EFP: Efficient Frontier-Based Autonomous UAV Exploration Strategy for Unknown Environments

Hong Zhang, *Member, IEEE*, Songyan Wang, Yuanshuai Liu, Pengtao Ji, Runzhuo Yu, and Tao Chao, *Member, IEEE*

***Abstract*—The optimization of quadrotors for the efficient and autonomous exploration of complex, unknown environments and the construction of corresponding maps with integrity is of high priority in unmanned aerial vehicle(UAV) research. To overcome the challenges of inefficient and incomplete map construction in autonomous UAV exploration, this study propose EFP, an efficient frontier-based autonomous UAV exploration strategy for unknown environments. For this, the UFOMap algorithm was adopted to represent an entire environment and reduce the map construction time. Its accurate representation and hierarchical frontiers structure were then employed to rapidly extract frontiers. Simultaneously, a fast Euclidean clustering approach was implemented to process the frontiers and obtain the relevant viewpoints, an approximate trajectory optimization strategy was used to rapidly obtain a preferred trajectory that traverses all the viewpoints, and finally the RRT-based global planner and sampling-based local planner algorithms were utilized to perform autonomous exploration with a drone. The proposed algorithm was analyzed and validated in both simulation and real-world scenarios, demonstrating higher efficiency than state-of-the-art approaches and enabling quadrotors to autonomously explore and construct complete maps in complex and unknown environments.**

***Index Terms*—Aerial systems: applications, aerial systems: perception and autonomy, autonomous vehicle navigation.**

I. Introduction

OWING to their high mobility and degree of freedom, unmanned aerial vehicles(UAVs) are being increasingly used in different civilian and military applications, including search and rescue missions [1], exploration of unknown subterranean environments [2], infrastructure reconnaissance [3], Inspection of industrial facilities and data gathering [4] etc. Typically, quadrotors are operated by professionals to accomplish exploration and mapping of unknown or partially known environments, However, human subjectivity and skill considerably influence the effectiveness of this approach. Therefore, the autonomous exploration and mapping of these complex environments using quadrotors with a variety of high-precision, onboard sensors has become relevant for researchers. In addition, this method is economical and highly efficient, which can lead to unburdening manpower, moreover, the use of drones in hazardous environments can significantly reduce human risks.

Over the past decades, various suitable strategies have been proposed for the use of UAVs for exploring complex, unknown environments.However, completion and efficiency optimization is required for UAVs to adapt to larger, more challenging conditions. Fist, although many approaches utilize frontiers to guide the vehicles to cover the environment [5], identifying and clustering frontiers is computationally demanding, which makes the overall planning time too long to satisfy real-time requirements and restricts vehicles from promptly responding to environmental changes. Moreover, vehicles tend to repeatedly explore a region, which considerably reduces exploration strategy that involves directly navigating to the nearest frontier target position or location with the largest gain, disregarding the global exploration task.

To optimize the efficiency and completion of exploration and mapping with UAVs, this study proposes **EFP**, an **E**fficient **F**rontier-based autonomous ex**P**loration strategy that enables UAVs to rapidly explore unknown and complex environments. we adopt the UFOMap algorithm [6] as the environment representation of the whole exploration system, and an efficient map frontier extraction and clustering method was designed to obtain basic information about the exploration planning process. Simultaneously, To enable the quadrotor to expeditiously explore the whole environment, we employed an approximate trajectory optimization strategy that considers the coverage area, travel distance and yaw angle to obtain the optimal exploration trajectory of the quadrotor. Simulations and real-world experiments were conducted on a vision-based quadrotor platform and our proposed algorithm was compared with state-of-the-art related works, which showed that our work has the ability to efficiently explore the entire environment without a priori information. There are some improvements in terms of trajectory quality and overall exploration time as well. Operational segments of the real-world experiment is shown by Fig. 1, where the quadrotor trajectory is shown in gray, the yellow trajectory is for local planner, and the blue and green trajectories are for global planner. The main contributions of this study are as follows:

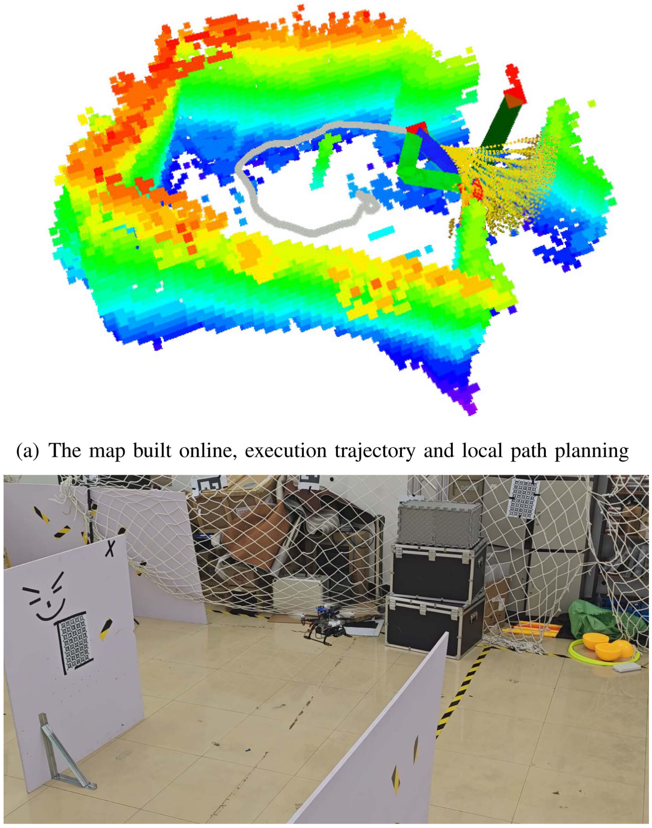
1) An efficient frontier detection and clustering method that enables real-time scheduling.

