2025 International Conference on
Unmanned Aircraft Systems (ICUAS)
May 14-17, 2025. North Carolina, USA

Multi-resolution UAV Path Replanning for Inspection of Tailings Dams

Paulo V. G. Simplicio and Guilherme A. S. Pereira

Abstract— **Autonomous inspection of large and complex structures with a commercial unmanned aerial vehicle (UAV) is a challenging problem that has been addressed in recent years. In this paper, we address the global motion planning problem of creating autonomous inspection missions for UAVs considering photogrammetry constraints. We focus on the inspection of large tailings dams, which are dam structures used to store waste byproducts of mining. Our method uses a prior sparse point cloud of the dam to generate a voxel grid, where paths satisfying photogrammetry constraints are tested for collisions. We then apply the A* algorithm as a local planner to avoid obstacles within the global mission. Moreover, we address the problem of changing routes online by using octree-based multi-resolution grids for efficient and fast pathfinding. Our results, obtained using tridimensional maps of an actual coal mine tailings dam, show that using octrees for multi-resolution motion planning is faster than using a fixed voxel grid in online missions while inspecting large structures.**

I. INTRODUCTION

In recent years, autonomous inspections using unmanned aerial vehicles (UAVs) have become an efficient, fast, and safe alternative to traditional inspection methods, especially in complex and hazardous environments such as tailings dams, underground mines, skyscrapers, bridges, and power lines (see, for instance, [1], [2]). Several factors contribute to the rapid increase of the number of drones in inspection tasks, including their high flexibility and their ability to navigate challenging environments. In addition, UAVs reduce operational costs and eliminate risks to human life, making them essential for inspecting difficult-to-reach areas. However, ensuring reliable and efficient motion planning for UAVs in these practical scenarios remains a significant challenge due to factors such as the quality of environmental maps, uncertain obstacles, computational limitations, and the requirement for online decision-making. This paper presents an approach for performing offline and online motion planning for autonomous UAVs performing inspection missions over large structures, with a focus on the inspection of tailings dams, as shown in Fig. 1. Tailings dams are large embankment dams that store waste byproducts of mining. Frequent inspection is essential to prevent accidents [3].

In structured and well-known environments, voxel grids are an alternative for representing 3D spaces and enabling fast collision checks during motion planning (see, for instance, [4], [5]). In summary, the voxelization process discretizes the environment into uniform cells called 3D voxels, allowing efficient pathfinding using grid-based algorithms like Dijkstra and A* [6], [7]. Based on the quality of the map, this approach ensures accuracy and resolution optimality, making it suitable for pre-defined inspection missions. However, the computational time to find an optimal path in large environments using small voxel sizes can be a limitation for online applications. To address this, hierarchical structures, such as octrees have been proposed to represent multi-resolution voxel grids [8].

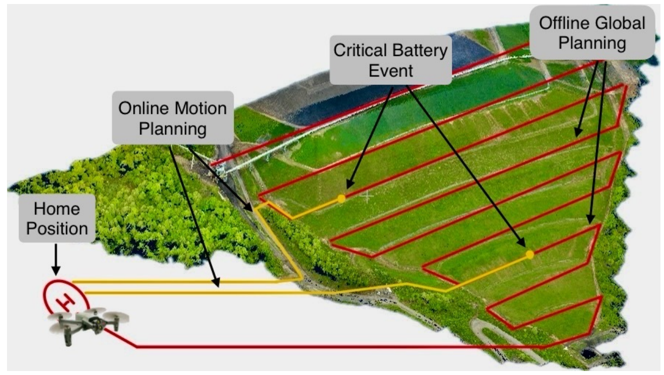


Fig. 1: Integrated motion planning framework for autonomous inspections of large structures using a commercial UAV: Global planning defines the primary coverage path (—) for surveying the field considering photogrammetry constraints, while online motion planning ensures dynamic obstacle avoidance and route adjustments. A critical battery event (●), for example, triggers immediate path replanning to make the drone return to its home position safely.

An octree is a hierarchical data structure used to partition the 3D space into smaller cubic regions, often referred to as voxels, cells, or nodes. It consists of multiple layers, each with varying resolutions. At the root (layer 0), the octree represents the entire 3D space as a single cube. Each subsequent layer subdivides each cell into 8 smaller cubes, continuing until the finest layer, with the highest resolution, is reached [9], [10]. In recent years, many researchers have shown that octrees can significantly reduce computational costs for storage, collision detection, and pathfinding. For example, the works of [10] and [11] demonstrated the effectiveness of combining octrees with probabilistic occupancy estimation, enabling UAVs to navigate uncertain spaces. Additionally, they provided practical experiments to demonstrate that the framework provides a fast response in

This work was supported by the Department of Energy of the United States of America (DOE), Grant DE-FE0032206.

Paulo V. G. Simplicio and Guilherme A. S. Pereira are with the Department of Mechanical, Materials and Aerospace Engineering, Benjamin M. Statler College of Engineering and Mineral Resources, West Virginia University. Morgantown, WV, 26501, USA. Emails: pvg00001@mix.wvu.edu, guilherme.pereira@mail.wvu.edu

online applications.

