Fast Multi-UAV Decentralized Exploration of Forests

Fast Multi-UAV Decentralized Exploration of Forests

Luca Bartolomei, Lucas Teixeira, Margarita Chli

***Abstract*—Efficient exploration strategies are vital in tasks such as search-and-rescue missions and disaster surveying. Unmanned Aerial Vehicles (UAVs) have become particularly popular in such applications, promising to cover large areas at high speeds. Moreover, with the increasing maturity of onboard UAV perception, research focus has been shifting toward higher-level reasoning for multi-robot missions. However, autonomous navigation and exploration of previously unknown large spaces still constitute an open challenge, especially when the environment is cluttered and exhibits large and frequent occlusions due to high obstacle density, as is the case of forests. Moreover, the problem of long-distance wireless communication in such scenes can become a limiting factor, especially when automating the navigation of a UAV fleet. In this spirit, this work proposes an exploration strategy that enables multiple UAVs to quickly explore complex scenes in a decentralized fashion. By providing the decision-making capabilities to each UAV to switch between different execution modes, the proposed strategy is shown to strike a great balance between cautious exploration of yet completely unknown regions and more aggressive exploration of smaller areas of unknown space. This results in full coverage of forest areas in multi-UAV setups up to 30% faster than the state of the art.**

***Index Terms*—Aerial Systems: Perception and Autonomy, Path Planning for Multiple Mobile Robots or Agents**

I. Introduction

The growing interest in Unmanned Aerial Vehicles (UAVs) has led to their extensive deployment in tasks such as inspection and search-and-rescue missions. In these applications, the capacity of the robot to quickly explore and map unknown environments autonomously is fundamental. The literature on this topic is extensive, and many different approaches have been proposed throughout the years [1], [2], [3]. However, one of the biggest challenges in the exploration of unknown environments is the capacity to achieve a good trade-off between the competing goals of shorter exploration times of an area of interest (i.e. pushing for high-speed navigation) and safety, which requires caps on the velocity of each robot. In fact, navigating in the vicinity of the boundaries between known and unknown space is challenging, as the robot can get stuck in dead ends, or needs to perform complex dodging maneuvers to avoid collisions. Consequently, to maintain the safety of both the platform and its surroundings, most path planners generate conservative start-and-stop motions, not fully exploiting the capacity of a UAV to fly at high speeds. This effect is exacerbated when the environment to explore is particularly cluttered, as is the case in forests, leading to inefficient and incomplete coverage. By design, these methods generally drive the exploration process by biasing exploration towards large areas of unexplored space. While this strategy could be advantageous in open and wide spaces, it can be detrimental when exploring cluttered scenes. In fact, the main pitfall of such strategies is that, while the exploration process attempts to cover as much unknown space as possible, when this is deployed in environments with many obstacles, thinner trails of unknown space are left unexplored (e.g. due to occlusions), imposing the need for a second sweep of the environment over mostly explored areas. Aiming to mitigate these issues, pushing for faster coverage of the areas of interest, multi-robot extensions for exploration have also been proposed [4], [5], [6], [7]. However, these focus on the problem of coordination at the system-level and, and while they can perform better from a global planning point of view, they suffer from the same limitations as the single-UAV case in obstacle-dense environments.

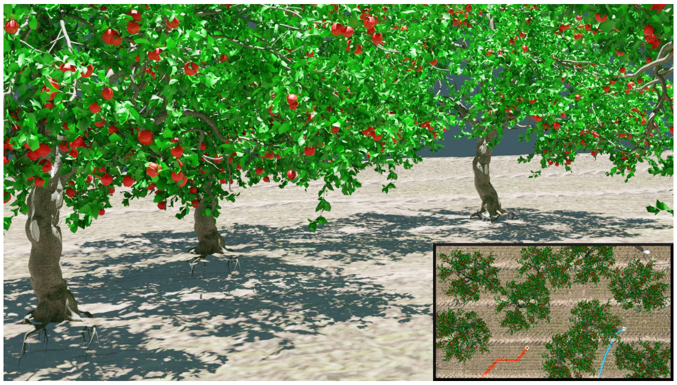


Fig. 1: The view of one of two drones exploring a synthetic forest with geometrically realistic tree models. The inset depicts the top view of the scene of the experiment and parts of the robots' trajectories. The proposed system guides the safe and successful exploration of the digital model of the forest by the two UAVs. The planner is able to avoid collisions between the UAVs and the scene while clearing frontiers on the go by balancing cautious and aggressive navigation in a bid to maximize the efficiency of the exploration. Coordination is achieved in a decentralized fashion, relying on direct peer-to-peer communication.

Motivated by these challenges, in this work we propose a decentralized multi-robot exploration strategy for autonomous UAVs aiming to explore forests of increasing tree density, as they pose some of the most difficult challenges for exploration

Manuscript received: May, 05, 2023; Revised June, 27, 2023; Accepted June, 27, 2023.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by NCCR Robotics, Unity Technologies, the Amazon Research Awards, and the HILTI group.

Authors are with the Vision for Robotics Lab, Department of Mechanical and Process Engineering, ETH Zurich, Switzerland, and University of Cyprus, Cyprus. {lbartolomei, pilucas, chlim}@ethz.ch

Video – https://youtu.be/F3aKqi5Q2LE

Code – https://github.com/VIS4ROB-lab/fast_multi_robot_exploration

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

planning. Our objective is to exploit the platform's agility despite the high density of obstacles, in order to achieve the complete coverage of the environment efficiently. To this end, the proposed strategy enables switching between two different behaviors for each robot; namely, cautious exploration of unknown space and more aggressive maneuvers when navigating in already explored areas to clear smaller portions of unknown space caused by occlusions. Multi-robot coordination is achieved in a decentralized fashion with direct peer-to-peer communication, as the agents collaborate and split the areas to explore. We evaluate the proposed approach in a series of challenging experiments in simulation of randomly generated forests, as well as in a 3D reconstruction of a real forest [8] and in a geometrically realistic forest model (Fig 1). Benchmarking against the state of the art reveals superior efficiency for the proposed approach, achieving higher overall UAV speeds and lower exploration times.

In summary, the contributions of this work are as follows:

- the design of a multi-robot decentralized exploration strategy, able to strike an effective balance between cautious exploration and aggressive exploitation of the explored map,
- extensive evaluations in simulation, demonstrating better performance than the state of the art, and
- the source code of the proposed system.