Manuscript received 24 October 2023; accepted 27 January 2024. Date of publication 7 February 2024; date of current version 15 February 2024. This letter was recommended for publication by Associate Editor S. Baidya and Editor A. Banerjee upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation (NNSF) of China under Grant 62273119. *(Corresponding author: Tao Chao.)*

The authors are with the National Key Laboratory of Modeling and Simulation for Complex Systems, School of Astronautics, Harbin Institute of Technology, Harbin 150001, China (e-mail: zhyn0515@163.com; sywang@hit.edu.cn; 13019706103@163.com; 17856504619@163.com; 23s104166@stu.hit.edu.cn; chaotao2000@163.com).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3363531, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3363531



(a) The map built online, execution trajectory and local path planning

(b) Quadrotor autonomous exploration in unknown environments

Fig. 1. Operational segments of a quadrotor for autonomous exploration in unknown complex environments, the video of the experiment is provided at: https://youtu.be/GJBGfaLJexs.

2) An approximate trajectory optimization strategy that considers coverage, travel distance, yaw angle and global information, rendering exploration more efficient.
3) Simulations and real-world experiments validated the proposed strategy, demonstrating its efficiency and lightweight of our proposed method for unknown 3D environment without a priori information.

The overall structure of this letter is shown as follows, Section II summarizes the previous state-of-the-art related works. Section III is the core part of this letter, which summarizes the specific details of the strategy of our proposed approach. Section IV discusses the simulation and real-world experiments as well as the analyzes their results. Finally, Section V presents the conclusions of this article.

II. Related Work

Over the past few decades, research related to autonomous exploration using quadrotors in unknown or partially known environments has evolved rapidly with the development of embedded computing. The related approaches can be broadly categorized into three groups: frontier-based approaches, next best view-based approaches, and deep reinforcement learning-based approaches.

Traditional frontier-based exploration was originally proposed by [7], which pioneered the concept of frontiers as boundary regions between open and unexplored spaces. The concept enabled the autonomous exploration and mapping of 2D environments using mobile robots, pioneering the direction of frontier-based exploration. Subsequently, [5], [8], [9], [10] and other related researchers extended and improved it by adapting frontier-based exploration and mapping to 3D complex environments instead of only 2D ones. Batinovic proposed a frontier detection method based on multi-resolution Octomaps [11] in [5], however, this frontier identification and clustering approach is extremely computationally demanding and cannot satisfy real-time planning requirements for UAV platforms with limited computing capabilities. In [9] Huang et al. introduced a UFOMap-based frontier detection strategy that considerably sped up frontier recognition. Nevertheless, the excessive number of frontiers in complex environments still led to the requirement of considerable computational still led to the requirement of considerable computational resources for subsequent processing.

The first autonomous exploration based on next best views [12] was proposed by [13]. These methods randomly sample some candidate viewpoints around the robot or at the frontier, then calculate their potential information gain, and select the viewpoint with the largest potential information as the next best viewpoint. This method was subsequently improved and optimized by among others, [14], [15], [16]. In [14], Bircher calculated a rapid random tree online and simultaneously extracted the optimal branch according to the known surroundings to guide the UAV into exploring unknown space volumes. Although this method can explore large environments, the nature of the rapid random tree makes its exploration trajectory suboptimal, which renders its overall exploration time unpredictable and extends the computation time in some environments. Subsequently, in [15], Zhu et al. dynamically extended the rapid random tree and global map, and preserved the nodes generated by the tree to achieve fast backtracking, thus improving efficiency. However, this type of method produces many invalid viewpoints which affects the efficiency of planning.

Autonomous exploration methods based on deep reinforcement learning have emerged in recent years from studies such as [17], [18], [19], [20]. In [17], Li described a grid-map based motion space to improve the migration velocity of the robot and designed a Fully convolutional Q-network with an Auxiliary task(AFCQN). The network evaluated the target points in the motion space based on known constructed map information and the current and previous robot positions. A deep reinforcement learning method was utilized to train the network. Ultimately, the robot was able to construct a map covering most of the environment with a shorter path, However, this method is less robust and less adapted to unfamiliar scenarios than the two types of approaches mentioned previously, Its performance deteriorates in sophisticated and unfamiliar environments.

