Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.075459

ARTICLE

Path Planning for Substation UAV Inspection Based on 3D Point Cloud Mapping

Yanping Chen[1], Zhengxin Zhan[1], Xiaohui Yan[1], Le Zou[1,*], Yucheng Zhong[1] and Hailei Wang[2]

[1]School of Artificial Intelligence and Big Data, Hefei University, Hefei, China
[2]Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China

*Corresponding Author: Le Zou. Email: zoule@hfuu.edu.cn

Received: 01 November 2025; Accepted: 23 January 2026; Published: 12 March 2026

ABSTRACT: With the increasing complexity of substation inspection tasks, achieving efficient and safe path planning for Unmanned Aerial Vehicles in densely populated and structurally complex three-dimensional (3D) environments remains a critical challenge. To address this problem, this paper proposes an improved path planning algorithm—Random Geometric Graph (RGG)-guided Rapidly-exploring Random Tree (R-RRT)—based on the classical Rapidly-exploring Random Tree (RRT) framework. First, a refined 3D occupancy grid map is constructed from Light Detection and Ranging point cloud data through ground filtering, noise removal, coordinate transformation, and obstacle inflation using spherical structuring elements. During the planning stage, a dynamic goal-biasing strategy is introduced to adaptively adjust the sampling direction, the sampling distribution is optimized using a pre-generated RGG, and collision detection is accelerated via a K-Dimensional Tree structure. After initial trajectory generation, redundant nodes are eliminated via greedy pruning, and a curvature-minimizing gradient-based optimization method is applied to smooth the trajectory. Experimental results conducted in a simulated substation environment demonstrate that, compared with mainstream path planning algorithms, the proposed R-RRT achieves superior performance in terms of path length, planning time, and trajectory smoothness. Comprehensive analysis shows that the proposed method significantly enhances trajectory quality, planning efficiency, and operational safety, validating its applicability and advantages for high-precision 3D path planning in complex substation inspection scenarios.

KEYWORDS: R-RRT algorithm; unmanned aerial vehicles; path planning; random geometric graph; 3D occupancy grid map; substation inspection

1 Introduction

In recent years, with the rapid advancement of Unmanned Aerial Vehicle (UAV) technology and multimodal remote sensing systems, UAVs have been widely applied in fields such as smart agriculture, urban traffic monitoring, emergency response, and logistics delivery [1–4], demonstrating strong adaptability in both industrial and civilian applications. In the power industry, UAVs have become a key technology for replacing traditional manual inspection due to their advantages of low cost, high precision, and high operational efficiency [5,6].

Substations, serving as essential hubs for power generation, transmission, and distribution, play a critical role in ensuring grid reliability [7]. However, substations are characterized by densely arranged equipment, limited space, and complex internal structures. Traditional manual inspection methods are labor-intensive, pose safety risks, and exhibit low efficiency, making it difficult to meet the requirements of modern smart grids [8]. UAV-based automated inspection systems, equipped with multiple sensors such as Light Detection and Ranging (LiDAR), thermal imagers, and high-definition cameras, can efficiently perceive and collect

data from power facilities [9]. Nevertheless, achieving high-precision and low-risk autonomous navigation in complex three-dimensional (3D) environments remains a major technical challenge—particularly in the path planning stage, which demands high real-time performance, accuracy, and obstacle avoidance capabilities from the algorithm.

It is worth noting that path planning systems can be broadly categorized into two paradigms: those relying on a priori maps and those based on Simultaneous Localization and Mapping (SLAM). SLAM-based systems excel in unknown or dynamically changing environments by concurrently building a map and localizing within it, offering great flexibility. Recent surveys have shown that modern SLAM solutions increasingly rely on multi-sensor fusion—such as LiDAR, IMU, and camera integration—to improve robustness and mapping accuracy in complex environments [10]. However, for structured and relatively static industrial sites like substations, where high-precision, stable maps can be pre-acquired and safety is paramount, the use of a priori maps presents distinct advantages. This approach decouples the computationally intensive mapping process from the online planning task, allowing the planning algorithm to utilize a consistent, optimized environmental model. This eliminates the potential for cumulative errors and map inconsistencies inherent in real-time SLAM, thereby providing a more reliable and deterministic foundation for generating safe inspection paths. Consequently, this work focuses on the path planning challenge given a high-fidelity pre-existing map, a common and practical scenario in scheduled industrial inspections.

The core of UAV path planning lies in constructing a high-precision 3D environmental model and generating feasible, safe, and smooth flight trajectories. Currently, various mapping approaches have been proposed, including octree-based [11], probabilistic occupancy grid-based [12], and point cloud map-based [13] methods, each with distinct advantages. Octree maps are widely adopted due to their sparse representation and global mapping capability. The Robocentric Occupancy Grid Map (ROG-Map) proposed by Ren et al. integrates multi-resolution occupancy probabilities and an adaptive sliding window mechanism to achieve efficient map updates and accurate perception representation in complex outdoor environments [12]. However, a trade-off between detailed modeling and real-time performance still exists, limiting their adaptability to highly dynamic scenarios.

In contrast, point cloud-based mapping approaches have gained increasing attention. By directly utilizing raw 3D spatial data obtained from LiDAR or vision sensors, this method offers independence from grid assumptions, retains rich geometric details, and allows flexible resolution adjustment. For example, the Efficient Point Cloud-Based Exploration and Mapping (EPIC) framework integrates topological structures with lightweight point clouds, enabling rapid mapping and high-quality trajectory generation in large-scale environments [14]. Additionally, Zhao et al. proposed an improved Rapidly-exploring Random Tree Star (RRT*) algorithm based on point cloud maps to enhance UAV path search in complex 3D spaces using adaptive sampling and expansion strategies, thereby significantly improving planning efficiency and path quality [15]. In substation environments characterized by dense equipment and complex spatial layouts, point cloud maps provide superior realism and detail preservation, making them the preferred solution for environmental modeling.

This paper adopts a point cloud-based mapping framework. The input is a pre-scanned, high-density LiDAR point cloud of the substation. The methodology involves processing this raw point cloud into a 3D occupancy grid map through steps including ground filtering, noise removal, and obstacle inflation. This hybrid approach leverages the geometric fidelity of point clouds for accurate obstacle representation while converting it into a voxelized occupancy grid to enable efficient, deterministic collision checking during the planning phase—a critical requirement for real-time performance in dense environments. This choice is justified as it strikes an optimal balance for the substation inspection task: it maintains the environmental

detail necessary for safe navigation around complex apparatus while providing the computational structure needed for fast and reliable path planning.

The compact structure and dense obstacles of substations introduce significant spatial complexity, imposing higher requirements on UAV path planning algorithms in terms of environmental modeling accuracy, obstacle avoidance, and computational efficiency. The first category includes graph search-based algorithms such as A* search algorithm (A*) [16] and Dijkstra [17], which ensure global optimality but perform poorly in dynamic environments. The second category comprises sampling-based algorithms such as Rapidly-exploring Random Tree (RRT) [18] and Probabilistic Roadmap (PRM) [19], which are effective in high-dimensional spaces but often generate non-smooth or directionally random paths. The third category involves optimization-based methods, including Artificial Potential Field (APF) [20] and Gradient Descent (GD) [21], which produce smooth trajectories but are susceptible to local minima and may fail to guarantee global feasibility.

The RRT algorithm has been widely used for global path planning due to its scalability and efficiency in high-dimensional environments. Its improved version, RRT* [22], enhances path quality by optimizing parent node selection to ensure local path optimality, gradually refining the overall tree structure. However, most existing research remains limited to two-dimensional applications, while challenges persist in extending these algorithms to complex 3D environments such as substations, where densely packed obstacles and narrow spaces increase planning difficulty.

In recent years, several enhanced RRT-based algorithms have been developed to address these issues. The Bidirectional Adaptive Multi-objective RRT* (Bi-AM-RRT*) improves planning efficiency and search performance in dynamic environments through adaptive node expansion metrics [23]. The algorithm that employs a K-Dimensional Tree (KD-Tree) to accelerate searches in dense environments with point cloud data is termed the KD-Tree-based RRT (BTO-RRT) [24]. The Alternative Paths and Triangular Sampling RRT* (ATS-RRT*) optimizes convergence speed and path quality via alternative paths and triangular area sampling strategies [25]. The Continuous Bidirectional Quick-RRT* (CBQ-RRT*) algorithm further enhances search efficiency and path smoothness through continuous bidirectional reconnection [26]. Given our focus on static environments with prior maps, it should be noted that these approaches still face difficulties in the specific context of dense and static substation environments—such as maintaining sufficient obstacle clearance, avoiding deadlocks in narrow spatial searches, and ensuring smooth, continuous paths suitable for automated inspection.

