Upgraded trajectory planning method deployed in autonomous exploration for unmanned aerial vehicle

Tong Zhang¹, Jiajie Yu², Jiaqi Li² and Jianli Wei¹

Abstract
Autonomous exploration is grounded on target decision and trajectory planning, which is widely deployed on unmanned aerial vehicles. However, existing methods generally only focus on the exploration effect of target decision but neglect the environment information gained with trajectory planning during flight, resulting in redundant exploration trajectories and low exploration efficiency. This article proposes an upgraded method of trajectory planning for autonomous exploration work. We design a fresh cost term considering the frontier information in the part of trajectory optimization. Besides, yaw angles are planned independently to catch more environment information during flight. We present extensive simulations and real-world tests. The results show that our proposed method reduces the exploration cost time by 10–15% compared with the previous one.

Keywords
Autonomous exploration, trajectory planning, trajectory optimization, yaw planning

Introduction
Unmanned aerial vehicles (UAVs), especially quadrotors, are popular for their agility and ability to work at heights. Therefore, they are widely used in various fields, such as search and rescue, intelligence reconnaissance, and so on. In the above applications, autonomous exploration work plays an important role. As Figure 1 shown, autonomous exploration requires UAVs to make decisions for local targets to complete, plan effective trajectories to reach, and get the global environment information to map.

Exploration planning methods have recently received increased attention. Autonomous exploration can be divided into three parts, namely viewpoints decision, global tour path planning, and trajectory planning and optimization.

Viewpoints decision means robots autonomously determine the next best viewpoints (NBVs) (obtained position and yaw angle) that can gain the maximum environment information according to the current local map. Some of the previous work only focus on the local exploration speed as fast as possible. Yamauchi¹ first introduces a new approach for exploration based on the concept of frontiers and simply treats the nearest frontier point as a local target. Cieslewski et al.² extend the classical frontier-based exploration method which enables

¹ Unmanned System Research Institute, Northwestern Polytechnical University, Xi’an, China
² School of Astronautics, Northwestern Polytechnical University, Xi’an, China

Corresponding author:
Jiajie Yu, School of Astronautics, Northwestern Polytechnical University, 127 West Youyi Road, Beilin District, Xi’an, Shaanxi 710072, China.
Email: yujiajienpu@foxmail.com
exploration at high speeds. It selects a goal frontier in a way that minimizes the necessary change in velocity, making the total exploration time shorter at the cost of a longer path. Above methods\textsuperscript{1–3} have been proposed to improve the efficiency of autonomous exploration. Among them, exploring motions planners decide on the next viewpoints in different manners, such as greedily navigating to the closest unknown region for maximizing the immediate information gain\textsuperscript{1} or refining a local set of viewpoints for a more efficient tour path.\textsuperscript{3} Yu et al.\textsuperscript{4} further propose a TSSEAE (Two-Sided Cross-Domain Collaborative Filtering model based on Selective Ensemble learning considering both Accuracy and Efficiency) method based on the work\textsuperscript{5,6} to convert the problem of selecting combinations to that of selecting classifiers, which provides new ideas for clustering and classification of frontiers. In fact, the concept of NBVs has proposed a long time ago, which was explained as a good sequence of range-image views for obtaining a scene as complete as possible it can.\textsuperscript{7} Bircher et al.\textsuperscript{8} first used NBVs in 3D exploration work in which it expands rapidly exploring random trees (RRTs) with free space. Based on it, some extensive work was proposed, which considered the visual importance of different objects,\textsuperscript{9} the uncertainty of localization,\textsuperscript{10} and inspection tasks.\textsuperscript{11}

In the classic frontier-based exploration method, Zhou et al.\textsuperscript{3} recognize frontiers as known-free voxels right adjacent to unknown voxels and generates candidate viewpoints for a frontier cluster (black circle in Figure 2). Note that, unlike previous approaches which simply navigate to the center of a cluster, points are sampled uniformly at the average position of the cluster. For each of the sampled points lying within the free space, with the yaw optimization method similar to Witting et al.,\textsuperscript{12} the yaw
angle is determined according to the maximizing sensor coverage to the cluster by using. In this article, we follow this strategy to select and decide NBVs.