II. Related Works

Autonomous exploration of unknown environments with UAVs has been an active field of research over the past few decades. The most popular approach to exploring an area of interest is to use frontiers [9], defined as the boundary between known and unknown space [10]. These can be utilized to identify potentially informative spatial regions to drive the exploration process efficiently until no new frontiers are found and the exploration process can be considered complete [11]. However, while frontier-based approaches have been proven to yield satisfactory performance in terms of coverage [1], they generally lead to inefficient motions, especially in the case of UAVs. This is mostly caused by the sensing modalities used to generate the map of the environment to explore, as the most common sensors used onboard UAVs, such as RGB-D and stereo cameras, have a limited detection range. Consequently, these platforms need to fly cautiously to ensure safety. Cieslewski *et al.* [2] tackle this limitation, by proposing an exploration strategy that generates velocity commands based on newly detected frontiers, in a bid to maximize the UAV's speed. This method is shown to outperform classical methods [10] but focuses only on local frontiers. Instead, Zhou *et al.* [3] propose a hierarchical planner which generates efficient global paths, while encouraging safe and agile local maneuvers for exploration. However, this strategy requires maintaining a list of active frontiers. This additional bookkeeping becomes prohibitive in cluttered environments such as forests, since in this type of scenery, the number of frontiers quickly increases due to occlusions caused by tree trunks, branches, and shrubs.

To boost the efficiency in exploration, various multi-robot cooperative frontier-based methods have also been proposed in the literature, both in centralized [12], [13] and decentralized formats [14]. Centralized methods feature a ground station where a global multi-robot plan is first computed, and then broadcast to the agents under the assumption of perfect communication. To this end, Burgard *et al.* [12] aims at reducing the overlap in explored areas, by down-scaling the information gain of a candidate frontier if another robot is assigned to a different frontier in its vicinity, while Tian *et al.* [15] solve a Multiple Travelling Salesman Problem (mTSP) to allocate each agent to a candidate frontier. Instead, the approach in [16] is able to generate efficient trajectories for 3D reconstruction. However, this method requires a prior overhead flight over the area of interest, making it unsuitable for the exploration of forests. The approach proposed in [4] puts more focus on the problem of navigating forests, but the emphasis is on collaborative state estimation and mapping rather than on path planning.

Nevertheless, by design, centralized systems assume reliable long-range communication; moreover, they suffer from poor scalability, as the amount of data to process in the central unit increases with the number of agents. Decentralized approaches [17], [18] are instead more robust, since, given the nature of these methods based on direct peer-to-peer communication, each robot is able to operate independently from the others. This advantage comes at the cost of higher difficulty of coordinating the agents, as they have access to local information only. Yamauchi [17] propose a strategy where robots move to the closest frontier and exchange map information with the other agents. However, this approach is inefficient, as robots may move to the same frontier. Kabir and Lee [19] mitigate this limitation by basing their approach on optimal transport theory, while Yu *et al.* [20] utilize a multi-robot multi-target potential field to assign robots to different frontiers. To this end, [21] and [22] coordinate the agents using a distributed assignment of single targets, while Corah and Michael [7] assign robots to groups of frontiers. However, these methods assume stable communication between agents. The approach proposed by Zhou *et al.* [18], dubbed RACER, aims at mitigating these issues by dividing the navigation area in a grid, and ensuring that the agents explore distinct regions while flying coverage paths. RACER is robust to limited communication ranges, as it is based on direct pairwise robot interaction; nevertheless, it suffers from sub-optimal coordination in extremely cluttered environments such as forests, since it assumes a homogeneous distribution of obstacles.

Motivated by these limitations, in this work, we propose a multi-robot decentralized strategy that allows a fleet of robots to explore complex forest-like environments while flying at high speeds, thanks to the freedom and flexibility that our planner provides to each UAV to switch between different navigation modes online. Coordination is achieved by direct, pairwise communication and a less restrictive area splitting than [18]. While slower, cautious exploration is performed using a frontier-based approach, we efficiently clear trails of unexplored space caused by occlusions by employing a more aggressive local exploration strategy, boosting the efficiency of the mission and pushing the overall time to cover a given area of interest down.

III. Methodology

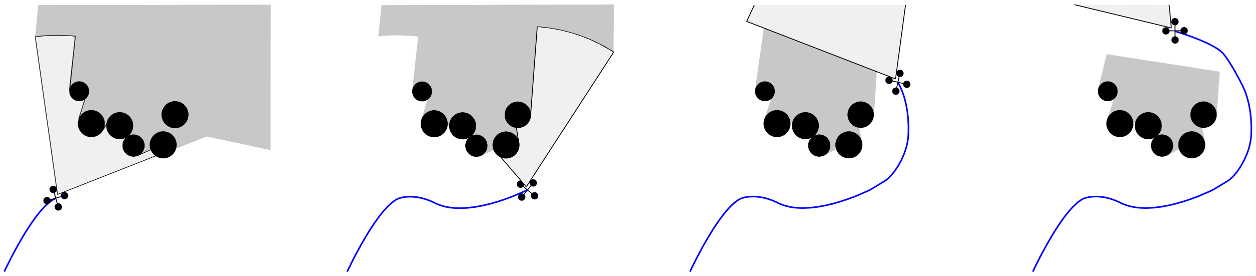
The overall problem considered in this work is the minimum-time exploration of cluttered environments, such as forests, using a fleet of UAVs. We assume that the robots are equipped with a front-looking depth camera with limited sensing range and that they perform decentralized state estimation, using e.g. [23], to localize themselves and their peers in a common reference frame. Given the estimated relative poses, each UAV exchanges map information with nearby team members within the communication range for more informed decision-making. The principal difficulty with forest-like scenes is the presence of a high number of obstacles in a variety of dimensions (e.g. trunks, leaves, and branches) that make standard frontier-based exploration approaches inefficient. In fact, during the exploration process, many islands of unknown space are usually left behind, as illustrated in Fig. 2, necessitating subsequent passes of exploration on a nearly completely explored map. This happens because generally these small islands are ignored initially because they carry a low information gain in terms of explorable volume. Therefore, standard planning approaches concentrate on frontiers that allow clearing of bigger regions. To tackle this limitation, we propose an exploration pipeline that can change the exploratory behavior of the robot depending on the frontiers in its vicinity. In particular, we propose to define two different modes of operation for the robot, namely the *Explorer* and the *Collector* modes. In the *Explorer* state, the robot is driven by frontiers and it is tasked to explore large unknown areas. Consequently, it predominantly operates on the most external boundaries between known and unknown regions. Conversely, the robot in the *Collector* mode clears small islands of unknown space generated by occlusions, that are left behind during the exploratory phase. The objective of a Collector is to clear these portions of space on the go, avoiding the need for subsequent revisits of the map, at the expense of short local detours. However, notice that these can be performed at high speed, since, when in Collector mode, the robot operates in mostly explored areas. By allowing a robot to switch between these two different modes and by finding the right trade-off between map exploration and exploitation, we can quickly reach full coverage of large cluttered environments.