Apart from classical and improved sampling-based methods, the field of path planning has also shown growing interest in data-driven approaches, particularly deep learning techniques. A survey focusing on mobile robots has systematically summarized the applications of Deep Reinforcement Learning (DRL) in SLAM and autonomous navigation, providing a reference for related technology trends [27]. Such methods are capable of learning navigation policies directly from sensor inputs to address path planning challenges in complex 3D obstacle-ridden environments. In addressing path planning with complex constraints, research has applied DRL to generate 3D UAV paths that satisfy specific connectivity requirements [28]. Concurrently, to meet the real-time demands of online planning, work has been done to improve response efficiency in unknown environments by dynamically adjusting the iterative process of classical Q-learning [29]. However, for structured and safety-critical tasks such as substation inspection, these methods still face fundamental challenges. Firstly, they heavily rely on large-scale, high-quality training data. Yet, substations feature complex layouts, diverse equipment, and safety constraints, making it difficult to construct comprehensively covered, annotated datasets. Secondly, the decision-making process of deep learning models lacks interpretability, making it difficult to provide the formal safety assurances required for industrial-grade applications. The robustness of their performance is also hard to guarantee in environmental configurations

not covered by the training set. Therefore, this paper focuses on industrial inspection scenarios with known high-precision prior maps and adopts a deterministic, geometry-based planning approach. Within highly structured environments, the proposed method achieves efficient search and obstacle avoidance through rigorous geometric computations and deterministic logic, thereby avoiding the uncertainties inherent in data-driven methods and providing a verifiable, reproducible, and reliable solution for automated inspection.

To address the aforementioned challenges, this study first develops a 3D occupancy grid mapping framework specifically designed for substation environments. The framework systematically processes LiDAR point cloud data for ground filtering, noise removal, and safety-aware obstacle expansion using spherical structuring elements, providing a reliable spatial representation for UAV navigation in equipment-dense environments.

At the path planning level, an improved algorithm termed the Random Geometric Graph (RGG)-guided Rapidly-exploring Random Tree (R-RRT) is proposed, which integrates multiple strategies with the following key enhancements: (1) dynamic goal-biasing strategy: adaptively adjusts sampling probability based on the history of successful expansions, effectively balancing global exploration and local convergence; (2) RGG-guided sampling: optimizes the growth direction of the random tree, significantly reducing invalid node generation; (3) KD-Tree-based fast collision detection: enhances real-time performance in complex environments; (4) two-stage path optimization: employs greedy pruning to remove redundant nodes and gradient-based smoothing to improve trajectory quality.

2 Construction of Navigation Grid Maps for Substation Inspection

This section presents a complete technical solution for constructing navigation grid maps specifically tailored for substation environments. The framework systematically integrates three core components: (1) point cloud acquisition and preprocessing, involving LiDAR data collection with ground point filtering and noise removal; (2) 3D voxel space and occupancy map generation, converting processed point clouds into discrete voxel grids through coordinate mapping; and (3) equipment safety expansion, incorporating UAV physical dimensions and safety margins using morphological dilation. This pipeline transforms raw point cloud data into a navigable environment model for UAV path planning.

2.1 Map Data Acquisition and Preprocessing

The data acquisition phase employs LiDAR and 3D scanners to capture 3D point clouds of the substation environment. This study designs a point cloud preprocessing pipeline that systematically integrates data acquisition, hierarchical ground filtering, and statistically-based noise removal methods to address environmental modeling requirements in substation inspection.

2.1.1 Ground Point Cloud Filtering

To address the challenge of mixed terrain and equipment in substation environments, a hierarchical ground-non-ground separation framework is designed. The approach integrates filtering algorithms based on complementary principles: a height threshold method for fast coarse segmentation, followed by a RANdom SAmple Consensus (RANSAC) plane fitting strategy for robust refinement. This two-stage workflow ensures both computational efficiency and segmentation accuracy.

- **Height Threshold Method:** A height threshold $z_{\text{threshold}}$ is defined to distinguish ground from non-ground points. Points with heights greater than this threshold are retained:

$$P_{\text{filtered}} = \{p_i = (x_i, y_i, z_i) \mid z_i \geq z_{\text{threshold}}\} \tag{1}$$

where p_i denotes the i-th point in the raw point cloud, and P_{filtered} represents the intermediate non-ground subset. Based on the criterion defined in Eq. (1), this coarse filtering step rapidly excludes the majority of obvious ground points, thereby reducing the computational load for the subsequent refinement stage.

- **Plane Fitting Method:** Since height thresholding alone is insufficient when ground surfaces exhibit slight slopes or undulations, a RANSAC-based plane fitting algorithm is employed to model the ground surface robustly in the presence of outliers. The ground plane is expressed as:

$$ax + by + cz + d = 0 \tag{2}$$

The Euclidean distance between each point and the estimated plane is computed as:

$$d_i = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{3}$$

where d_i measures how far point p_i deviates from the plane. A point is classified as ground when $d_i < d_{\text{threshold}}$. The choice of $d_{\text{threshold}}$ must balance accuracy and robustness: overly small values may fail to accommodate natural terrain variations or noise, while overly large values risk erroneously removing low-lying structural components.

Given the relatively flat terrain and limited sensor noise in typical substation environments, this study experimentally selects $d_{\text{threshold}} = 0.03\,\text{m}$, which provides a reliable compromise between precision and stability. This refined segmentation, based on the plane model in Eq. (2) and the distance criterion in Eq. (3), significantly improves the quality of the obstacle map used in downstream path planning.

2.1.2 Noise Removal

To mitigate noise introduced by sensor measurement errors and environmental interference, this study adopts a statistical filtering method based on local spatial consistency. The method assumes that points belonging to real object surfaces exhibit strong local density and geometric continuity, whereas noise points lack such structural support and therefore appear spatially isolated.

- **Statistical Filtering:** Noise identification is performed by inspecting the neighborhood characteristics of each point. The neighborhood $\mathcal{N}(p_i)$ of a point p_i (the i-th point in the original cloud) is defined as the set of all points p_j that lie within a radius r:

$$\mathcal{N}(p_i, r) = \{p_j \mid \|p_j - p_i\| \le r\} \tag{4}$$

Based on this definition, the denoised point cloud is obtained as:

$$P_{\text{denoised}} = \{p_i \mid N_i \ge N_{\min}\} \tag{5}$$

where $N_i = |\mathcal{N}(p_i)|$ denotes the neighborhood size, and $N_{\min}$ is the minimum required neighbor count. Points with $N_i < N_{\min}$ are classified as noise.

The radius parameter is set to $r = 0.05\,\text{m}$, corresponding to approximately 2–3 times the average point spacing (0.01–0.03 m). This ensures that meaningful local geometric structures are captured without introducing unnecessary neighborhood expansion.

The threshold for neighbor count is determined through empirical analysis. Since isolated noise points typically exhibit fewer than 8–15 neighbors, this study sets $N_{\min} = 10$. This parameter choice effectively removes sparse noise points while preserving structural edges and fine details crucial for accurate obstacle modeling in complex substation environments.

Therefore, the denoising filter defined by Eqs. (4) and (5) effectively removes spatially isolated points based on the chosen parameters $r = 0.05\,\text{m}$ and $N_{\min} = 10$.

2.2 3D Voxel Space and Occupancy Map Construction

To transform the continuous 3D point cloud space into a discrete representation suitable for path planning, a structured voxel grid space is constructed. The dimensions of this space are determined by the spatial distribution range of valid point cloud data. Assuming the grid map has a uniform resolution R in all three dimensions, the required number of voxels n_x, n_y, n_z in each dimension is calculated by:

$$n_\theta = \left\lceil \frac{\max\limits_{i=1,2,\ldots,N} \theta_i - \min\limits_{i=1,2,\ldots,N} \theta_i}{R} \right\rceil, \quad \theta = x, y, z \tag{6}$$

where $\lceil \cdot \rceil$ represents the ceiling function, ensuring that any fractional part is allocated to a complete voxel unit, and N is the number of valid points. The span of point cloud in each dimension $(\max \theta_i - \min \theta_i)$ is divided by the resolution R to obtain the theoretical number of voxels required in that dimension, as described in Eq. (6). The ceiling function ensures all point cloud data are contained within the voxel space.

The mapping relationship from point cloud coordinates to voxel coordinates is defined by:

$$\hat{\theta} = \left\lceil \frac{\theta - \min\limits_{i=1,2,\ldots,N} \theta_i}{R} \right\rceil, \quad \theta = x, y, z \tag{7}$$

where $\hat{\theta}$ is the voxel index along the θ-axis, θ is the original point cloud coordinate, R is the voxel resolution, and $\lceil \cdot \rceil$ denotes the ceiling function.