The proposed approach is inspired by that of Huang et al. [9], To make their approach applicable to our system, we extended their procedure by clustering the obtained frontier information

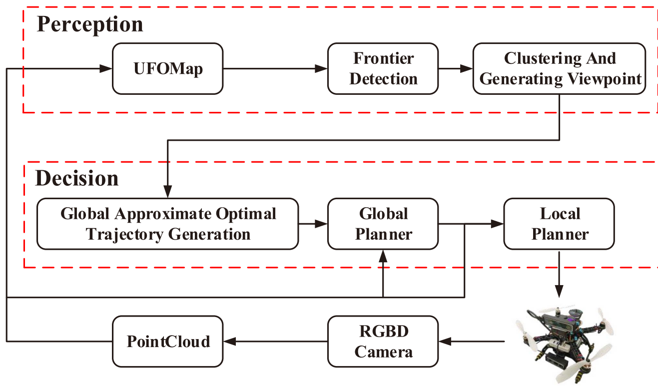


Fig. 2. Overview of the proposed exploration strategy framework.

and extracting the optimal viewpoints, thus optimizing frontier information and improving the subsequent planning time.

III. Methodology

A. Algorithm Overview

Fig. 2 shows the framework proposed in this study, which mainly consists of components for environment perception and path planning decisions. The perception component constructs the environment around the quadrotor using an onboard RGBD camera and analyzes frontiers, viewpoints and their rewards based on already known surroundings. The decision part mainly relies on the information from the perception part to plan the shortest trajectory that traverses all the unknown regions, using global planner and local planner to execute the trajectory. The representation of entire environment and extraction of frontiers are detailed in Section III-B. The procedure for processing the frontier information is introduced in Section III-C. Section III-D describes the trajectory optimization method to rapidly traverse all unknown regions.

B. 3D Representation and Frontier Detection

In this work, a 3D occupancy grid map based on the Octomap extension UFOMap [6] was used to represent the environment of the whole system. UFOMap explicitly represents all three states on the map: unknown, free and occupied.

$$V \equiv V_{free} \cup V_{occ} \cup V_{un} \tag{1}$$

where V_{free} is the free voxels from the free space, V_{occ} is the occupied voxels from the occupied space, V_{un} is the unknown voxels from the unknown space. This expression enable obtaining the frontier of the map more intuitively. In addition, UFOMap utilizes Morton codes to accelerate the traversal of the entire octree, and can obtain elements whose states change over time. Frontiers are usually present in these changed elements, hence, they can be detected in them, which further expedites obtaining frontiers.

This study reduces frontiers to a series of voxels with the following properties:

$$f(V) = \{V \in V_{free} : \exists neighbor(V) \in V_{un}\} \tag{2}$$

Algorithm 1: Frontier Detection and Updating.

Require: point cloud $\mathcal{M}$
Ensure: local frontier list $\mathcal{F}_{local}$ and global frontier list $\mathcal{F}_{global}$

```
F_local, F_global, V_update ← ∅
while exploration is not finished do
    UFOMap M ← UpdateMap(point cloud)
    V_update ← CheckUpdate(M)
    F_local ← CheckFrontier(V_update)
    F_global.append(F_local)
    F_global ← CheckFrontier(F_global)
end while
return F_local, F_global
```

Our algorithm adopts hierarchical frontiers maintenance and maintains a list of local F_{local} and global F_{global} frontiers, instead of only using a global frontiers list like previous similar works. Therefore, our method can only consider the local update range during frontiers detection, without traversing the entire map to update the frontiers, which significantly improves the efficiency of frontiers detection. Specifically, the list of local frontiers is updated according to the changing elements in each UFOMap and simultaneously inserted into the list of global frontiers. It is then determined whether the elements in the global list satisfy the definition of frontiers according to (2). Algorithm 1 illustrates the whole process of frontier detection and updating.

Experimental have shown that there are typically serveral frontiers in the exploration process and that, given the complexity of the subsequent decision-making, evaluating them is expensive. Therefore, fast clustering, which produces many clusters whose geometric centers are potential exploration centers, was used to cluster the global frontiers.

C. Fast Frontier Clustering

The presence of numerous frontiers can critical slow down subsequent decisions and planning, hence, many frontier-based exploration algorithms utilize clustering algorithms to deal with the frontier. Similarly, after investigating the state-of-the-art clustering methods, this study selected Fast Euclidean Clustering(FEC) to process the frontier information we proposed to. The FEC clustering algorithm was first proposed by Cao in [21], who cleverly adopted a point clustering algorithm over existing clustering methods and applyed a point-by-point scheme to the cluster-by-cluster scheme used in the existing works, This increased the clustering speed by two orders of magnitude with small quality variation. In order to balance exploration efficiency and computational consumption, we carefully select the appropriate clustering bandwidth, Since our map representation resolution is set as 0.1 m, we choose $n = 20$ as the minimum number of clusters, which can not only eliminate wrong frontiers, but also improve subsequent computational efficiency of the process.