A. System Overview

As shown in Fig. 3, we adopt a decentralized coordination strategy, where the agents exchange map information, current desired destinations, and execution modes, together with planned future trajectories, in order to assign the areas to explore and build a map collaboratively. The pipeline run by every single robot is composed of three main components: a mapping system, a mode selector, and a path planner. The mapping system uses odometry and depth information to generate a voxel grid map $\mathcal{M}$ of the environment [25]. At every update, the newly observed voxels are stored in chunks and communicated to other nearby robots [18]; similarly, information received from peers is integrated in $\mathcal{M}$. In the next step, frontiers are extracted from $\mathcal{M}$ and clustered. For each cluster c, we adopt the sampling strategy from [3] to generate viewpoints $\xi_c := \{\mathbf{x}_c, \gamma_c\}$, where $\mathbf{x}_c \in \mathbb{R}^3$ is the position of the viewpoint and $\gamma_c \in \mathbb{R}$ its orientation, which are later on used as target poses during the exploration process. The positions $\mathbf{x}_c$ of candidate viewpoints are sampled uniformly in the cylinder centered at the centroid of the cluster, while γ_c is chosen such that it maximizes the sensor coverage over the cluster. Moreover, each cluster undergoes a binary classification step, where unconnected islands of frontiers, or *trails*, are identified. This is necessary to identify those regions that are likely to require an additional revisiting phase towards the end of the mission if a traditional frontier-based exploration method is utilized. Here, a cluster is considered a *trail* if its convex hull is surrounded by free space, or when it has only another neighboring cluster. This implies that most clusters at the corners of the area to be explored are classified as trails. We motivate this design choice by arguing that corners are generally problematic for exploration due to their low informative value. In fact, they are rarely covered in a first sweep of the map, implying the need for a revisiting step. The labeled clusters and the other agents' positions are then utilized by the Mode Selector to choose the best exploration strategy for the robot, deciding whether it has to persevere in its current mode (Explorer or Collector) or switch mode. The mode assignment is regulated according to the frontiers in the vicinity of the UAV. Given that our objective is to clear trails locally to avoid large detours on the map, we assign the role of Collector if a minimum number of trails is close to the robot. Similarly, we adopt a more exploratory strategy once all smaller islands of unknown space are cleared, or when the trails are far away from the drone. However, in order to avoid duplication of the work in the same area, we enforce an agent to be in Explorer mode if a robot acting as Collector is already present in the local area. Since Explorers are more likely to generate new trails as they drive the exploration process toward unknown regions, they generate new tasks that can be directly assigned to nearby Collectors. Once a strategy is selected, the viewpoint of the most promising cluster is selected as the new target pose. This is fed to a path planner [24] that generates the shortest trajectory flying the UAV toward its destination while avoiding collisions with the environment and the other UAVs. Additional constraints, such as maximum inter-robot communication ranges, are not considered in the trajectory-generation phase.

B. Multi-robot Assignment of the Areas of Interest

In order to coordinate the robots and avoid duplication of exploration efforts within the same region, we assign separate areas of influence to every agent by performing pairwise coordination during target selection. This task distribution is modeled as an optimization problem, where, given the currently assigned areas $\tilde{\mathbf{x}}_R^i$ and $\tilde{\mathbf{x}}_R^j$ of robots i and j, the objective is to identify distinct regions of attractions $\tilde{\mathbf{x}}_R^{i*}$ and $\tilde{\mathbf{x}}_R^{j*}$. In the first iterations between robots, when regions are not assigned yet, the current target positions are used. The optimal values are computed by minimizing the following cost



(a) Time-instant 1 **(b)** Time-instant 2 **(c)** Time-instant 3 **(d)** Time-instant 4

Fig. 2: A schematic example demonstrating the problem with greedy frontier-based exploration, at progressive time-instants, generating islands of unknown space surrounded by free regions. The field of view of the robot is depicted as a light-gray shaded area delimited by black solid lines, while the obstacles and the unexplored space are in black and dark gray, respectively (a). The robot navigates towards the most informative frontiers (b); however, due to the limited sensor range, the space occluded by the obstacles is not cleared (c). Consequently, since the exploration process is biased towards larger, more informative frontiers, the naïve planner flies the UAV robot ignoring the smaller portion of unexplored space shown in dark gray (d).

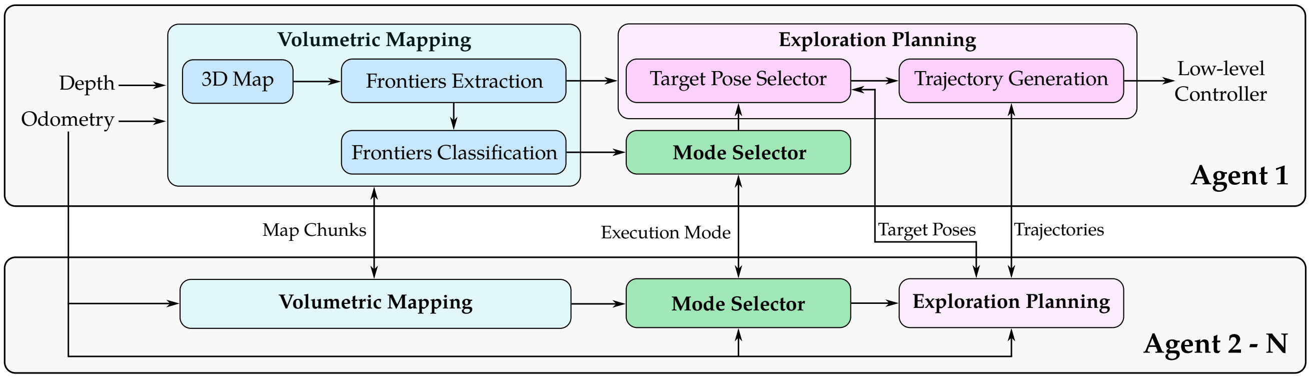


Fig. 3: Overview of the pipeline, where the task is to coordinate a fleet of UAVs to collaboratively build a map of the area of interest. Coordination is achieved by exchanging odometry and map information, as well as current execution modes and the target poses in order to assign areas of interest. Each robot generates a 3D grid-based map of the environment using depth measurements and the map chunks received from its peers within communication range. In the next step, frontiers are extracted and clustered. The trails of frontiers are identified and used to select the adequate exploration mode for the agent. The next target pose is chosen and a trajectory towards the goal is generated using [24], avoiding collisions with the scene and the other UAVs.

function:

$$J_{coord}(\tilde{\mathbf{x}}^i_R, \tilde{\mathbf{x}}^j_R) := d^*||\tilde{\mathbf{x}}^j_R - \tilde{\mathbf{x}}^i_R||_2 + \frac{1}{2}\frac{{d^*}^4}{||\tilde{\mathbf{x}}^j_R - \tilde{\mathbf{x}}^i_R||_2^2} - \frac{3}{2}{d^*}^2, \quad (1)$$

where d^* represents the desired distance between areas of interest. The optimization is triggered only in one of the two communicating robots, while the other waits for the response. If another team member k is encountered, the process is repeated, and $\tilde{\mathbf{x}}^{i*}_R$ is updated using its previous value to hot-start the optimization. Once the optimal values are computed, each UAV uses them in the remainder of the target pose selection process as described in the next section, where the different exploration modes are detailed.