The use of octrees has proven essential for applications that integrate global path planning with local motion planning, especially when the UAV needs to adjust its precomputed path suddenly [8], [12]. In online applications, when the UAV needs to change the course of the mission, either to return home due to a critical battery level or to inspect a potential hazard closely, the motion planning algorithm is applied first to a coarse layer and then to deeper layers, iteratively. This approach aims to speed up pathfinding by eliminating regions of the area where the optimal path is not located.

The latest research has demonstrated numerous techniques for UAVs to inspect various structures, as presented in [13] and [14]. A few research groups, such as [15] and [16], have explored photogrammetry techniques during the inspection of large areas. However, the literature lacks an integration of motion planning algorithms with photogrammetry techniques that utilize a previously acquired sparse point cloud of the area to plan both offline and online missions. By integrating both techniques, we can efficiently address the inspection of large areas using a commercial drone while taking photogrammetry constraints into account.

In photogrammetry-based autonomous UAV inspection missions, it is crucial to consider specific constraints, such as maintaining a fixed distance from the area of interest to ensure accurate 3D map reconstruction. In these cases, it is necessary to plan the global mission around photogrammetry constraints, while ensuring total coverage of the area and navigability for the UAV. A few recent papers in the literature have developed path-planning algorithms that consider photogrammetry constraints and use a prior model of the area to plan the mission [17], [18]. However, these papers do not consider the adaptability of the mission or the online change of route, which is essential for the inspections of large structures where the drone must return home multiple times to replace the battery.

In our previous work [19], we developed a motion planning approach for autonomous dam inspections using a commercial UAV. In [20], we proposed a behavior tree for battery-aware inspection of large structures, enabling the drone to return home for battery replacement and resume the mission as needed. In both works, path planning was assumed to occur in free space, allowing the drone to return home in a straight line from any point. In practical scenarios, however, the UAV must fly close to the area of interest, which may contain obstacles such as trees, power lines, and towers. In such cases, the motion planning algorithm must account for these obstacles. Building on this foundation, this paper presents a motion planning approach for autonomous UAV inspections of large structures. Planning is based on a previous sparse point cloud of the area and considers photogrammetry constraints. Additionally, it enhances mission adaptability by performing online multi-resolution motion planning, allowing the UAV to change its route safely when needed.

The remainder of this paper is organized as follows. Section II defines the problem addressed in this work and provides insights into our proposed solution. Section III discusses the methodology for global motion planning and online pathfinding in inspection missions. Additionally, it analyzes the computational complexity of pathfinding using A* in both a voxel grid and an octree. Next, Section IV describes the experimental setup, presents the results, and compares online motion planning using voxel grids with multi-resolution approaches based on octrees. Finally, Section V concludes the paper and outlines perspectives for future work.

II. PROBLEM DEFINITION

The overall problem solved in this paper is motion planning for the inspection of large areas using a commercial UAV that lacks built-in obstacle avoidance capabilities. Our primary goal is to accelerate online pathfinding by using an efficient representation of the baseline point cloud that represents the area to be inspected. This is crucial in situations where the drone must deviate from a pre-planned (offline) mission during an autonomous inspection operation. The key motivations to solve this problem include:

- Battery constraints: UAVs have limited flight time, especially in severe weather conditions, and must periodically return to a home base for battery replacement, requiring efficient planning to minimize downtime;
- Reactive dynamic inspection: The UAV may encounter unexpected regions of interest (e.g., cracks, seepage) that require immediate investigation, necessitating online adaptation of its path;
- Obstacle avoidance: The environment may contain obstacles that must be avoided to ensure safe navigation, requiring both offline and online path-planning methods that balance efficiency and safety;

Thus, in summary, our problem lies in developing an autonomous navigation strategy that enables a commercial UAV to efficiently perform an inspection mission while safely navigating the obstacles in the environment. The proposed strategy must ensure comprehensive coverage based on photogrammetry requirements while maintaining adaptability to dynamic conditions.

III. METHODOLOGY

In this section, we outline our proposed methodology. The process begins with creating a sparse point cloud using photogrammetry, where a UAV flies at the maximum allowable altitude of 400 feet (FAA regulations in the US). The initial flight follows a back-and-forth path based on parameters such as ground sampling distance (GSD), coverage rows, and overlap, as presented in [19]. This preliminary mission does not account for detailed structure geometry and is used solely for mission planning. Due to the high altitude, the resulting sparse point cloud lacks the detail necessary for hazard detection but serves as a foundation for planning subsequent missions. These future missions refine the UAV's path, bringing it closer to the area of interest while accounting for obstacles.

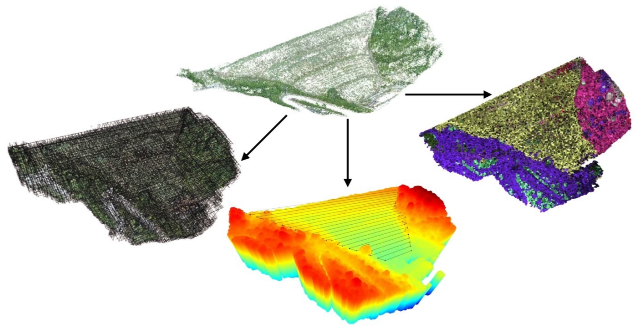


Fig. 2: Illustration of the prepossessing steps for tailings dam inspection: A sparse point cloud (top) is used as the basis for creating a voxel grid (bottom), an octree (left), and identifying the main plane to be inspected (right). The main plane is essential for extracting critical information about the structure to generate paths that satisfy photogrammetry constraints, such as the one illustrated above the voxel grid.

