UAV-Based Target Search in Maze-Like Terrain

Guanyu Huang [1,*], Jiayin Wang [2], Haozhe Xu [3]

[1] School of Mechanical Engineering, Hebei University of Technology, Tianjin, 300130, China,

[2.]Beijing Aidi School, Chaoyang District, Beijing, 100018, China

[3.]Shenzhen College of International Education, Shenzhen, 518048, China

* Corresponding Author Email: 235731@stu.hebut.edu.cn

Abstract. Owing to their being less expensive to operate, greater accessibility, and inherent ability to maneuver over tough terrain, unmanned aerial vehicles (UAVs) have become an indispensable tool for rescue missions. However, in the cases where the missions require real-time decision-making without prior maps, existing approaches to UAV path planning still lack efficiency. The problem addressed in the paper is the design and assessment of a hybrid path planning algorithm for UAV-supported search and rescue operations adopted in maze-like, two-and-a-half-dimensional (2.5D) environments. We compared the classical algorithms, such as Random DFS, map-based, A-star, and D, with our suggested hybrid approach, which uses both the Frontier and Greedy algorithms, in Python-based simulations. Through this process, we achieved a total success rate (100%) across all algorithms, but we noticed profound differences in path efficiency, which required a lower time budget and computational resources among them. In particular, the hybrid advice had remarkable benefits over the baseline, such as 22% in Steps-to-Target, 46% in coverage speed, and more than 50% in Total Mission Steps. The only point is that the algorithm preserves its real-time feasibility and that the duration of each step is less than 50 ms. This study concludes that the UAV navigation strategy with hybridized frontier and greedy search is a promising and computationally efficient solution for time-critical SAR missions in maze-like environments, with a strong potential application in time-sensitive search and rescue responses.

Keywords: UAV Path Planning, Search-and-Rescue (SAR), Real-time navigation.

1. Introduction

With the rapid technological development of UAVs, their use in search and rescue has become increasingly significant, with UAVs providing better accessibility to areas where the terrain is complex, lower cost in operation, and superior efficiency when operating in comparison with ground vehicles [1]. For instance, in a practical test, a UAV identified 38 out of 42 concealed individuals in the dense forest trials at 17 test sites. Based on the onboard detection and adaptive flight planning in real time, it is shown that this design has an accessibility advantage for environments and operational efficiency in rugged terrains [2].

Many other studies reveal that frontier-based exploration is effective in GNSS-denied areas, which UAVs equipped with lightweight sensors can autonomously navigate to without prior location data [3]. These studies make it evident that UAVs could be systematically deployed for the exploration of unknown terrains.

Many SAR UAV systems are still dependent on map data, pre-planning, and assumptions about the environment, which is largely not the case when using UAVs in search and rescue scenarios. In the case of rescue missions, the UAV would be forced to navigate entirely in real-time without additional GPS data, prior mapping, or obstacle layout predictions, to which SAR drones have yet to be fully developed.

The crucial issue that calls for new UAV path planning techniques in terrains of complicated 2.5D configurations stems from the recognition of human survivors operations in disaster relief, for which the UAV is essential in getting to otherwise inaccessible sites to launch time-critical operations [4]. In case the area is littered with obstacles of all sizes, a quick access to the victims is ensured through efficient navigation by a UAV.

Path planning for UAV SAR cases implies dealing with strengths as well as weaknesses. Complex terrain and an abundant number of obstacles lead to significant inefficiencies in conventional algorithms, as unknown target whereabouts and extensive search areas accompany uncalculated expense of time and electricity.

Although Dijkstra's algorithm, the A* algorithm, and the D algorithm are subjected to wide penetration in the domain of path planning, each of these is characterized by distinct disadvantages. The well-known Dijkstra algorithm is highly time-consuming but restricted to certain static environments, where the number of nodes is very large or where the environment is dynamic [5]. While A* does an acceptable job to create heuristic function, its ability to do so in high-dimensional (or dynamic) scenarios may be very limited, restricting its performance. This is why variants like Theta* have been proposed to improve path smoothness in grid-based implementations [6]. The execution of D* is found to be more suitable for dynamic environments, but when the environment becomes too dynamic or experiences frequent changes, it may result in significant computational overhead [7].

These challenges are more prominent in 2.5D terrain. Current research mainly focuses on 2D or simple dynamic terrains, and there is a lack of efficient algorithms for 2.5D terrain with high obstacles and unknown target locations. However, an efficient path planning algorithm is crucial, especially for disaster relief, missing person search, military reconnaissance, and monitoring tasks.