After getting a series of NBVs, the crux of the problem becomes how to plan a path as short as possible to connect the above points. This is what path planning does and is generally modeled as a traveling salesman problem (TSP).

Inspired by Meng et al., Zhou et al. reduce it to a standard asymmetric TSP (ATSP) and recompute an open-loop tour starting from the current position and passing viewpoints at all clusters. Nonetheless, the initial tour path involves only a single viewpoint of each cluster, which is not necessarily the best combination among all viewpoints. Therefore, by using the graph search approach, Zhou et al. optimize the previous global tour path to further improve the exploration rate.

Discrete viewpoints and rough global tour paths can’t be used for UAV motion execution, so continuous trajectories are required for smooth navigation, which is the work of trajectory planning and optimization.

Quadrotor online planning methods require a faster generation of a more safe, smooth, aggressive trajectory. Among the methods proposed, the local trajectory planning method based on the gradient information becomes the mainstream for UAV local trajectory generation, for its simple but efficient function to smooth a trajectory and improve its clearance.

Euclidean signed distance field (ESDF) is first introduced in robotic motion planning by Ratliff, which calculates the information of distance to the closest obstacle. Ratliff et al. and Kalakrishnan et al. further proposed discretization optimization of trajectory based on ESDF but is not applied to drones for its dynamical constraints. Therefore, Oleynikova et al. proposed a continuous-time polynomial trajectory optimization method for UAV planning. However, it causes a heavy computation burden because of the involved integral of the potential function and has a low success rate of around 70%. Gao et al. significantly increases the success rate by initializing a collision-free path as the front-end. And the performance is further improved after taking the initial collision-free path into account kinodynamic constraints.

A robust and efficient quadrotor motion planning method for fast flight in 3D clutter environments was adopted by the frame. It adopts a kinodynamic path searching method to find an initial trajectory and improves the smoothness and clearance of the trajectory by a B-spline optimization using ESDF information. Zhou et al. proposed an ESDF-free gradient-based planning framework. Its collision term in the penalty function is formulated by comparing the trajectory in collision with a collision-free guiding A* path instead of ESDF, making the computation time reducing mostly. Unfortunately, the planning trajectories are not suitable for autonomous exploration because of the instability caused by aggression, which acquires an extremely high demand for control tracking. Once tracking is poor, collisions and even crashes may occur, which can be fatal to autonomous exploration. Therefore, in this article, our planner takes inspiration from the work at a cost of more planning time in exchange for security, and takes consideration of unknown environment information into cost function in trajectory optimization, which makes the planner better for autonomous exploration.

Besides, we change the way of yaw planning to keep the field of view (FOV) of UAVs heading to an unknown area as possible it could. However, the above planning methods ignore the critical role of the yaw angle in the acquisition of environmental information as it is only applied in purely autonomous flight. But it is important to note that yaw is the window for environment-UAV interaction in autonomous exploration, which is focused on in this article.

Benefiting from the flatness of the quadrotor dynamics, yaw can be planned independently of the 3D position. However, since autonomous navigation work only has strict requirements on position for obstacle avoidance, the current trajectory planning work is simplified for yaw planning to improve efficiency.

A representative local planner obtains the yaw angle by sampling the position of the neighboring trajectory points and solving the yaw by the arctangent. And the yaw will be parallel to the UAV’s velocity. Another local planner is almost the same as Zhou et al., but the difference is that Zhou et al. impose a strict limit on the yaw at the end. Based on the yaw limit, the classic framework for autonomous exploration adopts the method in trajectory planning and can use the yaw information of the viewpoints.

The above work has completed autonomous exploration to a certain extent and improved exploration efficiency. However, existing methods ignore the role of the flight process to viewpoints but only focus on the viewpoint environment gain, resulting in repeated inefficient planning and redundant trajectory. It wastes the positive function of trajectory planning by reducing the exploration cost time and making exploration less efficient.

Motivated by the above facts, we implemented a smarter method in trajectory planning, which is based on a hierarchical framework using an incremental frontier structure. By designing a fresh cost term called environment cost in the step of trajectory optimization and planning yaw angles independently with a more flexible strategy, we maximize the gain of environment information during flight.