The frontier information extracted in the previous step was transformed into point cloud information and input it into Fast

Euclidean Clustering. The geometric center of the cluster was output as the candidate viewpoint, which is defined as $V_k, k > 0$. In related research, the information gain is usually calculated by the number of frontiers in the clusters, For simplification, this study used the area of clustered clusters as the information gain, which is easily obtainable and more common, defined as $G(V_k), k > 0$. In addition, the current position of the quadrotor was taken as v_0, and its information gain was set to 0.

D. Topology Mapping and Trajectory Optimization

For the system to traverse all candidate viewpoints more efficiently, a trajectory optimization strategy that simultaneously considers relative distance, relative yaw and global information utility was proposed. First, an undirected topology graph $G = (V, E)$ connecting candidate viewpoints was employed to accelerate trajectory search and improve exploration strategies. To ensure that each candidate viewpoint was added to the undirected topology graph while avoiding distant, ineffective viewpoints, the computationally efficient KD-TREE was used to search for the n nearest neighbors of each candidate viewpoint to the current robot position of the robot. They were then associated to form topology graph edges and eliminate duplicate edges. Subsequently, their Euclidean distances were calculated to represent the utility of the two candidate viewpoints. However, this was incomplete, The experimental results showed that when two neighboring viewpoints of the quadrotor have the same utility, the quadrotor may fall into a local optimal solution, which moves the quadrotor back and forth and reduces exploration efficiency. Therefore, unlike previous research that directly use the Euclidean distance of the current position to set the utility of the edges of the neighboring viewpoints at the current position, the motion consistency cost was adopted to introduce the relative yaw of the current position of the quadrotor. To prevent the quadrotor from having a large steering direction, the utilities were calculated as follows:

$$U_{i,j}(V_i, V_j) = \frac{length(V_i, V_j)}{\nu_{\max}}, i, j \in 1, 2, \ldots, N \tag{3}$$

$$U_{0,j}(V_0, V_j) = \frac{length(V_0, V_j)}{\nu_{\max}} * \log_2(\frac{\theta}{\Pi} + 1), \tag{4}$$

$$\theta = \arccos \frac{(V_j - V_0) \cdot v_0}{\|V_j - V_0\| \cdot \|v_0\|}, j \in 1, 2, \ldots, N \tag{5}$$

where $length(v_i, v_j)$ is the Euclidean distance between the two candidate viewpoints, and $\nu_{\max}$ is the maximum speed of the quadrotor. θ is the angle between the current position of the quadrotor and any other candidate viewpoint adjacent to it. The information gain obtained from the previous step was used as the reward for each candidate viewpoint.

In contrast to algorithms that navigate directly to the nearest viewpoint or navigate to the area with the greatest information gain, this work introduces relative yaw into the cost function so that the above-mentioned back-and-forth movement problem can be taken into account in the overall trajectory optimization process. At the same time, when encountering a fork in the road, the UAV will be more inclined to kinematically turn to the path with the least consumption and maximum profit to execute, making it easier for the UAV to obtain a higher benefit ratio in the exploration problem and improve the overall exploration efficiency. At the same time, the algorithm proposed in this study synthesizes global information gain and abstracts the exploration problem as a variant of the Traveling Salesman Problem (TSP) problem using an undirected topology graph to find an open-looped global path that traverses all candidate viewpoints and efficiently explore the entire environment. It is well known that such problems are NP-complete. Therefore, most studies used heuristic solutions. Because the problem at hand is a small-scale TSP problem and does not require an exact solution, this study referred to the relevant state-of-the-art techniques, inspired by [9], [22]. A 3-opt local search heuristic was designed considering the relative distances, relative angles and global information utility to found an approximate solution to the trajectory optimization problem. First the information for each viewpoint is sorted in ascending order to derive an initialized path, defined as $P_{init} = [v_0, v_1, \ldots, v_n]$. Then three different nodes v_i, v_j, v_k were randomly selected, where $1 \le i \le n-2, i \le j \le n-1, j \le k \le n$. Finally, two of the viewpoints were arbitrarily exchanged to obtain the five new trajectories, and their gain utility was calculated as follows:

