajectory length and deviations from the robot's maximum velocity, selecting the NBV as the initial segment of the highest-scoring trajectory. By penalizing long trajectories and deviations from maximum velocity, the objective function establishes the link between NBV selection and the robot's kinodynamics, ensuring informative and efficient exploration. Within this framework, we introduce two planners: the Kinodynamic Receding-Horizon Next-Best-View Planner (KRH-NBVP), a local kinodynamic planner, and the Kinodynamic Autonomous Exploration Planner (KAEP), a global kinodynamic planner that ensures full environment coverage. In KAEP, we also propose the Iterative Minimum Gain (IMG) for improved global coverage and an Informed Yaw Optimization (IYO) for improved computational efficiency.

We validate our approach through extensive experiments in realistic simulated environments and real-world outdoor deployments, providing quantitative and qualitative performance evaluations of the proposed methods against state-of-the-art algorithms. Additionally, the impact of various global cost formulations and the effects of the developed IYO and IMG are studied and quantitatively compared. The full framework is released as an open-source package[1] to facilitate integration and further research. The main contributions of this work are:

- Two kinodynamic trajectory planners for efficient exploration of unknown environments that integrate the robot's kinodynamic model and constraints directly into the planning process through a KRRT, outperforming other state-of-the-art methods.
- A novel informed yaw optimization strategy that speeds up gain computation during trajectory evaluation.
- Extensive simulation and real-world outdoor experiments that validate the performance and efficiency of our approach. In these real-world deployments, we implement and directly compare all algorithms, including baselines, providing a comprehensive evaluation of our methods under realistic conditions.

II. RELATED WORK

The exploration problem consists of finding feasible paths that allow a robot to map and explore an unknown environment in real-time. While numerous approaches have been proposed to tackle this problem, most can be classified into two categories: frontier-based and sampling-based planning methods.

The frontier-based approach, first introduced in [11], defines a frontier as the boundary between known free space and unknown space. This method involves continuously moving the robot towards the closest frontier until the full map is explored. This approach was extended in [12] and [13] to tackle some of its limitations. In [12], frontier voxels are clustered to reduce computation, while [13] adapts the method to high-speed exploration by moving the robot towards the frontiers within its perception sensors' current Field of View (FoV) that minimize its change in velocity. Frontier-based methods, however, are limited to exploring boundaries between free and unknown space, ignoring other informative regions.

Sampling-based approaches overcome this by directly sampling candidate viewpoints in free space, also avoiding frontier computation. An example is the Receding Horizon Next-Best-View Planner (RH-NBVP) [3], which samples viewpoints using an RRT and executes the first node of the branch maximizing an objective function. This method was later extended in [7] to handle localization and mapping uncertainty using a two-step process: an RH-NBVP first determines the NBV, and a secondary RRT then generates a path minimizing uncertainty. Another sampling-based method, the Exploration-RRT (ERRT) [6], formulates exploration as a next-best-trajectory problem that integrates control effort into trajectory selection. The ERRT combines a tree-based strategy with a Nonlinear Model Predictive Control (NMPC) framework: it generates robot-safe candidate branches, refines them using NMPC to evaluate control effort and trajectory feasibility, and selects the trajectory maximizing exploration efficiency while minimizing cost. However, these approaches remain prone to local minima and may fail to achieve full coverage.

To escape local minima and ensure global coverage, other sampling-based methods have been proposed. In [4], a history-aware RH-NBVP that tracks previously visited spaces as possible areas for future reseeding was proposed. In [8], Selin et al. introduced the Autonomous Exploration Planner (AEP), combining the RH-NBVP for local exploration with a frontier-based global planner. This global planner caches the sampled nodes from the local planner, and if the robot is in a low-information area, it expands an RRT* [14] from the current position to the closest informative cached node. In [9], an RRT*-inspired method is presented, where a single tree is continuously expanded and maintained to ensure global coverage, using a polynomial trajectory optimization [15] to generate smooth paths to selected viewpoints.

Alternative approaches to frontier and sampling-based methods include using motion primitives and Artificial Potential Fields (APFs). In [16], a primitives-based planner is proposed for Unmanned Aerial Vehicle (UAV) exploration in confined spaces. The planner samples accelerations to generate a fixed set of collision-free trajectory segments, optimized to maximize information gain and minimize trajectory time and deviations from the current direction. However, it is prone to local minima and not asymptotically complete as it uses a fixed set of motion primitives. In [17], an informative sampling method was proposed using APFs to rank candidate viewpoints generated from frontier voxel normals. The top-ranked viewpoints are then evaluated using an objective function. After selecting a viewpoint, an A* is expanded, and a B-spline optimization is used to generate smooth paths.

While methods such as [6] and [16] consider kinodynamic constraints, they lack global coverage guarantees. In contrast, methods that achieve global coverage [4], [9], [17], decouple trajectory optimization from viewpoint selection, ignoring kinodynamic considerations in NBV decisions. Our approach bridges this gap through a KRRT-based framework. In particular, the KAEP combines local and global planning to

[1]https://github.com/IRSg-ARG/UAV_3d_reconstruction

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

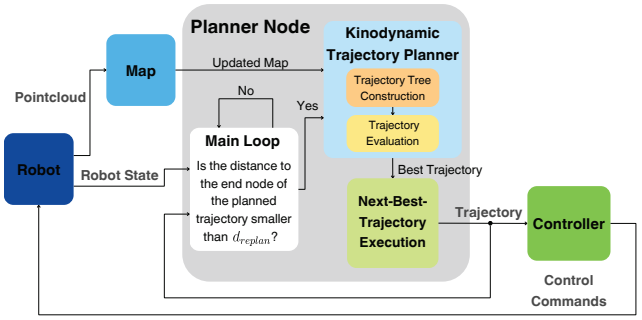


Fig. 2. Overview of the planning framework. A trajectory tree is iteratively expanded and evaluated, and the next best trajectory is executed by the robot's controller. State estimates and map updates are sent to the planner, triggering replanning when the robot is within d_{replan} of the current trajectory's end.

maintain kinodynamic feasibility while ensuring full environment coverage, overcoming the key limitations of prior work.