This work makes several contributions. First, it puts forward a hybrid exploration method, and this method combines systematic frontier expansion with greedy prioritization at the same time. Second, it sets up a strict, index-based evaluation plan, which is specifically designed for 2.5D maze-like terrains. Third, it verifies the performance of the proposed algorithm, and this algorithm can work in real time under the hardware conditions of UAVs.

2. Methods

2.1. UAV Platform and Experimental Environment Setup

Simulation only (Python). All experiments are conducted in Python (NumPy, SciPy, Matplotlib). Grid and units. Grid size: N × N (reported in Results). Cell size: Δ meters per cell (all x, y, z coordinates are in meters). Occupancy grid G: 0 = free, 1 = wall. Height map Z (2.5D): free cells range from 0 to 13.5 m (0 to 0.9 × 15); all walls are set to 15 m (highest obstacles, non-flyable). UAV geometry & safety. 4-inch class quadrotor; safety diameter ≈ 0.22 m. Inflation radius: Obstacles are expanded by to enforce a keep-out buffer. Sensing & motion. 8-connected moves (north, east, south, west, NW, SW, NE, SE), one cell per step (one "action"). Local sensor reveals cells within Manhattan radius (default) as FREE or WALL on the agent's internal map. The agent has no prior map. Start/target and validity. Start at a free cell near the origin (fallback to nearest free if needed). Target is sampled from free cells with minimum Manhattan separation ≥ αN (default). A hidden connectivity check ensures a path exists on the ground-truth grid. Reproducibility. Each dataset run records N, Δ, ρ, start/target, and the random seed. Figures and NPZ/CSV outputs are saved.

2.2. Maze-Like Terrain Modeling (2.5D Representation)

To achieve a compromise between realistic elevation representation and computational efficiency, the models utilize 2.5D representations, preserving realistic elevations while avoiding the 3D volumetric planning requirements. Specifically, each cell in the model retains occupancy properties and is assigned a specific height: 15 m for walls and 13.5 m for free-space ground terrain to accommodate vertical clearance. This modeling approach surpasses any 2D representation in addressing the undulating nature of SAR scenes while also alleviating the heavy computational requirements associated with 3D planning systems.

In fact, semi-3D modeling representations similar to this have recently been suggested for UAV exploration, as 2.5D representations can provide a viable compromise between accurate geometry and real-time performance in complex terrains [8][9].

In order to estimate the robustness of each algorithm under different density of obstacle, our research generated three representative maze-like terrain configurations.

As can be seen in Figure 1, the dense condition features clustered obstacles and narrow corridors that create high navigation difficulty.

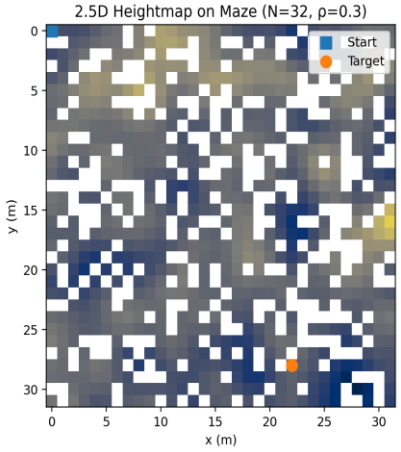

Figure 1. Map sample of obstacles in ‘dense’ conditions

Figure 2 shows the core condition, where obstacle distribution is moderate and provides a balanced exploration environment.

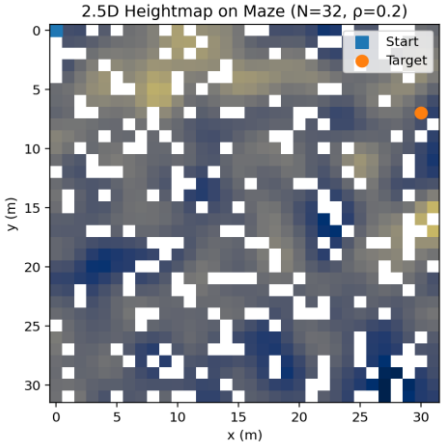

Figure 2. Map sample of obstacles in ‘core’ conditions

In contrast, Figure 3 illustrates the sparse condition, which contains wide open spaces and fewer barriers, used to assess performance in low-density terrains.