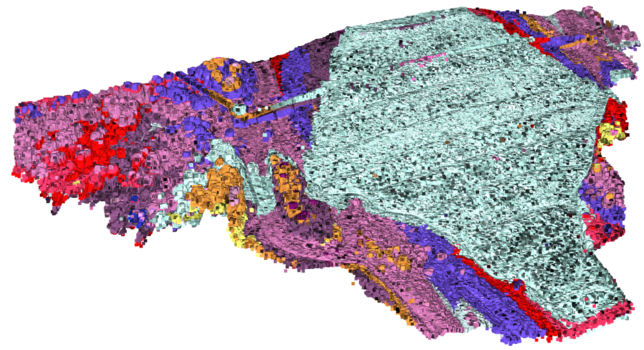


Fig. 3: Plane segmentation applied to a sparse point cloud of a coal mine tailings dam using the RANSAC algorithm. The main plane, which corresponds to the face of the downstream slope is the largest detected surface, presented in a lighter color (●).

Using the sparse point cloud, we perform several key steps for motion planning, including plane segmentation, voxelization, collision checking, and local path planning to avoid obstacles as illustrated in Fig. 2. The segmentation process identifies the primary plane, which is crucial for defining the drone's main inspection path using [19]. This initial path is usually a back-and-forth coverage path, which is overlaid onto the voxel grid, generated from the point cloud, to detect collisions. On the voxel grid, the A* algorithm is used to locally re-plan paths where collisions occur. Additionally, we construct an Octree structure to enable fast online motion planning. The next subsections focus on presenting the two main components of our methodology: global motion planning and online multi-resolution motion planning.

A. Global Motion Planning

This section describes the steps performed to generate the global motion planning mission offline.

1) Plane Segmentation: From a sparse point cloud, we automatically segment the main plane of the structure to be inspected, which will be used to define the inspection boundaries. For this operation, we apply Open3D's [21] RANSAC-based plane detection method iteratively. The algorithm detects a dominant plane using Random Sample Consensus (RANSAC) [22], extracting its equation coefficients a, b, c, and d and the points belonging to the plane. The identified planes are saved, and the boundaries of the main plane are then obtained by connecting the maximum and minimum points at the left and right in both the bottom and upper parts of the plane (see Fig. 3). These boundaries are later used for path generation. Additionally, we calculate the normal vector of the plane $n = [a,\, b,\, c]$, and compute the slope of the plane with respect to the horizontal axis using

$$\theta = \arcsin\left(\frac{c}{||n||}\right), \tag{1}$$

where $||n|| = \sqrt{a^2 + b^2 + c^2}$ is the magnitude of the normal vector and θ, which represents the angle of the plane relative to the horizontal plane, is used to rotate the generated initial path to match the slope of the plane (see [19]).

2) Waypoints Generation: Considering the plane boundaries obtained in the previous step, the goal is to create a back-and-forth path to generate a photogrammetry-based 3D map. For this step, the methodology discussed in [19] is applied, incorporating parameters such as the proximity of the UAV to the area of interest, the GSD, the camera footprint, the number of rows of coverage, the distance between the rows of coverage, the overlap and the lateral speed. Based on this, we generate a path consisting of a sequence of waypoints, which is used to navigate the UAV.

3) Voxelization and Local Planning: The sparse point cloud is discretized into a voxel grid using the voxelization process available in the Open3D library. Specifically, the points are converted into voxels with a defined size. This discretization allows collision checking and facilitates the application of motion planning algorithms.

After generating the initial path, it is incorporated into the voxel grid, where each segment is discretized, and collisions with obstacles are checked. When a collision is detected, the algorithm records the last point before the collision and the first point after it. The A* algorithm is then applied to compute an optimal path around the obstacle using a 6-connected voxel grid, with the Euclidean distance as the heuristic.

For each collision detected, Algorithm 1 is used to locally change the path. This algorithm takes as input the waypoints before and after the collision (p_{before} and p_{after}), the voxel grid (v_{grid}), and the resolution of the voxel grid (δ), given by the size of each voxel. The output consists of path ($\mathcal{T}_p$), which is a smoother version of the path initially found by the A* algorithm. In lines 4–5 of the algorithm, the corresponding points before and after the obstacle are mapped

Algorithm 1 Path Planning Using Voxel Grid and A*

1: **Input:** p_{before} (point before), p_{after} (point after), v_{grid} (voxel grid), δ (resolution)
2: **Output:** $\mathcal{T}_p$
3: Initialize $\mathcal{T}_p \leftarrow \emptyset$
4: $x_{\text{init}} \leftarrow \lfloor p_{\text{before}}/\delta \rfloor$
5: $x_{\text{goal}} \leftarrow \lfloor p_{\text{after}}/\delta \rfloor$
6: $V \leftarrow$ A*(x_{init}, x_{goal}, v_{grid}, δ)
7: $n \leftarrow$ length(V), $i \leftarrow 0$, $V_p \leftarrow \{x_n\}$
8: **repeat**
9:     **if** ObstacleFree(x_n, x_i) **then**
10:         $V_p \leftarrow V_p \cup \{x_i\}$
11:         Remove intermediate points between x_n and x_i
12:         $n \leftarrow i$, $i \leftarrow 0$
13:     **else**
14:         $i \leftarrow i + 1$
15:     **end if**
16: **until** $n = 0$
17: $\mathcal{T}_P \leftarrow \mathcal{T}_P \cup V_p$
18: **Return** $\mathcal{T}_p$