III. PROPOSED APPROACH

Our approach introduces two exploration planners that generate feasible trajectories using KRRTs [10]. While inspired by RH-NBVP [3] and AEP [8], our method reformulates the framework in several ways. First, instead of expanding a tree of geometric nodes, our planners expand a tree of trajectories, where each trajectory is a time-parametrized sequence of states subject to the robot's kinodynamics. Second, NBV selection is formalized as an objective function over trajectories, combining information gain with two new costs: trajectory length and deviations from maximum velocity. These costs directly couple NBV selection to the kinodynamic model, encouraging faster and smoother motion. Third, the IMG strategy introduced in the KAEP ensures full coverage, gradually relaxing the minimum gain threshold to map low-information regions that would otherwise remain unmapped. Finally, an IYO replaces uniform yaw sampling, reducing gain computation time while maintaining optimality. Figure 2 summarizes the framework. The planners use the robot's perception data to map the environment and build a trajectory tree based on the updated map and robot state. Each trajectory is evaluated using the objective function, and the first segment of the highest-scoring trajectory is executed. Once the robot is within a distance d_{replan} of the trajectory's endpoint, replanning begins.

A. Trajectory Formulation

A trajectory associates a discrete time parametrization with a sequence of geometric poses, where each pose is defined at fixed time intervals. We define each trajectory τ in a tree $\mathbb{G}$ as a sequence of m nodes N_i, where $i = 1, \ldots, m$, with N_1 and N_m representing the start and end of the trajectory, respectively. Each node N_i represents a time-parametrized state vector $S(t_i)$, defined as:

$$N_i = S(t_i) = [p(t_i),\ v(t_i),\ \psi(t_i),\ \dot{\psi}(t_i)], \tag{1}$$

where $p(t_i) = [x(t_i),\ y(t_i),\ z(t_i)]$ is the position, $v(t_i) = [v_x(t_i),\ v_y(t_i),\ v_z(t_i)]$ the linear velocity, $\psi(t_i)$ the yaw, and $\dot{\psi}(t_i)$ the yaw rate at time t_i. Each trajectory is associated with an information gain $\mathrm{g}(\tau)$, two cost terms $c_1(\tau)$ and $c_2(\tau)$, and an objective function $\mathrm{o}(\tau)$. A trajectory τ is established as:

$$\tau = \{(S(t_i))_{i=1}^{m},\ g(\tau),\ c_1(\tau),\ c_2(\tau),\ o(\tau)\}. \tag{2}$$

We define the information gain $\mathrm{g}(\tau)$ as the volume of observable unknown voxels computed from the last node N_m of a trajectory τ. Formally, it is given by:

$$g(\tau) = g(N_m) = \sum_{\mathbf{v} \in \mathrm{Obs}(N_m)} \begin{cases} \mathbf{v}_{\mathrm{size}}^3, & \mathbf{v} \in U, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathbf{v}$ is a voxel, $\mathrm{Obs}(N_m)$ the set of observable voxels from node N_m, $\mathbf{v}_{size}$ the voxel size and U the set of unknown voxels.

We introduce two cost functions: a trajectory-related term that penalizes deviations from the maximum velocity c_1 and a trajectory length term c_2. These are defined as:

$$c_1(\tau) = c_1\left(\mathrm{parent}(\tau)\right) + \left(v_{\max} - \bar{v}(\tau)\right), \tag{4}$$

$$c_2(\tau) = c_2\left(\mathrm{parent}(\tau)\right) + \sum_{n=1}^{m-1} \left\| p(t_{n+1}) - p(t_n) \right\|, \tag{5}$$

where $\mathrm{parent}(\tau)$ is the parent trajectory of τ, $v_{\max}$ the maximum velocity and $\bar{v}(\tau)$ the average velocity of τ.

The objective function $o(\tau)$ merges gain and cost functions to choose informative trajectories that avoid changes from the maximum velocity and high path lengths. It is given by:

$$o(\tau) = o(\mathrm{parent}(\tau)) + g(\tau) e^{-\lambda_1 \cdot c_1(\tau) - \lambda_2 \cdot c_2(\tau)}, \tag{6}$$

where λ_1 and λ_2 are the penalizing coefficients of the costs.