We have evaluated our method with previous work in simulation with unknown maps, showing that in all cases our method could finish autonomous exploration work in a shorter time. Furthermore, we have verified our method in a cluttered environment in the real world, proving that it can be stably and efficiently applied to real-world scenarios. The contributions of our work are summarized as follows:
1. A new cost term in trajectory optimization, which optimizes a tour trajectory by gaining more environment information in flight.
2. A yaw planning strategy, which simply gets a set of yaws heading to the unknown area and reduces waste of exploration resources.
3. Extensive simulation and real-world test that verifies our method. The results show that this method can effectively reduce the exploration cost time by 10–15%.

The article is organized as follows: The first and second sections describe the introduction and overview of the autonomous exploration frame. The third section explains a fresh environment cost term in trajectory optimization. The fourth section explains a creative yaw planning strategy. The fifth section describes the experimental verification including simulation and real-world. The final section is the conclusion of our work.

**Frame overview**

In this article, we refer to the hierarchical framework proposed by Zhou et al. and follow its processing steps to improve autonomous exploration work.

As illustrated in Figure 3, the framework is composed of an incremental update of the frontier information structure and a hierarchical exploration planning approach. Once the map is indicated by the grid, the framework will detect whether any frontier clusters are changed and update the frontier information. After ensuring the sets of viewpoints by frontier clusters, it will plan a global exploration tour path, optimizes local viewpoints, and generates the trajectory navigating to the best viewpoint. When there is no frontier on the map, the exploration is considered fully completed.

As Figure 4 shown, we will focus on the work of trajectory planning, which includes trajectory optimization and yaw planning, after receiving the selected viewpoint information.

**Upgraded trajectory optimization deployed in exploration**

In this section, we will reveal our work that sets a new environment cost to make UAV gain more environmental benefits during the flight.

The framework we referred to Zhou et al. generates smooth, safe, and dynamically feasible B-spline trajectories and further optimizes all parameters of B-splines based on the method.

The method uses the classical front-end and back-end planning way to finally generate a high-quality trajectory, which is a polynomial in mathematical essence. The hybrid-state A* search, which is first applied to autonomous vehicles, is used for the front-end kinodynamic path searching. It searches for a safe, kinodynamic feasible but suboptimal trajectory that is minimal concerning time duration and control cost in a voxel grid map. Unfortunately, the path is so close to obstacles that can’t be executed due to the danger. So it’s necessary to improve the smoothness and clearance of the path in the back-end trajectory optimization.
A B-spline is a piecewise polynomial uniquely determined by its degree $p_b$, a set of $N + 1$ control points \( \{ Q_0, Q_1, \ldots, Q_N \} \), and a knot vector \( \{ t_0, t_1, \ldots, t_{M} \} \), in which $Q_i \in \mathbb{R}^d$, $t_n \in \mathbb{R}$, and $M = N + p_b + 1$. A B-spline trajectory is parameterized with time $t$, where $t \in [t_0, t_{M-p_b}]$.

For a uniform B-spline, each knot $\Delta t_m = t_{m+1} - t_m$ has the same value. This is the basic principle of the B-spline representing polynomial trajectories. Our work focuses on the cost function in optimization, so we recommend readers to read\(^2\)\(^0\) for more details.

In our work, the optimization problem which bases on Zhou et al.\(^3\) and creatively adds a fresh environment cost term $f_e$ is formulated as follows:

\[
\arg \min f_{all} = \lambda_s f_s + \lambda_c f_c + \lambda_d f_d + \lambda_b f_b + w_t T + \lambda_e f_e
\]

where $\lambda_s$, $\lambda_c$, $\lambda_d$, $\lambda_b$, $w_t$, $\lambda_e$ are the weights for each cost term.

$f_s$ is the band smoothness cost similar to Quinlan and Khatib\(^2\)\(^4\) and Zhu et al.\(^2\)\(^5\)

\[
f_s = \sum_{i=p_b-1}^{N-p_b+1} ||(Q_{i+1} - Q_i) + (Q_{i-1} - Q_i)||^2
\]

This formulation views a trajectory as an elastic band,\(^2\)\(^4\) where each term $F_{i+1,i} = Q_{i+1} - Q_i$ and $F_{i-1,i} = Q_{i-1} - Q_i$ are the joint force of two springs connecting the nodes $Q_{i+1}, Q_i$ and $Q_{i-1}, Q_i$, respectively.