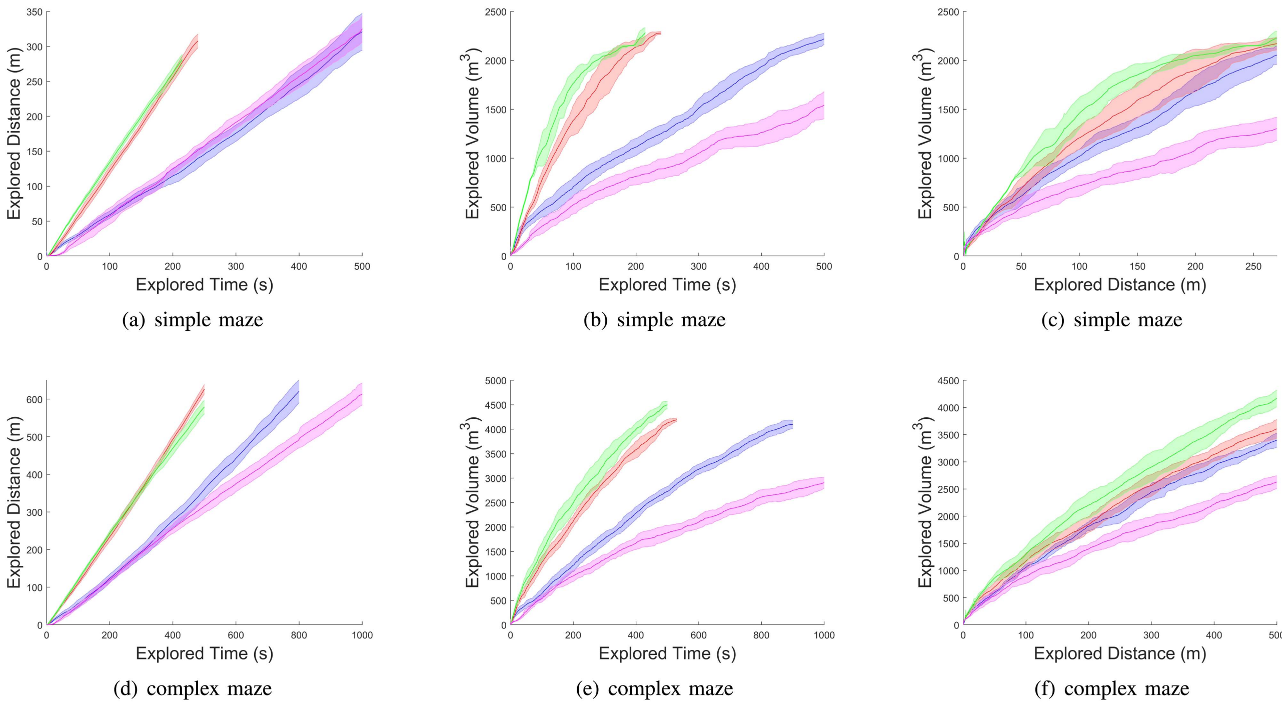
$$V^* = \max \Sigma_{k=1}^{n} G(v_k) \cdot \exp(-c \cdot C_{k-1,k}(k-1, k)) \tag{6}$$

where $G(v_k)$ is the information gain of the viewpoint calculated according to step 1, c is a penalty factor, and $C_{k-1,k}(k-1, k)$ is the consumption of two neighboring viewpoints according to (3). The maximum utility of the trajectory was calculated to update the optimal trajectory, and then three different viewpoints were selected again until a better trajectory could not be generated. Finally, an optimal path that traverses all viewpoints was obtained. This algorithm can update the path in real-time with a control frequency of up to 10 Hz to compensate for the rapid changes in the environment around the quadrotor, thus improving exploration efficiency. Algorithm 2 describes the entire process of topology mapping and trajectory optimization.

Algorithm 2: Topology Mapping and Trajectory Optimization.

Require: candidate viewpoints and current position $\mathcal{V}_k$, information gain $\mathcal{G}(\mathcal{V}_k)$
Ensure: undirected topology graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, approximately optimal path $\mathcal{P}_{optimal}$

```
1:  P_optimal, P_new, P_init, G(V, E) ← ∅
2:  G(V, E) ← TopologyMapping(V_k, G(V_k))
3:  P_optimal, P_init ← Sort(G(V_k))
4:  U(P_optimal) ← CalculateUtility(P_optimal)
5:  while no greater utility do
6:    P_new ← RandomlyExchange(P_init)
7:    U(P_new) ← CalculateUtility(P_new)
8:    P_optimal ← Compare(U(P_optimal), U(P_new))
9:  end while
10: return G(V, E), P_optimal
```



(a) simple maze (b) simple maze (c) simple maze

(d) complex maze (e) complex maze (f) complex maze

Fig. 3. Comparison of the four algorithms in two environments, solid line is the mean, shaded is the standard deviation.

When the optimal path is obtained, the robot starts to following the planned path and navigates towards the first viewpoint. To control the quadrotor, we applied a local path planning and trajectory tracking method based on a trajectory library [23]. This method loads the trajectory library based on the current speed of the robot and performs a check to select the optimal collision-free trajectory for execution. The airOMPL planner is employed to plan the route for the aerial vehicle. The algorithm uses a point cloud-based map representation and the RRT* algorithm [24] to guide the aerial vehicle toward a specified viewpoint. The whole exploration process is carried out until no frontier exists on the map, a complete global map is then created.

