COOPERATIVE AERIAL ROBOTS INSPECTION IMPLEMENTATION IN PYTHON

Zinonas Kortas

UNIVERSITY OF CYPRUS

Department of Computer Science

May 2025

UNIVERSITY OF CYPRUS
Department of Computer Science

COOPERATIVE AERIAL ROBOTS INSPECTION IMPLEMENTATION IN PYTHON

by

Zinonas Kortas

Supervisor

Prof. Panayiotis Kolios

Co-Supervisor

Andreas Anastasiou

The Thesis was submitted in partial fulfillment of the requirements for obtaining the Computer Science degree of the Department of Computer Science of the University of Cyprus

May 2025

Acknowledgements

I would like to express my sincere gratitude to Professor Panayiotis Kolios for giving me the opportunity to be part of his team. Working under his supervision has been a rewarding and formative experience that has significantly contributed to both my academic and personal development.

I am also deeply thankful to Research Assistant Andreas Anastasiou for his continuous support and valuable advice throughout the entire academic year. His willingness to assist whenever I needed help and his consistent guidance were truly invaluable to the completion of this work.

Abstract

Rapid advancement in the field of Unmanned Aerial Vehicles (UAVs) has significantly expanded their use in critical real-world applications, including infrastructure inspection, search and rescue operations, and environment monitoring. Their ability to navigate complex environments and operate autonomously has made them ideal candidates for inspecting large-scale or hazardous 3D structures. However, autonomous inspection in unknown and cluttered spaces remains a challenging problem, particularly when coordination among multiple heterogeneous UAVs is required.

This work introduces a cooperative inspection system that efficiently controls and coordinates a swarm of distributed UAV agents to perform autonomous 3D infrastructure inspection in unknown environments. The proposed system employs a two-stage methodology. In the first stage, UAVs collaboratively map the environment using their complementary sensing capabilities. In the second stage, optimized collision-free inspection paths are generated for each agent to ensure full surface coverage of the target structure.

The system was developed using the Robot Operating System (ROS) and was implemented entirely in Python. Integrates autonomous exploration, real-time mapping, and trajectory generation modules that allow each UAV to operate without human intervention. The system was tested in realistic simulation environments created with the Gazebo simulator. The evaluation was performed both qualitatively and quantitatively, demonstrating the effectiveness of the system in terms of inspection completeness, safety, and swarm coordination. The general goal of this work is to increase the autonomy and efficiency of UAV swarms in complex inspection scenarios.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CARIC Cooperative Aerial Robot Inspection Challenge

CDC Conference on Decision and Control

GCS Ground Control Station

LOS Line-of-Sight

MAV Micro Aerial Vehicle

MIQP Mixed Integer Quadratic Programming

NBV Next-Best-View

ROS Robot Operating System

SAR Search and Rescue

SLAM Simultaneous Localization and Mapping

TSP Traveling Salesman Problem

UAS Unmanned Aerial System

UAV Unmanned Aerial Vehicle

WiSAR Wilderness Search and Rescue

1 Chapter 1

Introduction

- Purpose of this work
- Problem structure
- Thesis structure

1.1 Purpose of this work

In recent years, the need for fast, reliable and safe environment inspection and monitoring has become more urgent than ever. Whether it is after a natural disaster, such as an earthquake or a wildfire, or during routine infrastructure maintenance, there are many situations where accessing a damaged or complex area is either extremely difficult or dangerous for human operators. Traditional inspection methods often require physical access through scaffolding, cranes, or manned vehicles, which is time consuming, costly, and poses serious safety risks.

This problem becomes even more critical when dealing with large-scale or complex 3D environments like collapsed buildings, bridges, power plants, or industrial zones, where visibility is limited, the environment may still be unstable, and fast response is essential. In such conditions, deploying human teams to perform manual inspection or search and rescue is not only inefficient but can also put lives at further risk.

To address these challenges, UAVs have emerged as an essential technological solution. Equipped with sensors, cameras, and wireless communication, UAVs can quickly navigate and explore hazardous or hard to reach areas from above. They are lightweight, autonomous, and can provide real-time data.

This work focuses on building a cooperative system that uses multiple UAVs working together to inspect complex 3D environments. The proposed system is developed in Python using the ROS framework, enabling each drone to autonomously explore, map the environment, and carry out efficient inspection tasks while sharing information with the rest of the swarm. Given the importance of safety and the need to avoid inter-agent collisions, it is essential to build an accurate and complete map of the environment. Proper environment mapping allows each UAV to plan safe and efficient flight paths, ensuring thorough area coverage while minimizing the risk of accidents

1.2 Problem structure

This work is inspired by the Cooperative Aerial Robot Inspection Challenge (CARIC), which took place during the IEEE Conference on Decision and Control (CDC) in 2023. The goal of the challenge was to provide a simulation-based benchmark for evaluating multi-UAV infrastructure inspection methodologies under realistic conditions.

Despite the numerous advantages that UAV swarms offer for infrastructure inspection, several challenges remain unresolved. These include the difficulty of navigating complex and partially unknown indoor and outdoor environments filled with static or dynamic obstacles, as well as the need to inspect structures of high geometric complexity. Furthermore, ensuring real-time coordination and efficient utilization of multiple UAVs especially when they are equipped with heterogeneous sensing capabilities, adds another layer of complexity to the problem.

However, during testing and evaluation using predefined simulation scenarios such as the MBS, hangar, and crane environments, frequent UAV collisions were observed at various time intervals. These collisions occurred both during the first stage, where UAVs collaboratively map the environment using their complementary sensing capabilities, and during the second stage, in which optimized, collision-free inspection paths are generated for each agent to ensure full surface coverage of the target structure.

As a result, the swarm was often unable to fully explore the environment or achieve complete inspection coverage, leading to reduced overall scores and several missed points of interest.

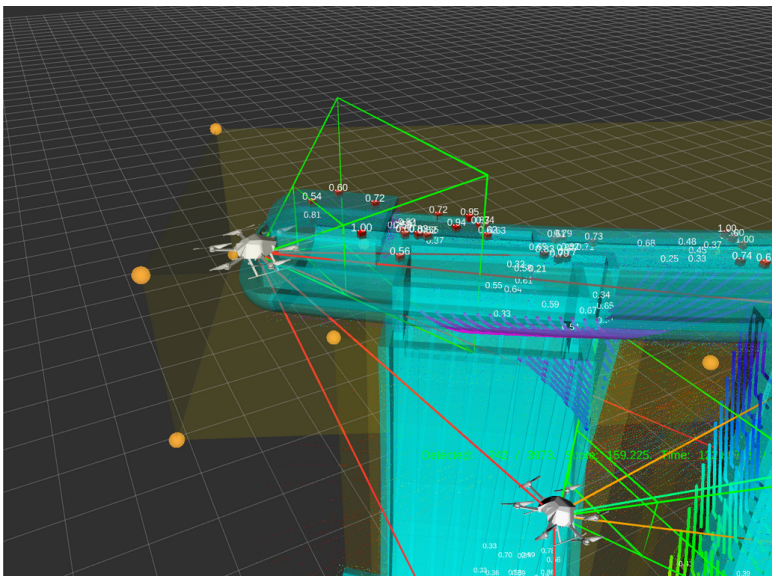

Figure 1.1: Example of UAV collisions during the inspection process.

1.3 Thesis structure

The rest of our work is structured as follows. Chapter 2 is a literature review that summarizes other previous works on UAVs and similar problems that has been studied. Chapter 3 presents the various problems identified during the execution and testing of the initial implementation, and Chapter 4 is a detailed review of the changes that were made to eliminate the problems. Furthermore, Chapter 5 analyzes the results and compares system performance before and after the applied changes. Finally, Chapter 6 concludes this work.

2 Chapter 2

Literature Review

- Related Work
- How this work differs from related work

2.1 Related Work

- [1] **CARIC Challenge – Cooperative Aerial Robot Inspection.**

  The CARIC Challenge, organized during the IEEE Conference on Decision and Control (CDC) in 2023, provides a standardized, simulation-based benchmark for evaluating cooperative aerial robot inspection strategies in complex 3D environments. The challenge focuses on assessing the ability of multi-UAV systems to perform accurate environmental mapping and structure inspection under realistic constraints, including limited sensor ranges, partial observability, and the need for autonomous coordination between heterogeneous UAVs.

  Participants are evaluated over two mission stages: Environmental Mapping and Cooperative Inspection. During the first stage, UAVs must build a global occupancy map using onboard sensors (i.e., LiDAR, RGB-D), while in the second stage, they must inspect specific regions of interest using vision-based coverage. The challenge emphasizes real-time decision-making, effective task allocation, collision avoidance, and full 3D coverage.

- [2] **Towards Automated 3D Search Planning for Emergency Response Missions.**

  This paper presents a unified framework for automating collision-free UAV search trajectories in fully 3D environments. The approach formulates the problem as a Mixed Integer Quadratic Programming (MIQP) model that incorporates UAV dynamics, sensor limitations (e.g. field of view, probabilistic detection), obstacle avoidance, energy constraints, and mission-specific objectives such as time and cost minimization.

  One of the key contributions is the integration of a probabilistic sensing model that ensures a guaranteed level of detection for potential targets. The planner takes into account both static and dynamic constraints and produces globally optimal trajectories with respect to energy and coverage objectives.

The authors validate their approach in Gazebo-based simulation environments, closely replicating post-disaster scenarios with partially known or cluttered terrain. The evaluation demonstrates that the proposed planner can significantly improve mission efficiency by reducing the total execution time while ensuring sufficient area coverage and detection success rates. The results highlight the potential of using optimization-based techniques for real-time, risk-aware UAV deployment in emergency response applications.

- [3] **Uav autonomous indoor exploration and mapping for sar missions: Reflections from the icuas 2022 competition**

  This paper presents a comprehensive perception-aware autonomous exploration framework tailored for indoor Search and Rescue (SAR) operations using UAVs. The proposed system is designed to operate in GPS-denied environments, where rapid and safe exploration is vital to locate victims and deliver emergency payloads. The UAV operates without prior knowledge of the environment, utilizing onboard perception to dynamically build a map, avoid collisions, and identify targets using fiducial markers. A key feature of this framework is the integration of real-time exploration and mapping using a 2D SLAM-based approach, which enables the UAV to navigate through unknown, cluttered environments while optimizing its search trajectory.

  One of the most practical contributions of the paper is the incorporation of a payload delivery mechanism, which simulates the drop of medical supplies upon detecting a target. The UAV performs a full pipeline of tasks: autonomous takeoff, map building, exploration, target detection, and safe payload deployment. This closed-loop system is tightly coupled with obstacle avoidance logic that reacts to both static and dynamic hazards, ensuring reliable operation even in unpredictable environments. The architecture also considers hardware limitations and real-world constraints, aiming to deploy the solution on compact and lightweight aerial robots suitable for indoor use.

  The results highlight strong performance in exploration coverage, target localization accuracy, and mission execution under time and resource constraints. The paper also offers a detailed discussion of the lessons learned from the competition, identifying practical limitations of onboard processing, challenges with fiducial detection in poor lighting conditions, and potential areas for improving robustness and reliability.

- [4] **UAV Intelligent Path Planning for Wilderness Search and Rescue**:

  In this paper, the authors propose a path planning framework for UAVs in the context of wilderness search and rescue (WiSAR) missions. The approach is motivated

by the need to maximize the cumulative probability of finding a missing person, based on a predefined probability distribution over the search area. The problem is formalized as a variant of the Orienteering Problem, which balances the trade-off between maximizing information collection and respecting time constraints. To address this, several algorithms are proposed.

A key contribution is the integration of ”search effectiveness” into the cost model. Instead of simply using Euclidean distance or travel time, the cost function incorporates a penalty for proximity to previously visited points, encouraging the UAV to prioritize unexplored or less-covered areas. This modification helps to avoid redundancy and promotes information gain. The Evolutionary Algorithms include crossover and mutation operators specifically designed for path evolution, ensuring feasible and efficient path generation under time constraints.

- [5] **Detecting, Localizing, and Tracking an Unknown Number of Moving Targets Using a Team of Mobile Robots.**

  This paper addresses the challenge of tracking an unknown number of moving targets using a team of mobile robots operating in uncertain environments. The authors propose a decentralized algorithm that enables each robot to detect, localize, and track targets autonomously, while sharing information with teammates to improve the overall effectiveness of the group. This approach enhances scalability and robustness compared to traditional centralized methods.

  The system combines particle filtering with cooperative decision-making and a hybrid communication model, allowing robots to adapt their actions based on target dynamics and environmental changes. The proposed method is validated through both simulations and real-world experiments, demonstrating strong tracking performance, even in cluttered scenarios and high target mobility. This work is highly relevant to search and rescue, surveillance, and other real-time multi-robot applications.

- [6] **Re-routing UAVs in the Wild: Preemptive Path Planning for Efficient Wilderness Search**:

  This paper builds upon previous work in UAV-based WiSAR, introducing an online rerouting mechanism that enhances the adaptability of UAV path planning to dynamic mission updates. The system enables the UAV to preemptively interrupt its current plan in favor of a more promising one, based on evolving knowledge about the environment or updated probability maps. By leveraging an efficient rerouting strategy that evaluates potential gains from switching paths, the authors aim to improve mission success while minimizing redundant coverage.

The planner relies on a modified version of the Orienteering Problem formulation, where the UAV must maximize information gain under time constraints. A key feature is the use of penalty-based cost functions for the evaluation of the path. Specifically, paths that revisit areas already covered by the UAV or that are deemed less probable to contain the target are penalized, allowing the planner to favor more informative routes. The rerouting mechanism triggers when the expected gain of a new candidate path, after accounting for switching costs, exceeds a predefined threshold, ensuring that the UAV does not overreact to small changes, but remains responsive to significant updates.

Experimental simulations show that rerouting improves overall search efficiency, especially in environments with nonuniform or changing probability distribution.

- [7] **Rapid exploration with multi-rotors: A frontier selection method for high speed flight**:

  This paper introduces a novel frontier-based exploration strategy tailored for high-speed navigation with quadrotors. Traditional frontier-based methods tend to minimize the path length by directing the robot to the nearest unexplored frontier, often resulting in stop-and-go movement that limits velocity. In contrast, this paper introduces a reactive frontier selection approach that prioritizes frontiers within the current field of view and aligned with the robot's direction of motion. This minimizes abrupt changes in velocity and allows continuous high-speed exploration. Although the traveled path may be longer, the overall exploration time is significantly reduced.

  The paper evaluates this method through both simulation and real-world experiments, comparing it against classical frontier-based exploration and NBV planning. The results show that the proposed method consistently achieves lower exploration times, particularly in 3D environments or scenarios requiring frequent directional changes. The authors also derive an optimal flight velocity that maximizes energy efficiency for multirotors and integrate this model into their planning strategy. Overall, the work demonstrates how local, reactive planning aligned with visibility constraints can yield global performance benefits for fast and efficient exploration.

- [8] **The determination of next best views**:

  In this foundational work, Connolly introduces the concept of determining the NBV for constructing complete 3D models from range images. The paper presents two algorithms that utilize partial octree models to compute viewpoints that maximize visibility of unseen regions. The first algorithm, termed the ”Planetarium Algorithm,” samples a spherical shell around the object and identifies the viewpoint that

exposes the largest unseen volume. The second, called the ”Normal Algorithm,” uses exposed faces between unseen and empty nodes to rapidly estimate optimal view directions.