C. Exploration Strategies

1) Explorer: Driven by frontiers, the objective of an Explorer is to cover large areas of previously unknown space. Similarly to [2], we process the incoming clusters of frontiers $\mathcal{C}$ from the most recent map update and store the ones in front of the robot in the set $\mathcal{C}_f$. Notice that in a multi-robot setting $\mathcal{C}_f \subseteq \mathcal{C}$, since the map update uses also the chunks received from other team members. The clusters in $\mathcal{C}_f$ are mostly aligned with the direction of the motion, implying that, if one of these is selected as the target, the robot avoids abrupt changes in the flight direction or aggressive maneuvers. The objective of the planner is to select as the target the viewpoint ξ_{c^*} with the lowest cost J_E, where $c^* \in \mathcal{C}_f$.

If we consider robot i, given the positions $\mathcal{X}^i_R := \{\mathbf{x}^k_R\}^{N-1}_{k=0}$ of the other $N-1$ robots in the team with $k \neq i$, and the assigned areas $\mathcal{A}_R := \{\tilde{\mathbf{x}}^{k*}_R\}^{N-1}_{k=0}$ of all agents, this cost is defined as:

$$\begin{aligned} J_E(\xi_c, \mathcal{X}^i_R, \mathcal{A}_R) := \omega_D J_D(\xi_c) + \omega_V J_V(\xi_c) + \\ \omega_L J_L(c) + \omega_C J_C(\xi_c, \mathcal{X}^i_R, \mathcal{A}_R), \quad (2) \end{aligned}$$

where the cost J_D is the length of the path in $\mathcal{M}$ between the current robot's position and $\mathbf{x}_c$, and it is calculated using the A* algorithm. Instead, J_V is associated with the change in direction of travel, while J_L with the label of cluster c and J_C to the distance from the assigned area of interest. The terms ω_D, ω_V, ω_V and ω_C are constant weights.

The cost $J_V(\xi_c)$ is calculated as

$$J_V(\xi_c) := acos(\frac{{\mathbf{v}^i_R}^T}{||\mathbf{v}^i_R||_2}\frac{\mathbf{x}_c - \mathbf{x}^i_R}{||\mathbf{x}_c - \mathbf{x}^i_R||_2}), \quad (3)$$

where $\mathbf{v}^i_R \in \mathbb{R}^3$ and $\mathbf{x}^i_R \in \mathbb{R}^3$ are robot i's current velocity and position, respectively. This cost is directly associated with the angle between the velocity and the direction vector towards the candidate position $\mathbf{x}_c$ covering cluster c. However, it may happen that the cluster is labeled as *trails*, e.g. in the case of occlusions caused by thin obstacles, such as tree trunks.

Since an Explorer should focus on actual frontiers, we assign a penalty to these clusters:

$$J_L(c) = \begin{cases} 0 & \text{if } c \text{ is } frontier \\ p_{trail} & \text{if } c \text{ is } trail \end{cases}, \tag{4}$$

where p_{trail} is the constant penalty associated with trails. The cost associated to collaboration is instead defined as:

$$J_C(\xi_c, \mathcal{X}_R^i, \mathcal{A}_R) := \\ \kappa_a U_a(\mathbf{x}_c, \tilde{\mathbf{x}}_R^{i*}) + \kappa_r \sum_{k=0,\ k\neq i}^{N-1} U_r(\mathbf{x}_c, \tilde{\mathbf{x}}_R^{k*}) + U_r(\mathbf{x}_c, \mathbf{x}_R^k). \tag{5}$$

The costs U_a and U_r act as attractive and repulsive potential fields, respectively. While the first term in Eq. 5 rewards viewpoints near i's area of interest, the second component penalizes them if they are close to the regions of influence of other robots. Given the Euclidean distance $d_{AB} := ||\mathbf{x}_A - \mathbf{x}_B||_2$, the cost functions U_a and U_r are calculated as follows:

$$U_a(\mathbf{x}_A, \mathbf{x}_B) := \begin{cases} 0 & \text{if } 0 \leq d_{AB} < d_0 \\ -(d_{AB} - d_0)^2 & \text{if } d_0 \leq d_{AB} < d_f \\ -\frac{(d_f - d_0)^2}{\tan^{-1}(0.1 d_f)} \cdot & \\ \tan^{-1}(d_{AB} - 0.9 d_f) & \text{if } d_{AB} \geq d_f \end{cases} \tag{6}$$

and

$$U_r(\mathbf{x}_A, \mathbf{x}_B) := \begin{cases} \frac{(d_c - d_0)^2 \sqrt{d_c d_0}}{\sqrt{d_0} - \sqrt{d_c}} \cdot & \\ \left(\frac{1}{\sqrt{d_{AB}}} - \frac{1}{\sqrt{d_0}}\right) & \text{if } 0 \leq d_{AB} < d_c \\ (d_{AB} - d_0)^2 & \text{if } d_c \leq d_{AB} < d_0 \\ 0 & \text{if } d_{AB} \geq d_0 \end{cases}. \tag{7}$$

The parameter κ_r and κ_a are constant weights, while d_0 represents the minimum distance between positions A and B to have a collision. The parameter d_c represents the distance after which the agents should not approach any closer, and d_f is the distance at which the attractive fields begin.