The collision cost $f_c$ is the important term to force the trajectory to avoid a collision, which is formulated as the repulsive force emerged from the distance to the closest obstacle acting on each control point

\[
f_c = \sum_{i=0}^{N} F_c(d(Q_i))
\]

where $d(Q_i)$ is the distance information given by ESDL. A $d_{thr}$ can be set to specify the threshold of obstacle clearance, so the differentiable $F_c$ can be defined as follow

\[
F_c(d(Q_i)) = \begin{cases} (d(Q_i) - d_{thr})^2 & d(Q_i) \leq d_{thr} \\ 0 & d(Q_i) > d_{thr} \end{cases}
\]

As the quadrotor dynamics are differentially flat,\(^2\)\(^2\) we can ensure the feasibility by restricting the higher order derivatives of the trajectory on every single dimension. $v_{ij}$, $a_{ij}$ are the $i$th velocity and acceleration in each independent dimension. $f_d$ is the penalty to ensure dynamic feasibility involving $f_c$ and $f_a$ which penalize infeasible velocity and acceleration

\[
\lambda_v f_v + \lambda_a f_a
\]

\[
\sum_{\mu \in [x,y,z]} \sum_{i=0}^{N-1} F_v(v_{ij}), \quad f_a = \sum_{\mu \in [x,y,z]} \sum_{i=0}^{N-2} F_a(a_{ij})
\]

\(\lambda_v, \lambda_a\) are the weights of responding dynamic feasibility cost terms. And the velocity and acceleration are penalized along with the exceeding maximum allowable value $v_{max}$ and $a_{max}$, which can be preset according to the agile of the drone in reality.

\[
F_v(v_{ij}) = \begin{cases} (v^2_{ij} - v^2_{max})^2 & v^2_{ij} > v^2_{max} \\ 0 & v^2_{ij} \leq v^2_{max} \end{cases}, F_a(a_{ij}) = \begin{cases} (a^2_{ij} - a^2_{max})^2 & a^2_{ij} > a^2_{max} \\ 0 & a^2_{ij} \leq a^2_{max} \end{cases}
\]

$f_{hr}$ is the boundary cost for smooth motion at the start and end state. It denotes the start of the instantaneous by 0th and 2nd order derivatives are set at the start of the instantaneous state. $Q_{c,i}$ is the $i$th control point of the $c$th piecewise polynomial trajectory and $Q_{c,i}$. $Q_{c,i}$ are its first and second-order derivatives. $X_0$ is the current starting position. And the viewpoint $X_{opt}$ is taken into account the penalty of the boundary cost. The final associated cost is

\[
f_{hr} = \frac{||Q_{c,0} + 4Q_{c,1} + Q_{c,2} - X_0||^2}{6} + \frac{||Q_{c,0} + 4Q_{c,1} + Q_{c,2} - X_0||^2}{2} + ||Q_{c,N-2} + 4Q_{c,N-1} + Q_{c,N} - X_{opt}||^2
\]

\(T\) is the total trajectory time composed of $\Delta t_b$ and the number of B-spline segments is $N + 1 - p_b$

\[
T = (N + 1 - p_b) \cdot \Delta t_b
\]

The above are all the original cost terms of Zhou et al.\(^3\) They are inspired by trajectory planning, which means paying more attention to how to reach the observation point more safely, smoothly, and feasibly. However, they all ignore the environment gain during flight, which is the difference between exploration and planning. Flight—the process of executing trajectory, accounts for more than 90% of the whole exploration work. Therefore, we creatively propose a method considering the in-flight environment in the optimization, which makes the trajectory closer to the unknown area.

The incremental frontier information structure of the framework can get the frontier information representing the area of interest in the exploration space. And we can get a cubic defined by a series of frontier points. We use the center of the cubic $X_{ct}$ to specifically indicate the area can get a lot of environment gains and optimize the control points in the middle to make the trajectory closer to $X_{ct}$ without influencing the original trajectory’s start and end points.