The key objective of this work was to automate the selection of camera views in 3D reconstruction tasks by proposing heuristics for maximal information gain. Connolly’s approach laid the groundwork for later developments in active vision and robotic exploration, where determining efficient sensor placements is critical. Although the algorithms assume idealized conditions (e.g., free camera placement, sensor always pointing at the origin), they highlight the fundamental trade-off between computational cost and completeness in model acquisition.

- [9] **Occlusions as a Guide for Planning the Next View**:

  This paper presents the problem of acquiring complete 3D data from unknown scenes using a laser range sensor. Their strategy focuses on intelligently planning sensor movements by analyzing occlusions detected in range images. The core idea is that occluded regions, which appear as gaps in the acquired data, provide valuable guidance for selecting the next best viewpoint. The authors divide the data acquisition task into two stages based on two types of occlusions: one caused by lack of visibility to the camera, and the other by lack of illumination.

  The main goal of the work is to exploit occlusion information, along with knowledge of the sensor’s geometry, to compute new scanning directions that maximize data completeness while minimizing unnecessary scans. The proposed algorithms are based on a two-phase process: first rotating the sensor within the scanning plane to resolve camera occlusions, and then planning perpendicular scans to capture parts of the scene hidden due to illumination constraints. The research contributes significantly to the field of active vision by demonstrating how purposive sensing strategies can lead to efficient 3D scene reconstruction without needing full a priori models.

- [10] **Navigation Strategies for Exploring Indoor Environments**:

  In this paper, the authors investigate strategies for building accurate and complete maps of unknown indoor environments using a mobile robot with imperfect sensing and control. While prior approaches such as SLAM focus on integrating sensor data into a coherent map, this work addresses the often-overlooked challenge of deciding where the robot should move next to acquire the most useful information. The paper introduces the concept of a ”safe region,” which defines the largest area guaranteed to be free of obstacles based on current sensor readings. This notion forms the basis of a next-best-view (NBV) algorithm that iteratively guides the robot to explore new

areas while maintaining alignment with the existing map and ensuring collision-free navigation.

The key contribution is a unified framework that combines map-building, viewpoint planning, and safe motion. The NBV algorithm selects candidate positions within the safe region based on expected information gain, required overlap with the current map for alignment, and motion cost. Through both simulation and real-world experiments, the authors demonstrate that their method enables efficient, automatic construction of polygonal maps, reducing the number of sensing operations while maintaining mapping accuracy. This work significantly bridges the gap between SLAM and active sensing by offering a principled way to choose sensor placements in an exploratory context.

- [11] **Planning Exploration Strategies for Simultaneous Localization and Mapping**:

  This paper presents a planning framework for guiding mobile robots during the construction of environmental maps, particularly in the context of SLAM. Unlike traditional SLAM approaches which mostly emphasize estimation and sensor fusion, the focus here is on *where* the robot should go next to maximize mapping efficiency. To this end, the authors introduce a novel utility function that balances information gain, localization accuracy, motion cost, and visibility of features such as corners and landmarks.

  The utility function considers a variety of factors including proximity to unexplored frontiers (free edges), probability of landmark recognition, expected localization uncertainty based on trajectory shape, and distance from obstacles. A multiplicative formulation ensures that configurations with poor scores on any one dimension are automatically penalized, thereby guiding the robot to paths that are both informative and robust. The paper further proposes a randomized sampling algorithm that enables multi-step lookahead planning without incurring excessive computational cost.

  The framework is implemented in both single- and multi-robot scenarios. Experiments with real robots and simulations demonstrate the efficacy of the approach in producing reliable, feature-rich maps. The strategy supports dynamic replanning, backtracking, and landmark-based registration using partial Hausdorff distance. Overall, the work bridges motion planning and active perception in SLAM, offering a flexible system that adapts to robot sensing capabilities and uncertainty constraints.

- [12] **A Frontier-Based Approach for Autonomous Exploration**:

  This paper introduces the concept of *frontier-based exploration*, where a mobile robot incrementally explores unknown environments by navigating to the boundaries (frontiers) between known free space and unexplored regions. These frontiers are detected within occupancy (evidence) grids, allowing the robot to always target the most promising regions for acquiring new information. The method effectively balances exploration and safety, as it guides the robot through accessible routes without relying on prior maps.

  A key innovation is the integration of laser-limited sonar, a sensor fusion technique that uses laser range readings to correct errors from sonar reflections, significantly improving the accuracy of the evidence grids. Frontier detection is achieved by classifying each cell based on its occupancy probability and identifying open cells adjacent to unknown ones. The robot then autonomously selects and navigates to the nearest accessible frontier, updating its map upon arrival and repeating the process.

  The method was tested on a real Nomad 200 robot in complex office environments filled with irregular obstacles, such as furniture and narrow passages. The results demonstrate that the approach can handle both cluttered and open spaces efficiently, outperforming previous reactive methods limited by rigid environmental assumptions (e.g., right-angled walls).

- [13] **Frontier-Based Exploration Using Multiple Robots**:

  This paper extends the concept of frontier-based exploration to multi-robot systems. Each robot builds its own global evidence grid, detects frontiers at the boundaries between known and unexplored areas, and navigates independently to these frontiers. Although robots maintain separate decision-making processes, they share local perceptual data, allowing them to avoid redundant exploration and achieve better overall coverage. This decentralized and cooperative approach enhances robustness: if a robot fails, the others can continue exploring without interruption. The strategy relies on laser-limited sonar for accurate environment mapping, minimizing errors caused by specular sonar reflections.

  Through experiments in real office environments using Nomad 200 robots, the system demonstrated effective and efficient multi-robot exploration. The robots explored independently but merged local information to improve each other's maps, using a log-odds fusion technique for efficient evidence grid integration. The study highlights that even with minimal coordination, multiple robots can cooperatively explore complex spaces with arbitrary wall orientations, adapting dynamically to discovered frontiers and robot failures. This work laid groundwork for decentral-

ized robotic exploration with shared environmental understanding.

- [14] **Coordinated Multi-Robot Exploration**:

  This paper presents a decision-theoretic approach for coordinating a team of mobile robots to efficiently explore unknown environments. The key idea is to assign frontier cells (i.e., the borders between known and unknown space) to robots by maximizing the expected utility and minimizing travel cost. Utility is reduced for frontiers already assigned to other robots or likely to be covered by their sensors, thus promoting spatial distribution of exploration tasks. The authors also consider real-world limitations such as sensor noise and communication range, and incorporate path planning based on occupancy grid maps using a smoothed value iteration algorithm.

  The system supports both unlimited and limited communication scenarios. In limited-range settings, robots locally coordinate in sub-teams and retain memory of others' last known targets to minimize redundant exploration. The approach was validated through extensive simulations and real-world experiments, demonstrating superior performance compared to uncoordinated or greedy strategies. The authors show that even partial coordination enabled by intermittent communication is sufficient to achieve exploration efficiency near that of fully connected systems.

- [15] **Evaluating the Efficiency of Frontier-Based Exploration Strategies**:

  This paper provides a thorough analysis and enhancement of frontier-based exploration strategies, which are widely used in mobile robotics for autonomous mapping of unknown environments. The authors begin by reviewing the classical closest-frontier approach, where the robot always navigates toward the nearest boundary between explored and unexplored space. While this method is simple and efficient, it has known drawbacks such as redundant movement and inefficient revisiting of already explored areas, particularly in indoor, room-structured environments.

  To address these issues, the authors introduce two lightweight but impactful extensions. The first is *repetitive re-checking*, a mechanism that continuously monitors whether the robot's current frontier target is still valid as the map updates. If the targeted frontier becomes obsolete before arrival, due to new sensor information, the robot re-evaluates its goal in real-time. This technique minimizes wasted motion toward irrelevant locations. The second improvement is *map segmentation*, which aims to restrict exploration within the currently occupied room or region before moving to others. This is achieved through Voronoi-based segmentation of the occupancy grid, enabling more coherent and efficient room-wise exploration.

  The authors conduct an extensive evaluation of these strategies in multiple simulated

environments, including open spaces and office-like indoor layouts. Metrics such as total path length are used to assess exploration efficiency. The study shows that the enhanced frontier-based strategies are applied more complex decision-theoretic strategies and significantly outperform random frontier selection. In structured settings, segmentation prevents unnecessary revisits, while re-checking improves reactivity without adding computational burden. In less structured, open environments, repetitive re-checking still offers benefits, while segmentation has a more neutral or even slightly negative effect.

- [16] **A Comparison of Path Planning Strategies for Autonomous Exploration and Mapping of Unknown Environments**:

  This paper offers a comprehensive review and empirical comparison of several widely used exploration strategies in autonomous robotics, focusing on how these strategies balance exploration efficiency and mapping quality. The authors first outline the theoretical background of path planning in unknown environments, including non-coordinated and coordinated methods, as well as those that incorporate SLAM into the planning process. They highlight that while many strategies exist, few have been systematically compared under a unified framework, which motivates the study.

  The authors implement and evaluate seven distinct strategies, ranging from the classic nearest frontier and cost-utility methods to market-based and hybrid coordinated approaches, using a common simulation platform. Metrics such as total exploration time, landmark-based SLAM map accuracy, and occupancy grid consistency are used to assess performance across different environments and team sizes. The results show that techniques integrating SLAM uncertainty, such as the hybrid integrated coordinated strategy, tend to yield higher map quality, although they often incur longer exploration times. In contrast, cost-focused approaches are more time-efficient but may sacrifice mapping accuracy.

  Importantly, the paper emphasizes that no single strategy is optimal in all scenarios; instead, the best choice depends on application-specific requirements such as the need for rapid deployment, high-fidelity mapping, or robustness in multi-robot coordination. The study also notes the importance of scalability, revealing that centralized coordination becomes computationally expensive as team sizes grow, whereas distributed models like market-based strategies offer a balance between coordination and efficiency. This comprehensive evaluation provides valuable insight into the design of exploration systems for real-world robotic deployments.

- [17] **A Hybrid Solution to the Multi-Robot Integrated Exploration Problem**:

This paper presents a fully hybrid reactive/deliberative architecture for solving the integrated exploration problem in multi-robot systems. Unlike prior methods that give precedence to either reactive or deliberative planning, this approach assigns equal importance to both layers. The architecture is built upon the concepts of the expected safe zone and gateway cells. The reactive layer allows robots to safely explore locally visible regions, while the deliberative planner builds an exploration decision tree to decide whether the robot should remain within its current safe zone, navigate to a gateway to explore a new zone, or revisit known areas to reduce localization uncertainty. This ensures robust and efficient behavior in environments with potential navigation traps, like local minima.

The model is particularly notable for introducing a localized, map-limited reactive process that avoids local minima by ignoring unreachable areas behind obstacles. It also supports decentralized execution, which increases robustness and scalability. The deliberative layer, on the other hand, makes strategic decisions using an exploration tree structure that balances information gain, localization accuracy, and multi-robot coordination. Robots share information through a centralized SLAM system but rely primarily on local maps for reactive behavior. This division reduces computational load while preserving the benefits of global planning.

Experimental results in simulation with up to eight robots across multiple environments demonstrate that the hybrid approach improves exploration efficiency and localization quality over traditional methods, especially in cluttered or structurally complex environments. Compared to classic frontier-based methods, the proposed model yields better performance in both exploration time and mapping error. Moreover, the architecture's flexibility and robustness make it a strong candidate for real-time, scalable robotic deployments in indoor scenarios.

- [18] **Vision-Based Autonomous Mapping and Exploration Using a Quadrotor MAV**:

  This paper presents a fully autonomous quadrotor system capable of mapping and exploring unknown environments using only vision-based sensors. The platform relies on a front-facing stereo camera for visual odometry and 3D perception, combined with a downward-looking camera for optical flow-based motion estimation. A key contribution is the integration of all critical components pose estimation, local navigation using the VFH+ algorithm, frontier-based exploration, and even wall-following behavior via the Bug algorithm, making the system suitable for GPS-denied environments. For large-scale mapping, image data is streamed to a ground station where loop-closure detection and pose-graph SLAM are performed off-board.

The authors demonstrate that the system can navigate autonomously through structured indoor corridors and generate accurate 3D occupancy maps. The fusion of visual odometry with optical flow improves pose robustness, while the ability to switch between frontier-based and wall-following strategies ensures adaptability to both dense and sparse environments. Experimental results show effective exploration and large-scale SLAM, with re-localization capabilities using a visual vocabulary tree for loop detection. The work highlights the potential of lightweight camera-based solutions for fully autonomous MAV operation without reliance on heavy or infrastructure-dependent sensors like laser scanners or motion capture systems.

- [19] **OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees**:

  This paper introduces OctoMap, an open-source 3D mapping framework designed to create compact, probabilistic volumetric representations of the environment using octrees. OctoMap explicitly models free, occupied, and unknown space, which is essential for autonomous navigation, exploration, and manipulation tasks. By relying on a probabilistic occupancy estimation and efficient memory usage, the framework allows robots to maintain consistent and updateable maps while coping with noisy sensor data. Key features include multi-resolution queries, lossy and lossless map compression, and support for dynamic environments.

  A core innovation of OctoMap lies in its use of bounded confidence (clamping) in occupancy probabilities, which enables compression through pruning of homogeneous map regions without significantly sacrificing accuracy. The framework can incorporate data from various types of range sensors, and supports incremental map updates using raycasting. Its probabilistic formulation ensures consistent integration of multiple uncertain measurements, and the log-odds representation simplifies updates while allowing for fusion from different sources and even multiple robots.

  The authors validate OctoMap across a diverse set of real-world datasets, demonstrating scalability, accuracy, and efficiency. They also show how OctoMap outperforms traditional grid-based and point cloud methods in memory usage while retaining high fidelity and navigational relevance. Integration into ROS and widespread adoption across robotics tasks like 3D SLAM, humanoid navigation, and mobile manipulation highlight the versatility of the framework. Overall, OctoMap represents a significant contribution to the field of 3D mapping in mobile robotics.

- [20] **Fast Frontier-based Information-driven Autonomous Exploration with an MAV**:

This paper presents a hybrid exploration strategy designed for fast, onboard computation by Micro Aerial Vehicles (MAVs). The approach combines frontier-based and sampling-based planning by selecting candidate next-views directly from frontier voxels and evaluating them using a utility function based on map entropy and estimated travel time. Unlike conventional frontier methods, this strategy avoids computationally expensive voxel clustering by leveraging the octree map structure, treating voxel blocks as implicit clusters. Information gain is approximated efficiently via sparse raycasting, removing the need for full map updates or heavy computation.

The system runs entirely onboard the MAV and uses the ‘supereight‘ octree-based volumetric mapping framework to build and maintain the occupancy map in real-time. Candidate viewpoints are sampled uniformly from updated frontier voxel blocks and evaluated based on their expected entropy contribution. Optimized yaw angles for each candidate are computed using 360° sparse raycasting to maximize visible uncertainty, resulting in a utility-driven path planner that selects informative viewpoints while minimizing time. The MAV follows full computed paths (rather than just initial segments), which avoids oscillatory behavior common in receding-horizon strategies.

Simulation results across multiple 3D environments, including an apartment, a maze, and a powerplant. Proposed method significantly reduces both computation time and exploration duration compared to the widely used NBVP planner. A real-world experiment with a DJI hexacopter further confirms feasibility, showing onboard real-time performance using only RGB-D sensing. This work offers an efficient and scalable solution to autonomous 3D exploration for MAVs operating in real-time and computationally constrained conditions.