IV. Experimental Evaluation

The proposed strategy is simultaneously tested in simulation and real-world scenarios, and to verify the efficiency of our algorithm, we qualitatively and quantitatively compared it with RH-NBVP [14] and AEP [25], which are based on next best view methods, and FUEL [8], which is based on frontier strategy, in two complex simulation environments. All simulation were run on an Intel Core i7-9700 CPU @ 3.00 GHz x 8 and 16 GB of RAM with the Ubuntu 18.04 system, ensuring equality between experiments and a fair comparison process.

A. Simulation Experiment

We conduct experimental analysis with a fixed starting point and do not set any end point. First we will provide size of the map to be explored to the exploration algorithm, including length, width, and height. The exploration algorithm then performs path planning based on known information, and automatically determines whether the map still has unknown areas or whether the passable area has been explored.

TABLE I
Indentical System Constraints

Parameter	Value	Parameter	Value
Linear Velocity	1.5m/s	Sensor FoV	115° x 60°
Angular Velocity	1rad/s	Sensor Range	5m

To better evaluate the efficiency of the proposed algorithm and its applicability to large and complex environments, we constructed two different maze-like environments using Gazebo, as shown in Fig. 4(a) and (b), The Gazebo environment modes were obtained from 3DGEMS [26]. Both maps have numerous dead-ends and small loops, the simple maze is $20 \times 40 \times 3m^3$, and the large, complex maze is $40 \times 40 \times 3m^3$. The experiments satisfy the identical system constraints to ensure equal conditions, as presented in Table I. Meanwhile, in order to prevent longer experimental time, we took twice the time of our proposed algorithm as the experimental abort time, in which the experimental abort time is 1000 seconds for the large and complex maze, and 500 seconds for the simple maze. To better analyze the efficiency of the strategy, we conducted ten experiments in two mazes with the same initial conditions and compared the exploration times, movement distances, and the ratios of exploration volume to exploration distance with the other three strategies.

Fig. 3 illustrates all the data obtained from our simulation experiments, where the solid line and the shaded area represent the average of the ten experiments and the standard deviation, respectively. The proposed strategy performed the best in all

(a) complex maze (b) simple maze (c) complex maze (d) simple maze

Fig. 4. Two simulation environments and the results of EFP where the white line showing the path travelled. The red point is the starting point and the green point is the end point of the experiment. Note that we have not set a target point, the end point is when the system automatically determines that there are no unknown areas in the entire map, the task is completed and the exploration ends.

Fig. 5. Compact quadrotor for real-world experiments.

TABLE II
AVERAGE CONSUMPTION TIME OF SYSTEM COMPONENTS

Average consumption time(s)					
scenario	Construct UFOMap	Detect frontiers	Cluster frontiers	Optimize trajectory	Total
simple	0.0297	0.0069	0.0038	0.0030	0.0434
complex	0.0313	0.0081	0.0074	0.0071	0.0539

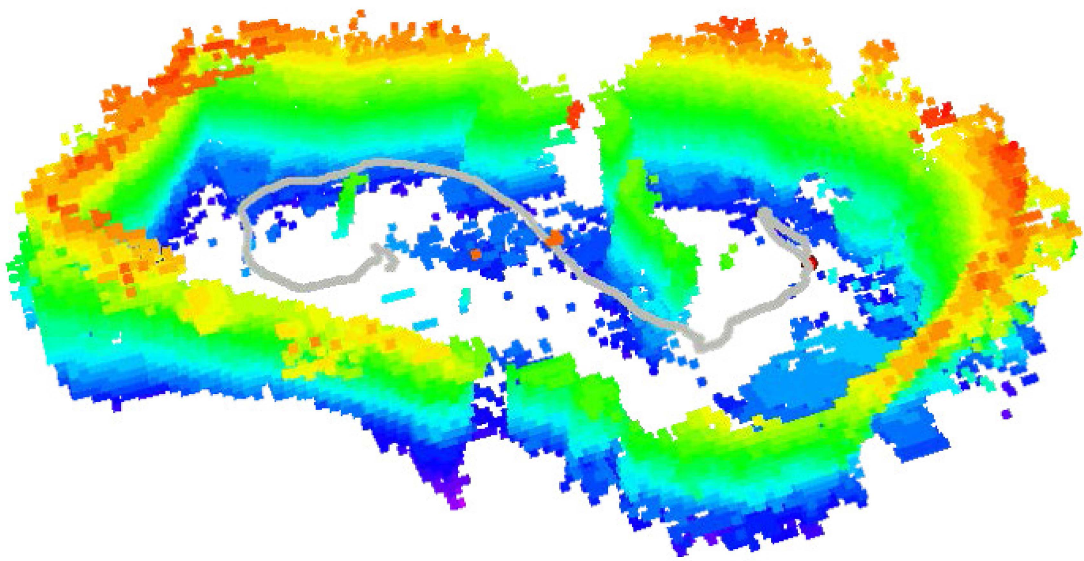
three aspects. Fig. 3(a) and (d) demonstrate the ratio of explored distance to explored time. Both EFP and FUEL operated at an average speed of approximately 1.25 m/s, whereas that of AEP and NBVP were significantly slower. The maximum speed set was 1.5 m/s. Fig. 3(b) and (e) illustrate the ratio of explored Volume to explored time. In the simple maze environment, the proposed algorithm considerably outperforms FUEL by 10% efficiency (214 s compared to 240 s) whereas AEP barely completes the task in the specified time 500 s, and the NBV does not complete it. In the complex maze environment, the proposed algorithm EFP, with an average value of 498 s, far outperforms the average running times of 530 s for FUEL and 897 s for AEP, and the NBV algorithm does not complete the task within the specified time. Therefore, the proposed algorithm improves the efficiency by 5% and 45% compared to FUEL and AEP, respectively. The NBV algorithm cannot be compared because it did not complete the task within the specified time, however, our algorithm is clearly more efficient than NBV, and the completeness of the constructed maps exceeds the state-of-the-art algorithms. Fig. 3(c) and (f) illustrate the ratio of explored volume to explored distance, which proves that EFP can tend to the area with more information under the same condition and explore more volume information for the same explored distance. Fig. 4(c) and (d) show the EFP's simulation performance in the following two scenarios. The white line is the exploration trajectory, which shows that the quadrotor is far away from obstacles and rarely explores an area repeatedly. The map obtained from exploration is very satisfactory, and the whole map is excellently detailed, which proves that our algorithm can manage different sophisticated scenarios.