B. Kinodynamic Model and Constraints

A robot's kinodynamic model and constraints are used to generate feasible trajectories. The model describes the evolution of the robot's state over time, while the constraints arise from its physical limits, like maximum velocity and acceleration. Assuming a decoupled approach of linear and angular motion, the kinematic model follows the linear and angular equations under constant acceleration, computed as:

$$v(t_{i+1}) = v(t_i) + a\, dt, \tag{7}$$

$$p(t_{i+1}) = p(t_i) + v(t_i)\, dt + \frac{1}{2} a\, dt^2, \tag{8}$$

$$\dot{\psi}(t_{i+1}) = \dot{\psi}(t_i) + \ddot{\psi}\, dt, \tag{9}$$

$$\psi(t_{i+1}) = \psi(t_i) + \dot{\psi}(t_i)\, dt + \frac{1}{2} \ddot{\psi}\, dt^2, \tag{10}$$

where dt is the time step between sequential nodes N_i and N_{i+1}, a the constant linear acceleration, $\dot{\psi}(t_i)$ the yaw rate at time t_i, and $\ddot{\psi}$ the constant yaw acceleration. Then, the kinematic constraints are given by:

$$|v(t_i)| \leq v_{max}, \tag{11}$$

$$\left|\dot{\psi}(t_i)\right| \leq \dot{\psi}_{max}, \tag{12}$$

where $\dot{\psi}_{max}$ is the maximum yaw rate. The dynamic constraints are defined by:

$$\left|\ddot{\psi}\right| \leq \ddot{\psi}_{max}, \tag{13}$$

$$|a| \leq a_{max}. \tag{14}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1 KRH-NBVP

1: $S_0 \leftarrow$ Current vehicle state
2: Initialize Tree $\mathbb{G}$ with S_0 and previous best trajectory branch, unless it is the first planner iteration
3: $o_{best} \leftarrow 0$
4: $\tau_{best} \leftarrow \tau_0(S_0)$
5: $N_{\mathbb{G}} \leftarrow$ Number of initial trajectories in $\mathbb{G}$
6: **while** $N_{\mathbb{G}} < N_{max}$ **or** $o_{best} \leq 0$ **do**
7: $\quad \xi_{random} \leftarrow$ Sample random pose
8: $\quad \tau_{nearest} \leftarrow$ Nearest(ξ_{random})
9: $\quad u_{control} \leftarrow$ Sample random control input
10: $\quad \tau_{new} \leftarrow$ Simulate$(\tau_{nearest}, u_{control}, \Delta t_{traj})$
11: $\quad N_{\mathbb{G}} \leftarrow N_{\mathbb{G}} + 1$
12: $\quad$ **if** $o(\tau_{new}) > o_{best}$ **then**
13: $\quad\quad \tau_{best} \leftarrow \tau_{new}$
14: $\quad\quad o_{best} \leftarrow o(\tau_{new})$
15: $\quad$ **if** $N_{\mathbb{G}} > N_{termination}$ **then**
16: $\quad\quad$ Terminate Algorithm
17: $\tau_{NBT} \leftarrow$ ExtractNextBestTrajectory(τ_{best})
18: Delete $\mathbb{G}$
19: **return** τ_{NBT}

where $\ddot{\psi}_{max}$ is the maximum yaw acceleration and a_{max} the maximum linear acceleration.

In the yaw motion defined in (9)–(10), quickly aligning the robot with the intended direction is critical for fast exploration. To achieve this, the yaw acceleration is set to its maximum, $\left|\ddot{\psi}\right| = \ddot{\psi}_{\max}$, and the robot moves along the direction with minimal yaw difference to the target orientation.

In our exploration framework, each trajectory is generated as a segment of fixed duration Δt_{traj} propagated using the model in (7)–(10) under the constraints of (11)–(14). Each segment applies a unique linear acceleration $u_{control} = [a_x, a_y, a_z]$ held constant throughout the interval Δt_{traj}, so that the KRRT is composed of multiple trajectory segments with different accelerations, as further detailed in subsection III-C.

C. KRH-NBVP Details

In the KRH-NBVP (Algorithm 1), a KRRT defined as $\mathbb{G}$ is first initialized with the current robot state S_0, represented as a single-node trajectory $\tau_0(S_0)$. A random pose $\xi_{random} = [p_{random}, \psi_{random}]$ is sampled, and the nearest trajectory $\tau_{nearest}$ is identified as the one whose last node is closest to p_{random} (lines 7–8). From the last node of $\tau_{nearest}$, a new trajectory τ_{new} is generated by applying a randomly sampled control input $u_{control} = [a_x, a_y, a_z]$ over a fixed duration Δt_{traj}. The resulting motion is propagated using the kinodynamic model in (7)–(10) under the constraints of (11)–(14) (lines 9–10). If τ_{new} is collision-free, it is added to the tree and its cost, gain, and objective values are computed. The tree expands until the number of trajectories $N_{\mathbb{G}}$ exceeds N_{max} and an informative trajectory with $o_{best} > 0$ is found. Finally, the best trajectory branch τ_{best}, maximizing the objective function, is selected (lines 12–14) and its first trajectory segment τ_{NBT} is executed (line 17). The tree is then deleted and, when replanning starts, reinitialized with the previous best branch. If no informative trajectory is found within N_{max} expansions, the tree continues expanding until either one is found or $N_{termination}$ is reached, after which the algorithm terminates. The KRH-NBVP, like RH-NBVP [3], assigns a random yaw ψ_{random} to each trajectory's end node.

Each trajectory uses a fixed time step Δt_{traj}, and acceleration is chosen as a randomly sampled control input $u_{control} = [a_x, a_y, a_z]$. This ensures probabilistic completeness, which would not occur under a best-control input strategy. Although steering trajectories toward the sampled poses using an optimal acceleration would be ideal, [18] showed that this approach becomes probabilistically incomplete under fixed time steps.