- [21] **Adaptive-Resolution Octree-Based Volumetric SLAM**:

  This paper proposes a dense SLAM pipeline capable of integrating depth data into a volumetric map at adaptive resolutions, addressing issues such as aliasing, memory inefficiency, and detail loss inherent in single-resolution systems. The system dynamically selects the appropriate integration scale depending on the sensor’s distance from the scene, and fuses information using a hierarchical octree-based representation. Built upon the ‘supereight‘ framework, it supports real-time performance on CPUs and enables scale-consistent up- and down-propagation of voxel information. A key innovation is the lazy propagation mechanism, which ensures consistent map updates across scales only when needed for tracking or rendering, significantly reducing computational overhead.

  The authors demonstrate that their approach improves both runtime and reconstruc-

tion quality, especially in cluttered environments. Compared to voxel hashing and fixed-resolution octrees, the adaptive-resolution method achieves up to 6× speed-ups and superior reconstruction accuracy on standard SLAM benchmarks. Fine scene details are better preserved at close range, while faraway surfaces are mapped more efficiently, enabling scalable dense SLAM suitable for mobile robotics, AR/VR, and aerial exploration tasks.

- [22] **An Improved Frontier-Based Approach for Autonomous Exploration**:

  This paper presents a refined frontier-based exploration method that improves efficiency in cluttered environments, such as office spaces with narrow corridors and cubicles. The authors enhance the classic utility function used for selecting frontiers by integrating a new rotation cost component that considers the robot's current heading. This encourages exploration in the direction the robot is already facing, minimizing unnecessary turning and improving coverage speed. The method combines robot odometry with a topological graph of previously visited frontiers, allowing the robot to both prioritize reachable unexplored areas and backtrack systematically when needed.

  Real-world experiments conducted in a complex office environment demonstrate the superior performance of the proposed method over conventional frontier selection. The robot autonomously maps the environment using SLAM and ROS navigation tools, while dynamically updating a topological graph to track visited and unvisited nodes. Compared to the baseline, the improved utility function leads to faster completion times, avoids local minima, and ensures more consistent coverage. The study highlights the importance of heading-awareness and memory-based backtracking in enhancing the autonomy and efficiency of frontier-based exploration systems.

- [23] **Applying Frontier Cells Based Exploration and Lazy Theta* Path Planning over Single Grid-Based World Representation for Autonomous Inspection of Large 3D Structures with an UAS**:

  This paper proposes a lightweight, unified framework for autonomous 3D inspection using unmanned aerial systems (UAS), specifically designed for large-scale and GPS-denied environments like offshore platforms. The authors combine frontier-based exploration with the Lazy Theta* path planning algorithm, both implemented over a single sparse octree structure (OctoMap). This design choice allows memory-efficient representation and planning within large volumes, without falling back on local 2D grids or requiring downsampling of 3D data. The system emphasizes modularity and test-driven development, enabling repeatable experiments and facilitating debugging through unit test automation.

The exploration algorithm identifies frontier cells (free cells adjacent to unknown space) directly on the octree structure, and benefits from the ability to skip over unknown nodes, significantly reducing computational cost. For path planning, Lazy Theta* is adapted to work directly on the sparse grid, avoiding unnecessary line-of-sight checks and taking advantage of the multi resolution hierarchy of the octree. The authors also present enhancements to the heuristic for path-finding, ensuring more efficient traversal even in sparsely mapped or partially known spaces. The same world model is shared between both exploration and planning components, reducing data redundancy and system complexity.

Experimental results on both simulated and real-world datasets, including complex offshore-like structures, demonstrate notable improvements in runtime, memory usage, and exploration efficiency when compared to regular grid-based methods. The octree-based approach required one to two orders of magnitude fewer iterations to detect frontiers and significantly reduced execution times in 3D planning. The Lazy Theta* planner showed robust obstacle avoidance and path optimality under tight memory constraints.

- [24] **Multi-robot Exploration and Coverage: Entropy-based Adaptive Maps with Adjacency Control Laws**:

  This paper introduces a novel framework for multi-agent exploration and coverage (E-C) missions that combines three key elements: entropy-based world modeling, adaptive occupancy grids, and adjacency-aware control laws. The authors design a system where agents use local sensor data to construct a shared entropy grid that reflects spatial uncertainty, guiding the group toward regions with high information gain. The grid structure adapts in real-time using a quadtree representation, dynamically refining resolution in areas of high entropy and coarsening it where uncertainty is low. This enables memory-efficient mapping with minimal loss of environmental detail.

  A critical contribution of the work is the integration of the adaptive entropy grid with decentralized control laws that consider both local and neighboring agent states. The proposed TVD-D1 (time-varying density, 1-hop) controller enables each agent to compute control inputs based on the density and centroid of its Voronoi region, as well as those of adjacent agents. This leads to more coordinated agent movement, avoids redundant exploration, and allows the system to converge toward a target agent distribution in environments with time-varying density functions. The entropy grid itself is generated using Shannon entropy derived from occupancy probabilities, and drives both control and discretization.

  The system is evaluated in large-scale simulations with up to 75 agents operat-

ing in an unknown environment. Results show that the agents successfully explore and cover the space while maintaining grid adaptivity and communication efficiency. The approach achieves near-optimal coverage with reduced computational cost compared to constant-resolution occupancy grids. The modular design and demonstrated scalability make the method well-suited for real-time multi-robot deployments in tasks such as search and rescue, surveillance, and environmental monitoring.

- [25] **Effective Exploration for MAVs Based on the Expected Information Gain**: This paper presents an autonomous exploration strategy for MAVs that selects the next-best-view by maximizing expected information gain while minimizing flight cost. The approach accounts for sensor uncertainty, abrupt motion changes, and safety concerns such as battery life and predictable flight paths. A utility function balances the expected reduction in 3D model uncertainty with the cost of reaching candidate viewpoints, factoring in path smoothness and proximity to the mission's endpoint. A key novelty is the time-dependent cost that ensures the MAV returns safely to its starting point before battery depletion, without sacrificing exploration performance.

  The MAV's movement is constrained to a dynamically refined "hull" around the region of interest, allowing uniform sampling of candidate viewpoints. Each voxel in the octree-based map stores uncertainty estimates, and measurement updates are performed via covariance intersection from stereo or monocular observations. The system uses a probabilistic model of the environment and computes the information gain through an approximation based on camera geometry. Efficient online evaluation of candidate poses is performed using sparse raycasting. The algorithm avoids abrupt turns by penalizing sharp directional changes, aiding human supervision in legal scenarios where manual oversight is required.

  Extensive simulation and real-world experiments validate the proposed method. Compared to state-of-the-art alternatives, it yields better uncertainty reduction and safer paths, and faster planning times. In a realistic building-scale scenario, the system produced accurate 3D reconstructions with significantly lower voxel uncertainty. Real-time capability was demonstrated using a stereo camera on an MAV running the algorithm onboard.

- [26] **Autonomous Quadrotor 3D Mapping and Exploration Using Exact Occupancy Probabilities**:

  This paper proposes a framework for 3D mapping and autonomous exploration using quadrotors, which combines an exact inverse sensor model with entropy-based exploration. Unlike conventional occupancy grid mapping methods that use ap-

proximate inverse sensor models, the proposed approach computes exact Bayesian occupancy probabilities for each voxel by fully integrating prior beliefs and sensor characteristics. This allows the robot to make more accurate probabilistic estimates of unknown environments while fusing data from multiple depth sensors of different types and accuracy levels.

To manage the computational burden of 3D exploration, the authors project key aspects of the 3D map into 2D planes for planning purposes. Specifically, entropy and collision maps are generated in 2D to guide motion while maintaining a full-resolution 3D map for mapping. The robot selects exploration poses by maximizing expected information gain, using predicted entropy updates from simulated rays. The system also incorporates real-time constraints, sensor synchronization, and computational limitations through ROS-based modular software and external computing resources, enabling flexible deployment on resource-constrained quadrotors.

The framework is validated through both simulation and real-world experiments using an Asus Xtion, Hokuyo lidar, and onboard Nvidia Jetson computing. Results show effective 3D reconstruction, safe collision avoidance, and successful entropy-driven exploration in cluttered environments. The authors also explore the tradeoffs between map completeness and exploration efficiency, showing that careful integration of planning heuristics and map projections can improve both. This work demonstrates how exact probabilistic mapping can be scaled to real-time robotic systems in practical 3D navigation scenarios.

- [27] **A Comparison of Volumetric Information Gain Metrics for Active 3D Object Reconstruction**:

  This paper presents a detailed comparison of different volumetric information gain (IG) metrics used for selecting the next best view in 3D reconstruction tasks. The authors focus on dense camera-based sensors and probabilistic voxel-based mapping frameworks. They propose several new IG formulations—such as Occlusion Aware, Unobserved Voxel, Rear Side Voxel, Rear Side Entropy, and Proximity Count—each capturing different aspects of how visibility and uncertainty influence the informativeness of a viewpoint. These are evaluated against existing state-of-the-art metrics from Kriegel et al. (2015) and Vasquez-Gomez et al. (2014).

  The evaluation is performed in simulation using 11 different object models and a static candidate viewspace around each model. The authors isolate the effect of each IG formulation by using idealized sensors and environments, ensuring the only varying factor is the metric itself. Performance is assessed based on surface coverage, map entropy reduction, and computational cost. Results indicate that the Proximity

Count VI and Vasquez-Gomez's Area Factor VI consistently achieve higher surface completion in fewer views, while the Average Entropy VI excels at reducing global map uncertainty, though this doesn't necessarily correlate with reconstruction completeness.

The proposed metrics are implemented in a modular ROS-based software framework, which the authors release as open source. The system is hardware-agnostic and easily extensible, enabling integration with real-world platforms for tasks such as object modeling and inspection. The findings show that no single VI metric is best for all objectives, but that task-specific choices e.g., prioritizing surface coverage vs. entropy minimization, can significantly influence reconstruction efficiency and quality. This work provides the first systematic comparison of IG formulations in active 3D reconstruction and establishes useful guidelines for selecting appropriate metrics in practice.

- [28] **Autonomous 3D Exploration of Large Structures Using an UAV Equipped with a 2D LIDAR**:

  This paper proposes a deterministic and modular system for fully autonomous 3D exploration using an unmanned aerial vehicle (UAV) equipped with a 2D laser scanner. By leveraging the lightweight and cost-effective nature of 2D LIDAR, the authors introduce a flyby sampling manoeuvre that extends its capability for 3D mapping. The system uses an OctoMap-based sparse volumetric representation and integrates frontier-based exploration, the Lazy Theta* path planner, and local/global switching strategies. A key contribution is the generation of safe, smooth flight paths that obey sensor visibility and obstacle clearance constraints, while allowing the UAV to operate entirely onboard and online.

  The framework was validated through extensive Hardware-in-the-Loop simulations using flight-ready components. The results demonstrate that the system can explore up to 93% of the predefined search volume in under 30 minutes, while generating non-repetitive, efficient paths. Compared to nearest-neighbor and surface-frontier heuristics, the proposed octomap-based frontier selection strategy achieved higher space coverage and lower final map entropy. The architecture is ROS-based, reusable across UAV and AUV platforms, and requires minimal operator input.

- [29] **Efficient 3D Exploration with Distributed Multi-UAV Teams: Integrating Frontier-Based and Next-Best-View Planning**:

  This paper proposes a distributed framework for 3D exploration using multiple UAVs equipped with LiDAR, combining frontier-based and NBV planning. Each UAV independently builds and updates a local OctoMap, detects frontiers, and eval-

uates candidate viewpoints based on information gain and distance metrics. A simplified auction-based assignment strategy coordinates exploration targets among UAVs to minimize redundancy. The system emphasizes adaptability and computational efficiency by using adaptive Octree depth for frontier detection and a modified A* planner for safe and responsive path planning. Real-time collaboration is achieved through OctoMap sharing and log-odds map fusion among UAVs.

The proposed framework is validated in ROS-based simulation environments, including urban and forest settings, using one to three UAVs. Extensive benchmarking shows that the method achieves superior exploration rates and path efficiency compared to classical frontier-based and greedy information-gain approaches. Ablation studies demonstrate the importance of tuning key parameters such as Octree depth, distance-weighting constant λ, and clustering bandwidth to balance accuracy and performance. Overall, the method shows strong scalability, efficient coordination, and adaptability across a range of environments, providing a practical solution for real-time, cooperative 3D exploration.

- [30] **Fast Multi-UAV Decentralized Exploration of Forests**:

  This paper introduces a decentralized exploration strategy tailored for multi-UAV teams operating in dense and occlusion-heavy environments such as forests. The method allows each UAV to switch dynamically between two operational modes, Explorer and Collector, based on the structure of the surrounding frontier clusters. While Explorers aim to cover large unknown regions, Collectors are tasked with clearing narrow, occlusion-induced trails left behind. This switching mechanism ensures high-speed exploration with minimal need for costly revisits and significantly improves overall mission efficiency. Coordination is achieved via peer-to-peer communication, and a cost-based assignment function ensures effective region partitioning among the robots without relying on centralized infrastructure.

  The proposed framework is evaluated in both simulated and realistic forest environments, including high-fidelity 3D reconstructions. Experiments with up to ten UAVs show that the system consistently outperforms both centralized and decentralized frontier-based approaches in terms of mission completion time and average UAV velocity. Even under limited communication ranges, the method maintains its advantage by optimizing exploration based on local maps and area assignments. The results confirm the scalability, robustness, and high-speed capability of the system, making it suitable for complex missions such as disaster response and large-area inspection in unstructured terrains.

- [31] **Automated Real-Time Inspection in Indoor and Outdoor 3D Environments with Cooperative Aerial Robots**:

  This work presents a robust and integrated real-time framework for infrastructure inspection using a team of cooperative heterogeneous UAVs operating in complex 3D indoor and outdoor environments. The proposed methodology follows a two-stage approach: The first stage involves environmental mapping based on the complementary sensing capabilities of UAV agents, some equipped with LiDAR and gimballed cameras, while the second stage focuses on the generation and execution of collision-free cooperative inspection paths.

  As illustrated in Figure 3.1, during the first stage, all UAV agents, regardless of sensor type, are tasked with visiting scenario-dependent fixed exploration waypoints aligned along the longest axis of the environment. This strategy enables efficient spatial coverage and initial map construction, which forms the basis for the subsequent inspection planning.

  Experimental evaluations conducted in simulation environments closely similar to real-world scenarios, including building and crane inspections, demonstrate the effectiveness of the system in achieving high inspection quality and full surface coverage under communication and time constraints.

2.2 How This Work Differs From Related Work

A large body of research has been conducted in the area of 3D exploration using aerial robots, focusing on efficient path planning, environment representation, collision avoidance and coordinated multi-agent behavior. The following section outlines and reflects on the research we reviewed, highlighting how it contributed to shaping and optimizing our implementation.

Several works have adopted frontier-based exploration to incrementally expand the map by directing UAVs toward the boundary between known and unknown space. Some of these focus purely on classical frontier algorithms without integrating higher-level decision-making, such as those in [7],[12],[13] which emphasize efficient expansion and coverage. Other approaches combine frontier-based strategies with more advanced mechanisms such as utility scoring and cost-aware planning. These works [15],[20],[22],[23] aim to improve exploration efficiency and data quality.

Other research efforts adopt Next-Best-View (NBV) strategies[8],[9] to guide UAVs towards viewpoints that are expected to maximize information gain or inspection quality.

Efficient 3D environment representation is also a core aspect of UAV exploration systems. Works such as [19],[21] utilize OctoMap or Octree-based mapping techniques to build

memory-efficient occupancy grids in real-time.

Path optimization is frequently modeled as a Traveling Salesman Problem (TSP) or its variants[4],[6]. In this work, a cost-aware variant of the TSP is employed, where the path planning strategy penalizes revisits to previously explored areas in order to enhance overall route efficiency.