The average running time of each component of our system is presented in Table II, Which clearly reveals that our system consumes time in milliseconds, with a total time of 43.4 ms in the simple maze environment, and as the complexity of the scenario increases, it still maintains a high computing power of 53.9 ms in the complex maze environment. It is worth noting that the time consumption of construction of the map is very high, which is the key to the exploration of the problem. Simulation experiments demonstrate that the proposed algorithm can make exploration more efficient with limited arithmetic power and has the ability to explore complex environments without a priori information.

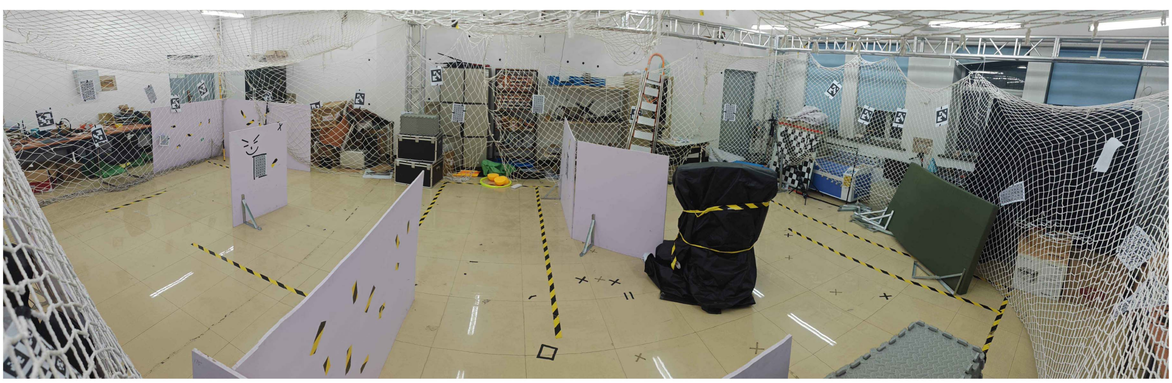
Based on the results of the simulation experiments, we believe that our algorithm outperforms other algorithms due to the fact that we uses more distinct frontier features to guide the UAV, takes into account global information, and does not have redundancy information that affects the speed of decision making, such as viewpoints obtained from random sampling, and so on.

B. Real Word Experiment

The EFP was implemented into real devices to further verify the efficiency and feasibility of the proposed algorithm. The compact quadrotor used for real-world experiments utilized PX4



(a) The map built online and execution trajectory for real-world experiments



(b) Indoor environments for real-world experiments

Fig. 6. Operational results and experimental environment of real-world experiments.

CUAV V5 nano as the flight controller, IntellNUC11TNKi5-1145G7@2.6 GHz and 16 GB of RAM with the Ubuntu 18.04 system to run the proposed algorithms in this article, an Intel T265 camera for positioning, and a D435i RGBD-camera for path planning and perception, as shown in Fig. 5. It is worth mentioning that all our algorithms run on the quadrotor, whereas the localization was derived from the on-board sensor and does not rely on external localization equipment.

The panorama of the real-world experimental environment is shown in Fig. 5(b), where we built a small experimental environment indoors with specific dimensions of 10x5x3m^3. baffle boards were used to construct simulated obstacles that quadrotor may encounter while exploring, increasing the complexity of the experiment. Note that the ArUco markers in the environment are only present to increase the feature points required for localization, and do not actually participate in the entire algorithm.