Algorithm 2 KAEP - Global Planner

1: $S_0 \leftarrow$ Current vehicle state
2: Initialize Tree $\mathbb{G}$ with S_0
3: **while** $g_{zero} > 0$ **do**
4: $\quad N_{cached} \leftarrow$ Cached nodes with $g(N_i) > g_{zero}$
5: $\quad$ **if** size$(N_{cached}) > 0$ **then**
6: $\quad\quad$ Expand $\mathbb{G}$ toward N_{cached} using KRRT
7: $\quad\quad \tau_{best} \leftarrow \arg\max_{\tau \in \mathbb{G}} o_{global}(\tau)$
8: $\quad\quad$ **break** ▷ Informative trajectory found
9: $\quad$ **else**
10: $\quad\quad g_{zero} \leftarrow g_{zero}/2$ ▷ Relax threshold (IMG)
11: $\quad$ **if** $g_{zero} = 0$ **then**
12: $\quad\quad$ **Terminate Algorithm** ▷ No informative nodes
13: Delete $\mathbb{G}$
14: **return** τ_{best}

D. KAEP Details

The KAEP introduces a two-level planner composed of a local and a global planner, with yaw optimization applied to all trajectories. It extends KRH-NBVP to avoid getting trapped in local minima and to ensure global coverage, similar to how AEP [8] extends RH-NBVP [3]. The local planner is a modified KRH-NBVP (Algorithm 1) that incorporates three key modifications. First, the end node N_m of each generated trajectory τ_{new} is cached for later use by the global planner. Second, the zero-gain condition (line 6) is replaced with a tunable minimum gain g_{zero}, which defines the threshold separating informative and low-information regions. Third, when no trajectories satisfy $o_{best} > g_{zero}$, the global planner is invoked instead of terminating exploration (lines 15–16).

The global planner (Algorithm 2) is responsible for escaping local minima and ensuring global coverage. It initializes a KRRT, $\mathbb{G}$, with the current vehicle state S_0 (lines 1–2) and retrieves all cached nodes with $g(N_i) > g_{zero}$ (line 4). When such nodes exist, the tree expands toward them until at least one is reached (line 6), and each trajectory is evaluated using the global objective function $o_{global}(\tau)$ (line 7), formulated similarly to (6), which balances information gain and trajectory costs through the weights $\lambda_{1_{global}}$ and $\lambda_{2_{global}}$. The trajectory τ_{best} maximizing $o_{global}(\tau)$ is then executed (lines 7–8, 14).

Finally, to guarantee robustness across environments without manual parameter tuning, the IMG adaptively relaxes the g_{zero} threshold in the global planner when no cached nodes

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 3 Informed Yaw Optimization

1: $g_{best} \leftarrow 0$; $\psi_{best} \leftarrow 0$
2: **for** $i = 0$ to $Y_{min} - 1$ **do**
3: $\quad {}^{i}\psi_{min} = i * H_{min}$
4: $\quad$ **if** $g({}^{i}\psi_{min}) > g_{best}$ **then**
5: $\quad\quad g_{best} \leftarrow g({}^{i}\psi_{min})$
6: $\quad\quad \psi_{best} \leftarrow {}^{i}\psi_{min}$
7: **for** $j = 0$ to $Y_{min} - 1$ **do**
8: $\quad$ **if** $g({}^{j}\psi_{min}) + g({}^{j+1}\psi_{min}) > g(\psi_{best})$ **then**
9: $\quad\quad$ **for** $k = 1$ to Y_{Region} **do**
10: $\quad\quad\quad \psi_{add} \leftarrow {}^{j}\psi_{min} + k * H$
11: $\quad\quad\quad$ **if** $g(\psi_{add}) > g_{best}$ **then**
12: $\quad\quad\quad\quad g_{best} \leftarrow g(\psi_{add})$
13: $\quad\quad\quad\quad \psi_{best} \leftarrow \psi_{add}$
14: **return** g_{best}, ψ_{best}

satisfy $g(N_i) > g_{zero}$. Specifically, g_{zero} is initialized with a conservative value (set in table I) and iteratively halved (lines 3, 9–10) until at least one informative node is found or g_{zero} reaches zero, which ends exploration (lines 11–12). This process allows the planner to gradually map lower-information regions that would otherwise remain unmapped, ensuring full coverage.

E. Informed Yaw Optimization

The planners in [4] and [8] introduce a uniform yaw optimization strategy, sampling yaw angles uniformly at each viewpoint with a step size of $H = \frac{2\pi}{Y_{samples}}\text{rad}$, where $Y_{samples}$ is the number of samples. For each sampled yaw, the gain is computed, and the yaw with the highest gain is selected. However, as shown in [3], [8], [9], most computation time is spent evaluating the gain, making this approach inefficient, as significant computation is wasted evaluating yaws with small gains compared to more promising ones at the same position.

We propose the IYO method (Algorithm 3) to improve computational efficiency. First, the gain is evaluated for a minimum number of yaw angles, Y_{min}, chosen so that the combined horizontal FoV, FoV_h, of the perception sensor covers at least 2π rad, i.e., $Y_{min} = \text{ceil}(\frac{2\pi}{\text{FoV}_h})$. The corresponding angles are spaced by $H_{min} = \frac{2\pi}{Y_{min}}\text{rad}$, forming the set $\{{}^{i}\psi_{min}\}_{i=0}^{Y_{min}-1}$. Then, the angle ψ_{best} maximizing the gain g_{best} is selected (lines 4–6). Next, for each pair of adjacent minimum yaw angles, their combined gain is compared to g_{best}. If it exceeds g_{best}, the region between them is selected for further optimization (lines 7–8). Within each selected region, $Y_{Region} = \frac{(Y_{samples} - Y_{min})}{Y_{min}}$ additional angles are evaluated with a step size of H, updating the best gain and yaw accordingly (lines 9–13). When the combined gain of two adjacent minimum yaw angles does not surpass g_{best}, the angles between them are skipped, thus reducing unnecessary evaluations. Note that this method assumes $Y_{samples}$ is a multiple of Y_{min} so that Y_{Region} is a positive integer.