Our implementation focuses primarily on the use of the frontier algorithm combined with a customized variation of the TSP algorithm. This approach aims to produce an efficient solution that satisfies the constraints defined in Chapter 3. By adapting the underlying heuristics, we seek to improve resource utilization.

3 Chapter 3

Problem Statement

- Environmental Mapping and Pre-Inspection Collisions
- Traveling Salesman Problem Limitations in Inspection Task

3.1 Environmental Mapping and Pre-Inspection Collisions

The UAV Fleet:

The mission configuration consists of a fleet of five UAVs and one centralized Ground Control Station (GCS). The fleet is divided into two categories: two Explorer drones (named *jurong* and *raffles*) and three Photographer drones (named *changi*, *sentosa*, and *nanyang*). Each UAV type is tailored for a specific role within the cooperative inspection process.

Explorers are larger and heavier than Photographers, which allows them to carry a rotating LiDAR sensor in addition to a gimballed camera. Their primary task is to generate an initial map of the environment using the LiDAR, although their increased mass results in slower flight speeds.

In contrast, Photographer UAVs are more lightweight and agile, equipped solely with gimballed cameras. Their higher speed makes them well suited for covering large, previously mapped regions in less time, enabling efficient image capture for inspection.

The Ground Control Station (GCS) acts as the central processing unit for evaluating collected data. As each interest point may be inspected multiple times by different UAVs, the GCS selects the highest-quality image captured for each location to assign a score. This centralized comparison ensures that the final inspection score reflects the best available observations for each point of interest.

Environmental Mapping and Pre-Inspection Collisions:

The CARIC is structured in two primary stages: the first focuses on Environmental Mapping, while the second emphasizes Cooperative Inspection. As illustrated in Figure 3.1, during the first stage, all UAV agents, regardless of their sensor types, are assigned to visit fixed exploration waypoints that are scenario-dependent and distributed along the longest axis of the environment. The purpose of this stage is to build an initial occupancy map that identifies a high percentage of traversable, inspection-relevant voxels. Let $\mathbf{V_{valid}}$ denote the set of valid voxels for inspection, and $\mathbf{V_{total}}$ be the total number of voxels in the

environment; the objective is to maximize the mapping ratio $\frac{|V_{valid}|}{|V_{total}|}$ prior to the inspection stage.

The rationale is that by generating a detailed map during Stage 1, the UAVs, particularly the Photographer agents that are not equipped with LiDAR and therefore lack the ability to sense their surroundings, will benefit from a preprocessed map that allows them to navigate safely and inspect structures without relying on local obstacle detection. This initial mapping should allow UAVs to operate without collisions during Stage 2, leveraging shared information to reduce spatial uncertainty.

However, through extensive empirical testing across all provided simulation scenarios (i.e. MBS, hangar, crane), a persistent issue was observed: UAVs, and especially the Photographer agents, frequently experience collisions even before the inspection phase has been fully initiated. These collisions occur despite the existence of a partial environment map and result in the early loss of agents that would otherwise contribute significantly to the efficiency and completeness of the inspection task. An example of this behavior is illustrated in Figure 1.1, where a Photographer UAV collides with surrounding geometry due to a lack of accurate obstacle information in its vicinity.

This recurring issue highlights the need for a more robust and strategic approach to stage 1, particularly in terms of how waypoints are selected and how shared mapping information is utilized by agents with limited sensing capabilities.

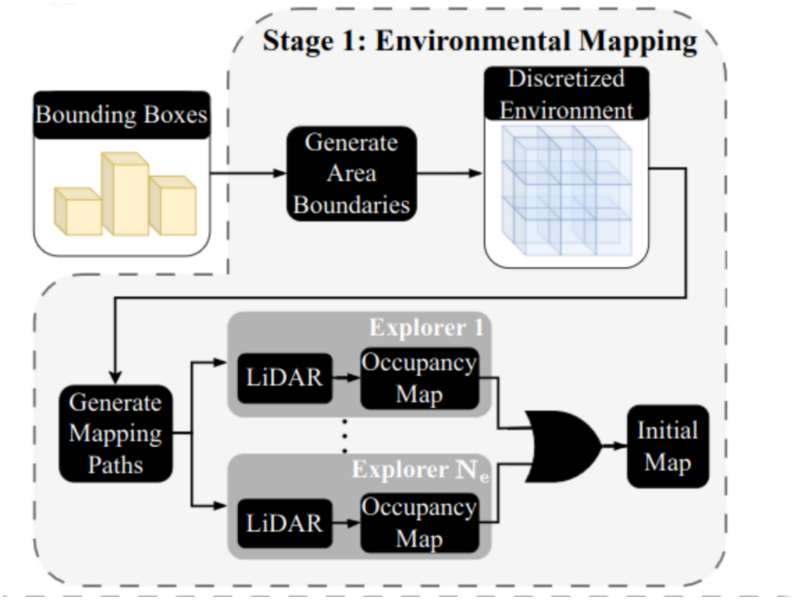

Figure 3.1: Stage 1 Environmental Mapping Process

Problem Constraints and Environmental Structure

Let $V \subset \mathbb{R}^3$ denote a bounded 3-dimensional volume, discretized into voxels, each associated with an occupancy probability $P_o(v)$ for all points $v \in V$. Initially, all voxels are assumed unknown, therefore $P_o(v)$ is not defined or neutral, and UAV agents must actively explore the environment to classify voxels into occupied, free, or unobservable categories. The goal of autonomous exploration during the Environmental Mapping stage can therefore be formulated as the process of constructing an occupancy map M of the observable subset of voxels within V. It's worth noting that there might be some voxels that can't be observed due to practical limitations, such as hollow spaces or narrow pockets. The exploration process is regarded as complete when there are no "unknown" voxels left in the observable section of the space.

The status of each voxel $M(v)$ is categorized based on its occupancy probability $P_o(v)$, into:

$$\mathbb{M}(v) = \begin{cases} \text{free}, & P_o(v) = 0.1 \\ \text{unknown}, & 0.1 < P_o(v) < 0.98 \\ \text{occupied}, & P_o(v) = 0.98 \end{cases}$$

A core component of the mapping task is the notion of frontiers. Frontiers represent boundaries between the known free space and unknown space regions:

$$F = \{v \in V : v \text{ is a free cell and has at least one neighboring cell that is unknown}\}$$

Frontiers are critical to exploration as they identify regions that, when visited, can maximize the information gained about the environment. The autonomous exploration process continues iteratively until no frontiers remain:

$$F = \emptyset$$

During this process, UAV explorers equipped with LiDAR sensors generate partial occupancy maps M_i:

$$M_i : V \to \{\text{occupied, free, unknown}\}, \quad i = 1, 2, \ldots, N_e$$

where N_e denotes the number of explorer UAVs participating in the mapping task.

These individual occupancy maps are merged into a global initial occupancy map:

$$M_{\text{init}} = \bigcup_{i=1}^{N_e} M_i$$

This merging process is not performed only once, but occurs continuously during the mission. As long as exploration is ongoing and UAVs come into communication range or line-of-sight with one another, their local occupancy maps are synchronized in real-time. This dynamic fusion contributes to a progressively more complete and up-to-date version of M_{init}. Once the frontier-based exploration is concluded and the environment has been fully mapped, further merging becomes unnecessary, as no new information is generated.

The problem arises when UAV agents, especially those without local sensing capabilities (i.e., Photographer UAVs), utilize M_{init} for path planning without adequate collision checking, resulting in frequent collisions. Hence, our problem formulation improves both the accuracy and completeness of the occupancy map, as well as efficient frontier-based exploration, to reduce collisions and enhance mission efficiency.

3.2 Traveling Salesman Problem Limitations in Inspection Task

Before introducing the travel salesman problem limitations, it is important to first explain how the overall mission score is calculated, as this metric serves as the foundation for evaluating inspection performance.

Explanation of the mission score

Let us denote the set of the interest point as $\mathcal{I}$, and the set of UAVs as $\mathcal{N}$. At each simulation update step k, we denote $q_{i,n,k}$ as the score of the interest point i captured by UAV n. Specifically $q_{i,n,k}$ is calculated as follows:

$$q_{i,n,k} = \begin{cases} 0, & \text{if } (k=0) \text{ or } (q_{\text{seen}} \cdot q_{\text{blur}} \cdot q_{\text{res}} < 0.2), \\ q_{\text{seen}} \cdot q_{\text{blur}} \cdot q_{\text{res}}, & \text{otherwise.} \end{cases}$$

where $q_{\text{seen}} \in \{0,1\}$, $q_{\text{blur}} \in [0,1]$, $q_{\text{res}} \in [0,1]$ are the LOS-FOV, motion blur, and resolution metrics, which are elaborated in the subsequent sections. The above equation also implies that an interest point is only detected when its score exceeds a threshold, which is chosen as 0.2 in this case.

At the GCS, the following will be calculated:

$$q_{i,\text{gcs},k} = \max \left[\max_{n \in \mathcal{N}_{\text{gcs}}} (q_{i,n,k}),\ q_{i,\text{gcs},k-1} \right],$$

where $\mathcal{N}_{\text{gcs}}$ is the set of UAVs that have LOS to the GCS. This reflects the process that the GCS receives the images captured by the drones and selects the one with the highest score and keeps that information in the memory. Moreover, the stored data will also be compared against the future captures.

The score of the mission up to time k is computed as follows:

$$Q_k = \sum_{i \in \mathcal{I}} q_{i,\text{gcs},k}.$$

Hence, **the mission score will be Q_k at the end of the mission**.

The Traveling Salesman Problem

During the second stage Cooperative Inspection, the UAV agents utilize path-planning strategies inspired by variations of the TSP. In our current implementation, the selection of the next inspection point is determined solely by a cost function that prioritizes minimizing the distance from the UAV's current position to available inspection points. Specifically, if we denote the current UAV position as U_{pos} and a set of candidate inspection points as P_{inspect}, the existing cost function is formulated as:

$$C_{\text{current}} = \min_{p \in P_{\text{inspect}}} d(U_{\text{pos}}, p) \tag{3.1}$$

where $d(U_{\text{pos}}, p)$ denotes the Euclidean distance between the UAV position and each potential inspection point.

However, extensive experimentation during simulation scenarios has highlighted a critical limitation with this approach. Often, UAV agents repeatedly select inspection points surrounded by regions of interest points with high scores, thereby neglecting other inspection points with potentially higher cumulative value that remain unexplored. As a consequence, towards the end of the inspection stage, UAVs frequently face scenarios in which numerous critical inspection points-those with initially high-value points of interest around them-remain unattended, severely limiting the final overall score and effectiveness of the inspection mission.

The underlying reason for this issue is the inadequacy of the current cost function definition. The current implementation fails to incorporate information regarding the interest points' potential inspection value in the decision-making process, thus causing suboptimal resource allocation and reduced coverage quality.

To address this limitation, an additional criterion should be introduced to enhance the decision-making process during inspection point selection. This new criterion should factor into the cost function not only the spatial proximity but also the potential inspection value of nearby points of interest around each inspection candidate. By integrating such a measure, UAV agents could dynamically prioritize inspection points that maximize both spatial and informational coverage, significantly improving overall inspection efficiency and mission outcomes.

Details about the specific method chosen for integrating this additional criterion into the TSP-based selection process are discussed comprehensively in the subsequent chapter, where our proposed approach is outlined and validated.

4 Chapter 4

Proposed Approach / Solution

- Frontier-Based Exploration Algorithm
- Traveling Salesman Problem Limitations in Inspection Task
- General Modifications for Enhanced Performance

4.1 Frontier-Based Exploration Algorithm

Frontier-based exploration arises from the necessity to autonomously investigate environments that are hazardous, inaccessible, or simply unexplored by humans. These environments may include deep-sea regions, outer space, or areas affected by natural disasters on Earth. Key challenges in this domain involve constructing accurate maps of the area, determining optimal exploration paths, and dynamically adjusting strategies in response to evolving and uncertain conditions

A key challenge in exploration lies in how an autonomous agent determines its next movement in order to maximize knowledge of the environment. Frontier-based exploration focuses on detecting and analyzing the boundaries between known free space and yet-unexplored regions-commonly referred to as ”frontiers.” These frontiers represent zones with partial or uncertain information, which may potentially include important features, obstacles, or areas of interest. The core objective of frontier exploration is to autonomously investigate these uncertain boundaries, progressively expanding the mapped area and uncovering useful environmental insights.

4.1.1 Virtual Boundary Generation

Initially, the GCS constructs a conceptual “visual boundary” (illustrated in Figure 4.3) comprised entirely of occupied cells, which encloses the boundaries of the given exploration scenario. This virtual boundary effectively transforms any open environment into a closed, finite region. The occupied cells defining this boundary correspond to the outermost coordinates of the environment along the X-axis (X_{min}, X_{max}), the Y-axis (Y_{min}, Y_{max}), and the upper limit along the Z-axis (Z_{max}). The lower limit of the Z-axis, denoted as Z_{min}, naturally represents the ground level, which is inherently occupied.

Upon generating this virtual boundary, the GCS publishes these boundary points as a `PointCloud2` message to the topic `/$(explorer_namespace)/cloud_in`. This topic serves as an input channel where LiDAR-equipped UAV explorers continuously publish

detected occupied points during their mission. Hence, this boundary seamlessly integrates into the exploration data flow, acting as a predefined occupied boundary layer.

The motivation behind the construction of this virtual boundary stems from a fundamental limitation inherent to LiDAR sensors. Specifically, if a LiDAR sensor ray passes through free (unoccupied) voxels without eventually encountering an occupied voxel, all points detected along the ray path may be considered uncertain or invalid, thereby causing valuable environmental data to be disregarded. By introducing this occupied boundary, each LiDAR sensor ray is guaranteed to intersect with at least one occupied voxel, regardless of the scenario or environment type. Thus, each sensor measurement is effectively validated, ensuring maximum utilization of sensor data for accurate and comprehensive environment mapping.

Mathematically, let r be a ray originating from the LiDAR sensor position p_{sensor}. The presence of the virtual boundary ensures that, for any direction vector $\vec{d}$, the ray defined by:

$$r(t) = p_{\text{sensor}} + t\vec{d}, \quad t \geq 0$$

will always intersect with at least one occupied voxel (v_{occupied}) within the predefined virtual boundary limits:

$$\exists\, t' \geq 0 : r(t') \in v_{\text{occupied}}$$

This ensures consistent and accurate frontier determination throughout the exploration process.

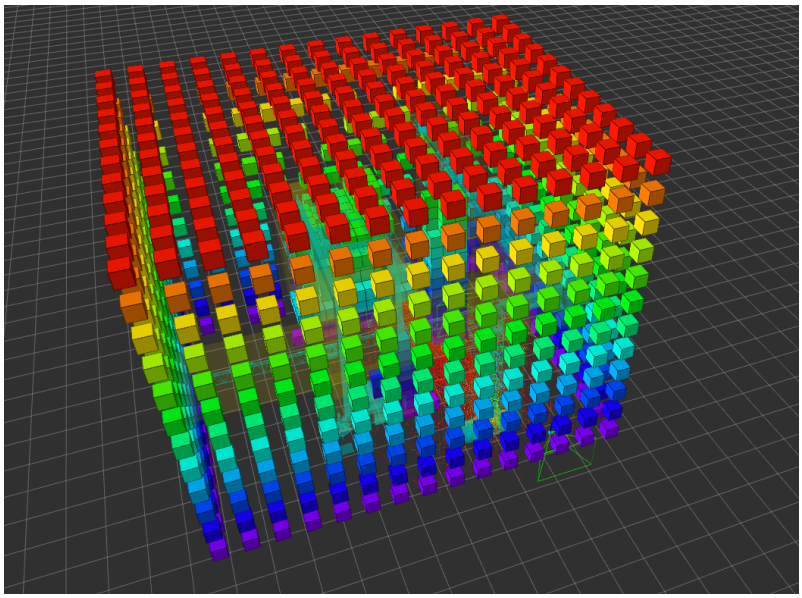

Figure 4.1: Illustration of the virtual occupied boundary used for frontier exploration.