to voxel indices using the size of the voxel, δ. Then, A* is executed in line 6 to find a collision-free path between them. This ensures that the original path is efficiently adapted while avoiding obstacles. Then, the A*-generated path undergoes post-processing to remove unnecessary waypoints while maintaining obstacle-free connections in lines 7-16. This process begins by attempting to connect the first and last waypoints directly. If a collision is detected, the algorithm incrementally checks intermediate waypoints, starting from the last waypoint to the second, then the third, and so on, until it finds the closest waypoint that allows a direct, collision-free connection, excluding those in between. This process is repeated iteratively from the newly connected waypoint until the entire path is optimized. This approach smooths the path while preserving obstacle avoidance.

B. Online Multi-resolution Motion Planning

For online planning, instead of voxel grids, we propose the use of octrees to represent the 3D area of interest. The octree is constructed by recursively subdividing the space into cubic nodes, creating a multi-resolution representation. With a depth of n layers, an octree starts with 1 root node at Layer 0 and expands to 8^n nodes at Layer n. The size of each node decreases from S_{root} meters at the root to $S_{\text{root}}/2^n$ meters at the deepest layer. Each node is assigned an occupancy probability (P_{occ}), which is calculated based on the point density within the node, enabling the classification of nodes as either free or occupied. Given an octree leaf node containing N points and a cubic volume of $V = l^3$, where l is the edge length of the node, the occupancy probability is given by $P_{\text{occ}} = \min(1, N/V)$ to ensure that the probability is bounded within $[0, 1]$. A node is considered occupied or free based on whether the computed occupancy probability

Algorithm 2 Multi-resolution Path Planning Using A*

1: **Input:** $x_{\text{init}}, x_{\text{goal}}$, O (octree with layers $l_1, \ldots, l_n$), δ_0 (initial resolution)
2: **Output:** $\mathcal{P}_{l_n}$ (path)
3: Initialize $\mathcal{P}_{l_1} \leftarrow \emptyset$
4: Project $x_{\text{init}}, x_{\text{goal}} \rightarrow l_1$ with resolution $\delta_0/2$
5: $\mathcal{P}_{l_1} \leftarrow$ A*(x_{init}, x_{goal}, l_1, $\delta_0/2$)
6: **for** $i = 1$ to $n - 1$ **do**
7:     Initialize $\mathcal{T}_{l_{i+1}} \leftarrow \emptyset$
8:     **for** each node $v \in \mathcal{P}_{l_i}$ **do**
9:         $\mathcal{T}_{l_{i+1}} \leftarrow \mathcal{T}_{l_{i+1}} \cup$ Subdivide(v)
10:     **end for**
11:     Project $x_{\text{init}}, x_{\text{goal}} \rightarrow l_{i+1}$ with $\delta_0/2^{i+1}$
12:     $\mathcal{P}_{l_{i+1}} \leftarrow$ A*(x_{init}, x_{goal}, $\mathcal{T}_{l_{i+1}}$, $\delta_0/2^{i+1}$)
13: **end for**
14: **Return** $\mathcal{P}_{l_n}$

(P_{occ}) exceeds a threshold (P_{thresh}), as follows:

$$\text{Node status} = \begin{cases} \text{Occupied}, & \text{if } P_{\text{occ}} \geq P_{\text{thresh}} \\ \text{Free}, & \text{if } P_{\text{occ}} < P_{\text{thresh}} \end{cases}. \tag{2}$$

In our system, path planning uses the A* algorithm, applied iteratively across layers, to improve path resolution. Let l_1 denote the coarser layer and l_n the finest layer. First, the start and goal positions x_{init} and x_{goal} are projected into the grid resolution of l_1. After computing a path $\mathcal{P}l_1$, only the nodes that form $\mathcal{P}l_1$ are subdivided into 8 smaller nodes for l_2. This creates a localized "tunnel" in l_2 with a resolution $\delta_0/2^2$, significantly reducing the search space for A*. This refinement process continues iteratively across layers until the final layer l_n is reached at the finest resolution, ensuring both optimality and computational efficiency.

Algorithm 2 presents the proposed approach. Lines 6-10 describe the iterative refinement process that improves path resolution across successive octree layers. At each iteration i, the path $\mathcal{P}_{l_i}$ found in the current layer is subdivided to generate a localized search space, $\mathcal{T}_{l_{i+1}} \subseteq \mathcal{P}_{l_i}$, for the finer layer l_{i+1}. The x_{init} and x_{goal} positions are projected into l_{i+1} with a refined resolution of $\delta_0/2^{i+1}$. The A* algorithm computes the path $\mathcal{P}_{l_{i+1}}$ within $\mathcal{T}_{l_{i+1}}$, iteratively optimizing and narrowing the search space.

C. Computational Complexity Analysis

This section presents the computational analysis of the proposed planning methods.