IV. SIMULATED EXPERIMENTAL EVALUATION

The MRS UAV system [19] and its built-in Gazebo simulated environments within the Robot Operating System (ROS) were used as a simulation framework. The DJI F450 UAV equipped with an Intel RealSense D435i depth camera was utilized in simulation, using Voxblox [20] for mapping and the MRS built-in RTK-GPS estimator for pose estimation. Trajectories were executed by the MRS Model Predictive Controller (MPC), which tracked the planned trajectories. The ESDF layer in Voxblox was used for collision checking, ensuring all trajectory nodes maintained a minimum distance $d_{collision}$ from the nearest surface. Trajectories that fail this criterion are marked non-traversable. Table I shows all parameters chosen for the simulation. The parameters were selected based on prior UAV exploration works [3], [8], [9] and refined through preliminary simulations to ensure efficient exploration. Common parameters were kept identical for both the proposed and baseline algorithms to enable a fair comparison. Parameters were consistent across environments, except in the maze, where tighter obstacle spacing required some adjustments: $d_{collision}$ =0.7 m, d_{replan} =0.4 m, and Δt_{traj} =1 s.

TABLE I
SIMULATED EXPERIMENT PARAMETERS.

Parameter	Value	Parameter	Value
$v_{xy_{max}}, v_{z_{max}}$	$1\,\text{m}\,\text{s}^{-1}$	FoV	[87°, 58°]
$\dot{\psi}_{max}$	$2\,\text{rad}\,\text{s}^{-1}$	Camera range	5 m
$a_{xy_{max}}, a_{z_{max}}$	$1\,\text{m}\,\text{s}^{-2}$	Camera pitch	10°
$\ddot{\psi}_{max}$	$2\,\text{rad}\,\text{s}^{-2}$	$Y_{samples}$	15
N_{max}	50	$N_{termination}$	300
$\mathbf{v}_{size}$	0.2 m	g_{zero}	$5\,\text{m}^3$
λ_1	0.2	$\lambda_{1_{global}}$	0.02
λ_2	0.5	$\lambda_{2_{global}}$	0.05
dt	0.1 s	Δt_{traj}	2.0 s
d_{replan}	0.8 m	$d_{collision}$	1.5 m

We compare the performance of our methods with the RH-NBVP [3] and the AEP [8] in the reconstruction of three simulated scenarios: a maze ($20\,\text{m} \times 18\,\text{m} \times 3\,\text{m}$), a police station ($20\,\text{m} \times 20\,\text{m} \times 15\,\text{m}$), and a school ($50\,\text{m} \times 35\,\text{m} \times 20\,\text{m}$). The environments are available in our repository for benchmarking.

The performance of the algorithms is evaluated using four independent metrics: exploration rate, which quantifies how quickly the environment is reconstructed over time; Full Exploration (FE), which measures the overall completeness of the 3D reconstruction at the end of each experiment; Path Length (PL), that indicates the distance traveled by the UAV; and Average Velocity (AV) that reflects how efficiently the UAV utilizes its physical capabilities during exploration. The exploration rate is reported at three progress thresholds: Exploration at 25 %, 50 %, and 95 % completion (denoted as E25 %, E50 %, E95 %). Each algorithm was run five times, recording the mean and standard deviation for all metrics.

A. Efficiency of Informed Yaw Optimization

The computational efficiency of the proposed IYO was evaluated in the school environment using KAEP, comparing the average gain computation time per node against the uniform yaw optimization. Four yaw sample sizes were tested: 15, 120, 360, and 720 samples. Table II summarizes the results. The informed method consistently achieves lower computation times,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II
IMPACT OF DIFFERENT YAW OPTIMIZATION METHODS.

Yaw samples	Uniform [s]	Informed [s]	Speedup [×]
15	0.0077	0.0050	1.53
120	0.044	0.023	1.89
360	0.150	0.075	2.02
720	0.320	0.150	2.10

TABLE III
IMPACT OF DIFFERENT GLOBAL COST FUNCTIONS.

Planner	Metric	Cost	Gain	Objective
AEP	E25% [min]	3.77 ± 0.48	$\mathbf{3.17 \pm 0.25}$	3.25 ± 0.49
	E50% [min]	7.20 ± 0.86	$\mathbf{7.00 \pm 0.42}$	7.04 ± 0.13
	E95% [min]	20.85 ± 0.74	20.67 ± 1.67	$\mathbf{19.63 \pm 0.68}$
KAEP	E25% [min]	$\mathbf{2.32 \pm 0.44}$	2.77 ± 0.50	2.40 ± 0.28
	E50% [min]	5.59 ± 0.34	6.06 ± 0.47	$\mathbf{4.69 \pm 0.44}$
	E95% [min]	19.54 ± 0.82	16.12 ± 0.68	$\mathbf{12.32 \pm 0.31}$

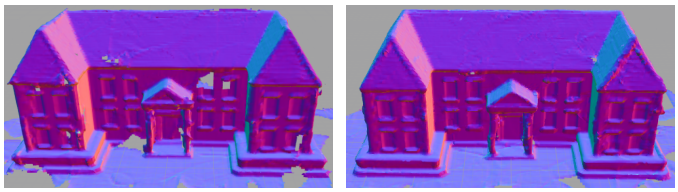
achieving up to a $2.1\times$ speedup over the uniform method while producing identical optimal yaw angles. Also, the informed method shows a trend of increased efficiency with bigger yaw sample sizes. However, the rate of improvement diminishes at larger sample sizes, i.e., although using more yaw samples continues to enhance the efficiency, the incremental gains gradually plateau.