We then select as target pose the next best viewpoint ξ_{c^*} covering the cluster $c^* \in \mathcal{C}_f$ with the lowest cost:

$$\xi_{c^*} := \arg \min_{\xi_c\ \forall c \in \mathcal{C}_f} J_E(\xi_c). \tag{8}$$

In case the UAV is trapped in a dead-end, or if no new clusters are available in front of the robot, we employ a greedy approach to select the new target pose. We find the most informative cluster in the vicinity of the robot at a maximum distance d_{max} using the following cost function:

$$\xi_{c^*} := \arg \min_{\xi_c\ \forall c \in \mathcal{C}} \omega_D J_D(\xi_c) + \omega_C J_C(\xi_c, \mathcal{X}_R^i, \mathcal{A}_R) \\ s.t.\ ||\mathbf{x}_c - \mathbf{x}_R^i||_2 \leq d_{max}. \tag{9}$$

2) Collector: The objective of a Collector is to clear as many trails as possible, in order to avoid the need for a revisiting step in poorly explored regions of the map at the end of the mission. Since this task implies a detour from the main direction of exploration, the UAV's speed needs to be maximized in order to go back to Explorer mode as soon as possible. To reach this objective, we plan a local tour clearing all nearby trails using the Asymmetric Travelling Salesman Problem (ATSP) algorithm [3]. Given N_T nearby trails, we design the cost matrix $\mathbf{M} \in \mathbb{R}^{(N_T+1)\times(N_T+1)}$ as follows. The cost to travel from viewpoint ξ_i to ξ_j considers the path length in $\mathcal{M}$ from ξ_i to ξ_j, calculated using the A* algorithm, as well as the difference in the potential field J_C. With some abuse of notation to improve readability, this cost is computed as:

$$\mathbf{M}(i,j) := \omega_D J_D(\xi_i, \xi_j) + \omega_C (J_C(\xi_i) - J_C(\xi_j)), \tag{10}$$

where J_D and J_C follow the definitions of Sec. III-C1. The first row and column of $\mathbf{M}_{TSP}$ are instead associated to the robot's current position. The cost to reach a viewpoint ξ_i from $\mathbf{x}_R$ is calculated as follows:

$$\mathbf{M}(0,i) = \omega_D J_D(\mathbf{x}_R, \xi_i) + \omega_C J_C(\xi_i). \tag{11}$$

Following the ATSP formulation, we set the connections costs $\mathbf{M}(i,0)$ from any viewpoint i to the current position to zero. In this way, the cost of any closed-loop path starting from the robot's position is equivalent to the corresponding open-loop tour, allowing the use of a standard TSP solver to find the optimal sequence of trails. Moreover, we double the robot's maximum velocity compared to when in Explorer mode, since the trails are surrounded by free known space. Consequently, the UAV is able to maximize its velocity, leading to fast motions that allow it to quickly cover all the viewpoints associated with the trails.

IV. Experiments

To assess the performance of the proposed method, the pipeline is run in a series of forest-like scenarios with varying tree densities. The dynamics simulator of [18] is used with ground-truth odometry of the UAVs, assuming that each agent is equipped with a front-looking depth camera with resolution 640×480 pixel and field of view $80° \times 60°$. Depth images are rendered using the pipeline introduced in [26] with a maximum sensing range of $4.5\,\mathrm{m}$. The UAV's maximum linear and angular velocities are set to $1.5\,\mathrm{m/s}$ and $0.9\,\mathrm{rad/s}$, respectively. Our planner is compared against two state-of-the-art algorithms, namely RACER [18] and Burgard *et al.* [12], in terms of the time needed to complete the exploration of the scene and the average velocity of the UAVs during each experiment. These methods are based on frontier assignment; however, while RACER is a decentralized coordination strategy, Burgard *et al.* follow a centralized approach and assume the presence of a central unit that coordinates the agents by assigning target frontiers sequentially. At each iteration, the best pair of robot and frontier is computed greedily and the information gain of the remaining frontiers is decreased according to the previous assignment. This process continues until all UAVs are assigned a target. RACER is instead based on peer-to-peer communication and a more flexible coordination strategy, where the environment is discretized in a grid and agents mutually agree on the allocation of the areas to explore. Once each UAV is assigned an area, a coverage path is computed and successively refined considering the robot's dynamics, with the objective to find the best trajectory that clears unexplored regions.

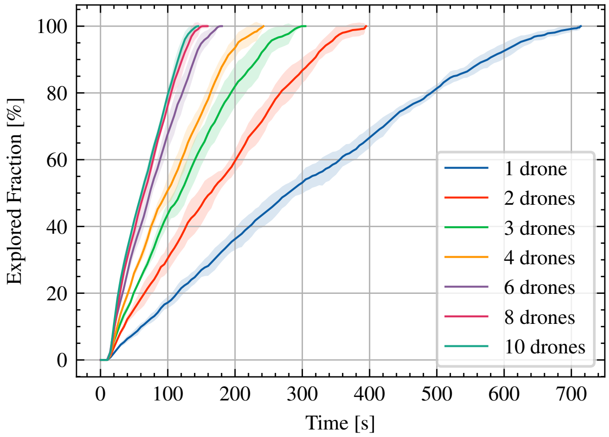


Fig. 4: Exploration rates over different numbers of UAVs in the forest with density 0.1 trees / m^2. The exploration process becomes more efficient as the size of the fleet increases.

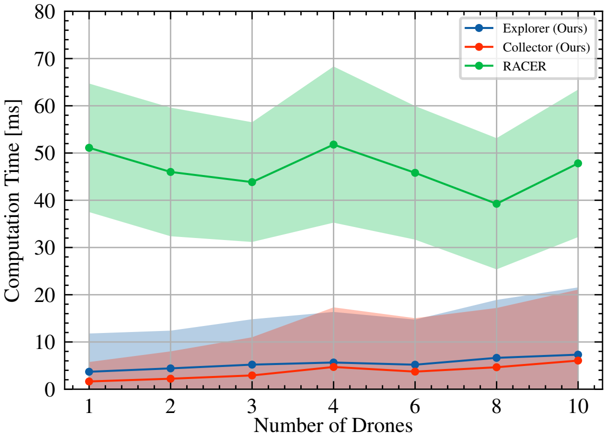


Fig. 5: Computation times per agent over varying team size during an experiment when using the proposed exploration modes and RACER [18]. We measure the time required for collaborative map splitting and for selecting a target viewpoint. Our approach reduces the total planning time by 70% on average.

A. *Multi-robot Forest Exploration*

We simulate RACER [18], Burgard *et al.* [12] and the proposed strategy in the digital 3D reconstruction of a real forest (0.05 trees / m^2) [8] and in four synthetic forests with size $50\,\mathrm{m} \times 50\,\mathrm{m} \times 2\,\mathrm{m}$. Three of these models have homogeneous obstacle density (0.1, 0.15, and 0.2 trees / m^2), while the fourth map presents a varying number of obstacles across different regions. In these synthetic scenes, trees are modeled as pillars representing their trunks. The environment is reconstructed with a voxel resolution of $0.15\,\mathrm{m}$, and the experiments are terminated when no new frontiers are extracted from the 3D volumetric map.