1) Planning over a Voxel Grid: A voxel grid uniformly discretizes the environment into fixed-size cubic voxels. The total number of voxels, N, grows cubically with the dimensions of the environment:

$$N = \frac{n_x}{\delta} \times \frac{n_y}{\delta} \times \frac{n_z}{\delta}, \tag{3}$$

where n_x, n_y, n_z are the dimensions of the environment in the x, y, and z axis, respectively, and δ is the resolution of the voxel. For pathfinding with A*, the complexity is well defined in the literature as $O(N \log N)$.

Fig. 4: Environment used for autonomous inspection missions. The figure shows the slope of a coal mine tailings dam (approximately 205,152 m²), located in Greene County, Pennsylvania, USA.

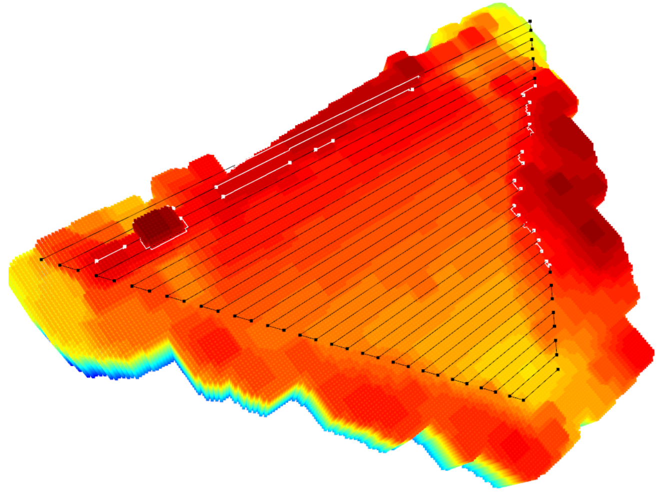


Fig. 5: Sequence of processes applied under photogrammetry constraints: path visualization over the voxel grid, collision detection showing points before and after collision events (highlighted in white), and motion planning results using the A algorithm applied to the voxel grid, depicted in black.

2) Planning over an Octree: The computational complexity of planning at each layer of the octree, starting from the root, is a function of $N_o^l = 8^l$, which is the number of nodes at each layer (l). This number increases as the resolution increases. The final computational complexity is the sum of all layers, where the time increases as the number of nodes increases exponentially. For each layer, the complexity of A* can be calculated as

$$O\left(N_o \log N_o\right) . \tag{4}$$

For the worst-case scenario, where the A* algorithm needs to search for a path in the entire layer, the final complexity is the sum of all layers given by

$$\sum_{l=1}^{n} O\left(N_o^l \log N_o^l\right) , \tag{5}$$

where n is the total number of layers. However, considering that at each layer, the A* algorithm searches only among the children of the path found in the previous layer, the search space is reduced by a factor γ, where $0 < \gamma < 1$. In this case, the complexity is proportional to

$$\sum_{l=1}^{n} \left(\gamma N_o^l \log(\gamma N_o^l)\right) , \tag{6}$$

which represents a geometric sum due to the iterative nature of the path refinement. By assuming a decreasing number of nodes at each layer, the final computational complexity is given by:

$$O(K \log K), \tag{7}$$

where $K = N_o^n$ represents the number of nodes in the final layer of the path tunnel. Since the search is restricted only to the necessary regions, this is much smaller than the full octree complexity. As a result, the A* algorithm can find a path more efficiently compared to its application in a standard voxel grid.

IV. EXPERIMENTS

This section presents experiments that illustrate and evaluate our motion planning approaches. The first subsection introduces our experimental setup, while the second subsection presents the results of global path planning for autonomous inspection missions considering photogrammetry constraints to generate a 3D map of the area. Finally, the third subsection presents the results of multi-resolution online local planning, used when the drone needs to interrupt its current path and return home from a random position. A comparison of computational time is provided between the paths found using a grid and those found using an octree.

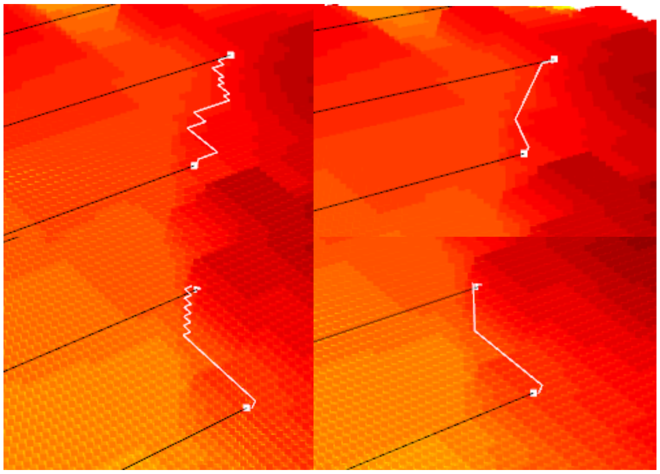
A. Experimental Setup

To test the developed approaches and create the path considering photogrammetry constraints, the Parrot Anafi USA Gov was selected as the commercial UAV. The camera parameters used were presented in the work of [19], along with the methodology to generate the path without considering obstacles. The inspection missions in this study were designed for a tailings dam located in Greene County, Pennsylvania, USA, and presented in Fig. 4.