B. Global Planner Performance in Large Scale Environments

Unlike KAEP, which uses an objective function in its global planner, AEP [8] originally uses a global cost function. To assess the impact of different global planning strategies on exploration performance, we evaluate the use of cost, gain, and objective functions in the global planner of AEP and KAEP. The results in Table III show that using a global planner with an objective function formulation improves the exploration rate for both algorithms. During the early stages of exploration (up to $50\,\%$), most planning is handled locally, given that there is a significant percentage of unexplored environment, resulting in similar performance across formulations. However, as the unexplored space shrinks, the global planner is more frequently used, leading to bigger performance gaps. Accordingly, the efficiency of the objective function is shown in the exploration of $95\,\%$ of the environment, where exploration is achieved within 13 and 20 minutes of simulation for the KAEP and AEP, respectively, and, at least, over one minute faster than planners with the gain formulation. In contrast, the cost formulation performs the worst, as it prioritizes low-cost areas even when they have limited information gain.

C. Completeness of IMG

To analyze the effects of the IMG, we compare KAEP performance with and without it, evaluating the full exploration percentage and path length in the reconstruction of the school environment. Without IMG, the algorithm reaches $99.08 \pm 0.28\,\%$ coverage with a path length of $1174.12 \pm 80.54\,\mathrm{m}$. With IMG, the coverage increases to $99.83 \pm 0.10\,\%$ and the path length to $1833.00 \pm 39.71\,\mathrm{m}$. These results indicate that IMG improves map completeness but increases path length by $56\,\%$, also resulting in longer algorithm termination times. Since the IMG is designed to complete reconstructions by filling in small, unexplored gaps in the map, its use should be carefully weighed against the cost of extended flight time. Also, although the quantitative improvement in full exploration is small, Fig. 3 shows that reconstruction is visibly more complete when using the IMG.



(a) Reconstruction without IMG (b) Reconstruction with IMG

Fig. 3. Comparison of reconstructed school surface without (a) and with (b) IMG activated in the KAEP.

D. Simulated Experiments

The obtained results are shown in Figs. 4 and 5 and Table IV. Across all environments, the proposed KAEP leverages its global planner and efficient kinodynamic planning to consistently achieve the fastest exploration rates, reaching $95\,\%$ exploration significantly earlier than all other algorithms. The kinodynamic considerations also allow it to achieve the highest average velocities, while the IMG ensures higher full exploration percentages at the cost of longer paths.

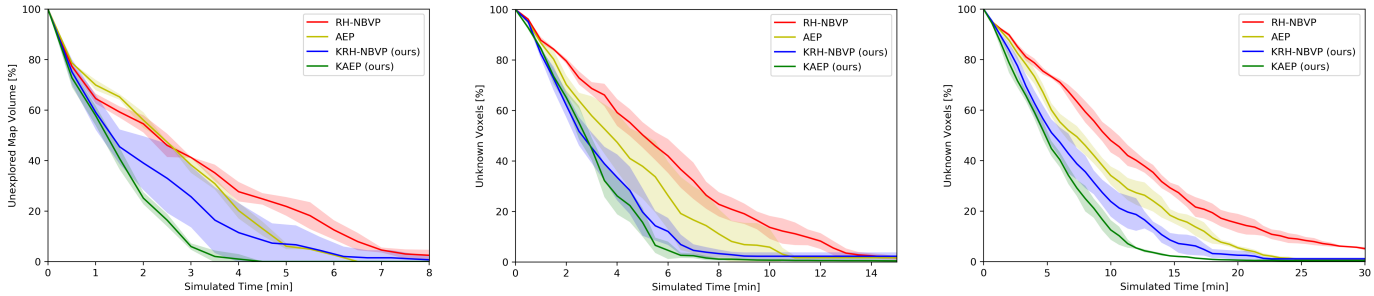
The KRH-NBVP outperforms both RH-NBVP and AEP in exploration rate and average velocity, particularly in open environments such as the police station and school, where its local planning quickly covers the large surrounding free spaces. In the police station, it even surpasses KAEP in the initial $50\,\%$ exploration. However, in more constrained scenarios like the maze, it remains more susceptible to local minima, which slows down exploration. Meanwhile, the RH-NBVP and AEP have lower velocities and exploration rates overall, confirming the benefit of incorporating kinodynamic considerations directly into the planning process.

Finally, the qualitative 3D reconstructions in Fig. 5 visually reinforce these trends, highlighting KAEP's better exploration rate, followed by the KRH-NBVP, AEP and RH-NBVP.