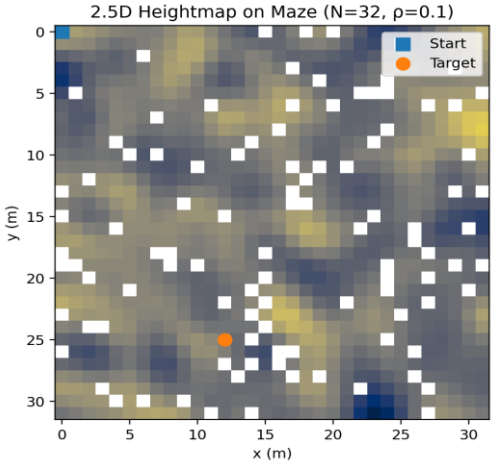

Figure 3. Map sample of obstacles in ‘sparse’ conditions

2.3. Path Planning Algorithm Design

2.3.1 Baseline Algorithm Selection

Nearest-Frontier (NF) is established as a classic frontier-based exploration baseline. For this method, frontiers are first defined as known-free cells that share a border with at least one unknown cell. At each exploration step, these frontiers are identified, and the exploration agent is directed toward the nearest frontier. The “nearest” frontier is determined by calculating the shortest path on the graph of currently discovered free space. In practice, the Breadth-First Search (BFS) algorithm is employed to compute this shortest path, which ensures the planning process adheres to all known obstacles and avoids making any assumptions about unseen space. The NF approach, characterized by its simplicity and resilience in extending the boundaries of environmental understanding, bears two main limitations: it is prone to oscillations when investigating cul-de-sacs and usually cannot differentiate the amount of information that can be acquired across different frontiers.

Randomized Depth-First Search treats the explored free-space as a graph and follows a depth-first policy with randomized neighbor ordering; when the search gets stuck at a dead-end, the algorithm backtracks to the last branching point. This baseline has a meager per-step compute requirement, is efficient in penetrating narrow corridors, and offers effective control for "uninformed exploration." Its disadvantages include bad detours before detection/return, coming across long loops, and higher revisit rates in cases of loopy layouts.

2.3.2 Algorithm Improvement Strategies (Frontier-Based + Greedy Hybrid)

We build on Frontier-Based exploration (systematically expanding known free space toward frontiers—free cells adjacent to unknown) [8], but add a Greedy frontier selector to avoid wasting steps on low-value frontiers.

Such notions are adopting the direction used in the latest UAV path-planning frameworks such as GO-FEAP [8], utilizing omission-aware frontier prioritization and structure hierarchy of frontiers for FUEL [9]: all these make effective exploration possible. These studies further support the effectiveness of combining systematic frontier expansion with heuristic-based prioritization in real-time UAV navigation.

Internal map. Maintain three states: UNKNOWN, FREE, WALL. Frontier detection. Mark FREE cells with at least one UNKNOWN neighbor as candidates. Scoring. For each frontier f: where d is the shortest path on the known-FREE map (via BFS), IG(f) is the information gain (number of UNKNOWN cells within r_s if standing at f), and $\lambda > 0$ balances “near” versus “informative.” Execution. Move one step toward the best frontier, reveal with the sensor, and re-evaluate. If the target becomes reachable on the known map, go directly; then compute a return-to-base path on the discovered map.

2.4. Simulation and Evaluation Metrics

2.4.1 Success Criterion and Budget

A trial is considered successful if the agent meets two criteria: first, detecting and reaching the target, and second, returning to the base within the predefined mission step budget (B) ； the budget B is set as $B = 2N^2$, where N^2 approximates the number of cells in the 16×16 grid used in the simulation, and this allocation roughly provides N^2 steps for the agent to explore the terrain and another N^2 steps for a potentially winding return path.

2.4.2 Path Efficiency

Steps-to-Target (ST) is defined as the number of actions the agent takes from the start until it first enters the target cell. Total Mission Steps (TMS) refers to the total number of actions the agent completes from the start to the target and then back to the start. Redundancy Ratio (RR) is calculated as the ratio of the agent’s total actions to the number of unique free cells it visits. Coverage Speed (CS) represents the number of new free cells the agent discovers per 100 actions until it detects the target. Normalized Excess (NE) is computed using the formula NE= "ST-L" /"L" , where L is the

shortest path from the start to the target on the map discovered by the agent at the time of target detection.