This mapping, formulated in Eq. (7), performs a translation and scaling transformation: first, subtract the minimum value $\min \theta_i$ of the dimension from coordinate θ to normalize to a zero-point relative coordinate system; then divide by the resolution R to discretize it to voxel indices; finally, use the ceiling function to ensure each point is uniquely assigned to a voxel. This linear mapping approach ensures efficient spatial queries and collision detection.

2.3 Equipment Dilation in Substation Environment

To ensure the flight safety of UAVs in complex substation environments, both the physical size of the UAV and a sufficient safety margin must be considered. This study employs a 3D morphological dilation algorithm to expand the safety boundaries of equipment point clouds. The core of this process is to construct a reasonable dilation radius calculation model that comprehensively considers the UAV's physical radius r_{uav} and the required minimum safety distance d_{safe}, which is defined as:

$$r = r_{\text{uav}} + d_{\text{safe}} \tag{8}$$

where r is the total dilation radius, r_{uav} denotes the physical radius of the UAV, and d_{safe} is the minimum required safety distance, as defined in Eq. (8).

In this work, the UAV used has a maximum diagonal dimension of approximately 0.4 m, corresponding to a circumscribed radius $r_{\text{uav}} \approx 0.2\,\text{m}$. An additional safety margin $d_{\text{safe}} = 0.3\,\text{m}$ is introduced to accommodate positioning errors and control uncertainties. Consequently, applying the model in Eq. (8), the dilation radius is set to $r = 0.5\,\text{m}$.

This design ensures that during flight, the UAV not only avoids physical contact with obstacles but also maintains an adequate buffer zone to account for operational uncertainties, thereby meeting the safety requirements for inspection operations in dense substation environments.

2.3.1 Kernel Construction

To uniformly expand the safety area in 3D space, a spherical structuring element is used for dilation, mathematically expressed as:

$$K = \left\{ (x, y, z) \middle| \sqrt{x^2 + y^2 + z^2} \leq r \right\} \tag{9}$$

where K is the spherical structuring element (kernel), (x, y, z) are the local coordinates relative to the kernel center, and r is the dilation radius. The kernel defined in Eq. (9) leverages its isotropic characteristics to most accurately reflect the physical reality that the UAV needs to maintain a safe distance in all directions in 3D space, thus forming a spherical safety envelope.

2.3.2 Obstacle Detection

The targets of the dilation operation are all obstacles in the original occupancy map. By traversing the voxel grid map M_{grid}, all voxel coordinates marked as occupied are extracted to form the initial obstacle set O:

$$O = \left\{ (x_i, y_i, z_i) \middle| M_{\text{grid}}(x_i, y_i, z_i) = 1 \right\} \tag{10}$$

where O is the set of all obstacle voxels, (x_i, y_i, z_i) represents the coordinates of the i-th obstacle voxel, and $M_{\text{grid}}(x_i, y_i, z_i) = 1$ indicates that the voxel at position (x_i, y_i, z_i) is occupied. The obstacle set O defined in Eq. (10) identifies all original obstacle source points that require dilation.

2.3.3 Safety Expansion

For each obstacle voxel O_i in the set O defined in Eq. (10), morphological dilation is performed within its neighborhood to update the safety map M_{safe}:

$$M_{\text{safe}}(x', y', z') = \max \left\{ M_{\text{safe}}(x', y', z'),\ K(x' - x_i, y' - y_i, z' - z_i) \right\} \tag{11}$$

where (x', y', z') represents the coordinates of the currently processed voxel, $O_i = (x_i, y_i, z_i)$ is the current obstacle voxel, and $K(\cdot)$ is the spherical kernel function defined in Eq. (9).

The search range for the dilation operation in Eq. (11) is limited to a cubic neighborhood centered on O_i with side length $2r$:

$$\begin{aligned} x' &\in [x_i - r, x_i + r] \\ y' &\in [y_i - r, y_i + r] \\ z' &\in [z_i - r, z_i + r] \end{aligned} \tag{12}$$

This operation evaluates, for each candidate position (x', y', z') in the neighborhood defined by Eq. (12), whether its offset relative to O_i falls within the spherical range of the dilation kernel K. If the condition is met, the max operation in Eq. (11) updates the state of this position in M_{safe} to 1 (occupied).

The dilation process handles obstacles sequentially, but the final result correctly merges multiple expansions through the maximum operation in Eq. (11), ultimately generating the dilated map that includes safety margins.

2.3.4 Boundary Constraints

During the dilation process, neighborhood searches may extend beyond the map boundaries. To prevent out-of-bounds access, a boundary constraint function is introduced:

$$B(v, v_{\min}, v_{\max}) = \min\left(\max(v, v_{\min}), v_{\max}\right) \tag{13}$$

where v represents the coordinate value to be constrained, $v_{\min}$ and $v_{\max}$ represent the minimum and maximum valid indices in that dimension, and $B(\cdot)$ is the boundary constraint function.

The constraint function defined in Eq. (13) achieves constraint through two-level truncation: first using $\max(v, v_{\min})$ to ensure the coordinate is not lower than the lower bound, then applying min to the intermediate result and $v_{\max}$ to ensure it does not exceed the upper bound. This strictly restricts all coordinates within the valid range $[v_{\min}, v_{\max}]$, ensuring the robustness of the dilation algorithm during boundary operations.

3 Path Planning for UAV Inspection Based on Grid Map

Building upon the 3D occupancy grid map constructed in Section 2, which provides a structured environmental representation for collision detection and navigation, an improved RRT algorithm named R-RRT is proposed for UAV inspection path planning. By integrating random geometric graph-guided sampling with multi-stage path optimization strategies, the proposed algorithm significantly enhances the planning performance of traditional RRT in complex substation environments.

Fig. 1 illustrates the overall workflow of the proposed UAV inspection path planning framework. The planning process is initialized with the voxelized grid map, which encodes obstacle information from 3D point cloud data. During the path search process, the algorithm introduces a dynamic goal-biasing strategy that adaptively adjusts the target sampling probability according to the search progress. This strategy enhances exploration in the early search stage to cover the global space, while reinforcing goal-directed guidance in the later stage to accelerate path convergence, effectively balancing global search capability and local convergence speed.

Secondly, to avoid the waste of numerous invalid sampling points in traditional random sampling, a sampling mechanism based on RGG is designed. In the initialization phase, a large number of uniformly distributed spatial sampling points are pre-generated. During the planning phase, optimal sampling points are selected from the RGG for tree expansion by combining current node direction information. This approach reduces invalid sampling and improves both sampling efficiency and path quality.

For collision detection, the algorithm constructs a KD-Tree-based data structure to accelerate path validity verification. During each path expansion, obstacle avoidance detection is performed only on key connecting segments through nearest-neighbor search, significantly reducing the computational complexity of collision detection.

Finally, to generate smooth trajectories suitable for actual flight, path post-processing optimization is performed after the search completion. Specifically, this includes: employing a greedy pruning strategy to remove redundant nodes in the path, and using the gradient descent method for path smoothing under safety constraints, thereby obtaining more continuous trajectories with smaller curvature that are executable by UAVs.

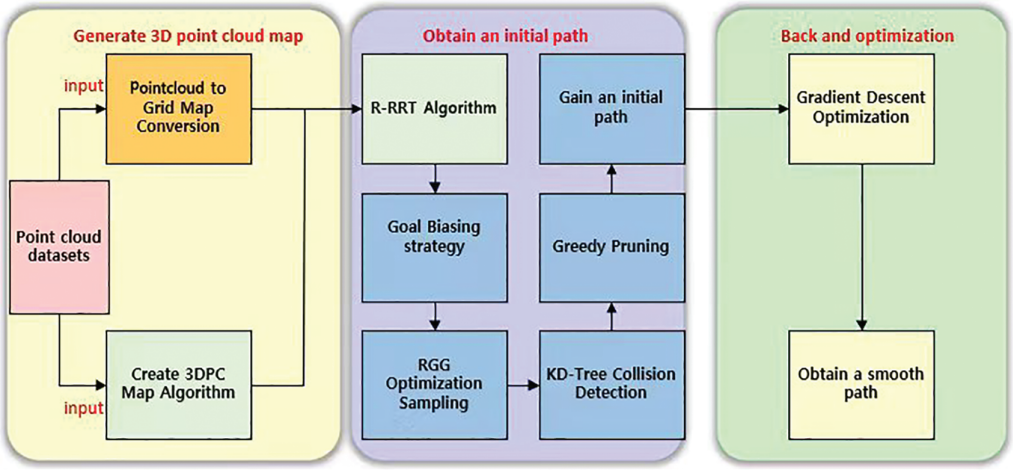


Figure 1: Flowchart of the proposed UAV inspection path planning framework.