1) Results and Discussion: In Table I we present a comprehensive overview of the results in each scene assuming an infinite communication range between robots. In the single-agent setting, RACER [18] reaches lower completion times thanks to its ability to plan long-horizon coverage paths, while our method suffers from sub-optimal frontier selection as decision-making is performed locally. Nevertheless, our exploration strategy consistently outperforms both RACER [18] and Burgard *et al.* [12] when multiple drones are deployed. These results demonstrate the benefit of using the proposed adaptive exploration strategy over fix-mode methods, thanks to its ability to fly UAVs at high velocities throughout the mission. However, the proposed design leads to 15% longer traveled distances compared to other approaches, albeit guaranteeing that there are no small unexplored areas left. In fact, decision-making both in Explorer and Collector modes is done on a local-map level, and this may cause the UAV to fly longer routes, deviating from the shortest path. Nonetheless, in the proposed strategy we compensate for this shortcoming by encouraging decisions leading to higher UAV velocities with 50% increase in average speed, and 30% shorter mission times. Fig. 4 reports the exploration rate of the proposed strategy with up to 10 agents in the synthetic forest with density 0.1 trees / m^2. The exploration rate is consistently improved as the number of robots increases, demonstrating that our approach is able to coordinate large fleets efficiently.

2) Experiments with Geometrically Realistic Tree Models: To further validate the proposed system, we benchmark it in a forest with size $20\,\mathrm{m} \times 20\,\mathrm{m} \times 2\,\mathrm{m}$ composed of geometrically realistic 3D models of trees, as shown in Fig. 1. This experiment has the objective of testing the proposed planner in a complex scene and partially closing the simulation-to-reality gap. The experimental results reported in Table II demonstrate that our strategy can guide the exploration of this complex scene more efficiently than our competitors.

3) Computation Times Analysis: The proposed strategy is computationally more efficient than RACER [18] as shown in Fig. 5, where we report the average total planning time per agent with varying team sizes. Here we measure the total time required to assign the areas of influence to each robot, as well as to select the next target viewpoint when different exploration modes are adopted. Despite the minor increase in computation times with larger fleets, the proposed approach is extremely efficient, reducing by 70% the overall timings compared to RACER [18].

B. *Effect of Communication Range*

To assess the robustness of the proposed method to communication losses, different communication ranges are simulated, namely $10\,\mathrm{m}$, $20\,\mathrm{m}$ and $30\,\mathrm{m}$. These values are selected given the sizes of the map used for testing, and without considering additional effects, such as obstacle density or the presence of occlusions that influence the quality of communication and the effective range of WiFi connections. Once out of communication range, the UAVs cannot exchange information or share map data, leading to redundant exploration and sub-optimal action selection. Since the method by Burgard *et al.* [12] is a centralized approach and requires continuous communication between the agents and the central server, we compare the proposed method only against RACER [18]. The experimental results are reported in Fig. 6, where the time to complete the mission averaged over all the maps is reported as a function of the number of robots. Despite worse performance than the ideal case with infinite-range communication, our approach consistently outperforms RACER [18] even under limited range. In fact, the proposed method considers the limited effective communication distance indirectly during

# Drones	Strategy	Sparse Forest (0.1 trees / m^2)		Mid-Density Forest (0.15 trees / m^2)		Dense Forest (0.2 trees / m^2)		Multi-density Forest		Real Forest Reconstr. [8]	
		Time [s]	Velocity [m/s]	Time [s]	Velocity [m/s]	Time [s]	Velocity [m/s]	Time [s]	Velocity [m/s]	Time [s]	Velocity [m/s]
1	Ours	703.2 ± 25.8	**1.36 ± 0.01**	744.5 ± 26.3	**1.30 ± 0.04**	776.6 ± 45.4	**1.30 ± 0.03**	820.4 ± 64.3	**1.24 ± 0.03**	384.4 ± 15.5	**1.48 ± 0.04**
	RACER [18]	**575.0 ± 11.9**	1.25 ± 0.02	**642.3 ± 36.3**	1.17 ± 0.07	**673.2 ± 61.1**	1.21 ± 0.06	**709.4 ± 49.6**	1.13 ± 0.04	**303.9 ± 16.0**	1.29 ± 0.02
	Burgard *et al.* [12]	859.4 ± 17.1	1.03 ± 0.02	954.2 ± 31.5	0.99 ± 0.02	1140.1 ± 61.5	0.97 ± 0.03	1361.3 ± 155.6	0.88 ± 0.05	541.0 ± 21.2	1.05 ± 0.04
2	Ours	**376.7 ± 14.2**	**1.38 ± 0.04**	**423.5 ± 9.9**	**1.33 ± 0.04**	**474.3 ± 29.3**	**1.26 ± 0.06**	**552.2 ± 40.3**	**1.19 ± 0.05**	**226.9 ± 12.7**	**1.41 ± 0.05**
	RACER [18]	448.7 ± 60.5	0.93 ± 0.10	555.1 ± 57.6	0.85 ± 0.07	654.0 ± 99.5	0.78 ± 0.08	678.9 ± 115.4	0.77 ± 0.08	268.1 ± 58.9	0.92 ± 0.17
	Burgard *et al.* [12]	460.4 ± 21.8	1.03 ± 0.03	492.1 ± 15.7	0.98 ± 0.01	608.9 ± 44.3	0.96 ± 0.03	702.0 ± 109.9	0.90 ± 0.05	341.0 ± 93.2	1.05 ± 0.03
3	Ours	**280.8 ± 18.2**	**1.33 ± 0.05**	**317.9 ± 9.9**	**1.28 ± 0.03**	**346.4 ± 26.1**	**1.29 ± 0.04**	**385.5 ± 53.8**	**1.25 ± 0.06**	**157.9 ± 12.4**	**1.45 ± 0.04**
	RACER [18]	426.2 ± 44.8	0.74 ± 0.06	465.2 ± 29.2	0.71 ± 0.05	526.2 ± 41.4	0.73 ± 0.03	584.1 ± 62.6	0.76 ± 0.09	238.4 ± 43.2	0.72 ± 0.12
	Burgard *et al.* [12]	313.1 ± 15.9	1.03 ± 0.03	356.4 ± 7.2	0.97 ± 0.03	481.4 ± 120.7	0.93 ± 0.06	488.7 ± 73.0	0.86 ± 0.07	213.9 ± 9.2	1.02 ± 0.04
4	Ours	**227.5 ± 13.0**	**1.28 ± 0.03**	**254.9 ± 8.7**	**1.28 ± 0.03**	**295.6 ± 21.8**	**1.27 ± 0.06**	**342.9 ± 29.9**	**1.17 ± 0.07**	**124.2 ± 10.1**	**1.38 ± 0.03**
	RACER [18]	368.4 ± 55.2	0.76 ± 0.12	383.3 ± 66.7	0.77 ± 0.19	480.2 ± 56.9	0.66 ± 0.10	492.7 ± 69.9	0.71 ± 0.13	196.2 ± 55.1	0.79 ± 0.16
	Burgard *et al.* [12]	252.9 ± 15.4	1.01 ± 0.03	272.1 ± 14.1	0.95 ± 0.05	316.8 ± 15.1	0.94 ± 0.03	764.1 ± 526.3	0.65 ± 0.32	168.5 ± 11.1	1.04 ± 0.03
6	Ours	**168.0 ± 15.5**	**1.27 ± 0.05**	**192.0 ± 15.6**	**1.28 ± 0.03**	**254.0 ± 41.7**	**1.20 ± 0.04**	**267.5 ± 31.9**	**1.15 ± 0.06**	**126.3 ± 11.9**	**1.25 ± 0.05**
	RACER [18]	287.0 ± 59.3	0.70 ± 0.18	285.0 ± 40.5	0.77 ± 0.12	336.0 ± 47.7	0.79 ± 0.14	372.8 ± 57.7	0.78 ± 0.10	199.6 ± 50.9	0.68 ± 0.12
	Burgard *et al.* [12]	191.8 ± 22.9	0.96 ± 0.03	227.2 ± 8.0	0.94 ± 0.05	364.0 ± 201.2	0.82 ± 0.17	362.9 ± 90.8	0.84 ± 0.10	157.1 ± 9.5	0.93 ± 0.03
8	Ours	**139.0 ± 8.6**	**1.25 ± 0.05**	**164.2 ± 14.2**	**1.20 ± 0.03**	**207.5 ± 10.4**	**1.18 ± 0.02**	**222.0 ± 3.5**	**1.13 ± 0.04**	**98.6 ± 6.7**	**1.25 ± 0.03**
	RACER [18]	271.0 ± 35.4	0.64 ± 0.05	233.3 ± 22.1	0.74 ± 0.08	286.0 ± 21.6	0.72 ± 0.08	305.9 ± 27.0	0.76 ± 0.09	161.9 ± 36.8	0.71 ± 0.11
	Burgard *et al.* [12]	169.0 ± 3.3	0.94 ± 0.02	201.5 ± 7.0	0.92 ± 0.04	329.9 ± 40.1	0.74 ± 0.09	262.9 ± 46.6	0.86 ± 0.04	124.7 ± 7.2	0.99 ± 0.05
10	Ours	**132.7 ± 6.2**	**1.18 ± 0.02**	**144.0 ± 3.5**	**1.15 ± 0.01**	**200.7 ± 15.2**	**1.12 ± 0.02**	**222.8 ± 20.2**	**1.06 ± 0.04**	**87.7 ± 5.2**	**1.17 ± 0.03**
	RACER [18]	193.1 ± 34.0	0.72 ± 0.09	215.9 ± 30.7	0.72 ± 0.10	212.6 ± 31.1	0.84 ± 0.13	307.5 ± 31.6	0.65 ± 0.08	124.5 ± 17.8	0.78 ± 0.11
	Burgard *et al.* [12]	158.8 ± 4.5	0.90 ± 0.04	174.0 ± 6.0	0.85 ± 0.02	251.6 ± 65.5	0.84 ± 0.08	299.6 ± 60.6	0.79 ± 0.11	102.4 ± 6.6	0.89 ± 0.05