2.4.3 Computational Cost

Decision Time (ms/step) is defined as the median wall-clock time the agent consumes for each action selection during the mission. Nodes Expanded per Decision refers to the median number of grid cells expanded by the agent in local planning when making each decision. Peak Working Set represents the maximum size of internal data structures including the frontier list and the open/closed sets used in path planning algorithms. Replan Frequency is calculated as the number of local replans the agent performs per 100 actions, where a very high frequency may indicate oscillation in decision-making and a very low frequency may suggest rigidity in adapting to new environmental information.

2.4.4 Target Search Success

Success Rate (SR) is defined as the percentage of trials where the agent both detects the target and returns to the base within the preset mission step budget (B)； Timeout Rate refers to the percentage of trials in which the agent's total actions exceed the mission step budget B； Failure Modes are categorized into three types. First: timeout occurring before the agent detects the target. Second: the agent detects the target but fails to find a valid return path to the base. Third, failures caused by other reasons.

3. Results

3.0 Sample trajectories under the core environment. To visually illustrate the navigation behavior of different algorithms in the same core maze environment, sample trajectories are shown in Figures 4–6.

Figure 4 presents the exploration pattern of the normal Frontier method, which follows a systematic boundary-expansion strategy but occasionally revisits previously explored corridors.

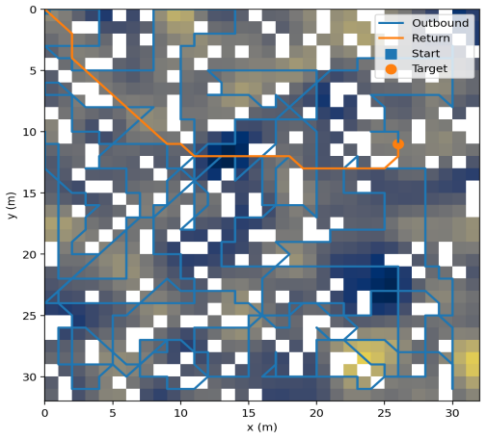

Figure 4. Normal Frontier

Figure 5 demonstrates the Frontier–Greedy Hybrid algorithm, whose trajectory appears more directed toward informative frontiers, avoiding unnecessary detours.

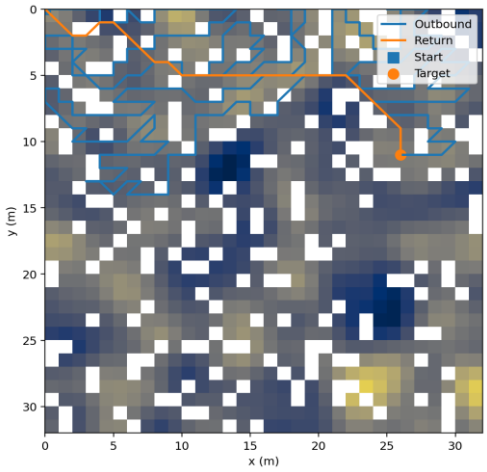

Figure 5. Frontier-Greedy Hybrid

Figure 6 shows the randomized DFS baseline, where path wandering and backtracking are frequent, resulting in a longer overall route.

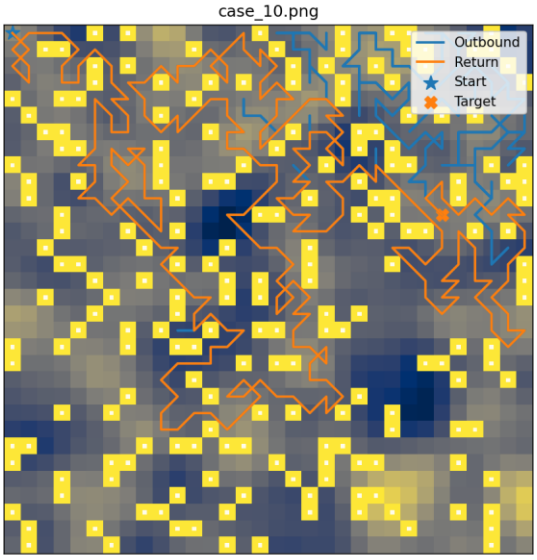

Figure 6. rDFS

3.1. Algorithm Performance in Simulated Maze-Like Terrains

In all three types of terrain (Core, Dense, Sparse), the success rate for detection as well as for return was 100% for all algorithms. The implication is that it was corroborated even with the baseline RDFS that task completion under budget was guaranteed. Nevertheless, the algorithms differed considerably in terms of path efficiency and computational cost.