V. REAL WORLD EXPERIMENTAL EVALUATION

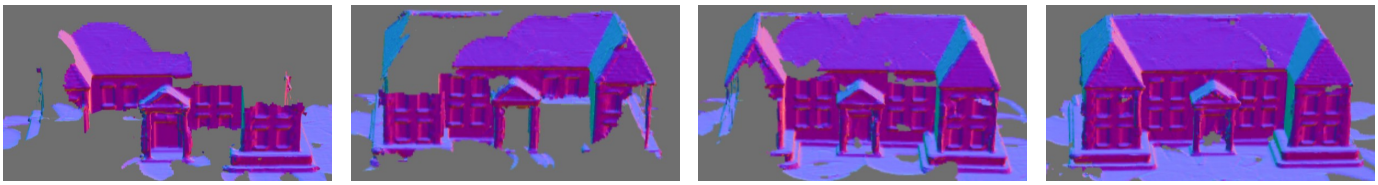
Our proposed methods, along with the baseline approaches, were tested and deployed outdoors using a DJI f550 UAV (Fig. 6a) to reconstruct an outdoor lamp structure (Fig. 6b). The UAV was equipped with an Intel RealSense Depth Camera D455, using voxblox [20] for real-time mapping. A Pixhawk flight controller running ArduPilot provided low-level control, while state estimation was performed onboard by ArduPilot through sensor fusion of GPS (from a DroneCAN H-RTK F9P Rover RTK-GPS module) and IMU data. An Intel NUC (7th Gen, featuring an Intel Core i7-7567U processor and 16GB RAM) was mounted onboard as the companion computer. It

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) Maze Exploration Rate (b) Police Station Exploration Rate (c) School Exploration Rate

Fig. 4. Comparison of exploration rate obtained by the four evaluated algorithms in maze (a), police station (b) and school (c) environments.



(a) RH-NBVP [3] (b) AEP [8] (c) KRH-NBVP (ours) (d) KAEP (ours)

Fig. 5. Reconstruction of school environment after 15 minutes with RH-NBVP (a), AEP (b), KRH-NBVP (c) and KAEP (d).

TABLE IV
SIMULATED EXPERIMENT RESULTS.

Environment	**Planner**	**E25%** [min]	**E50%** [min]	**E95%** [min]	**FE%** [%]	**PL** [m]	**AV** [$\mathrm{m\,s^{-1}}$]
Maze	RH-NBVP	0.60 ± 0.05	2.27 ± 0.27	6.94 ± 0.50	97.52 ± 2.20	$\mathbf{198.30 \pm 5.12}$	0.31 ± 0.01
	AEP	0.71 ± 0.06	2.33 ± 0.09	5.50 ± 0.03	$\mathbf{100.00 \pm 0.00}$	270.63 ± 7.70	0.43 ± 0.02
	KRH-NBVP	0.51 ± 0.18	1.34 ± 0.60	5.58 ± 2.16	99.48 ± 1.04	251.30 ± 86.46	0.55 ± 0.18
	KAEP	$\mathbf{0.46 \pm 0.09}$	$\mathbf{1.23 \pm 0.12}$	$\mathbf{3.13 \pm 0.27}$	$\mathbf{100.00 \pm 0.00}$	375.72 ± 76.64	$\mathbf{0.71 \pm 0.10}$
Police Station	RH-NBVP	2.34 ± 0.22	5.04 ± 0.73	12.64 ± 0.61	94.86 ± 0.01	495.65 ± 6.02	0.53 ± 0.10
	AEP	1.82 ± 0.19	3.93 ± 1.46	10.43 ± 3.65	98.21 ± 0.01	$\mathbf{458.41 \pm 30.18}$	0.61 ± 0.04
	KRH-NBVP	$\mathbf{1.38 \pm 0.23}$	$\mathbf{2.64 \pm 0.48}$	6.91 ± 1.03	97.74 ± 0.02	476.74 ± 40.04	0.83 ± 0.03
	KAEP	1.44 ± 0.16	2.75 ± 0.30	$\mathbf{5.91 \pm 0.51}$	$\mathbf{99.55 \pm 0.36}$	728.44 ± 160.38	$\mathbf{0.89 \pm 0.08}$
School	RH-NBVP	4.81 ± 0.34	9.63 ± 1.00	-	94.91 ± 0.66	$\mathbf{1098.03 \pm 13.01}$	0.54 ± 0.01
	AEP	3.77 ± 0.48	7.20 ± 0.86	20.85 ± 0.74	99.22 ± 0.26	1140.69 ± 22.45	0.62 ± 0.01
	KRH-NBVP	2.88 ± 0.37	5.53 ± 1.22	17.24 ± 2.35	98.90 ± 0.54	1322.51 ± 41.11	0.93 ± 0.01
	KAEP	$\mathbf{2.40 \pm 0.28}$	$\mathbf{4.69 \pm 0.44}$	$\mathbf{12.32 \pm 0.31}$	$\mathbf{99.83 \pm 0.10}$	1833.00 ± 39.71	$\mathbf{0.97 \pm 0.02}$

handled all high-level tasks, including planning, mapping, and control logic, entirely onboard and in real time. Each algorithm was repeated three times for consistency in a $5\,\mathrm{m} \times 4\,\mathrm{m} \times 6\,\mathrm{m}$ environment. Most parameters used matched those from simulation (Table I), with minor adjustments for real-world safety and environmental considerations: maximum linear velocities were reduced to $v_{xy_{max}} = v_{z_{max}} = 0.5\,\mathrm{m\,s^{-1}}$, minimum gain to $g_{zero} = 1\,\mathrm{m}^3$, and tree expansion limits to $N_{max} = 20$ and $N_{termination} = 200$. Note that the IMG was disabled for KAEP due to the relatively smaller environment and lower g_{zero}. Exploration results are shown in Fig. 6c.