4.1.2 Occupied and Free Space Integration

The next stage in the Frontier-Based Exploration Algorithm involves handling data acquisition and integration from LiDAR-equipped UAV explorers. The algorithm utilizes two primary ROS subscribers:

- `occupied_Callback`: This subscriber receives `PointCloud2` messages from the topic `/$(explorer_namespace)/octomap_point_cloud_centers`. The points received represent voxels detected by the UAV's LiDAR as occupied.
- `free_Callback`: Similarly, this subscriber handles messages from the topic `/$(explorer_namespace)/free_cells_vis_array`. The received points indicate voxels detected by the LiDAR as free. To ensure reliability, only free voxels within a certain proximity radius $r = \gamma \times$ `area_details.resolution.data` are considered, where γ is a user-defined constant.

  The role of γ is to scale the observation radius according to the voxel resolution of each scenario. Since resolution defines the size of each voxel, using a multiplicative factor allows us to adapt the effective sensing radius proportionally.

The exploration algorithm employs two primary threads to manage continuous and efficient processing:

1. **merge_map Thread**
2. **published_cells Thread**

merge_map Thread Operation

The `merge_map` thread runs continuously until the Frontier-Based Exploration process is complete. Initially, it updates an adjacency data structure based on occupied voxel information acquired from `occupied_Callback`. Subsequently, when a UAV explorer has line-of-sight (LOS) to another UAV, the thread performs a real-time merging of occupancy information. This coordination is realized through the `published_cells` thread, which is responsible for broadcasting the local occupancy data including occupied and free cells to neighboring agents within the communication range. This continuous exchange ensures that each UAV maintains an up-to-date map by integrating the observations of other agents in real time, thus improving global spatial awareness and reducing the set of unknown voxels in the environment.

Following this integration step, frontier points are computed. The frontier detection is executed via the following steps:

1. **Frontier Detection**: A voxel is classified as a frontier if it is adjacent to at least one unknown voxel and at least one free voxel. Mathematically, for each voxel $v \in V_{\text{free}}$ (set of free voxels), if there exists a neighboring voxel $v' \in V_{\text{unknown}}$, then voxel v is marked as a frontier:

$$F = \{v \in V_{\text{free}} \mid \exists v' \in V_{\text{unknown}}, v' \text{ is neighbor of } v\}$$

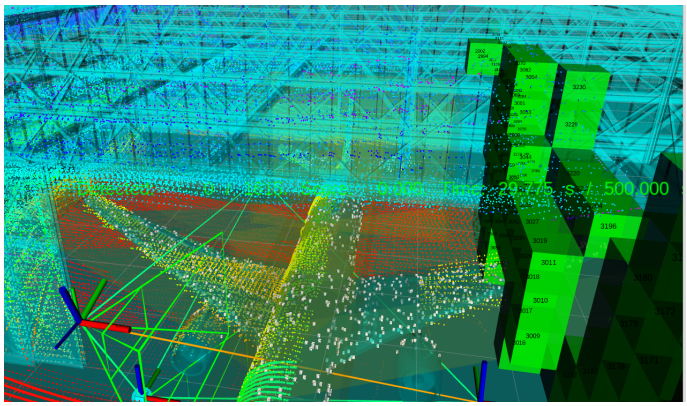

Figure 4.2: Detected frontier voxels visualized after each call to the `Frontier Detection` function.

A pseudocode representation of `Frontier Detection` approach follows:

```
function FindFrontiers(FreeCoords, UnknownCoords, NeighborOffsets):
    FrontierList = []
    for each coord in FreeCoords:
        for each offset in NeighborOffsets:
            neighbor = coord + offset * resolution
            if InBounds(neighbor) and neighbor in UnknownCoords:
                FrontierList.append(coord)
                break
    return FrontierList
```

2. **Frontier Clustering**: Detected frontier voxels are clustered using a MeanShift algorithm to group spatially adjacent frontiers, reducing the computational load and improving path planning efficiency:

$$F_{\text{clusters}} = \text{MeanShiftCluster}(F, \text{bandwidth})$$

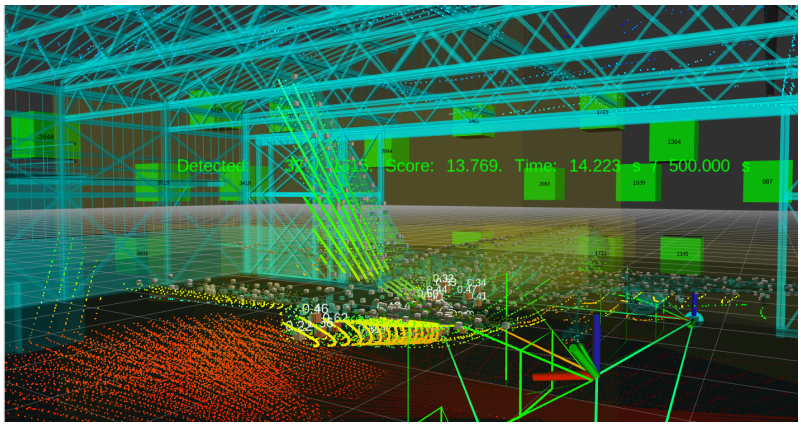

Figure 4.3: Visualization of clustered frontier regions used for target selection.

A pseudocode representation of `Frontier Clustering` function follows:

```
function ClusterFrontiers(FrontierCells, Bandwidth):
    if FrontierCells.empty():
        return []
    if FrontierCells.size() == 1:
        return [FrontierCells[0]]
    # Build a search tree over frontier points
    Tree = KDTree(FrontierCells)
    # Run Mean Shift clustering with given bandwidth
    Clustering = MeanShift(Bandwidth)
    Clustering.fit(FrontierCells)

    ClusterCenters = Clustering.cluster_centers_
    for center in ClusterCenters:
        _, idx = Tree.query(center)
        SnappedCenters.append(FrontierCells[idx])
    return SnappedCenters
```

3. **Optimal Frontier Selection**:

   To ensure efficient exploration in a multi-UAV environment, an optimal frontier selection strategy is employed. This strategy selects the next best cluster of frontier points for exploration based on a utility-based approach that balances information gain and distance cost, while also taking into account inter-UAV coordination.

   The process begins by evaluating all available frontier clusters. For each candidate cluster, the algorithm computes an *information gain* value, representing the expected number of unknown voxels around the cluster. This value is estimated by sampling the cluster's neighboring cells using a predefined offset pattern and checking against a list of currently unknown cells in the 3D map representation.
   Next, a *utility score* is computed for each cluster, which is a function of the following:

   - The distance from the current UAV to the cluster (self-distance)
   - The information gain around the cluster
   - A penalty if the cluster is closer to another UAV than to the current one (to avoid redundant effort)

The utility score is calculated using the formula:

$$\text{utility} = -(w_d \cdot d_{self} - w_g \cdot IG + \gamma \cdot \max(0, d_{self} - d_{other})) \tag{4.1}$$

where w_d and w_g are the weights for distance and information gain respectively, γ is a penalty multiplier, d_{self} is the distance from the UAV to the cluster, IG is the information gain, and d_{other} is the distance from the other UAVs to the cluster.

Once all candidate clusters are scored, the one with the highest utility score is selected. If this target has already been selected multiple times consecutively, indicating a possible deadlock or poor utility landscape, the algorithm attempts to choose an alternative cluster from the remaining top candidates. This ensures robustness against repetitive or suboptimal target selections.

After a best frontier is selected, it is not immediately assigned as the next target. Instead, a stabilization mechanism is introduced to avoid frequent switching between similar frontier candidates. The algorithm checks if the newly selected frontier differs from the currently assigned target. If a new best frontier is selected repeatedly over several cycles (specifically, three times), it is then accepted as the new target. This delay acts as a temporal filter, allowing the UAV to begin approaching the previously selected target without unnecessary re-planning due to minor fluctuations in utility values. This logic ensures that transient changes in utility do not cause abrupt target updates, contributing to smoother and more stable UAV movement planning.

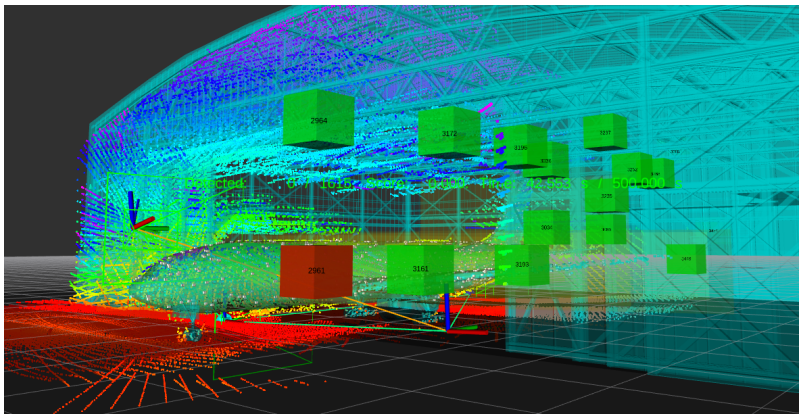

Figure 4.4: Exploration scenario with clustered frontiers shown in green and the chosen frontier highlighted in red, indicating the UAV's next navigation target.

A pseudocode representation of `Optimal Frontier Selection` function follows:

```
function SelectBestFrontier(Clusters, UAVPos, OtherUAVs, Tree, Threshold):
    if Clusters.isEmpty():
        return None

    Scores = []
    for each cluster in Clusters:
        IG = ComputeInformationGain(cluster, Tree)
        U  = ComputeUtility(cluster, UAVPos, OtherUAVs, IG)
        Scores.append((U, cluster))

    best = argmax(Scores)

    if best == LastTarget:
        RepeatCount += 1
    else:
        LastTarget   = best
        RepeatCount  = 0

        Alternatives = [(s, c) for (s, c) in Scores if c != LastTarget]
        best         = argmax(Alternatives)
        LastTarget   = best
        RepeatCount  = 0

    return best
```

This approach facilitates cooperative and efficient exploration by guiding each UAV to informative and non-redundant regions in the 3D environment.

4.1.3 Publishing Occupied and Free Cells

In a multi-agent exploration setting, coordination between UAVs is crucial. One such mechanism of coordination involves the continuous exchange of spatial information between agents, specifically, the occupied and free cells observed by each UAV. The thread serves this purpose by broadcasting spatial data to the other UAV in real-time.

The thread works as follows:

- It publishes the current lists of occupied and free grid cells using ROS topics: `/namespace/occupied_coords` and `/namespace/free_coords`, respectively.
- This transmission occurs repeatedly as long as the `flag` remains `False`, implying that the other UAV still requires updates.
- Once the other UAV completes its frontier-based exploration phase, it sends a flag

signal back (via a namespace-specific topic), which terminates this broadcast loop.

This thread-like mechanism ensures that data sharing remains consistent even under intermittent communication or temporary signal loss, while also allowing the receiving UAV to terminate the synchronization process once it no longer needs spatial updates.

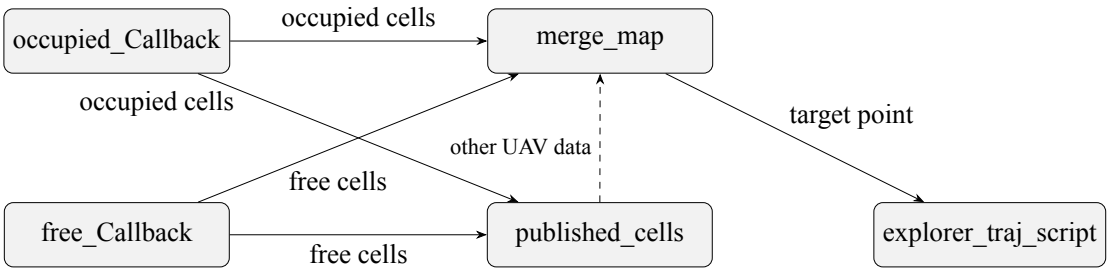

Figure 4.5: Dependency graph among the exploration routines.

4.2 Proposed TSP-based Inspection Path Refinement

To address the limitations previously discussed regarding the TSP formulation (3.1) for inspection planning, we introduce an enhanced strategy that incorporates both spatial cost and inspection value around each candidate point. This approach aims to mitigate the tendency of UAVs to overlook high-value inspection points that are not immediately adjacent to other clusters of interest.

Our solution modifies the cost matrix used in TSP-based path planning by assigning dynamic penalties to specific inspection points based on their contextual value. The method begins with evaluating the average interest score around each candidate inspection point, as computed by the `find_avg_score()` function. For each inspection point $p \in P_{\text{inspect}}$, the function uses a KD-Tree to identify all nearby interest points $i \in P_{\text{interest}}$ that lie within a radius r, where:

$$r = \text{area\_details.resolution.data}$$

Let $\mathcal{N}(p) \subset P_{\text{interest}}$ be the set of interest points within this radius. Each interest point $i \in \mathcal{N}(p)$ carries a scalar score value $s_i \in [0, 1]$. The average score for the inspection point p is then computed as:

$$\text{avg\_score}(p) = \begin{cases} 0, & \text{if } |\mathcal{N}(p)| = 0 \\ \frac{1}{|\mathcal{N}(p)|} \sum\limits_{i \in \mathcal{N}(p)} s_i, & \text{otherwise} \end{cases}$$

Finally, the average score is compared to a threshold. A point is considered important (score = 1) if its average score exceeds this threshold, and unimportant (score = 0) otherwise:

$$\text{binary\_score}(p) = \begin{cases} 1, & \text{if avg\_score}(p) > threshold \\ 0, & \text{otherwise} \end{cases}$$

The logic behind this adjustment is that high-value points, while important, should not be prioritized. The penalty defers their selection, allowing the UAV to initially cover less important points and leaving the high-value regions for more optimized integration later in the route.

Then, `calculateCircuits()` function, builds per-UAV inspection circuits. The function assigns targets to UAVs based on the minimum cost to unvisited points, while updating the source and destination sets for each UAV accordingly. The algorithm continues until all inspection points have been assigned, now taking into account both distance and interest-aware penalties.

This method results in more balanced and strategically valuable coverage patterns, avoiding the early exhaustion of high-value targets and increasing the total number of inspection points.

Key Enhancements:

- Spatial context of inspection points is captured via average interest score.
- A binary scoring mechanism flags high-value targets (score $>$ threshold).
- Penalties are added to discourage premature selection of critical points.
- UAV circuits are computed using an enhanced greedy assignment algorithm.

A pseudocode representation of `calculateCircuits()` function follows:

```
function CalculateCircuits(Positions, NumNodes, CostMatrix, InspectPoints):

    Scores = FindAvgScore(InterestPoints, InspectPoints)
    Penalty = 50

    for i, score in enumerate(Scores):
        if score > 0.65:
            CostMatrix[:, i + NumUAVs] += Penalty

    // Initialize circuit data structures
    CircuitSources = [ [] for each UAV ]
    CircuitDests   = [ [] for each UAV ]
    CircuitCosts   = [ [] for each UAV ]
    Visited = mark start Positions as visited

    // Greedy assignment until all nodes covered
    while not all Visited:
        for each UAV i:
            node = argmin_j (CostMatrix[Positions[i]][j], for j unvisited)
            mark Visited[node] = true
            CircuitSources[i].append(Positions[i])
            CircuitDests[i].append(node)
            CircuitCosts[i].append(CostMatrix[oldPos][node])
            Positions[i] = node

    return CircuitSources, CircuitDests, CircuitCosts
```

4.3 General Modifications for Enhanced Performance

To further improve the overall performance and responsiveness of the multi UAV exploration and inspection framework, we incorporated several general optimizations and architectural refinements:

Modifications to `gcs.py`

- Removed the unnecessary publication of static area details inside a loop, as this message only needs to be sent once.
- Resolved a critical type comparison error involving `rospy.Time` objects. All time variables were converted to seconds using `.to_sec()` for reliable comparison. Type checks were also added to ensure correct conversions, preventing runtime exceptions.