With respect to path efficiency, Frontier recorded the lowest mean ST (approximately 281 steps) and TMS (approximately 305 steps) in Core mazes, slightly ahead of Hybrid (approximately 289 steps / approximately 318 steps), which was attained through its capability to continue in a systematic manner to the frontier without wasting time in seeking out detours in relatively open layouts. In the mazes of Dense type, Frontier-wise reductions in ST were approximately 22% (266 vs. 341 steps), and TMS was approximately 17% (304 vs. 368 steps), which indicates that the greedy algorithm can avoid unnecessary visits to uninformative frontiers when obstacle density is high. Once more, Hybrid outperformed Frontier in terms of ST by about 12% (286 vs. 326 steps) in Sparse mazes, supporting that narrowing down the frontiers still led to better performance even when long corridors dominated. Particularly, RDFS incurred about 645 steps in each type of terrain, consisting of Core (ST = 645 steps).

Concerning the exploration quality, CS (Coverage Speed) showed the highest speed in the case of Hybrid, being about 46.5% faster discovery in Sparse terrains. Redundancy Ratio (RR) was the lowest for the Frontier in Core/Dense terrains, with an approximate range of 1.13–1.23, which shows stable coverage, while the Hybrid technology minimized redundancy in the Sparse terrains with the average ratio of redundancy around 1.10. Around the core, Rough (Core: 8.20, Dense: 5.70, Sparse: 12.0) NE (Normalized Excess) was almost constantly the lowest (Hybrid). Hence, the detected routes were closer to the theoretical optimum L*.

About the computational cost, Decision Time was marginal for the Frontier in Core/Dense terrains (≈5.1–5.2 ms/step) and for the RDFS in Sparse terrains (≈5.8 ms/step). Hybrid took slightly longer per step (≈11.6–33.4 ms) due to frontier scoring overhead but these values remain well within UAV real-time control cycles (<50 ms). Moreover, when examining the nodes expanded and peak size of frontiers/open list, the numbers remain fairly modest (≈56–68), which suggests that there was no computational tractability problem, even in Dense scenarios.

3.2. Comparison with Baseline Algorithms

The comparison highlights the trade-off between efficiency and computational complexity: RDFS is the basic lightweight approach that requires little computational resources but results in much longer paths (e.g., approximately 2.3 times higher ST in Core compared with Frontier). In particular, Frontier exhibits Core layouts that are quite robust, especially those that avoid systematic repeated trails, featuring expensive paths or going around when moving in Dense mazes. Hybrid, on the other hand, is a combination of the best features of them both, namely shorter paths and higher coverage speed in Dense/Sparse conditions, whilst requiring still moderately huge computational cost. Compared to Frontier, Hybrid delivered a 22% increased ST and 12% maximum in Dense and Sparse, respectively, all having significant CS improvements. Yet, it seems that in Core scenarios, Frontier continued to have a slightly superior ST and RR ratio.

Like this, as it concerns future guided UAV exploration hybrid scenarios, similar considerations have been noticed in the latter, meaning that the greedy-enhanced approaches outshine the conventional methods in both cluttered and partially observable conditions while having a practically appropriate computational cost within limits [10].

3.3. Statistical Analysis of Path Efficiency and Success

For the sake of estimating performance variances, the mean ± 95% CIs were set up: in Dense terrains, the confidence intervals of Hybrid and Frontier did not overlap, which statistically proved the significance in the detection time; in Sparse conditions, Hybrid had a small advantage, but the CIs still showed the same tendency for Hybrid to constantly gain in CS and NE; in Core environment, Frontier claimed significant ST and RR advances which fell inside overlapping intervals, thus it was concluded that both methods worked comparably in easy layouts; in almost all conditions, the success rate was 100%, and no timeouts appeared, confirming mission reliability within set budget parameters.

3.4. Key Findings

Hybrid reduces significantly the Steps-to-Target (ST) for complex terrains, where quick target identification is vital for UAV-SAR and Frontier remains efficient enough in simplified mazes, with no intensive computing process needed; RDFS indicates the trade-off idea of low computational load and inefficient environmental controls but untimely mission objectives, while insisting upon guaranteeing success of the mission; in general, Hybrid sees the balance between the idea of exploration efficiency and situation feasibility of real-time, arousing the possibility of UAV target search in maze-like environment.

The summary of quantitative results of all algorithms in different topologies can be seen in Table 1.

The means ± 95% confidence intervals of different metrics are presented in the table: Steps-to-Target (ST), Total Mission Steps (TMS), Coverage Speed (CS per 100 steps), Normalized Excess (NE), and Redundancy Ratio (RR).