3.1 Dynamic Goal Biasing Strategy

Goal biasing is a crucial mechanism in sampling-based path planning algorithms. Its main idea is to select the goal point as a candidate node with a certain probability during the random sampling process, thereby accelerating the convergence of the search tree toward the goal region. In the traditional RRT algorithm, this goal biasing probability is fixed (typically between 5% and 10%), which limits adaptability in complex or dynamic environments. A fixed probability may lead to insufficient exploration in the early stage or slow convergence in the later stage.

To overcome this limitation, this paper proposes a novel mechanism for dynamic goal biasing that utilizes feedback from successful tree extensions. This mechanism adaptively adjusts the sampling probability P_{goal} based on N_{success} to dynamically balance global exploration and local convergence in real-time. The mathematical formulation of this mechanism is given in Eq. (14) as follows:

$$P_{\text{goal}}(N_{\text{success}}) = 20\% + \frac{N_{\text{success}}}{1000} \times (40\% - 20\%) \tag{14}$$

Here, N_{success} represents the cumulative number of successful extensions. When $N_{\text{success}} = 1000$, the goal sampling probability reaches the upper limit of 40%. In the early stage ($N_{\text{success}} < 1000$), the probability remains between 20% and 30%, promoting extensive exploration and preventing premature convergence. As the search proceeds ($N_{\text{success}} \geq 1000$), the probability increases to 40%, guiding the search tree more aggressively toward the goal and reducing unnecessary random sampling.

This adaptive mechanism enables the algorithm to maintain strong exploratory behavior in the initial phase and achieve faster convergence in the later phase, thus improving the overall planning efficiency. Moreover, it enhances the algorithm's adaptability in dynamic and complex environments by continuously balancing global exploration and local refinement. The detailed procedure is shown in Algorithm 1.

Algorithm 1: Adaptive goal-biased sampling for RRT

1: Initialize `goal_bias` ← 0.2
2: Initialize `successful_searches` ← 0
3: **while** search not completed **do**

(Continued)

Algorithm 1 (continued)

```
goal_bias ← min(0.2 + 0.2 × (successful_searches/1000), 0.4)
if RANDOM() < goal_bias then
    sample ← goal
else
    sample ← RANDOM_POINT()
end if
EXPAND_TREE(sample)
if EXPANSION_SUCCESSFUL () then
    successful_searches ← successful_searches + 1
end if
end while
```

Algorithm 1 illustrates the adaptive goal-biased sampling mechanism. The sampling probability dynamically increases with the number of successful extensions, ensuring an optimal trade-off between exploration and exploitation throughout the planning process. Computationally, this strategy introduces minimal overhead: updating P_{goal} and maintaining N_{success} are both constant-time operations ($O(1)$ per iteration). In terms of space complexity, it requires only two scalar variables to store `goal_bias` and `successful_searches`, resulting in $O(1)$ space overhead. Consequently, it preserves the asymptotic time and space complexity of the base RRT while significantly improving practical convergence by reducing the total number of iterations N needed to reach the goal. This dynamic biasing strategy thus enhances the robustness and efficiency of RRT-based path planning in complex UAV inspection scenarios with negligible computational cost.

3.2 Incorporating RGG to Optimize Sampling

RGG is an effective tool in graph theory, widely used to optimize the sampling process in path planning algorithms. In the traditional RRT algorithm, sampling points are usually completely random. The distribution of sampling points may concentrate in certain areas while neglecting others, leading to a large number of invalid nodes, especially inside obstacles or redundant regions. This results in low search efficiency and poor path quality. To address this issue, this paper proposes an optimized sampling strategy based on RGG.

In this improved strategy, we pre-generate an RGG on the map, consisting of 1000 uniformly distributed sampling points. These points not only cover various regions of the map but also avoid the node redundancy and over-concentration problems that may appear in the traditional RRT algorithm. Then, during the sampling phase of path planning, the algorithm selects the goal point with a certain probability (using the `goal_bias` strategy). If the goal is not selected, the algorithm chooses a sample point from the RGG that aligns with the current target direction via the function `selectBestRGG`. This approach ensures that the sampling points guide the search tree to grow towards the target region as much as possible. The corresponding pseudocode is provided in Algorithm 2.

Algorithm 2: RGG-based adaptive sampling with KD-Tree (Function `selectBestRGG`)

Require: Goal point `goal`, KD-Tree `rggKDTree`, distance threshold `rgg_threshold`
Ensure: Best sampling point `best_rgg_sample`

```
[idx, dist] ← KNNSEARCH(rggKDTree, goal, 'K', 1)
best_rgg_sample ← rggKDTree.X(idx, :)
```

(Continued)

Algorithm 2 (continued)

3: **if** `dist` > `rgg_threshold` **then**
4: `best_rgg_sample` ← `goal` + RANDN$(1,3)$ × `rgg_threshold`/2
5: `best_rgg_sample` ← max(min(`best_rgg_sample`, `map_size`), 1)
6: **end if**
7: **return** `best_rgg_sample`

To improve search efficiency, we also integrate a KD-Tree structure (`rggKDTree`) to quickly retrieve and select the optimal candidate sampling points. The use of KD-Tree enables near-optimal fast search in high-dimensional space, effectively reducing invalid sampling and redundant computations. Computationally, the RGG-guided sampling involves two phases with distinct complexity profiles: the offline preprocessing requires $O(n_{\text{rgg}} \log n_{\text{rgg}})$ time and $O(n_{\text{rgg}})$ space to generate uniformly distributed points and construct the KD-Tree, while each online query performs a nearest-neighbor search in $O(\log n_{\text{rgg}})$ time with $O(1)$ additional space. This logarithmic query efficiency represents a significant improvement over the $O(n_{\text{rgg}})$ linear scanning of all candidate points.

3.3 Fast Collision Checking Optimization Based on KD-Tree

Collision checking is a critical component in 3D UAV path planning, as it directly affects both the feasibility of the generated path and the real-time responsiveness of the planning system. In the standard RRT algorithm, each path extension typically involves a voxel-by-voxel traversal to evaluate collisions along the entire candidate segment. This approach, while accurate in dense 3D environments reconstructed from point clouds, requires examining a large number of voxel units at each step, resulting in significant computational overhead and reduced efficiency, especially under real-time constraints.

To address this limitation, we propose a fast local collision checking strategy based on a KD-Tree structure, integrated into the improved R-RRT framework. This method constructs and dynamically maintains a KD-Tree to organize the nodes in the current search tree, enabling efficient nearest-neighbor queries in high-dimensional space. During each expansion, instead of checking the entire path globally, the algorithm performs collision detection only on the straight-line segment between the newly generated node and its nearest neighbor in the tree. This significantly reduces the number of voxel checks and accelerates the overall planning process.

More specifically, the search begins by initializing the search tree with the start node q_{start} and building the corresponding KD-Tree. In each iteration, a random sample q_{rand} is drawn from the free space. The algorithm then finds the nearest node q_{near} from the KD-Tree and attempts to steer toward q_{rand}, generating a new node q_{new}. The voxel map is queried to check whether the segment from q_{near} to q_{new} intersects any obstacles. If the segment is collision-free, q_{new} is added to both the search tree and the KD-Tree; otherwise, it is discarded. If q_{new} is sufficiently close to the goal node q_{goal}, the search is deemed successful.

This process is summarized in Algorithm 3, where the RRT tree and KD-Tree are initialized, and iterative sampling, extension, and collision checking are performed. If a feasible path is found, it is extracted by backtracking from the final node; otherwise, an empty path and infinite cost are returned.

The proposed KD-Tree-based local collision checking strategy is designed to accelerate the collision detection process compared to traditional voxel-by-voxel methods, with the aim of enhancing overall planning efficiency in dense 3D environments like substations. Its effectiveness will be quantitatively evaluated in the following experimental section.