To implement the algorithms and test the computational performance, we used a Dell Inspiron 14 laptop with an Intel Core i9-14900HX processor and NVIDIA GeForce RTX 4070 GPU. The machine is equipped with 32GB of DDR5 RAM and a storage unit of 1TB SSD.

B. Global Motion Planning Results

To test the global planning on a voxel grid, we designed autonomous inspection missions that would fly close to the area of interest at a distance of 30 m. The parameters of the mission are presented in Table I, while the graphical result illustrating the generated path after collision detection



(a) (b)

Fig. 6: A closer look at path planning in the voxel grid: (a) Motion planning results using the A* algorithm, and (b) post-processing of the path for removal of unnecessary waypoints.

and re-planning is shown in Fig. 5. The figure depicts the path planning based on photogrammetry constraints over a voxel grid, highlighting points before and after collisions. Moreover, it shows the local planning process using the A* algorithm.

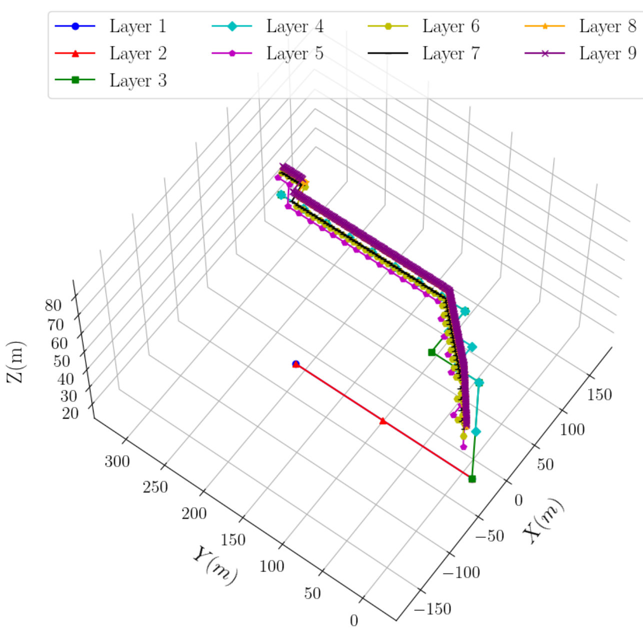


Fig. 8: Visualization of paths computed using the A* algorithm across 9 layers of the octree. Each path starts at the center of the cell where x_{init} is located and proceeds to the center of the cell containing x_{goal}, traversing through successive layers with increasing resolution. This multi-resolution approach ensures optimal pathfinding by refining the path progressively from the coarser to the finer layers.

As shown in Fig. 5, the motion planning algorithm on the voxel grid successfully handled the example mission by storing points before and after collisions and reconnecting them to generate an obstacle-free path. Fig. 6 presents a closer look of the local planning path, highlighting the points before and after collisions. The left side (a), presents the path found by the A* algorithm and the right side (b), presents the path after the post-processing step that removes unnecessary waypoints.

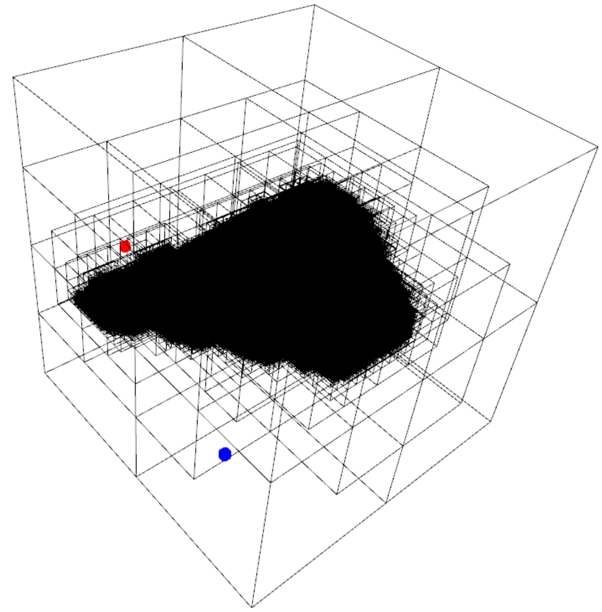


Fig. 7: Octree of the tailings dam used for online multi-resolution pathfinding. The red (■) and blue (■) squares indicate the start and goal points, respectively, which are approximately 311.13 meters apart.

TABLE I: Flight Plan Summary for the Mission

Parameter	Mission
Distance to Wall (m)	30.0
Coverage Lines (count)	31
Row Distance (m)	14.84
GSD (cm/px)	1.2
Overlap (%)	60

C. Online Multi-resolution Motion Planning Results

For the multi-resolution motion planning experiments, we simulated a scenario where the UAV needs to return home due to a critical battery event, diverging from the pre-planned path (see Fig. 1) and flying a long distance autonomously. For this case, We present the path refinement for each octree layer and compare with the computational time for pathfinding using a voxel grid. Additionally, we performed a quantitative analysis by conducting 8 different missions using both the octree structure and the voxel grid with varying start and goal positions, highlighting the path length and the pathfinding time.