This study shows that Frontier-Greedy Hybrid outperforms the commonly used baseline methods in terms of ST and TMS in dense and sparse terrains, as those methods had better path efficiency and exploration speed than Frontier and rDFS, respectively.

Besides that, the Hybrid showed comparatively moderate computational costs, as evidenced by rather equal RR and NE scores.

The consistency of both quantitative results and qualitative trajectory observations validates the model that was used and shows that Hybrid exhibited the fastest exploration time in which realistic UAV-SAR conditions were taken into consideration.

Table 1. Summary metrics (mean ± 95% CI) by scenario and algorithm

Terrain	Algorithm	ST (meters)	TMS (meters)	CS_per100	NE	RR
Core	frontier	280.71	305.08	130.44	8.66	1.13
Core	Hybrid	302.9	337.28	167.94	8.16	1.16
Core	RDFS	776.11	1118.97	110.83	23.6	1.99
Dense	frontier	341.08	368.38	112.16	9.45	1.23
Dense	Hybrid	281.53	326.56	138.94	5.66	1.27
Dense	RDFS	502.17	683	101.71	12.97	1.99
Sparse	frontier	345.7	371.22	139.85	13.61	1.11
Sparse	Hybrid	296.08	324.75	204.85	12.03	1.1
Sparse	RDFS	538.61	904.13	144.15	20.08	1.99

4. Conclusion

Hence, the work defined UAV path planning as a traversing task in maze-like 2.5D terrains. The simulation results demonstrated that the Frontier-Greedy hybrid, which we proposed, outperformed the classical methods for both the Dense and Sparse scenarios, even though all algorithms achieved 100% mission success. The accuracy was improved by 22% of Steps-to-Target, the coverage speed nearly doubled, and the Total Mission Steps halved in comparison with RDFS while complying with real-time decision-making requirements.

The paper provides a contribution by proposing a twofold exploration strategy based on systematic frontier expansion and greedily selected priorities, the main purpose of the method is verified by an efficient metric-driven evaluation of maze terrains in a 2.5D environment, and the hardware and time constraints for UAV-scale implementation of the algorithm are validated.

The next steps of research should be experimenting with the hybrid algorithm on more diverse and complex terrains, using inspiring UAV real platforms with onboard sensors while included, and extending the algorithm to multiple UAVs coordination that is needed for the large-scale SAR missions. These steps will fill the gaps between the design based on simulation and the actual deployment involving field conditions, which will make such UAVs SAR systems much more versatile in the response to real-world disasters.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] LYV M, ZHAO Y, HUANG C, et al. Unmanned Aerial Vehicles for Search and Rescue: A Survey[J]. Remote Sensing, 2023, 15(13): 3266.

[2] SCHEDL D C, KURMI I, BIMBER O. An Autonomous Drone for Search and Rescue in Forests using Airborne Optical Sectioning[J]. arXiv preprint, 2021.

[3] ALLAN S, BARCZYK M. A Low-Cost Experimental Quadcopter Drone Design for Autonomous Search-and-Rescue Missions in GNSS-Denied Environments[J]. Drones, 2025, 9(8): 523.

[4] PAL B, PAUL S, TURUK A K, et al. UAV aided post disaster relief networks using blockchain technology: Challenges and solutions[C]//2024 Eighth International Conference on Parallel, Distributed and Grid Computing (PDGC). IEEE, 2024: 804-809.

[5] MENG W, ZHANG X, ZHOU L, et al. Advances in UAV path planning: A comprehensive review of methods, challenges, and future directions[J]. Drones, 2025, 9(5): 376.

[6] NASH A, DANIEL K, KOENIG S, et al. Theta*: Any-angle path planning on grids[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2007: 1177-1183.

[7] FERGUSON D, STENTZ A. Using interpolation to improve path planning: The Field D* algorithm[J]. Journal of Field Robotics, 2006, 23(2/3): 79-101.

[8] ZHANG W, YU W, ZHUANG L, et al. GO-FEAP: Global optimal UAV planner using frontier-omission-aware exploration and altitude-stratified planning[J]. arXiv preprint arXiv:2310.15931, 2023.

[9] ZHOU B, ZHANG Y, CHEN X, et al. FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning[J]. arXiv preprint arXiv:2010.11561, 2020.

[10] ZHOU Y, YAN L, HAN Y, et al. HFCH: Hybrid frontier guided fast UAV autonomous exploration for complete and high-quality mapping in unknown environment[J]. Advanced Engineering Informatics, 2026, 69(Part A): 103834.