Algorithm 3: R-RRT path planning with KD-Tree-based local collision checking

Require: Start node q_{start}, goal node q_{goal}, voxel map map, maximum iterations max_iter
Ensure: Path r_rrt_path, path cost r_rrt_cost
Initialize RRTree $\leftarrow \{q_{start}\}$
Build initial KDTree from RRTree
pathFound $\leftarrow$ FALSE
for $i = 1$ to max_iter **do**
  $q_{rand} \leftarrow$ SAMPLEFREESPACE()
  $q_{near} \leftarrow$ NEARESTNEIGHBOR(KDTree, q_{rand})
  $q_{new} \leftarrow$ STEER(q_{near}, q_{rand})
  if COLLISIONCHECK(q_{near}, q_{new}) **then**
    continue
  end if
  Add q_{new} to RRTree
  Update KDTree with q_{new}
  if DISTANCE(q_{new}, q_{goal}) < threshold **then**
    pathFound $\leftarrow$ TRUE
    break
  end if
end for
if not pathFound **then**

  return $\varnothing, \infty$
end if
r_rrt_path $\leftarrow$ EXTRACTPATH(RRTree)
r_rrt_cost $\leftarrow$ COMPUTECOST(r_rrt_path)

return r_rrt_path, r_rrt_cost

The algorithm outlines the complete R-RRT planning process with KD-Tree acceleration. Computationally, the collision checking module operates in two stages with distinct complexity characteristics: offline construction of a KD-Tree from the obstacle map containing m points requires $O(m \log m)$ time and $O(m)$ space; online querying performs each collision check by querying the k nearest obstacle points within a safety radius using the KD-Tree, which reduces the per-check time complexity from $O(m)$ (linear scan in standard RRT) to $O(k \log m)$ on average, with $O(1)$ additional space per query. Since k is typically small and bounded, this represents a fundamental improvement that enables real-time performance in dense 3D environments.

3.4 Path Post-processing and Gradient Optimization

The paths generated by the traditional RRT algorithm are typically a series of discrete and discontinuous points, suffering from issues such as excessive redundant nodes, severe path zigzags, and poor trackability, making them difficult to directly apply in practical UAV systems. To improve the executability and energy efficiency of the paths, this paper proposes a two-stage path optimization mechanism: Greedy Pruning and Gradient-Based Smoothing.

Path Pruning Optimization: Redundant nodes in the original path are recursively removed. Specifically, starting from the initial point, the algorithm checks whether the current node can directly connect to

the k-th subsequent node without collision; if feasible, intermediate nodes are removed while preserving key turning points, until reaching the endpoint. This strategy effectively shortens the path length and reduces control jumps, thereby improving flight efficiency and path simplicity.

Gradient-Based Smoothing: We design a gradient descent-based path optimization method. The core idea is to minimize either the curvature variation or the jerk of the path as the objective function, while satisfying obstacle avoidance and dynamic constraints, locally adjusting the path points. By setting soft constraints (e.g., maximum offset radius) and hard constraints (e.g., no entry into obstacle regions), the algorithm gradually approaches a continuous, differentiable, and smooth trajectory, suitable for UAV control systems.

The optimization process can be expressed by the following objective function:

$$\min \sum_{i=2}^{n-1} \|x_{i-1} - 2x_i + x_{i+1}\|^2 \tag{15}$$

where $X = \{x_1, x_2, \ldots, x_n\}$ represents the complete set of path points, and $\|x_{i-1} - 2x_i + x_{i+1}\|^2$ denotes the discrete approximation of path curvature at point x_i. Minimizing the objective function in Eq. (15) reduces local path jerkiness while maintaining obstacle avoidance.

The optimization reduces the "zigzagging" in the path, resulting in a more coherent and smooth trajectory, which benefits UAV flight control and attitude maintenance.

The path optimization is subject to the following obstacle avoidance constraints:

$$x_i \notin O, \quad \forall i \in [2, n-1] \tag{16}$$

where x_i denotes the ith waypoint, O represents the obstacle space, and the constraint enforces collision-free intermediate path points between the fixed start (x_1) and goal (x_n) configurations.

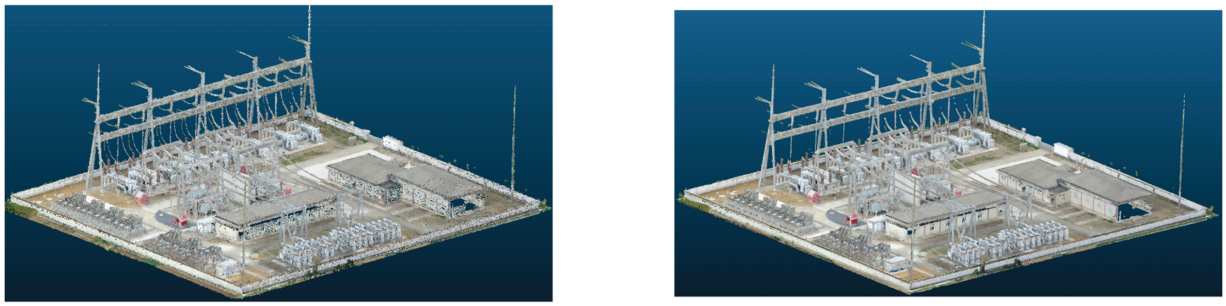
The gradient-based smoothing process, governed by the objective function in Eq. (15) and the constraints in Eq. (16), is applied to enhance path continuity and safety. Computationally, the two-stage post-processing demonstrates efficiency in both time and space: for a path with p nodes, greedy pruning performs a linear pass with $O(p)$ time complexity, while gradient-based smoothing with t iterations operates in $O(t \cdot p)$ time. In terms of space complexity, the processing requires $O(p)$ storage for the path waypoints. Both stages utilize the same KD-Tree structure for collision checking at $O(\log m)$ time per query, yielding an overall time complexity of $O((t+1) \cdot p \cdot \log m)$. Since the number of path nodes p and the iteration count t are bounded in practical applications—with p typically small due to effective pruning—this processing meets real-time requirements while maintaining modest memory footprint. The final optimized path is designed to fulfill the direct execution needs of flight control systems, providing a stable and reliable trajectory for practical deployment.

4 Simulation Experiments and Result Analysis

To verify the effectiveness of the proposed method, simulation tests were conducted using specific case studies. All experiments were conducted on a computational platform equipped with an AMD Ryzen 9 7945HX processor, an NVIDIA GeForce RTX 4060 GPU with 8 GB of VRAM, and 16 GB of DDR5 RAM. Algorithm implementation and testing were performed in MATLAB R2022b.

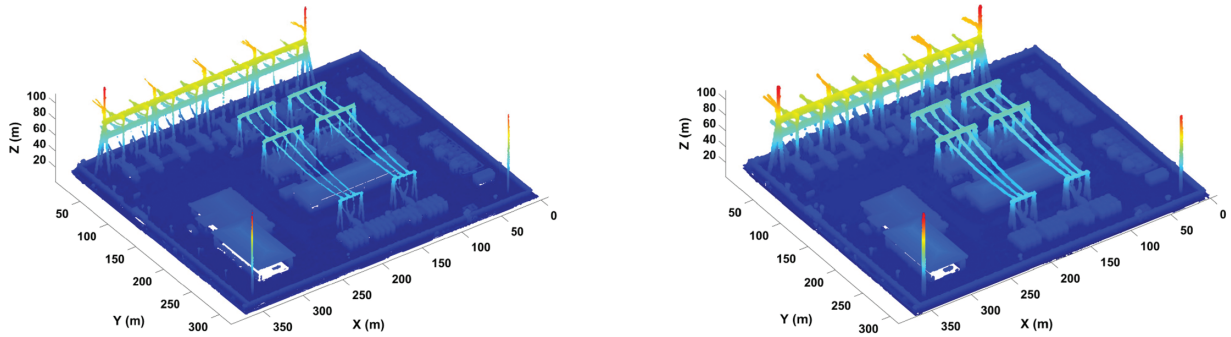
The point cloud data used in the experiments are shown in Fig. 2, including the raw input point cloud and the pre-processed point cloud. Fig. 3 presents the grid maps before and after obstacle inflation, where the 0.5, m dilation radius is derived from the safety model introduced in Section 2.3. The path-planning

algorithm prohibits trajectories from intersecting with the inflated virtual obstacles, thereby enforcing the required safety clearance and ensuring a safe operating margin for UAV inspection tasks.



(**a**) (**b**)

Figure 2: Point cloud data used in the experiments: (**a**) raw input point cloud; (**b**) pre-processed point cloud.



(**a**) (**b**)

Figure 3: Experimental environment modeling results: (**a**) initial voxel grid; (**b**) inflated voxel grid.

4.1 Computational Complexity Analysis

This section provides a comprehensive theoretical complexity analysis of all path planning algorithms evaluated in this study, including the proposed R-RRT and the baseline algorithms (A*, RRT*, Bi-AM-RRT*, BTO-RRT, and CBQ-RRT*). The analysis examines both time and space complexity, as these factors jointly determine the practicality of algorithms in resource-constrained UAV systems. Let n be the number of nodes in the search tree, m the number of obstacle points in the environment, V the number of vertices in the graph representation used by A*, γ the average number of nodes within the rewiring radius of RRT* variants, and n_{rgg} the number of pre-sampled RGG points, and C_{adapt} the constant per-iteration overhead for adaptive metric calculation in Bi-AM-RRT*, and C_{connect} the constant per-iteration overhead for the CreateConnectNode operation in CBQ-RRT*.