The real-world experimental results are shown in Fig. 5(a), where the gray trajectory is the trajectory of the quadrotor. The 3D grid map is represented by UFOMap, and the color only represents the height information. To ensure experimental safety with increasing difficulty, we set the sensor range of the quadrotor to 3 m, the maximum linear velocity to 0.5 m/s, the maximum angular velocity to 1 rd/s, owing to camera limitations, the Sensor FoV to 86° $\times$ 57°. The entire experimental environment was efficiently explored in approximately 80 seconds. These real-world experiments demonstrated that the compact quadrotor can avoid obstacles and efficiently complete exploration missions, and that utilizing depth camera to build out entire complex environments can helpful in handling hazardous tasks. The specific experimental video can be found at https://www.youtube.com/watch?v=GJBGfaLJexs. The video shows the running process of the real-world experiment and simulation experiment, the intermediate variables of the algorithm and the running results, so you can observe more details in this video.

V. Conclusion

This research presents an efficient, frontier-based UAV exploration strategy. An efficient representation of the surroundings was employed, based on which rapid frontier extraction was performed, Faster clustering methods were used to obtain viewpoints, which constituted the complete perceptual component. Based on this component, an undirected topology graph was constructed, and a global trajectory optimization strategy that considers the exploration distance, yaw angle and the information gain was proposed to quickly traverse all unknown regions. Meanwhile, the global path planning and local path plannings based on sampling were utilized to enable the quadrotor to explore unknown scenarios.

The proposed algorithm was analyzed and qualitatively and quantitatively compared with three other state-of-the-art, classical algorithms in two different simulation scenarios. Our algorithm provided a significant efficiency improvement. Real-world experiments were also conducted using a compact quadrotor to verify the feasibility of our algorithms. Both the simulation and real-world experiments proved that the algorithm has some applicability value. In future works, strategies related to multi-UAV and multi-UGV cooperative exploration will be explored, as well as algorithmic studies on multi-robot task allocation.

References

[1] H. Shakhatreh et al., "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[2] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *J. Field Robot.*, vol. 37, no. 8, pp. 1363–1388, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21993

[3] W. Greenwood, J. Lynch, and D. Zekkos, "Applications of UAVs in civil infrastructure," *J. Infrastructure Syst.*, vol. 25, 2019, Art. no. 04019002.

[4] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A UAV system for inspection of industrial facilities," in *Proc. IEEE Aerosp. Conf.*, 2013, pp. 1–8.

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

[6] D. Duberg and P. Jensfelt, "UFOMap: An efficient probabilistic 3D mapping framework that embraces the unknown," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6411–6418, Oct. 2020.

[7] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat., 'Towards New Comput. Princ. Robot. Automat.'*, 1997, pp. 146–151.

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

[9] J. Huang et al., "FAEL: Fast autonomous exploration for large-scale environments with a mobile robot," *IEEE Robot. Automat. Lett.*, vol. 8, no. 3, pp. 1667–1674, Mar. 2023.

[10] Z. Meng et al., "A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1680–1687, Jul. 2017.

[11] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, pp. 189–206, 2013.

[12] C. Connolly, "The determination of next best views," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1985, vol. 2, pp. 432–435.

[13] H. H. González-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robot. Res.*, vol. 21, no. 10-11, pp. 829–848, 2002.

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

[15] H. Zhu, C. Cao, Y. Xia, S. Scherer, J. Zhang, and W. Wang, "DSVP: Dual-stage viewpoint planner for rapid exploration by dynamic expansion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 7623–7630.

[16] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3105–3112.

[17] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2064–2076, Jun. 2020.

[18] S. Bai, F. Chen, and B. Englot, "Toward autonomous mapping and exploration for mobile robots through deep supervised learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2379–2384.

[19] H. Surmann, C. Jestel, R. Marchel, F. Musberg, H. Elhadj, and M. Ardani, "Deep reinforcement learning for real autonomous mobile robot navigation in indoor environments," 2020, *arXiv:2005.13857*.

[20] M.-F. R. Lee and S. H. Yusuf, "Mobile robot navigation using deep reinforcement learning," *Processes*, vol. 10, no. 12, 2022, Art. no. 2748.

[21] Y. Cao, Y. Wang, Y. Xue, H. Zhang, and Y. Lao, "FEC: Fast Euclidean clustering for point cloud segmentation," *Drones*, vol. 6, no. 11, 2022, Art. no. 325.

[22] S. Lin, "Computer solutions of the traveling salesman problem," *Bell Syst. Tech. J.*, vol. 44, no. 10, pp. 2245–2269, 1965.

[23] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "FALCO: Fast likelihood-based collision avoidance with extension to human-guided navigation," *J. Field Robot.*, vol. 37, no. 8, pp. 1300–1313, 2020.

[24] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

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

[26] A. Rasouli and J. K. Tsotsos, "The effect of color space selection on detectability and discriminability of colored objects," 2017, *arXiv:1702.05421*.