TABLE I: Mean and standard deviation of the robots' velocities, as well as the time required to complete the exploration of different maps with the proposed strategy (*Ours*), RACER [18] and Burgard *et al.* [12], over 5 runs and with varying fleet sizes. The best performance per team size is highlighted in bold.

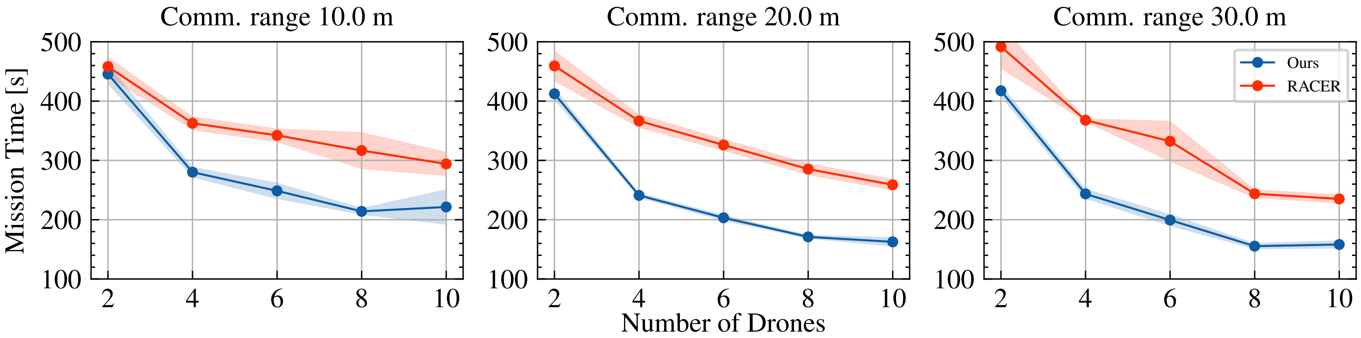


Fig. 6: Average mission times under different maximum communication ranges over a varying number of UAVs. The standard deviation is shown as a shaded area. We report the average performance in the 5 maps used for testing. The proposed approach outperforms RACER in all cases while showing improved performance with increasing connection ranges.

Experiment	Ours	RACER [18]	Burgard *et al.* [12]
1 drone	372.0 ± 20.0	**327.7 ± 12.3**	484.7 ± 18.5
2 drones	**204.8 ± 16.0**	277.5 ± 36.9	274.8 ± 12.1
4 drones	**116.0 ± 5.4**	203.6 ± 10.3	160.1 ± 20.3
8 drones	**86.6 ± 5.1**	100.6 ± 15.1	112.3 ± 10.2

TABLE II: Mean and standard deviation over 5 runs of the time required (in seconds) to complete the exploration of a map composed of photorealistic 3D models of trees. The best performance per team size is highlighted in bold.

map splitting, as the minimum of the cost function of Eq. 1 can be tuned to keep pairs of drones within the valid range.