So we design the environment penalty cost term as follow

\[
f_e = \sum_{i=1}^{N-1} F_c(Q_i)
\]
where $F_e$ is expressed as the square of the distance between the control point $Q_i$ and the center of the frontier cube

$$F_e(Q_i) = (Q_i - X_{ct})^2$$  \hspace{1cm} (11)

The formula is inspired by the artificial potential field method proposed by Khatib,\textsuperscript{26} which imagines that the center of the frontier cube generates a gravitational field that attracts the control point $Q_i$.

The formulation of the complete objective function is written as equation (1). And the optimal value of control points $Q_i$ is calculated by the gradient descent method with the general nonlinear optimization solver called NLopt. It is worth mentioning that increasing the weight of a parameter in equation (1) can improve the relevant characteristics of the trajectory accordingly, which we will explain in more detail in the “Simulation and analysis” section.

As shown in Figure 5, the control points are attracted to the region closer to the unknown environment after considering the frontier information.

In the real world, UAVs can get more environment information through sensors under the constraint of FOV. Verified by simulation, the fresh strategy can reduce repeated exploration in a certain area, especially the complex corner.

In Figure 6, we built one of the classic and common indoor exploration scenes to show the beneficial effect. In the original work, without the environment penalty term in optimization, UAVs can pass through a series of viewpoints smoothly and safely. However, there will be some small areas being ignored, which makes the UAV have to plan an additional trajectory to explore these areas again for fuller map coverage. In conjunction with the yaw planning strategy in the “Experiment results” section, our work solves this problem and reduces the whole path length and exploration time. The performance details in the simulation will be shown in the “Experiment results” section.

**Exploration yaw planning**

As the quadrotor dynamics are differentially flat,\textsuperscript{22} we can independently plan the yaw without being limited by the 3D position.

The yaw is the eye of the quadrotor to obtain environment information, and its planning should be valued in the exploration work.

According to what we explained above, it’s worth noticing that the best viewpoint includes not only position but also the next best view yaw angle information which is used as the end yaw angle. However, there is no special explanation for the yaw angles during trajectory execution. And Zhou et al.\textsuperscript{3} roughly plan the exploration yaw with the three steps: (1) sample several points in the trajectory; (2) obtain the yaw parallel to the direction of velocity by arctangent; (3) optimize in the same way we mentioned in the “Upgraded trajectory optimization deployed in exploration” section and get the smooth yaw angle of the whole trajectory.

Figure 7 represents a common situation in the original work:\textsuperscript{1} UAV has executed the trajectory and has explored the black frame region; therefore in the following
exploration work, the yaw angle, which faces the black frame still, is of little help in obtaining new environment information. Furthermore, due to the deficient strategy, the drone loses the opportunities to explore the red frame region in the process, making extra planning for this unknown region.

To solve this problem, we have proposed a smarter yaw planning method shown in Figure 8, which considers the beneficial effects of yaw in the flight. As we all know, there are two ways to accomplish the yaw angle transformation, that is, rotation around the inferior angle or the superior angle. So in this article, we judge the two rotation way through current environment information.

The unknown area we desire to explore, which is specified as the frontier center the same as the “Upgraded trajectory optimization deployed in exploration” section, will be first checked before yaw planning. And the yaw planner will determine the rotation direction (reflex or inferior) of the yaw based on the relative position of the center of the frontier. The specific process is shown in Figure 9: (1) ensure the direction (left or right) of the frontier cubic center $X_c$, mentioned in the “Upgraded trajectory optimization deployed in
exploration” section to the current position by current yaw; (2) sample several points in the trajectory according to the number of B-spline segments; (3) assign values to the average increase or decrease in yaw angle for each sampling point based on the start–end yaw difference and the determined direction, and (4) optimize.

Beneficial from this small but caring strategy, UAV can maximize the function of yaw to explore more unknown

---

**Figure 8.** Our proposed strategy considering the frontiers.

**Figure 9.** Proposed yaw planning steps. (a) Ensure the direction of the center $X_{ct}$ for the current position, (b) sample points, (c) assign values for each sampling point, and (d) optimize.
areas and avoid the waste of exploration resources. Then we test our proposed method by simulation in the “Experiment results” section.