Fig. 7 presents the 9-layers octree of the tailings dam along with the start and goal points used to test the approach. These points are approximately 311.13 meters apart from

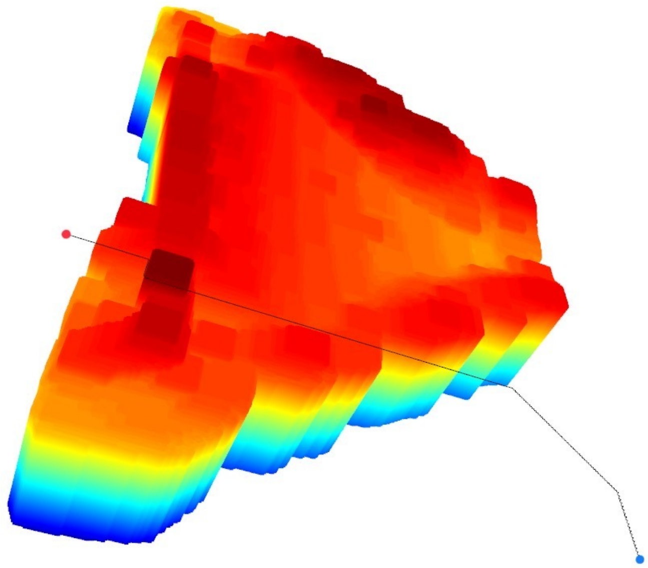


Fig. 9: Visualization of the path computed online using the A* algorithm over the voxel grid, with the same start (●) and goal (●) points as in the octree. The precomputed voxel grid is used online to find a path that avoids obstacles.

each other. Fig. 8 showcases the path found for each layer of the octree. As can be seen, even though we can find the optimal path for each layer, the first layers' paths are not good for practical application due to the lack of resolution. As the resolution increases in the final layers, the paths become more suitable for practical applications.

For the same start and goal points, we perform our motion planning algorithm using the voxel grid in order to compare the computational time of the approach. Here, the voxel grid is composed of voxels with a size of $1.0\,\text{m}$, same as the $9\,\text{th}$ layer (finest) of our octree. Fig. 9 presents the online path generated by the A* algorithm over the voxel grid. As clearly observed, when the UAV needs to return home, obstacles must be avoided, and the algorithm must provide a rapid response due to the online nature of the operation. It is worth mentioning that the path found using the voxel grid is the same as the one found at the finer layer (9^{th}) of the octree for the same starting and goal points. Furthermore, when using the 6-connected grid approach, the initial path shall be post-processed (lines 7-16 of Algorithm 1) to remove unnecessary waypoints before being sent to the drone, improving its route efficiency in real-world applications.

To compare the time spent by both planners, we ran every algorithm 50 times (to account for other processes running on the computer) and took the average. The computational time for finding a path in the voxel grid was $72\,\text{s}$, while the computational time for running the same algorithm in the octree was $46\,\text{s}$. This shows that the octre approach implemented in this work is 1.6 times faster than the voxel grid at the same resolution. This speedup suggests that hierarchical spatial partitioning significantly reduces the number of nodes that need to be processed during online motion planning. These results emphasize the advantage of using octree-based multi-resolution grids for efficient and fast pathfinding, particularly in large environments where long paths need to be calculated and computational efficiency is crucial.

Fig. 10: High-Resolution 3D map of the tailings dam generated through a photogrammetry-based autonomous mission. This 3D reconstruction was created while considering photogrammetric constraints, ensuring accurate and detailed mapping. The resulting model facilitates structural analysis, monitoring of potential deformations and hazards.

To perform a quantitative analysis, we ran 8 new missions with different start and goal positions. Table II presents the results for 8 different missions, presenting the length of the path and the time to find it using both an octree and a voxel grid. As can be seen, the octree outperformed the voxel grid in all cases, finding paths 1.41 to 2.62 times faster.

TABLE II: Online Motion Panning Results

Mission	Path Length (m)	Pathfinding Time (s)		Speedup
		Voxel	Octree	(Voxel/Octree)
1	134.00	42.28	16.10	2.62x
2	190.00	46.84	21.12	2.18x
3	210.00	47.35	20.04	2.36x
4	216.00	57.27	31.65	1.81x
5	227.00	68.96	46.04	1.49x
6	251.00	65.00	43.63	1.49x
7	337.00	77.58	53.44	1.45x
8	354.00	96.89	68.62	1.41x

Finally, Fig. 10 presents the high-resolution 3D map of the tailings dam considered in this study, generated through a photogrammetry-based autonomous mission. Such a map allows inspectors and engineers to better understand and inspect the area, making it possible to perform structural analysis, monitor potential deformations, assess environmental impacts, track vegetation growth, and identify any hazards that could lead to a further collapse of the structure. This process significantly accelerates the inspection process compared to traditional methods, where inspectors must physically traverse the entire structure to identify structural changes and other hazards. Moreover, it has proven to be a safer and more efficient solution, particularly for inspections that must occur under harsh weather conditions, such as extreme cold during winter. The UAV's ability to operate autonomously in such adverse conditions without exposing personnel to risk makes it a highly valuable approach, as we plan to demonstrate in a future study.