Table 1 demonstrates that the proposed R-RRT achieves a time complexity of $O(\log m + \log n + \log n_{\text{rgg}})$ by leveraging KD-Tree collision checking and RGG-guided sampling. This represents a balanced trade-off between computational efficiency and memory usage ($O(m + n + n_{\text{rgg}})$). Specifically, R-RRT trades modest increases in space ($O(n_{\text{rgg}})$) and sampling time ($O(\log n_{\text{rgg}})$) for a substantial reduction in overall search iterations, a strategy distinct from the constant overhead of CBQ-RRT*'s smoothing operations.

Table 1: Time and space complexity comparison of path planning algorithms.

Metric	Time Complexity	Space Complexity	Key Characteristics
A*	$O(\log V)$	$O(V)$	Graph-based search with priority queue
RRT*	$O(m + \log n + \gamma)$	$O(n)$	Linear collision check $O(m)$ dominates
Bi-AM-RRT*	$O(m + \log n + \gamma + C_{\text{adapt}})$	$O(2n)$	Inherits RRT* bottlenecks, bidirectional trees
BTO-RRT	$O(\log m + \log n)$	$O(m + n)$	KD-Tree collision checking ($O(\log m)$)
CBQ-RRT*	$O(\log m + \log n + C_{\text{connect}})$	$O(m + 2n)$	KD-Tree + dual-tree + CreateConnectNode smoothing
R-RRT (Ours)	$O(\log m + \log n + \log n_{\text{rgg}})$	$O(m + n + n_{\text{rgg}})$	KD-Tree check + RGG-guided sampling

The comparison reveals a key design insight: while algorithms like RRT* and Bi-AM-RRT* suffer from the $O(m)$ bottleneck of linear collision checking, those utilizing KD-Trees (BTO-RRT, CBQ-RRT*, R-RRT) achieve superior $O(\log m)$ scalability. This efficiency comes at the cost of higher memory for the tree structure. R-RRT's design justifies this memory investment for static inspection, as pre-computed RGG sampling dramatically reduces online planning time, outweighing the initial resource cost.

4.2 Simulation Experiment

To comprehensively evaluate the path planning performance of the proposed R-RRT algorithm, experimental scenarios are designed from two perspectives: local path optimization and global path generation. All tests were conducted on a voxel grid map with a resolution of 0.5 m. This resolution was chosen to balance the accuracy of environmental representation with computational efficiency for real-time planning: it matches the safety inflation radius (0.5 m) to ensure adequate obstacle representation while maintaining manageable map data volume. At the local level, the equipment-intensive area between Main Transformers 1 and 2 is selected to assess the algorithm's path smoothness and obstacle avoidance capability in dense environments. At the global level, a cross-regional path from Main Transformer 1 to the gantry structure is used to evaluate its sampling efficiency and path feasibility. Considering the stochastic nature of the algorithm, all experiments are repeated 50 times to ensure the statistical reliability of the results.

Intra-Region Path Planning: The positions of the starting and target observation points are listed in Table 2. Fig. 4 shows the path planning results for the A*, RRT*, Bi-AM-RRT*, BTO-RRT, CBQ-RRT*, and the proposed R-RRT algorithms. Table 3 provides a comparative performance analysis of the six algorithms.

Table 2: Target inspection points and coordinates in intra-regional simulation tests.

Target Inspection Point	X Coordinate	Y Coordinate	Z Coordinate
Start Position	246	263	7
Transformer #1 (Full View)	211	113	60
Transformer #1 Current Sensor Phase A	213	140	53
Transformer #1 Current Sensor Phase B	213	113	53
Transformer #1 Current Sensor Phase C	161	124	57
Transformer #2 (Full View)	147	115	60
Transformer #2 Current Sensor Phase A	145	141	55
Transformer #2 Current Sensor Phase B	147	141	55
Transformer #2 Current Sensor Phase C	93	122	53

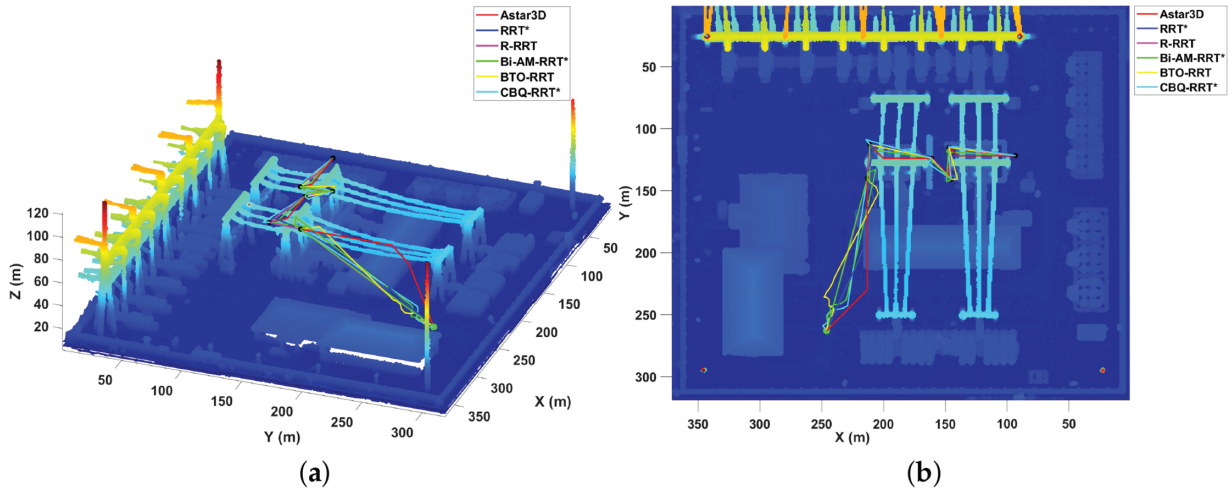


Figure 4: Path planning algorithms comparison in intra-regional scenarios: (**a**) three-dimensional view of path planning algorithms; (**b**) overhead plan view of path planning algorithms.

Table 3: Performance comparison of path planning algorithms (intra-regional test).

Metric	A*	RRT*	R-RRT	Bi-AM- RRT*	BTO- RRT	CBQ- RRT*	Best
Search Time (s)	0.3281	0.0700	0.0698	0.1125	0.0629	0.0723	BTO-RRT
Path Length (m)	342.6968	341.4562	323.8981	337.7726	368.6352	343.6285	R-RRT
Number of Path Points	297	27	13	32	40	26	R-RRT
Maximum Turning Angle (deg)	135.0000	176.1859	67.8309	147.8710	161.4403	176.3086	R-RRT
Turns Exceeding 45° (count)	11	4	1	5	10	5	R-RRT

Inter-Region Path Planning: The positions of starting, main transformer, and gantry observation points are given in Table 4. Fig. 5 shows the path planning results for the A*, RRT*, Bi-AM-RRT*, BTO-RRT, CBQ-RRT*, and the proposed R-RRT algorithms. Table 5 presents the performance comparison results of the six algorithms.

Table 4: Target inspection points and coordinates in inter-regional simulation tests.

Target Inspection Point	X Coordinate	Y Coordinate	Z Coordinate
Start Position	246	263	7
Transformer #1 (Full View)	211	113	60
Transformer #1 Current Sensor Phase A	213	140	53
Transformer #1 Current Sensor Phase B	213	113	53
Transformer #1 Current Sensor Phase C	161	124	57
Transformer #2 (Full View)	147	115	60
Transformer #2 Current Sensor Phase A	145	141	55
Transformer #2 Current Sensor Phase B	147	141	55
Transformer #2 Current Sensor Phase C	93	122	53
220kV Switchyard 4D533 Isolation	121	27	75
220kV Switchyard 4D543 Isolation	154	28	75
220kV Switchyard 4D563 Isolation	188	28	75
220kV Switchyard 4D573 Isolation	249	30	75
220kV Switchyard 4D583 Isolation	281	29	77