### Experiment results

#### Simulation and analysis

We verify our proposed method in simulation and compare it with the original framework FUEL. For the reliability of the comparison, we tested three large-scale maps, which are similar to the indoor office scenario and have complex structures that make exploration challenging. To bring the simulation closer to the real situation, we set the starting point of all tests in the center of the map. In addition, we performed all simulations and real-world tests (“Real-world tests” section) on an Intel Core i7-8550U CPU in real-time.

The details of all simulation tests are shown in Table 1. The parameters used by the compared method FUEL are the same as those in Table 1 (without \( \lambda_e \)) to ensure the fairness of the comparison.

Different parameters influence the characteristics of the trajectory. The particular feature of the trajectory can be changed by the parameters according to the requirements of the task. \( v_{\text{max}}, a_{\text{max}}, \) and \( \zeta_{\text{max}} \) are the dynamic limits of the quadrotor, and the trajectory will be limited to this range. We set specific values of the parameters, which are proper to fully verify the improvement effect of our method.

\( \lambda_s, \lambda_c, \) and \( \lambda_d \) influence the smoothness, safety (distance to obstacles), and dynamic feasibility. \( \lambda_{\text{hs}} \) and \( w_l \) are the set of parameters that need to be balanced, which affects the smoothness and rapidity of the trajectory between two viewpoints. And in our method, \( \lambda_e \) determines how close the whole trajectory is to the environment frontier information. The above parameters are all weight values, which means that if the value of a parameter is increased, the weight of the corresponding optimization item will be increased. And the parameters have been optimized through extensive simulation tests to meet the need for autonomous exploration of different clutter environments. In “Real-world tests” section, further real-world experiments have shown that these parameters also perform well in real exploration.

And the FOVs of the sensors are set as [80/60]° with a maximum range of 4.5 m.

To avoid the chance of results, each method is tested five times with the same initial parameter in every map. Performance and statistics of the two methods are shown in Figure 10 and Table 2.

The simulations verify our above conjecture about the redundancy of the trajectories due to repeated planning.

The original work FUEL, as shown in Figure 11, will miss part of the environment information when passing through complex environments such as corners, resulting in duplicate planning. But Ours, which has added the environment cost term, makes drones approach unexplored areas more intelligently, making local exploration more thorough. It avoids the incomplete exploration of some small and residual parts of the area, which causes the UAV to turn back and repeat exploration work after the exploration of the other areas is completed.

As shown in Figure 12, FUEL is always heading to the wall during trajectory execution resulting in a waste of exploration resources. However, our work notices the frontier information behind the UAV, especially in open areas, and during the flight, by planning the yaw separately, the UAV can discover more unknown areas at the same exploration time.

The above simulation experiments confirm the validity of our work which can improve the efficiency of exploration.

#### Real-world tests

To further validate the applicability of our method, we conduct extensive real-world experiments in a typical field application scenario as shown in Figure 13. The need for drones to efficiently explore through 16 × 16 × 2 m² dense woods imposes stringent planning requirements.

---

**Table 1. Parameter setting in all simulation test.**

| \( \lambda_s \) | \( \lambda_c \) | \( \lambda_d \) | \( \lambda_{\text{hs}} \) | \( w_l \) | \( v_{\text{max}} \) | \( a_{\text{max}} \) | \( \zeta_{\text{max}} \) | \( \lambda_e \) |
|---|---|---|---|---|---|---|---|---|
| FUEL | 20.0 | 10.0 | 2.0 | 1.5 | 1.0 | 3.0 m/s | 3.0 m/s² | 1.1 rad/s | \( \lambda_e \) |
| Ours | 1.5 | | | | | | | |

**Table 2. Exploration statistics in three maps.**

| Map | Method | Exploration time (s) | Flight distance (m) |
|---|---|---|---|
| | Max | Min | Avg | Max | Min | Avg |
| 1 | FUEL² | 199.83 | 188.26 | 195.31 | 238.92 | 226.21 | 233.87 |
| 30 m × 15 m × 3 m | Ours | 176.10 | 163.76 | 168.23 | 222.36 | 196.65 | 214.54 |
| 2 | FUEL³ | 249.28 | 226.90 | 234.51 | 282.67 | 263.32 | 275.20 |
| 30 m × 20 m × 3 m | Ours | 221.07 | 203.55 | 210.74 | 276.82 | 253.35 | 264.48 |
| 3 | FUEL³ | 165.29 | 162.12 | 164.07 | 195.83 | 190.22 | 192.78 |
| 20 m × 20 m × 3 m | Ours | 160.51 | 150.22 | 153.18 | 186.24 | 178.67 | 181.40 |
Figure 10. Both methods were tested in three indoor office scenes.