V. CONCLUSIONS AND FUTURE WORK

This paper presented motion planning approaches for autonomous UAV inspection of large tailings dams. Our planners rely on the sparse point cloud of the region of interest to create safe paths that will generate higher-resolution 3D maps. Our first planner incorporates photogrammetry constraints by checking for collisions a standard back-and-forth coverage paths against a voxel grid obtained from the space point cloud of the dam. Once collisions are detected, a local planner based on A* locally modifies the path to avoid obstacles. Additionally, an online multi-resolution motion planning method that leverages an octree structure was proposed for situations where the UAV must adjust its pre-planned route. A theoretical and practical comparative analysis of computational time between voxel grids and octrees was also provided, confirming that octree-based multi-resolution grids significantly enhance the speed of online pathfinding. These findings highlight the effectiveness of hierarchical structures in enabling reactive, scalable motion planning, providing a robust solution for large-scale UAV inspections.

Future work involves experimenting with the A* algorithm in a 26-connected grid and comparing the results with the post-processed path derived from a 6-connected voxel grid. While the 26-connected grid may initially yield higher-quality and shorter paths, it could also slow down pathfinding, further justifying the choice of the 6-connected grid in this study. Moreover, we plan to create missions that encompass all critical structures of a tailings dam requiring inspection, such as the downstream slope, main and emergency spillways, and the crest and pool. Additionally, we are developing a real-time localization system to track the UAV using a previously acquired sparse point cloud of the structure. This approach aims to mitigate errors associated with the Inertial Measurement Unit (IMU) and GPS measurements, to guarantee that the mission, whether planned offline or online, is executed with precision.

REFERENCES

[1] Z. Li, Y. Zhang, H. Wu, S. Suzuki, A. Namiki, and W. Wang, "Design and application of a uav autonomous inspection system for high-voltage power transmission lines," *Remote Sensing*, vol. 15, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/3/865

[2] C. Xu, Q. Li, Q. Zhou, S. Zhang, D. Yu, and Y. Ma, "Power line-guided automatic electric transmission line inspection system," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–18, 2022.

[3] W. Cornwall, "A dam big problem," *Science*, vol. 369, no. 6506, pp. 906–909, 2020. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.369.6506.906

[4] C. Toumieh and A. Lambert, "Shape-aware safe corridors generation using voxel grids," *ArXiv*, 2022.

[5] S. S. Harithas, R. D. Yadav, D. Singh, A. K. Singh, and K. M. Krishna, "Cco-voxel: Chance constrained optimization over uncertain voxel-grid representation for safe trajectory planning," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 11 087–11 093, 2021.

[6] L. Wenzheng, L. Junjun, and Y. Shunli, "An improved dijkstra's algorithm for shortest path planning on 2d grid maps," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2019, pp. 438–441.

[7] N. O. Eraghi, F. López-Colino, A. de Castro, and J. Garrido, "Path length comparison in grid maps of planning algorithms: Hctnav, a* and dijkstra," in *Design of Circuits and Integrated Systems*, 2014, pp. 1–6.

[8] Q. Massonnat and C. Verbrugge, "Efficient octree-based 3d pathfinding," in *2024 IEEE Conference on Games (CoG)*, 2024, pp. 1–8.

[9] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, 1982.

[10] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, p. 189–206, Apr. 2013. [Online]. Available: https://doi.org/10.1007/s10514-012-9321-0

[11] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap : A probabilistic , flexible , and compact 3 d map representation for robotic systems," 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:15581881

[12] M. shun Yuan, T. le Zhou, and M. Chen, "Improved lazy theta∗ algorithm based on octree map for path planning of uav," *Defence Technology*, vol. 23, pp. 8–18, 2023.

[13] S. Ivić, B. Crnković, L. Grbčić, and L. Matleković, "Multi-uav trajectory planning for 3d visual inspection of complex structures," *Automation in Construction*, vol. 147, p. 104709, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580522005799

[14] Z. Li, Q. Wang, T. Zhang, C. Ju, S. Suzuki, and A. Namiki, "Uav high-voltage power transmission line autonomous correction inspection system based on object detection," *IEEE Sensors Journal*, vol. 23, no. 9, pp. 10 215–10 230, 2023.

[15] S. Zhao, F. Kang, J. Li, and C. Ma, "Structural health monitoring and inspection of dams based on uav photogrammetry with image 3d reconstruction," *Automation in Construction*, vol. 130, p. 103832, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580521002831

[16] C. G. Arnaldo, F. P. Moreno, M. Z. Suárez, and R. D.-A. Jurado, "Autonomous route planning for uav-based 3d reconstruction," *Engineering Proceedings*, vol. 90, no. 1, 2025. [Online]. Available: https://www.mdpi.com/2673-4591/90/1/78

[17] S. Zhang, C. Liu, and N. Haala, "Guided by model quality: Uav path planning for complete and precise 3d reconstruction of complex buildings," *International Journal of Applied Earth Observation and Geoinformation*, vol. 127, p. 103667, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843224000219

[18] Q. Li, H. Huang, W. Yu, and S. Jiang, "Optimized views photogrammetry: Precision analysis and a large-scale case study in qingdao," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1144–1159, 2023.

[19] P. V. G. Simplicio and G. A. S. Pereira, "Mission planning for photogrammetry-based autonomous 3d mapping of dams using a commercial uav," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 464–471.

[20] B. M. Rocamora, P. V. G. Simplicio, and G. A. S. Pereira, "A behavior tree approach for battery-aware inspection of large structures using drones," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 234–240.

[21] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, Jun. 1981. [Online]. Available: https://doi.org/10.1145/358669.358692