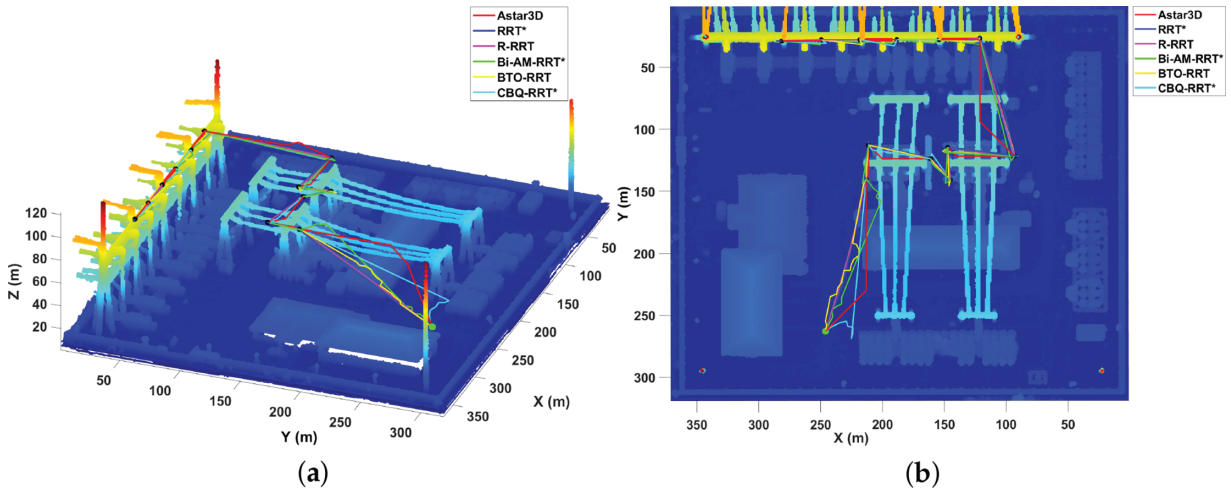


Figure 5: Path planning algorithms comparison in inter-regional scenarios: (**a**) three-dimensional view of path planning algorithms; (**b**) overhead plan view of path planning algorithms.

Table 5: Performance comparison of path planning algorithms (inter-regional test).

Metric	A*	RRT*	R-RRT	Bi-AM- RRT*	BTO- RRT	CBQ- RRT*	Best
Search Time (s)	0.4205	0.1311	0.0873	0.1002	0.0968	0.0957	R-RRT
Path Length (m)	621.5045	637.3423	579.3565	655.3751	633.8624	664.6073	R-RRT
Number of Path Points	717	44	25	61	66	50	R-RRT
Maximum Turning Angle (deg)	180.0000	178.4089	7.4880	151.0354	176.3454	163.7876	R-RRT

(Continued)

Table 5 (continued)

Metric	A*	RRT*	R-RRT	Bi-AM- RRT*	BTO- RRT	CBQ- RRT*	Best
Turns Exceeding 45° (count)	109	9	0	12	10	13	R-RRT

4.3 Qualitative Comparison

Figs. 4 and 5 present typical path planning results of six algorithms: A* [16], RRT* [22], the proposed R-RRT, Bi-AM-RRT* [23], BTO-RRT [24], and CBQ-RRT* [26] in the complex 3D substation environment. The path morphologies generated by these algorithms reflect their distinctive search strategy characteristics and environmental adaptability under complex obstacle distributions.

The A* algorithm demonstrates strong goal-directedness, enabling rapid approach toward the target in open areas. However, due to the inflexibility of its heuristic function in obstacle-dense regions, the path exhibits noticeable detours and sharp turns, resulting in high overall tortuosity and insufficient path smoothness.

The RRT* algorithm, expanding through random sampling, shows good obstacle avoidance capability. Nevertheless, the lack of adaptive guidance leads to "up-floating" or oscillatory paths during the search process, with relatively long path lengths and poor smoothness, which is unfavorable for stable UAV flight in complex 3D spaces.

The Bi-AM-RRT* algorithm, under the effect of bidirectional search and adaptive sampling, produces generally smoother paths. However, it still suffers from local oscillations in obstacle-intensive areas and has limited turning control capability. The BTO-RRT algorithm excels in search efficiency, achieving the shortest planning time, but its paths are longer with more frequent turns, and smoothness is slightly less smooth than R-RRT. Similarly, CBQ-RRT*, with its dual-tree and smoothing operations, enhances connectivity and smoothness to some extent; however, its paths can still show unnecessary spatial deviations and do not consistently converge to the shortest possible trajectory.

In contrast, the proposed R-RRT algorithm introduces RGG-guided sampling and a KD-tree structure into the traditional RRT framework, significantly enhancing sampling space representativeness and search efficiency. Meanwhile, the dynamic goal-biasing strategy based on successful expansion feedback maintains global exploration capability during the initial search phase and progressively strengthens convergence toward the goal direction in later stages, achieving a balance between global search and local optimality. In the post-processing phase, R-RRT employs a two-stage path optimization strategy comprising greedy pruning and gradient descent smoothing, effectively reducing redundant nodes and sharp turning points, thereby improving path continuity and flight executability.

Visually from the figures, the paths generated by R-RRT are more compact and smooth overall, capable of quickly returning to the main direction after obstacle avoidance, with minimal spatial fluctuations, gentle turning angles, and almost no sharp turns. Compared to A*, RRT*, and other algorithms, R-RRT demonstrates superior path controllability and spatial adaptability in complex obstacle environments, better meeting the requirements for smooth and executable paths in substation environments for UAVs.

4.4 Quantitative Comparison

The two sets of experimental data in Tables 3 and 5 quantitatively compare the performance of the six algorithms in terms of search efficiency, path quality, and smoothness metrics.

In terms of search time, BTO-RRT achieved the shortest planning time of 0.0629 s in the first experiment, while R-RRT recorded competitive times of 0.0698 and 0.0873 s in the two experiments, respectively, ranking among the fastest. Notably, in the second experiment, R-RRT's planning time surpassed all other compared algorithms, demonstrating its stable and efficient real-time planning capability across different scenarios. Overall, while BTO-RRT was the fastest in a single trial, R-RRT achieved a superior balance between computational speed and path quality, without significantly sacrificing path optimality for extreme speed.

Regarding path length, the R-RRT algorithm achieved the shortest paths in both experiments at 323.8981 and 579.3565 m, significantly outperforming the other algorithms. This indicates that its improved sampling mechanism and goal-biasing strategy effectively reduce redundant search and enable rapid generation of near-optimal feasible paths.

In terms of path point count and turning characteristics, R-RRT generated the fewest path points (13 and 25) and significantly reduced the number of turns. Its maximum turning angles were only 67.83° and 7.49°, with almost no sharp turns exceeding 45°, outperforming algorithms such as A*, RRT*, and BTO-RRT. This demonstrates that paths generated by R-RRT are smoother and structurally simpler, effectively reducing the energy consumption and time required for UAV flight attitude adjustments.

In summary, R-RRT performs best in terms of path length, smoothness, and turning control, while BTO-RRT holds an advantage in computational efficiency. Bi-AM-RRT* shows slight improvement in path smoothness but lacks the search stability of R-RRT. Although CBQ-RRT* aims to enhance path smoothness, it still falls short in path optimality and geometric simplicity. Overall, R-RRT achieves dual excellence in both planning quality and search efficiency in complex 3D scenarios.

Integrating the qualitative and quantitative results, the improved R-RRT algorithm demonstrates outstanding path planning performance in complex 3D substation environments. By introducing RGG and KD-tree structures to enhance search efficiency, combining a dynamic goal-biasing mechanism to strengthen goal-directedness, and employing a two-stage path smoothing optimization strategy to significantly improve path continuity, R-RRT outperforms A*, RRT*, Bi-AM-RRT*, BTO-RRT, and CBQ-RRT* in path length, smoothness, and spatial adaptability. The algorithm not only generates shorter and smoother feasible paths but also exhibits clear advantages in search stability and environmental adaptability, validating the effectiveness and practical value of R-RRT for UAV path planning tasks in complex 3D spaces.

5 Conclusion

To enhance the path planning performance of the RRT algorithm in dense three-dimensional environments, this paper proposes an R-RRT. The method builds upon a 3D occupancy grid map that integrates ground filtering and obstacle inflation, achieving performance improvements through the following mechanisms: First, a dynamic goal-biasing strategy is designed to adaptively adjust the sampling probability distribution based on the historical expansion success rate, effectively balancing global exploration and local convergence. Second, an RGG-guided sampling mechanism is introduced to optimize the growth direction of the random tree through a predefined spatial topological structure, significantly improving path search efficiency and reducing the generation of invalid nodes. To address the issue of redundant turning points in initial paths, a two-stage optimization strategy is proposed: initially removing redundant nodes using a greedy pruning algorithm, followed by trajectory smoothing via a gradient descent method, ensuring the

generated paths meet the practical flight requirements of UAVs. Experimental results demonstrate that in typical three-dimensional substation scenarios, R-RRT outperforms mainstream comparative algorithms in key metrics including path length, planning time, and trajectory smoothness.