Modifications to `traj` and `path script`

- **Dijkstra Enhancements**:
  - Re-enabled arrival message publishing for unreachable inspection points.
  - Introduced a `rate.sleep()` delay after publishing the `arrived` message to ensure synchronization with the trajectory planner thread.
- **Adjacency Graph Construction**:
  - Replaced linear nearest-neighbor search with a KDTree-based query in `construct_adjacency()`, significantly reducing the runtime of adjacency updates.
  - Removed redundant `enumerate()` since indices were not utilized.
- **Collision Risk Mitigation**:
  - Removed the condition that waits for more than 20 occupied cells before updating adjacency. This change allows faster reaction to obstacles and reduces collision probability.

Modifications to `photographer script`

- Removed the continuous loop that repeatedly fetched updated maps from the `explorer` nodes. In the previous implementation, the `photographer` continuously queried the latest map state throughout the exploration phase, resulting in unnecessary processing overhead and redundant data access.
- Instead, the updated map is now retrieved only once-at the end of the frontier exploration phase-when the UAV has completed mapping and is ready to initiate the inspection stage. This significantly reduces communication traffic and improves CPU utilization during runtime.
- This change assumes that the explorer has already constructed a sufficiently complete map before handing it over to the inspection module, thus allowing a one-time snapshot to suffice for the needs of the `photographer`.

Modifications to `run_solution.launch`

Grid Resolution Adaptations:

- The value of `grid_resolution` was updated to 8 (instead of 6) for general scenarios, and to 4 (instead of 5) specifically for the `hangar` scenario, enabling finer sampling and improved spatial representation.

Octomap Configuration Improvements:

- The resolution of the `octomap_server` was set to `grid_resolution/2` instead of a fixed value of 4, making the resolution dynamic and dependent on the grid configuration.
- The flags `publish_point_cloud_centers` and `publish_free_space` were enabled to provide both occupied and free-space data.
- The sensor model parameters were adjusted as follows:
  - `hit` probability: from 1.0 to 0.75,
  - `miss` probability: from 0.3 to 0.49,
  - `min`: from 0.3 to 0.10,
  - `max`: from 1.0 to 0.98,

  aiming for a more realistic representation of measurement uncertainty.

5 Chapter 5

Simulation Results

- Simulation Scenarios
- Coverage comparison

5.1 Simulation Scenarios

The simulation phase was designed to evaluate the system's performance across three distinct inspection scenarios, each representing different real-world environments. The primary objectives were: (i) to assess the coverage capabilities of the UAV team across varying geometric and semantic contexts, and (ii) to evaluate the impact of the proposed coordination and path-planning strategies on execution time, collision avoidance, and score efficiency.

Each scenario was simulated in Gazebo using a voxelized 3D environment managed by Octomap, and the UAVs operated via ROS-based multi-agent coordination modules. The simulation was visualized in real-time using RViz, allowing observation of UAV trajectories, frontier updates, and inspection progress. The three evaluation environments are:

- **Building Inspection (MBS)**: A vertical structure composed of three 60-meter towers connected at the top via a void deck. The full fleet of 5 UAVs was deployed to inspect both facades and connecting decks, as illustrated in Figure 5.1.

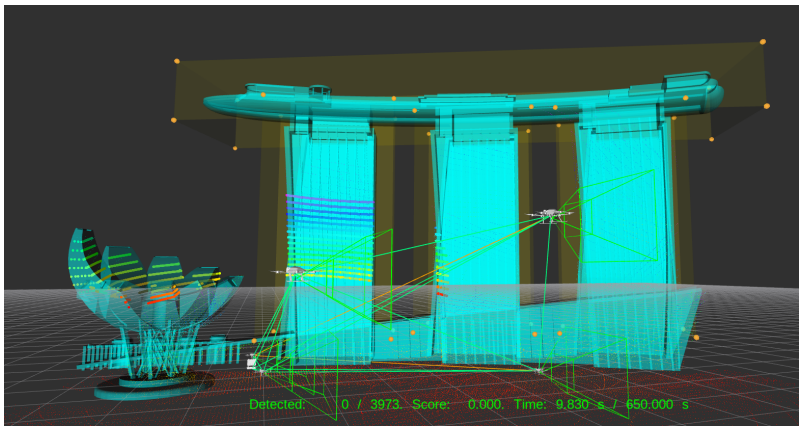

Figure 5.1: MBS Scenario

- **Crane Inspection**: A complex industrial scenario consisting of two construction cranes (60m and 80m tall) and a 50m gantry crane, emulating a seaport layout. The full 5-UAV fleet was deployed here, aiming to maximize coverage of structural elements from multiple angles, as illustrated in Figure 5.2.

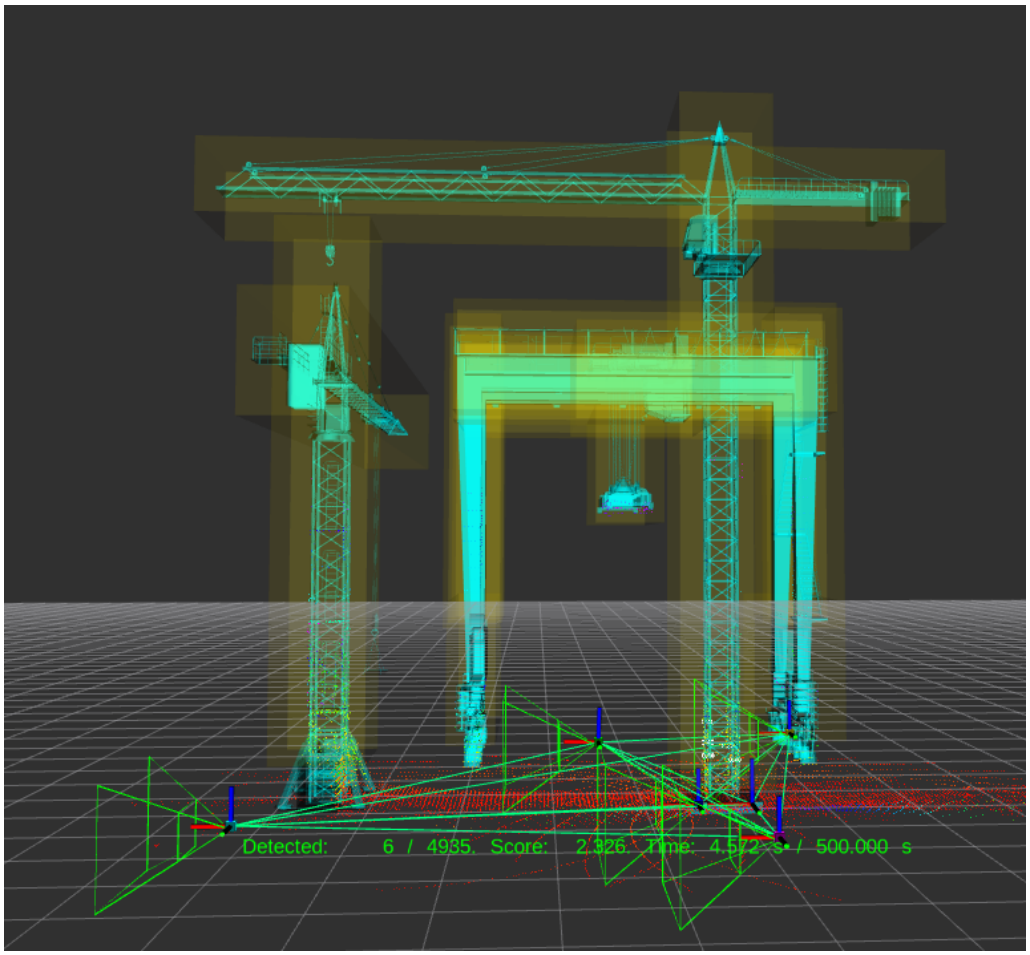

Figure 5.2: Crane Scenario

- **Hangar Inspection**: An aircraft placed at the entrance of a hangar served as the inspection target. The airplane dimensions are approximately 20m (height) and 70m (length). Due to the more constrained environment and reduced inspection volume, a smaller fleet of 3 UAVs (1 explorer and 2 photographers) was deployed, as illustrated in Figure 5.3.

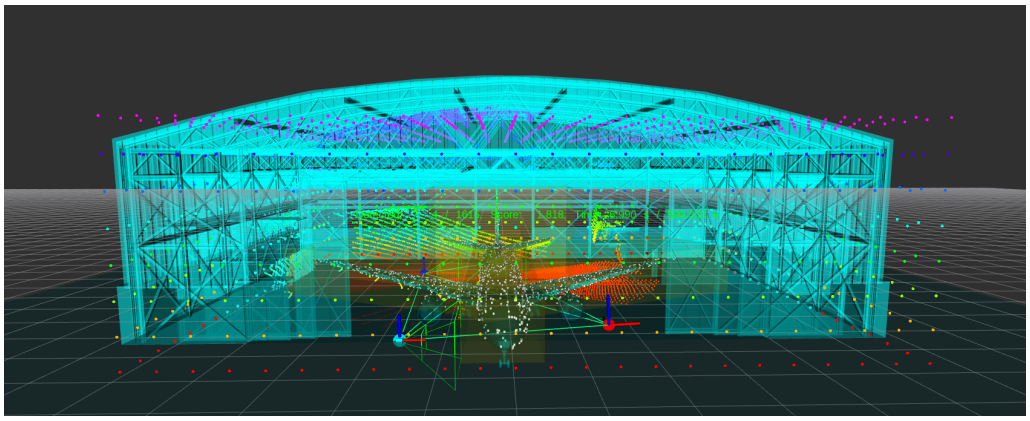

Figure 5.3: Hangar Scenario

Each simulation was run using a fixed number of UAVs per scenario, reflecting realistic task loads. No randomized trials were applied; instead, each environment was simulated using a deterministic setup to allow consistent evaluation of path strategies, target selection, and runtime behavior. The analysis in the following subsections presents metrics on inspection score, inter-agent collision rate, and total mission duration for both the baseline and improved implementations.

5.2 Coverage Comparison

To systematically evaluate the quality of exploration coverage across different implementations, we tested all three scenarios (MBS, Crane, Hangar) under four distinct execution durations: **250s**, **320s**, **500s**, and **650s**. The most notable improvements in mapping quality and final inspection readiness were observed after implementing a series of deeper architectural changes, such as the implementation of the Frontier-Based Exploration Algorithm and the inspection path planner (TSP refinement).

The observed improvements can be attributed to the ability of the proposed Frontier-Based Exploration Algorithm to allow each UAV to generate a more comprehensive and accurate environmental map prior to entering the inspection phase. By maintaining an up-to-date representation of free and occupied voxels, the UAVs are able to plan trajectories that avoid occupied areas more effectively, thus significantly reducing inter-agent collisions. Furthermore, the improved TSP inspection path planner prioritizes inspection points with lower average interest scores in the early stages. This scheduling defers high-value targets

to later phases of the mission when the UAVs are better positioned and the map is more complete, leading to more efficient coverage and better allocation of inspection efforts across the fleet.

5.2.1 From Baseline to Optimized: A Performance Evaluation

Before integrating the proposed improvements to frontier selection and TSP-based path planning, a set of general performance optimizations (as described in Section 4.3) had already been applied. These changes introduced both architectural and communication-level efficiencies, leading to observable improvements in overall mission performance. In particular:

- Inter-agent collisions continued to occur, but with slightly lower frequency compared to the baseline implementation.
- The final inspection score demonstrated noticeable improvement in nearly all tests conducted within our scenarios
- The number of interest points successfully visited per mission increased significantly, indicating better path planning effectiveness.
- The total time required to complete stage 1 (collaborative mapping) remained effectively unchanged.

These results indicate that even without modifying the core exploration and inspection algorithms, system-level optimizations contributed positively to mission outcomes.

However, in order to provide a complete understanding of the effects of the general optimizations, it is important to separately analyze their impact on computational performance and execution time.

Execution Time Improvements from General Optimizations

The modifications applied to the `gcs.py` script primarily improved communication stability and consistency, but did not introduce any measurable change in the execution times of the mission stages.

Similarly, changes made to the Dijkstra planning routine enhanced the synchronization performance between the trajectory and path planners, yet had no significant impact on raw computational times.

The most substantial improvement was observed in the adjacency graph construction phase. Specifically, by replacing the nearest-neighbor search with a KD-Tree-based structure, the runtime for adjacency construction when $n = 3000$ nodes was reduced from **12.86**

seconds to only **2.4 seconds**. This optimization significantly accelerated the environment mapping phase and reduced delays in updating the local navigation graph, thereby improving the responsiveness of the exploration system.

Comparison of Final Inspection Score

The following tables summarize the average final inspection scores, comparing the baseline implementation with the optimized one.

Table 5.1: Average Inspection Score before General Performance Optimizations

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	525	610	750	800
Crane	437	478	500	500
Hangar	330	402	420	485

Table 5.2: Average Inspection Score after General Performance Optimizations

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	581	640	760	790
Crane	810	1000	1200	1230
Hangar	423	478	540	576

Comparing the scores before and after applying general performance optimizations, significant improvements are evident across all tested scenarios (MBS, Crane, and Hangar). Specifically, the optimized implementation consistently achieved higher final inspection scores.

- **MBS scenario**: There is a modest improvement in the average inspection scores, particularly noticeable at shorter execution durations (250s and 320s). However, the improvement diminishes slightly at longer durations (500s and 650s), indicating diminishing returns from optimizations in scenarios already performing well.
- **Crane scenario**: Exhibited the most pronounced improvement, with substantial score increases observed across all durations. Particularly, at 500 seconds, scores improved dramatically from 500 points before optimization to 1200 points after optimization. This highlights the impact of the optimizations in geometrically complex environments, where UAVs benefited significantly from better navigation and collision avoidance.

- **Hangar scenario**: Although the environment was relatively simpler, the optimized implementation still resulted in consistent performance enhancements. Inspection scores steadily increased, for example from 420 to 540 at 500 seconds, demonstrating improved overall system efficiency even in less challenging environments.

These results underline that the applied general performance optimizations substantially enhanced exploration efficiency, mission reliability, and final inspection outcomes across diverse scenarios, highlighting their critical role in improving UAV fleet effectiveness in practical applications.

To complement the tabular data, Figure 5.4 presents a visual comparison of the average inspection scores across all scenarios and execution durations. Each line represents one scenario (MBS, Crane, Hangar), while dashed lines correspond to the baseline implementation and solid lines to the optimized one. The improvements are particularly pronounced in the Crane scenario, where the optimized system shows significantly higher scores at longer durations.