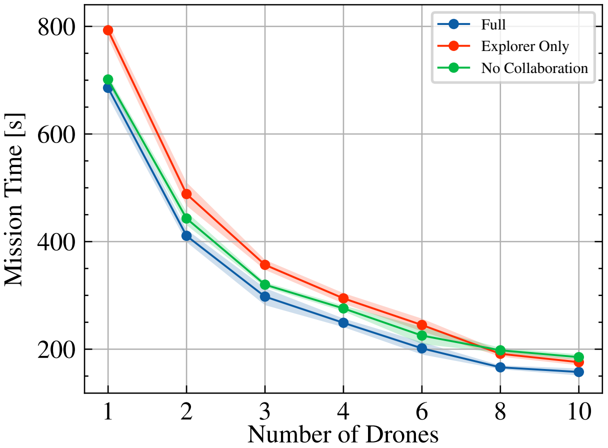


Fig. 7: Average exploration times in the reconstruction of a real forest [8] over 5 runs. The standard deviation is shown as a shaded area. We compare the proposed strategy (*Full*) against two variants, namely *Explorer Only*, where UAVs are not allowed to switch to Collector mode, and *No Collaboration*, where inter-agent coordination is ignored. The proposed pipeline shows the best performance.

C. Ablation Studies

In order to validate the different components of the proposed pipeline, we study their effect on the performance in exploration tasks. In particular, we compare two variants of our planning strategy. In the first approach, indicated with *No Collaboration*, we remove the area assignment component described in Sec. III-B, with the objective of validating the effectiveness of the proposed coordination algorithm. Instead, in the variant dubbed *Explorer Only*, exploration is carried out exclusively in Explorer mode. The aim is to check if switching between different exploratory behaviors is beneficial. As shown in Fig. 7, where we plot the average mission time over 5 runs when different numbers of agents are used in the 3D reconstruction of a real forest, our full pipeline shows the best performance. This demonstrates the validity of the proposed coordination strategy and map-splitting algorithm, as well as the advantages of having an adaptable exploration strategy. These effects become even more evident for large team sizes. Notice also that *Explorer Only* performs worse than *No Collaboration*, confirming the benefit of clearing the space while acting as a Collector. In fact, in Explorer mode smaller trails are ignored, leading to the necessity of an additional sweep over a mostly explored map and degrading the performance of the system.

V. Conclusion and Future Work

In this work, we propose a decentralized multi-robot exploration pipeline for autonomous UAVs operating in complex, cluttered environments, with a particular focus on forests. We choose this type of environment as one of the inherently most challenging for effective planning due to the increased number of obstacles and occlusions that they exhibit. To achieve collaboration, agents exchange map and odometry information, and area assignment is done by pairwise drone interaction. The proposed strategy allows UAVs to switch between different exploratory behaviors, autonomously balancing cautious exploration of unknown space and more aggressive maneuvers, exploiting already mapped space within a mission. This leads to faster completion times due to higher-speed flights and, consequently, to more efficient and faster map coverage than the state of the art, even under limited communication ranges. Following the push for automating higher-level decision-making in robotic missions, this work constitutes a key milestone toward effective exploration planning for robotic teams in cluttered scenes.

The natural next step for this work is to address the integration and deployment of the proposed pipeline onboard real platforms, while further investigations will push towards more advanced communication strategies to scale the approach to larger fleets.

References

[1] 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*, 2021.

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

[3] B. Zhou, Y. Zhang, X. Chen, and S. Shen, “FUEL: Fast UAV Exploration Using Incremental Frontier Structure and Hierarchical Planning,” *IEEE Robotics and Automation Letters*, 2021.

[4] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, “Search and rescue under the forest canopy using multiple UAVs,” *The International Journal of Robotics Research*, 2020.

[5] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanskỳ, D. Heřt, M. Petrlík, T. Báča, V. Spurnỳ, *et al.*, “Darpa subterranean challenge: Multi-robotic exploration of underground environments,” in *International Conference on Modelling and Simulation for Autonomous Systems*. Springer, 2019.

[6] L. Bartolomei, M. Karrer, and M. Chli, “Multi-robot Coordination with Agent-Server Architecture for Autonomous Navigation in Partially Unknown Environments,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[7] M. Corah and N. Michael, “Volumetric Objectives for Multi-Robot Exploration of Three-Dimensional Environments,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[8] A. Ahmad, V. Walter, P. Petráček, M. Petrlík, T. Báča, D. Žaitlík, and M. Saska, “Autonomous Aerial Swarming in GNSS-denied Environments with High Obstacle Density,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[9] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding Horizon ”Next-Best-View” Planner for 3D Exploration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[10] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1997.

[11] D. L. da Silva Lubanco, M. Pichler-Scheder, and T. Schlechter, “A novel frontier-based exploration algorithm for mobile robots,” in *International Conference on Mechatronics and Robotics Engineering (ICMRE)*, 2020.

[12] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” *IEEE Transactions on Robotics*, 2005.

[13] A. Mannucci, S. Nardi, and L. Pallottino, “Autonomous 3D Exploration of Large Areas: A Cooperative Frontier-Based Approach”,” in *Modelling and Simulation for Autonomous Systems*. Springer, 2018.

[14] M. S. Couceiro and D. Portugal, “Swarming in forestry environments: collective exploration and network deployment,” *Swarm Intelligence: Principles, Current Algorithms and Methods*, 2018.

[15] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, “Search and rescue under the forest canopy using multiple UAVs,” *The International Journal of Robotics Research*, 2020.

[16] D. Morilla-Cabello, L. Bartolomei, L. Teixeira, E. Montijano, and M. Chli, “Sweep-Your-Map: Efficient Coverage Planning for Aerial Teams in Large-Scale Environments,” *IEEE Robotics and Automation Letters*, 2022.

[17] B. Yamauchi, “Decentralized coordination for multirobot exploration,” *Robotics and Autonomous Systems (RAS)*, 1999.

[18] B. Zhou, H. Xu, and S. Shen, “RACER: Rapid Collaborative Exploration with a Decentralized Multi-UAV System,” *IEEE Transactions on Robotics*, 2023.

[19] R. H. Kabir and K. Lee, “Efficient, Decentralized, and Collaborative Multi-Robot Exploration using Optimal Transport Theory,” in *American Control Conference (ACC)*, 2021.

[20] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, “SMMR-explore: Submap-based Multi-robot Exploration System with Multi-robot Multi-target Potential Field Exploration Method,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[21] R. Zlot, A. Stentz, M. Dias, and S. Thayer, “Multi-robot exploration controlled by a market economy,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[22] A. J. Smith and G. A. Hollinger, “Distributed inference-based multi-robot exploration,” *Autonomous Robots*, 2018.

[23] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, “Omni-Swarm: A Decentralized Omnidirectional Visual–Inertial–UWB State Estimation System for Aerial Swarms,” *IEEE Transactions on Robotics*, 2022.

[24] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and Efficient Quadrotor Trajectory Generation for Fast Autonomous Flight,” *IEEE Robotics and Automation Letters*, 2019.

[25] L. Han, F. Gao, B. Zhou, and S. Shen, “Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[26] L. Bartolomei, Y. Kompis, L. Teixeira, and M. Chli, “Autonomous Emergency Landing for Multicopters using Deep Reinforcement Learning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.