Figure 11. Comparison of the two methods in a complex environment. (a) FUEL, (b) Ours, (c) FUEL, and (d) Ours.
In real-world experiments, we localize the quadrotor by a visual-inertial state estimator VINS-Fusion and adopt a classic controller for quadrotors to track the trajectories. The experiments were completed by our DIY quadrotor platform in Figure 14. The hardware configuration of the platform is shown in Table 3.

To verify the stability of the proposed method, we have conducted several experiments in the scenario. And

Table 3. Hardware configuration information.

| Hardware        | Name                  |
|-----------------|-----------------------|
| Autopilot       | CUAV V5               |
| Depth camera    | RealSense D435i       |
| Motor           | Tmotor F60            |
| ESC             | Hobbying Xrotor       |
| Onboard computer| Manifold 2C           |
\[ v_{\text{max}} = 1.0 \text{ m/s}, \quad a_{\text{max}} = 0.5 \text{ m/s}^2, \quad \text{and} \quad \zeta_{\text{max}} = 1.1 \text{ rad/s} \]

are set to limit the quadrotor’s dynamic. Similar to the simulation tests, the UAV quickly acquires information about the surrounding environment in open areas through reasonable yaw planning as shown in Figure 15. Besides, as shown in Figure 16, thanks to the proposed environment...
cost term, the UAV doesn’t head to other areas to continue exploring missions until thoroughly explores the corner areas, avoiding lengthy flights due to duplicate planning for small areas that have not been fully explored.

All in all, in real-world experiments shown in Figure 17, the upgraded trajectory planning method we proposed reduces trajectory redundancy caused by repeated planning and improves planning efficiency.

Conclusion

This article proposes an upgraded trajectory planning method for autonomous exploration work. Existing methods ignore the trajectory execution process but only focus on the viewpoint environment gain, resulting in repeated inefficient planning and redundant trajectory.

To address the problem, we present (1) a fresh environment cost term in trajectory optimization which makes trajectory closer to frontiers and gains more environment information in flight, (2) a more flexible yaw planning strategy which gets a set of yaws heading to the unknown area and reduces waste of exploration resources, and (3) test our proposed method in different simulation maps and further verified it in real-world experiments.

Based on the creative improvements, UAVs can explore more thoroughly in complex areas and faster in spacious spaces. And both simulation and real-world tests show that our method can effectively alleviate redundancy of trajectory caused by repeated planning and speed up the exploration process, thus reducing exploration time and flight distance.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by National Natural Science Foundation of
China (NSFC) (61603297) and Natural Science Foundation of Shaanxi Province (2020JQ-219).

**References**

1. Yamauchi B. A frontier-based approach for autonomous exploration. In: IEEE international symposium on computational intelligence in robotics & automation, Monterey, CA, USA, 10–11 July 1997. IEEE.

2. Cieslewski T, Kaufmann E, and Scaramuzza D. Rapid exploration with multi-rotors: a frontier selection method for high speed flight. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), Vancouver, BC, Canada, 24–28 September 2017. IEEE.

3. Zhou B, Zhang Y, Chen X, et al. FUEL: fast UAV exploration using incremental frontier structure and hierarchical planning. IEEE Robot Auto Lett 2021; 6(2): 779–786.

4. Yu X, Peng Q, Xu L, et al. A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm. Inform Proce Manag 2021; 58(6): 102691.

5. Yu X, Chu Y, Jiang F, et al. SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features. Knowl-Based Syst 2018; 141: 80–91.

6. Yu X, Jiang F, Du J, et al. A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains. Pattern Recog 2019; 94: 96–109.