In the lamp environment, KAEP achieved the fastest exploration, reaching $50\,\%$ and $95\,\%$ exploration in $8.88 \pm 0.42\,\mathrm{s}$ and $36.77 \pm 2.48\,\mathrm{s}$, respectively. Although KRH-NBVP was slower than AEP at $50\,\%$ exploration, it surpassed it at $95\,\%$ ($47.95 \pm 24.39\,\mathrm{s}$ compared to $50.68 \pm 9.67\,\mathrm{s}$). RH-NBVP was the slowest, needing $89.27 \pm 23.01\,\mathrm{s}$ to reach $95\,\%$ exploration. The larger standard deviations in RH-NBVP and KRH-NBVP reflect their tendency to spend time exploring local low-information areas, delaying the progress in some runs.

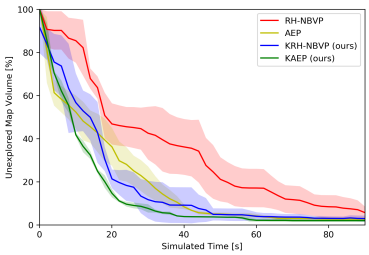
Average velocities followed the simulation trend: RH-NBVP and AEP were slower ($0.46\,\mathrm{m\,s^{-1}}$ and $0.43\,\mathrm{m\,s^{-1}}$), while KRH-NBVP and KAEP reached higher velocities ($0.61\,\mathrm{m\,s^{-1}}$ and $0.59\,\mathrm{m\,s^{-1}}$). Path lengths, however, differed from simulation: AEP has the shortest paths ($49.82 \pm 7.82\,\mathrm{m}$), followed by KAEP ($61.53 \pm 18.07\,\mathrm{m}$), whereas the RH-NBVP and KRH-NBVP traveled farther ($158.23 \pm 90.84\,\mathrm{m}$ and $120.64 \pm 28.39\,\mathrm{m}$). This difference arises from the small environment: once most of it is explored, the remaining unmapped regions are small and fall below the g_{zero} threshold, causing AEP and KAEP to stop earlier, while RH-NBVP and KRH-NBVP continue searching these minor gaps.

Figure 1 shows the post-processed reconstruction of the lamp using RTAB-Map [21] after $40\,\mathrm{s}$ of exploration. The figure highlights the two visible lamp faces, which were the most challenging to reconstruct. KAEP, KRH-NBVP, and AEP

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

(a) Real-world UAV

(b) Lamp Environment



(c) Lamp Exploration Rate

Fig. 6. UAV used for real-world deployment (a), real-world lamp environment (b), and corresponding exploration rate achieved by the evaluated algorithms during deployment (c).

all reconstructed the two non-visible faces, but KAEP achieved the most complete result, mapping both visible faces. KRH-NBVP and AEP each fully reconstructed the left visible face and partially reconstructed the right. RH-NBVP performed the worst, mapping only the right visible face while leaving the left and its adjacent non-visible face unmapped, reconstructing only half the structure. These qualitative results align with the quantitative findings, where KAEP achieved the fastest and most complete exploration, followed by the KRH-NBVP and AEP, with RH-NBVP performing the slowest.

VI. CONCLUSION

In this paper, we presented a trajectory planning framework for exploration with two novel methods that integrate kinodynamic models and constraints, allowing for higher velocities and faster exploration of unknown environments. We also introduced an IMG strategy for improved coverage and a novel informed yaw optimization method that achieved up to $2.10\times$ faster computation than state-of-the-art methods. The proposed algorithms were tested in simulated environments and real-world deployments, validating and showcasing the efficiency of the developed algorithms. For future work, we propose integrating a KRRT* into KAEP's global planner, as the algorithm spends valuable time following long, geometrically sub-optimal trajectories. Moreover, the kinodynamic framework provides the foundation to extend this work toward energy-aware planning by incorporating energy or control-aware cost terms, allowing a deeper analysis of energy efficiency and the trade-off between exploration speed and power consumption.

REFERENCES

[1] P. Petráček, V. Krátký, M. Petrlík, T. Báča, R. Kratochvíl, and M. Saska, "Large-scale exploration of cave environments by unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 2021.

[2] T. Dang, F. Mascarich, S. Khattak, H. Nguyen, H. Nguyen, S. Hirsh, R. Reinhart, C. Papachristos, and K. Alexis, "Autonomous search for underground mine rescue using aerial robots," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–8.

[3] 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.

[4] 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)*, 2018, pp. 1–9.

[5] 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)*, 2018, pp. 2526–2533.

[6] B. Lindqvist, A. Patel, K. Löfgren, and G. Nikolakopoulos, "A tree-based next-best-trajectory method for 3-d uav exploration," *IEEE Transactions on Robotics*, vol. 40, pp. 3496–3513, 2024.

[7] 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)*, 2017, pp. 4568–4575.

[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, Apr. 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, Apr. 2020.

[10] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[11] 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'*, 1997, pp. 146–151.

[12] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an mav," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9570–9576.

[13] 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.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[15] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Springer, 2016, pp. 649–666.

[16] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 179–185.

[17] Y. Kompis, L. Bartolomei, R. Mascaro, L. Teixeira, and M. Chli, "Informed sampling exploration path planner for 3d reconstruction of large scenes," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7893–7900, 2021.

[18] T. Kunz and M. Stilman, *Kinodynamic RRTs with Fixed Time Step and Best-Input Extension Are Not Probabilistically Complete*. Cham: Springer International Publishing, 2015, pp. 233–244.

[19] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, "The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, p. 26, 2021.

[20] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[21] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual slam library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.