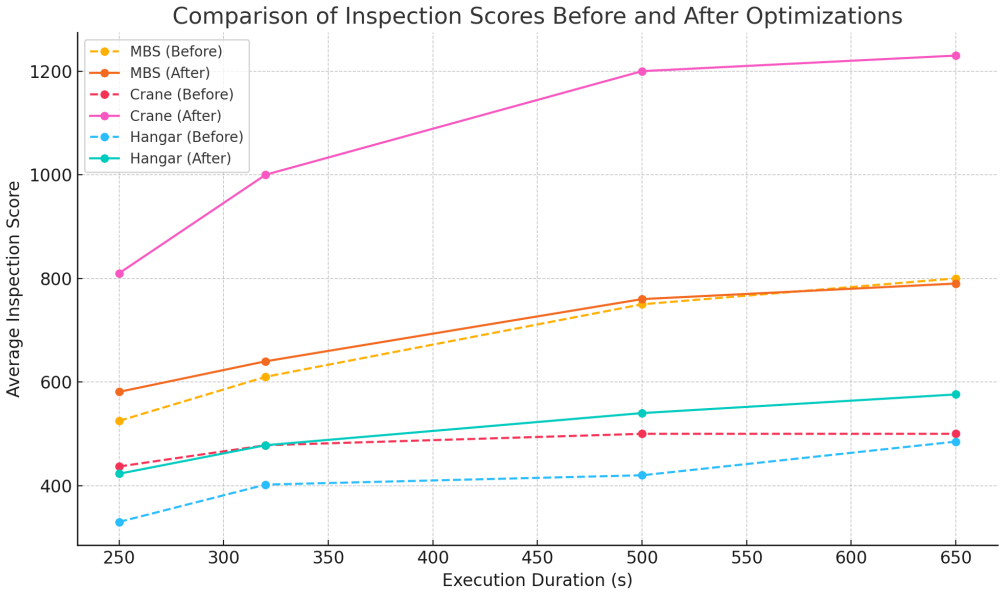

Figure 5.4: Comparison of inspection scores before and after general performance optimizations.

Comparison of Detected Interest Points

The following tables summarize the number of detected interest points successfully covered by the UAVs at each execution duration. The comparison includes both the baseline implementation and the optimized system that incorporates interest-aware planning strategies.

Table 5.3: Detected Interest Points before General Performance Optimizations

Scenario	250s	320s	500s	650s
MBS	860	1036	1176	1280
Crane	688	787	820	820
Hangar	587	689	720	779

Table 5.4: Detected Interest Points after the General Performance Optimizations

Scenario	250s	320s	500s	650s
MBS	1002	1128	1247	1260
Crane	1398	1826	2050	2100
Hangar	661	728	800	886

The results demonstrate a consistent increase in the number of detected interest points successfully covered by the UAVs after the application of general performance optimizations. These gains are evident across all scenarios and execution durations.

- **MBS Scenario:** The number of detected interest points improved modestly across all durations. While the pre-optimization performance was already relatively high, the optimized system offered noticeable gains e.g., from 1176 to 1247 at 500s. Interestingly, a small performance plateau is observed at 650s, suggesting that most interest points had already been covered by that stage.
- **Crane Scenario:** This scenario shows the most substantial improvement. The optimized implementation significantly outperformed the baseline, particularly at longer durations. For instance, the number of detected points increased from 820 to 2050 at 500s. This sharp increase illustrates the benefit of enhanced navigation and interest-aware planning in more complex geometries.
- **Hangar Scenario:** Improvements were moderate but consistent across all durations. The number of detected points rose from 720 to 800 at 500s and from 779 to 886 at 650s. This suggests that even in relatively simpler environments, the optimizations yielded tangible benefits.

In summary, the integration of general performance optimizations especially those targeting interest-aware planning resulted in more effective and efficient coverage of relevant areas, enhancing the UAVs' mission outcomes in both complex and simple scenarios.

To complement the tabular data, Figure 5.5 presents a visual comparison of the detected interest points across all scenarios and execution durations. Each line represents one scenario (MBS, Crane, Hangar), with dashed lines corresponding to the baseline implementation and solid lines to the optimized one.

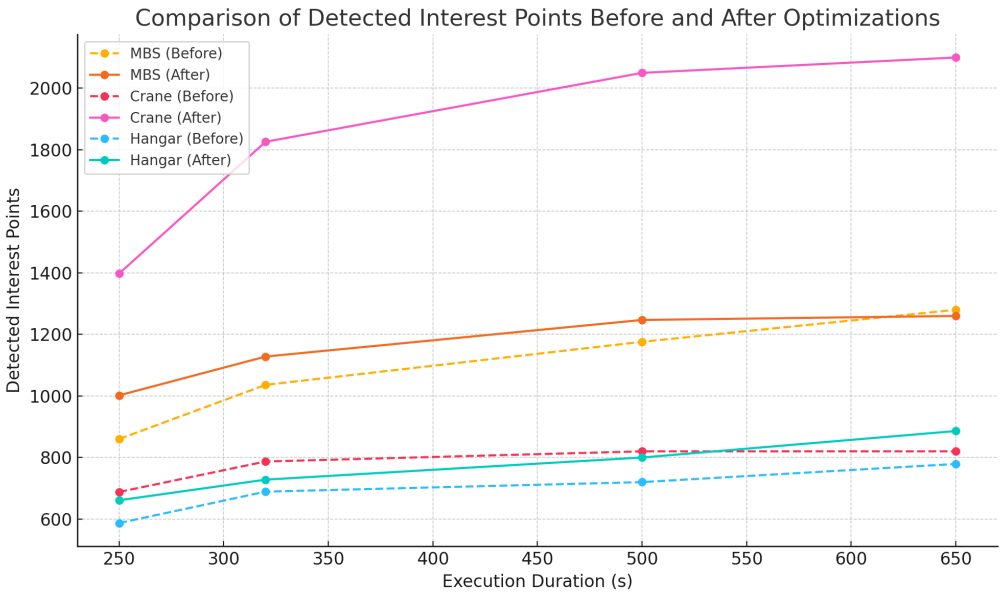

Figure 5.5: Comparison of detected interest points before and after general performance optimizations.

Comparison of UAV Collisions

The following tables summarize the number of UAV collisions recorded during each execution duration. Since collisions represent failures in coordination or spatial planning, fewer collisions generally indicate better system performance, improved trajectory planning, and more effective inter-agent cooperation.

Table 5.5: UAV Collisions before General Performance Optimizations

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	1	1	1	1
Crane	2	3	5	5
Hangar	1	1	1	2

Table 5.6: UAV Collisions after General Performance Optimizations

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	1	1	1	1
Crane	0	0	0	0
Hangar	0	1	1	1

The results indicate that the optimized implementation significantly reduced the number of UAV collisions, particularly in more complex environments.

- **MBS Scenario:** The number of collisions remained stable at one across all durations, both before and after the optimizations. This suggests that the scenario already had relatively low conflict potential due to open space or good baseline path planning.
- **Crane Scenario:** This environment showed the most dramatic improvement. Collisions dropped from as high as five (at 500s and 650s) to zero after optimizations. This clearly demonstrates the effectiveness of improved inter-agent coordination and collision avoidance strategies in complex geometries.
- **Hangar Scenario:** A moderate reduction in collisions was observed, especially in shorter durations. While some collisions still occurred at 320s and beyond, the overall reduction indicates more efficient spatial coordination among UAVs.

Overall, the reduction in collisions across scenarios confirms that the applied performance optimizations significantly enhanced the robustness and safety of multi-agent exploration missions.

To better illustrate the impact of the general performance optimizations on inter-agent coordination, Figures 5.6 and 5.7 show the number of UAV collisions per scenario, across all tested execution durations. Each scenario (MBS, Crane, Hangar) is represented with a different color, allowing for a clear comparison between the pre- and post-optimization states.

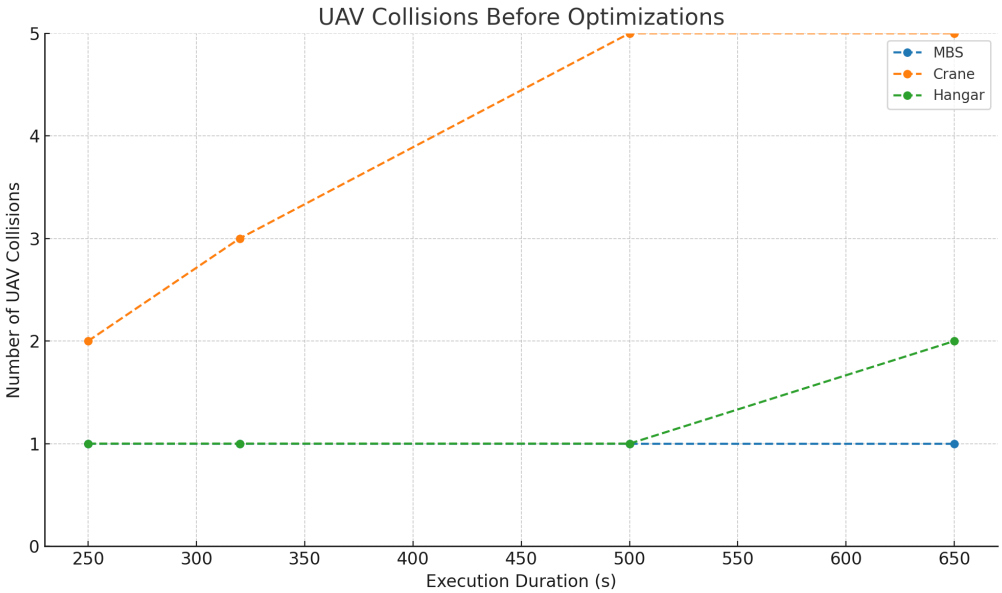

Figure 5.6: UAV collisions per scenario before applying performance optimizations.

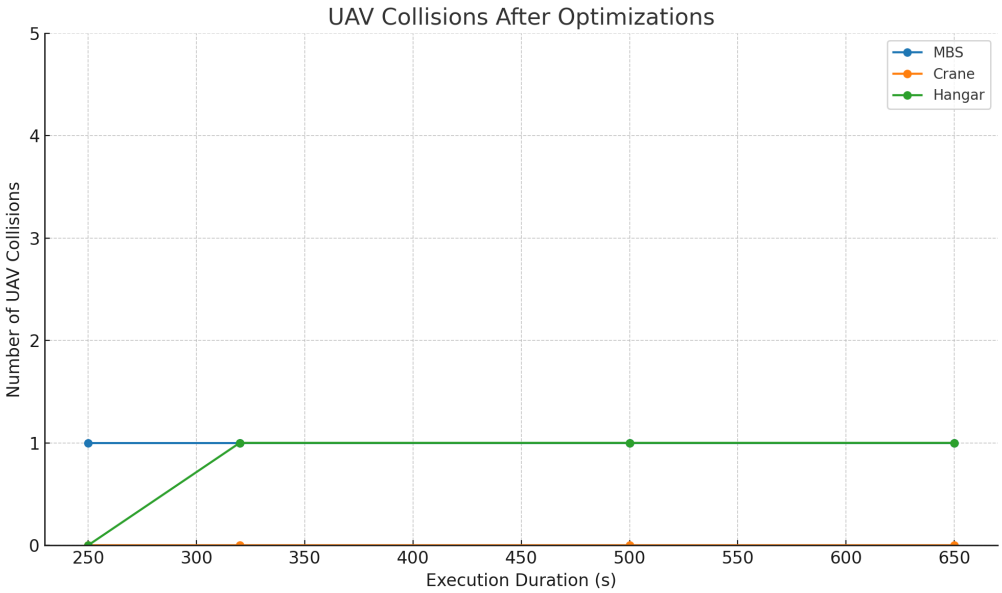

Figure 5.7: UAV collisions per scenario after applying performance optimizations.

Overall Conclusion

The comparative evaluation across all key performance indicators-final inspection scores, detected interest points, and number of UAV collisions-demonstrates that the applied general performance optimizations led to substantial improvements. Specifically, the optimized implementation consistently achieved higher inspection coverage, successfully detected more points of interest, and significantly reduced the number of collisions among UAVs. These enhancements confirm the effectiveness of the optimization strategies in increasing mission efficiency, safety, and coordination across diverse operational scenarios.

5.2.2 Exploration Optimized Planning Results

This section presents a comparison between the general performance optimized version and a more advanced implementation that integrates frontier-based exploration and an updated TSP algorithm. While the general optimization focused on improving system-level efficiency and coordination, the advanced planning version aims to enhance decision-making at the local exploration level through strategic goal selection and path refinement.

In the frontier-based approach, the main objective is to improve full-space coverage by selecting exploration goals at the boundary between known and unknown areas. This results in more accurate and complete mapping of the environment, enabling each UAV to better identify which voxels are free and which are occupied. Such enhanced spatial awareness significantly reduces the likelihood of collisions between agents, especially in cluttered or partially observable environments.

Moreover, the updated TSP algorithm prioritizes regions with higher potential for information gain, allowing UAVs to detect more interest points within the same execution duration. This planning refinement translates into better resource utilization and mission efficiency, especially when detecting sparse or scattered targets.

The following comparison evaluates the two approaches across multiple performance metrics including final inspection scores, detected interest points, and UAV collisions to assess the effectiveness of high-level planning enhancements on mission performance.

Comparison of Final Inspection Score

The following tables summarize the average final inspection scores reached per execution duration, comparing the general optimized implementation with the exploration-optimized planning approach that integrates frontier-based exploration and an improved TSP algorithm.

Table 5.7: Average Inspection Score with General Performance Optimizations

Scenario	250s	320s	500s	650s
MBS	581	640	760	790
Crane	810	1000	1200	1230
Hangar	423	478	540	576

Table 5.8: Average Inspection Score with Exploration-Optimized Planning Approach

Scenario	250s	320s	500s	650s
MBS	606	650	780	780
Crane	760	866	1020	1180
Hangar	527	545	597	627

The comparison between the general optimized version and the exploration-optimized planning approach reveals a mixed performance pattern across different scenarios.

- **MBS Scenario:** The inspection score remains relatively stable across all durations, with a small variation between the two approaches. This outcome is expected, as the frontier-based approach focuses more on accurate and safe exploration rather than aggressive coverage. In some cases, certain areas may be marked as unsuitable or unreachable due to strict voxel classification, which limits full inspection.
- **Crane Scenario:** The general optimized version outperforms the exploration-optimized approach in most durations. However, this difference can be attributed to the conservative nature of the frontier planner, which may avoid risky paths that the general version might take. Despite this, the exploration approach maintains competitive scores, while significantly improving safety and coordination.
- **Hangar Scenario:** Interestingly, the exploration-optimized approach surpasses the general one in all durations. This indicates that in simpler environments with less obstruction, the benefits of better spatial understanding and path selection lead to more effective inspection.

Overall, although the exploration-optimized approach may result in slightly lower or similar scores in some complex environments, this trade-off is justified by its emphasis on safety, structured exploration, and long-term coordination benefits.

To visually support the numerical comparison, Figure 5.8 presents the final inspection scores achieved by both the general optimized version and the exploration-optimized planning approach across all scenarios and durations. Dashed lines represent the general implementation, while solid lines correspond to the exploration-enhanced version. The comparison highlights how the exploration-focused strategy balances inspection performance with safer and more structured coverage.