It should be noted that R-RRT is primarily optimized for static environments with high-fidelity prior maps, and its performance may be limited in the following scenarios: sudden or moving obstacles may exceed the algorithm's current real-time replanning capability, leading to suboptimal or unsafe paths; frequent updates to environmental maps may cause planning delays or reduced path feasibility; measurement noise may degrade the accuracy of the occupancy grid, potentially resulting in collision risks or overly conservative path planning. To address these challenges, future research will focus on three main directions: (1) integrating a real-time replanning module with dynamic obstacle prediction; (2) enhancing environmental adaptability through incremental map updating and multi-sensor fusion; and (3) improving computational efficiency in highly complex scenarios via parallel computing and adaptive sampling. These enhancements will extend the applicability of the method to semi-dynamic industrial inspection tasks.

Acknowledgement: Not applicable.

Funding Statement: Funding for this research was provided by the Program for Scientific Research Innovation Team in Colleges and Universities of Anhui Province (No. 2022AH010095) and the Hefei Key Technology R&D "Champion-Based Selection" Project (No. 2023SGJ011).

Author Contributions: The authors confirm their contributions to this paper as follows: study conception by Yanping Chen and Zhengxin Zhan; research methodology design by Xiaohui Yan; software development by Zhengxin Zhan; validation by Le Zou; formal analysis by Hailei Wang; investigation by Yucheng Zhong; resources provision by Hailei Wang; data curation by Zhengxin Zhan; original draft preparation by Yanping Chen and Zhengxin Zhan; review and editing by Le Zou; visualization by Yucheng Zhong; supervision by Xiaohui Yan and Le Zou; project administration by Hailei Wang; funding acquisition by Yanping Chen. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bouzid Y, Siguerdidjane H, Bestaoui Y. Flight control boosters for three-dimensional trajectory tracking of quadrotor: theory and experiment. Proc Inst Mech Eng Part I J Syst Control Eng. 2018;232(6):709–27. doi:10.1177/0959651818757159.
2. Mukhamediev RI, Yakunin K, Aubakirov M, Assanov I, Kuchin Y, Symagulov A, et al. Coverage path planning optimization of heterogeneous UAVs group for precision agriculture. IEEE Access. 2023;11:5789–803. doi:10.1109/ACCESS.2023.3235207.
3. Khan NA, Jhanjhi NZ, Brohi SN, Usmani RSA, Nayyar A. Smart traffic monitoring system using unmanned aerial vehicles (UAVs). Comput Commun. 2020;157(4):434–43. doi:10.1016/j.comcom.2020.04.049.
4. Dissanayaka D, Wanasinghe TR, De Silva O, Jayasiri A, Mann GKI. Review of navigation methods for UAV-based parcel delivery. IEEE Trans Autom Sci Eng. 2024;21(1):1068–82. doi:10.1109/TASE.2022.3232025.
5. Du Q, Dong W, Su W, Wang Q. UAV inspection technology and application of transmission line. In: Proceedings of the 2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE); 2022 Sep 23–25; Dalian, China. Piscataway, NJ, USA: IEEE; 2022. p. 594–7. doi:10.1109/ICISCAE55891.2022.9927674.

6. Luo Y, Yu X, Yang D, Zhou B. A survey of intelligent transmission line inspection based on unmanned aerial vehicle. Artif Intell Rev. 2023;56(1):173–201. doi:10.1007/s10462-022-10189-2.
7. Gaspar J, Cruz T, Lam CT, Simões P. Smart substation communications and cybersecurity: a comprehensive survey. IEEE Commun Surv Tutor. 2023;25(4):2456–93. doi:10.1109/COMST.2023.3305468.
8. Yang X, Ye X, Cao J, Yan R, Guo X, Huang J. High-altitude inspection technology of substation based on fusion of unmanned aerial vehicle and multiple sensors. Sens Mater. 2022;34(8):3191–212. doi:10.18494/SAM3933.
9. Guan H, Sun X, Su Y, Hu T, Wang H, Wang H, et al. UAV-lidar aids automatic intelligent powerline inspection. Int J Electr Power Energy Syst. 2021;130:106987. doi:10.1016/j.ijepes.2021.106987.
10. Fan Z, Zhang L, Wang X, Shen Y, Deng F. LiDAR, IMU, and camera fusion for simultaneous localization and mapping: a systematic review. Artif Intell Rev. 2025;58(6):1–59. doi:10.1007/s10462-025-11187-w.
11. Hornung A, Wurm KM, Bennewitz M, Stachniss C, Burgard W. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton Robot. 2013;34(3):189–206. doi:10.1007/s10514-012-9321-0.
12. Ren Y, Cai Y, Zhu F, Liang S, Zhang F. ROG-Map: an efficient robocentric occupancy grid map for large-scene and high-resolution lidar-based motion planning. In: Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2024 Oct 14–18; Abu Dhabi, United Arab Emirates. Piscataway, NJ, USA: IEEE; 2024. p. 8119–25. doi:10.1109/iros58592.2024.10802303.
13. Peng CC, Wang CY. Design of constrained dynamic path planning algorithms in large-scale 3D point cloud maps for UAVs. J Comput Sci. 2023;67(2):101944. doi:10.1016/j.jocs.2023.101944.
14. Geng S, Ning Z, Zhang F, Zhou B. EPIC: a lightweight lidar-based UAV exploration framework for large-scale scenarios. IEEE Robot Autom Lett. 2025;10(5):5090–7. doi:10.1109/LRA.2025.3555878.
15. Zhao W, Wang H, Liu YJ, Liu L. An improved RRT* drone three-dimensional path-planning algorithm based on point cloud maps. Proc Inst Mech Eng Part I J Syst Control Eng. 2024;37:09596518241291182. doi:10.1177/09596518241291182.
16. Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybern. 1968;4(2):100–7. doi:10.1109/tssc.1968.300136.
17. Alshammrei S, Boubaker S, Kolsi L. Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance. Comput Mater Contin. 2022;72(3):5939–54. doi:10.32604/cmc.2022.028165.
18. LaValle SM, Kuffner JJ Jr. Randomized kinodynamic planning. Int J Robot Res. 2001;20(5):378–400. doi:10.1177/02783640122067453.
19. Kavraki LE, Svestka P, Latombe JC, Overmars MH. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans Robot Autom. 2002;12(4):566–80. doi:10.1109/70.508439.
20. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. Int J Robot Res. 1986;5(1):90–8. doi:10.1177/027836498600500106.
21. Pore E, Patle BK, Thorat S. Multi-drone path planning using gradient descent and potential fields for warehouse applications. In: Proceedings of the 2024 International Automatic Control Conference (CACS); 2024 Oct 31–Nov 3; Taoyuan, Taiwan. Piscataway, NJ, USA: IEEE; 2024. p. 1–6. doi:10.1109/CACS63404.2024.10773359.
22. Chen L, Shan Y, Tian W, Li B, Cao D. A fast and efficient double-tree RRT*-like sampling-based planner applying on mobile robotic systems. IEEE-ASME Trans Mechatron. 2018;23(6):2568–78. doi:10.1109/tmech.2018.2821767.
23. Zhang Y, Wang H, Yin M, Wang J, Hua C. Bi-AM-RRT*: a fast and efficient sampling-based motion planning algorithm in dynamic environments. IEEE Trans Intell Veh. 2023;9(1):1282–93. doi:10.1109/tiv.2023.3307283.
24. Zheng Z, Bewley TR, Kuester F. Point cloud-based target-oriented 3D path planning for UAVs. In: Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS); 2020 Sep 1–4; Athens, Greece. Piscataway, NJ, USA: IEEE; 2020. p. 1–8. doi:10.1109/icuas48674.2020.9213894.
25. Zhang ZW, Jia YW, Su QQ, Chen XT, Fu BP. ATS-RRT*: an improved RRT* algorithm based on alternative paths and triangular area sampling. Adv Robot. 2023;37(10):605–20. doi:10.1080/01691864.2023.2174817.
26. Ye L, Li J, Li P. Improving path planning for mobile robots in complex orchard environments: the continuous bidirectional Quick-RRT* algorithm. Front Plant Sci. 2024;15:1337638. doi:10.3389/fpls.2024.1337638.

27. Waga A, Benhlima S, Bekri A, Abdouni J, Saber FZ. A survey on autonomous navigation for mobile robots: from traditional techniques to deep learning and large language models. J King Saud Univ Comput Inf Sci. 2025;37(7):198. doi:10.1007/s44443-025-00216-x.
28. Xie H, Yang D, Xiao L, Lyu J. Connectivity-aware 3D UAV path design with deep reinforcement learning. IEEE Trans Veh Technol. 2021;70(12):13022–34. doi:10.1109/TVT.2021.3121747.
29. Rocha LGS, Caldas KAQ, Terra MH, Ramos F, Vivaldini KCT. Dynamic Q-planning for online UAV path planning in unknown and complex environments. Int J Intell Robot Appl. 2025;9(4):1654–74. doi:10.1007/s41315-025-00457-z.