7. Connolly C. The determination of next best views. In: Proceedings. 1985 IEEE international conference on robotics and automation, St. Louis, MO, USA, 25–28 March 1985, vol. 2, pp. 432–435. IEEE.

8. Bircher A, Kamel M, Alexis K, et al. Receding horizon “next-best-view” planner for 3D exploration. In: 2016 IEEE international conference on robotics and automation (ICRA), Piscataway, NJ, 2016, pp. 1462–1468. IEEE.

9. Dang T, Papachristos C, and Alexis K. Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics. In: 2018 IEEE international conference on robotics and automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018, pp. 2526–2533. IEEE.

10. Papachristos C, Khattak S, and Alexis K. Uncertainty-aware receding horizon exploration and mapping using aerial robots. In: 2017 IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–03 June 2017, pp. 4568–4575. IEEE.

11. Bircher A, Kamel M, Alexis K, et al. Receding horizon path planning for 3D exploration and surface inspection. Auto Robot 2018; 42(2): 291–306.

12. Witting C, Fehr M, Bähnhemann R, et al. History-aware autonomous exploration in confined environments using MAVs. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, Spain, 01–05 October 2018, pp. 1–9. IEEE.

13. Meng Z, Qin H, Chen Z, et al. A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction. IEEE Robot Auto Lett 2017; 2(3): 1680–1687.

14. Quan L, Han L, Zhou B, et al. Survey of UAV motion planning. IET Cyber-Sys Robot 2020; 2(1): 14–21.

15. Ratliff N, Zucker M, Bagonnell JA, et al. CHOMP: gradient optimization techniques for efficient motion planning. In: 2009 IEEE international conference on robotics and automation, Kobe, Japan, 12–17 May 2009, pp. 489–494. IEEE.

16. Kalakrishnan M, Chitta S, Theodorou E, et al. STOMP: stochastic trajectory optimization for motion planning. In: 2011 IEEE international conference on robotics and automation, Shanghai, China, 09–13 May 2011, pp. 4569–4574. IEEE.

17. Oleynikova H, Burri M, Taylor Z, et al. Continuous-time trajectory optimization for online UAV replanning. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), Daejeon, Korea (South), 09–14 October 2016, pp. 5332–5339. IEEE.

18. Gao F, Lin Y, and Shen S. Gradient-based online safe trajectory generation for quadrotor flight in complex environments. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), Vancouver, BC, Canada, 24–28 September 2017, pp. 3681–3688. IEEE.

19. Ding W, Gao W, Wang K, et al. An efficient B-spline-based kinodynamic replanning framework for quadrotors. IEEE Trans Robot 2019; 35(6): 1287–1306.

20. Zhou B, Gao F, Wang L, et al. Robust and efficient quadrotor trajectory generation for fast autonomous flight. IEEE Robot Auto Lett 2019; 4(4): 3529–3536.

21. Zhou X, Wang Z, Ye H, et al. Ego-planner: an ESDF-free gradient-based local planner for quadrotors. IEEE Robot Auto Lett 2020; 6(2): 478–485.

22. Mellinger D and Kumar V. Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE international conference on robotics and automation, Shanghai, China, 09–13 May 2011, pp. 2520–2525. IEEE.

23. Dolgov D, Thrun S, Montemerlo M, et al. Path planning for autonomous vehicles in unknown semi-structured environments. Int J Robot Res 2010; 29(5): 485–501.

24. Quinlan S and Khatib O. Elastic bands: connecting path planning and control. In: [1993] Proceedings IEEE international conference on robotics and automation, Atlanta, GA, USA, 02–06 May 1993, pp. 802–807. IEEE.

25. Zhu Z, Schmerling E, and Pavone M. A convex optimization approach to smooth trajectories for motion planning with car-like robots. In: 2015 54th IEEE conference on decision and control (CDC), Osaka, Japan, December 2015, pp. 835–842. IEEE.

26. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. In: Autonomous robot vehicles. New York, NY: Springer, 1986, pp. 396–404.

27. Qin T, Pan J, Cao S, et al. A general optimization-based framework for local odometry estimation with multiple sensors. arXiv preprint arXiv:1901.03638, 2019.