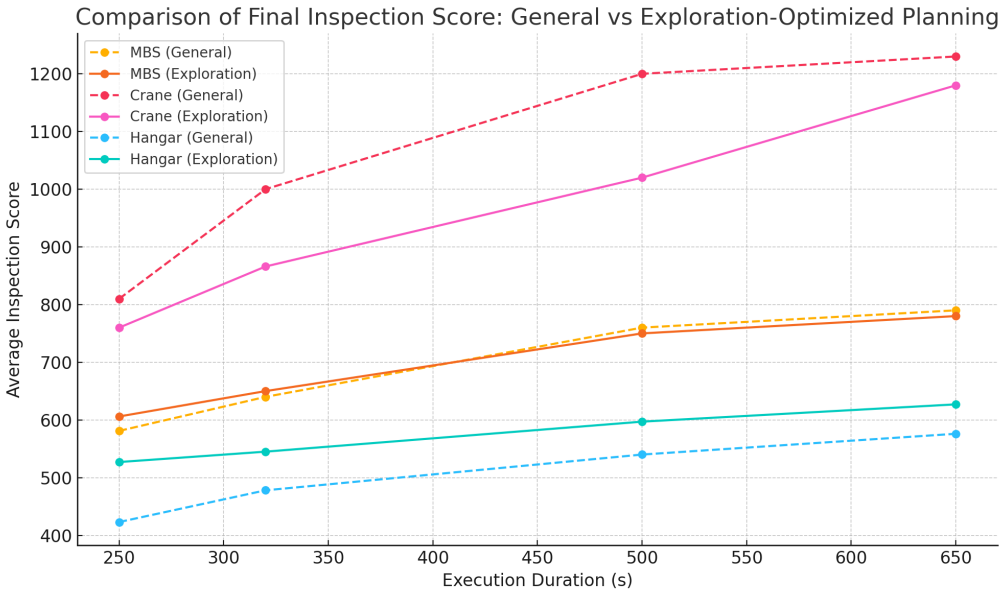

Figure 5.8: Comparison of final inspection scores between the general optimized version and the exploration-optimized planning approach.

It is also important to emphasize that although the inspection scores of the exploration-optimized approach are often comparable to those of the general optimized version, they are still significantly higher than the scores recorded before any optimizations were applied. This highlights the fact that the exploration-optimized planning not only preserves the benefits of the general improvements but also builds upon them, offering a safer and more structured exploration process without compromising overall mission performance. This observation is further supported by Figure 5.9, which presents a direct comparison between the baseline system and the exploration-optimized approach across all test scenarios and durations.

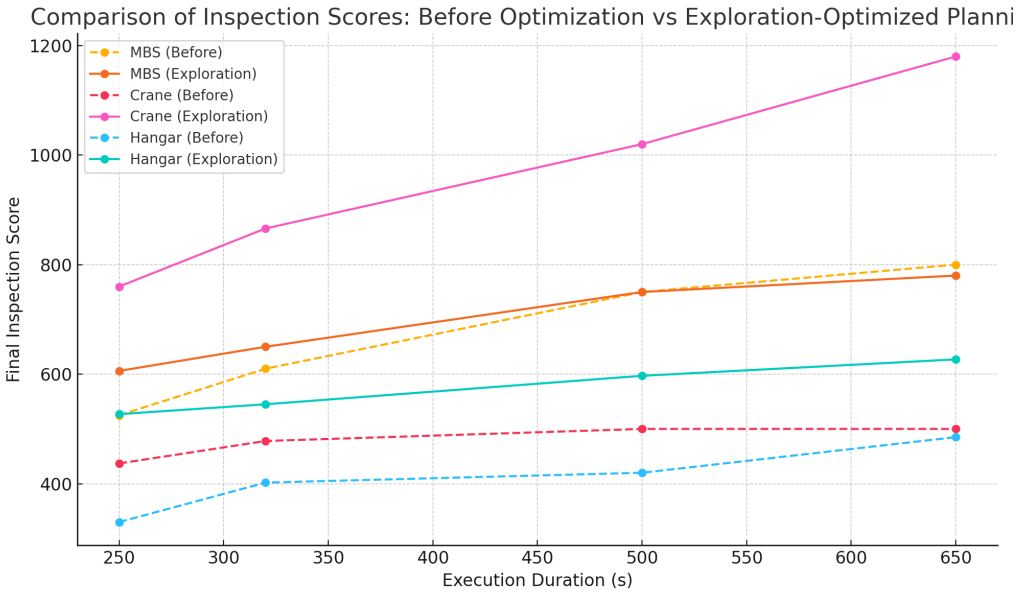

Figure 5.9: Comparison of final inspection scores between the baseline system (before optimization) and the exploration-optimized planning approach.

Comparison of Detected Interest Points

The following tables summarize the number of detected interest points successfully covered by the UAVs at each execution duration. The comparison is made between the general optimized implementation and the exploration-optimized planning approach, which integrates frontier-based mapping and an improved TSP algorithm to enhance coverage of high-interest areas.

Table 5.9: Detected Interest Points with General Performance Optimizations

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	1002	1128	1247	1260
Crane	1398	1826	2050	2100
Hangar	661	728	800	886

Table 5.10: Detected Interest Points with Exploration-Optimized Planning Approach

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	1118	1206	1280	1310
Crane	1684	2045	2336	2522
Hangar	920	963	1029	1034

The comparison highlights a significant improvement in the number of detected interest points when using the exploration-optimized planning approach, especially in the Crane and Hangar scenarios.

- **MBS Scenario:** A moderate improvement is observed across all durations, with the exploration-optimized version slightly outperforming the general one. This shows that even in relatively open environments, the advanced planning was able to target additional interest points.
- **Crane Scenario:** The largest gain is observed here, with over 400 additional points detected at 650 seconds compared to the general version. This considerable improvement is directly attributed to the integration of the upgraded TSP algorithm, which prioritized unexplored or high-interest areas more effectively, enabling UAVs to capture new data that was previously unreachable or deprioritized.
- **Hangar Scenario:** Similarly, a notable increase is recorded, particularly in longer durations. The improved path planning helped the UAVs systematically cover more interest-rich areas without overlapping paths.

Overall, these results confirm that the enhanced interest-aware planning and the refined TSP algorithm significantly boosted the UAVs' ability to detect more points of interest across diverse environments.

To support the tabular data, Figure 5.10 illustrates the detected interest points achieved by each method across all scenarios and durations. Dashed lines indicate the general optimized version, while solid lines represent the exploration-optimized planning approach.

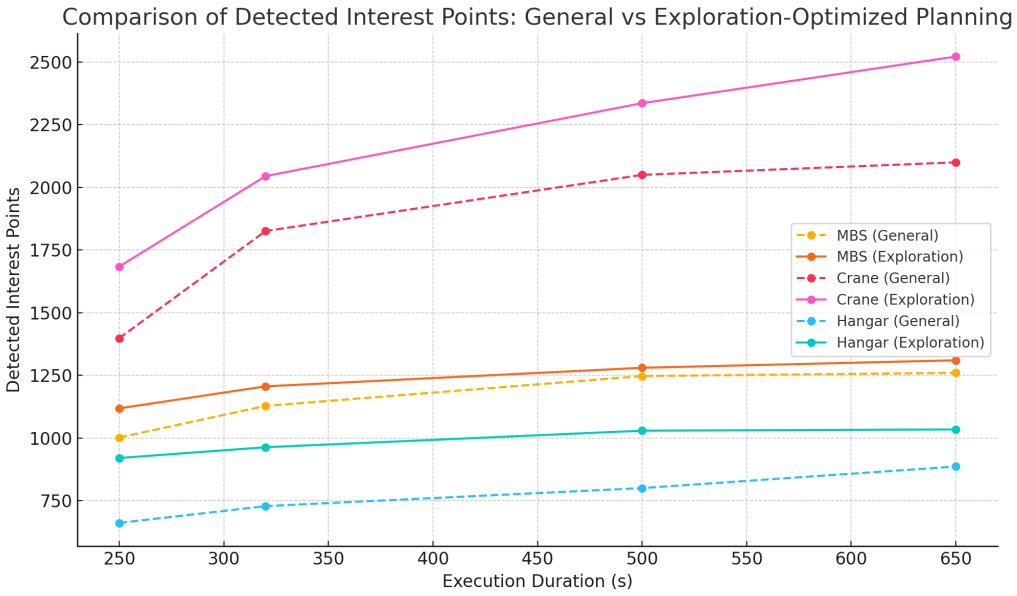

Figure 5.10: Comparison of detected interest points between the general optimized version and the exploration-optimized planning approach.

Comparison of UAV Collisions

The following tables summarize the number of UAV collisions recorded during each execution duration. The comparison is made between the general optimized implementation and the exploration-optimized planning approach, which leverages frontier-based mapping for safer navigation and an updated TSP algorithm for structured path planning. Reducing collisions is critical for ensuring mission reliability and safe multi-agent coordination.

Table 5.11: UAV Collisions with General Performance Optimizations

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	1	1	1	1
Crane	0	0	0	0
Hangar	0	1	1	1

Table 5.12: UAV Collisions with Exploration-Optimized Planning Approach

Scenario	**250s**	**320s**	**500s**	**650s**
MBS	0	0	0	0
Crane	0	0	0	0
Hangar	0	0	0	0

The collision data highlights a clear advantage of the exploration-optimized planning approach in ensuring safer navigation and improved inter-agent coordination.

- **MBS Scenario:** While the general optimized version recorded a constant one collision across all durations, the exploration-optimized approach managed to eliminate collisions entirely. This improvement demonstrates how even in relatively open environments, precise spatial reasoning can enhance safety.
- **Crane Scenario:** Both versions resulted in zero collisions, indicating that the environment itself may have been well-suited to the agents' paths. However, it is important to note that the exploration-optimized approach maintained this safety margin even while covering more interest points.
- **Hangar Scenario:** The most notable improvement is observed here. The general version showed occasional collisions, particularly at higher durations, while the exploration-optimized approach achieved complete elimination. This result is strongly linked to the frontier-based exploration method, which enabled more accu-

rate environmental mapping. By maintaining a clear distinction between free and occupied voxels, UAVs avoided potentially risky paths that could lead to conflicts.

In summary, the complete elimination of UAV collisions in the exploration-optimized approach confirms the effectiveness of frontier-based spatial awareness in enabling safer multi-agent operation.

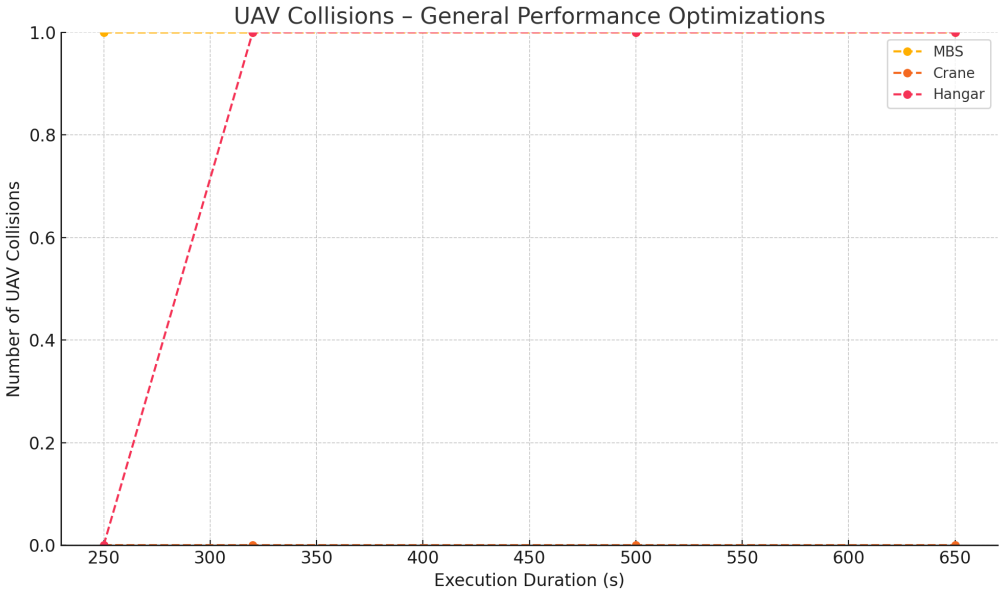

Figure 5.11: Number of UAV collisions per scenario using the general optimized version.

Overall Conclusion

The comparison between the General Optimized Version and the Exploration-Optimized Planning Approach highlights clear advantages introduced by the latter. Notably, the Exploration-Optimized approach achieved a substantial increase in the number of detected interest points and a significant reduction in UAV collisions. These improvements are primarily attributed to the integration of a frontier-based exploration strategy combined with a refined TSP solver for path planning. In addition, the overall inspection score experienced a slight but consistent improvement, reinforcing the effectiveness of the exploration-driven optimization in enhancing mission awareness, safety, and spatial coverage during inspection tasks.

6 Chapter 6

Conclusion

- Thesis Overview and Results
- Future work

6.1 Thesis Overview and Results

In this thesis, we addressed critical challenges in the efficient operation of UAV swarms for inspection and exploration missions. Initially, the implemented system exhibited considerable performance bottlenecks, characterized by prolonged execution times in critical functions and relatively low inspection scores, especially noticeable in geometrically complex scenarios. These inefficiencies significantly limited the UAVs' ability to complete missions effectively.

To overcome these issues, we introduced general performance optimizations aimed at refining system efficiency, improving computational resource management, and enhancing multi-agent coordination. These optimizations successfully addressed execution time problems and significantly boosted inspection scores across all scenarios. Notably, the Crane scenario experienced dramatic score improvements, highlighting the optimizations' effectiveness in challenging environments.

Further, recognizing persistent issues with UAV collisions and insufficient detection of interest points, we implemented a comprehensive exploration-optimized planning approach. This approach combined frontier-based exploration techniques with an upgraded TSP algorithm. The frontier-based method improved the accuracy and completeness of environment mapping, thereby dramatically reducing collision incidents by clearly distinguishing occupied and free voxels. Concurrently, the enhanced TSP algorithm strategically guided UAVs towards areas with higher interest point densities, significantly increasing the number of detected points compared to previous implementations.

These improvements not only optimized mission outcomes regarding data acquisition and safety but also ensured more reliable and systematic UAV swarm operations. By effectively eliminating collisions and enhancing inspection performance, the final system represents a robust and scalable solution for complex exploration missions involving multiple UAVs.

6.2 Future Work

Future work will focus on further enhancing and validating the current system through several key improvements and extensions:

- Scalability assessment of the system with a larger number of UAVs and extended operational areas. Further experimentation with additional environmental complexities and multi-sensor integration should be conducted to validate and improve robustness.
- Incorporating UAVs with different flight dynamics and sensor payloads in testing to refine the system's behavior and improve generalization across hardware platforms.
- Making the source code and system documentation publicly available, including detailed installation and setup guides, to facilitate community engagement and collaborative improvements.

These future enhancements will further establish the system as a practical, efficient, and widely applicable tool for UAV-based exploration and inspection missions.

Finally, the source code for the implementations developed in this thesis is available at the following Git repository:

- https://github.com/zinonas79/diplomatic.git

The repository contains three distinct branches, each corresponding to a different stage of the development process:

- `version1` – This branch includes the baseline implementation, which serves as the reference version without performance enhancements.
- `version2` – This branch incorporates general performance optimizations.
- `master` – The main branch contains the final version of the code, featuring the optimized planning strategy.

BIBLIOGRAPHY

[1] P. Kolios, "Caric challenge – cooperative aerial robot inspection," 2023, accessed: 2025-05-08. [Online]. Available: https://www.kios.ucy.ac.cy/pkolios/?p=679

[2] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Towards automated 3d search planning for emergency response missions," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 1, p. 2, 2021.

[3] A. Farooq, A. Anastasiou, N. Souli, C. Laoudias, P. S. Kolios, and T. Theocharides, "Uav autonomous indoor exploration and mapping for sar missions: Reflections from the icuas 2022 competition," in *Proceedings of the 2022 19th International Conference on Ubiquitous Robots (UR)*, 2022, pp. 621–626.

[4] L. Lin and M. A. Goodrich, "Uav intelligent path planning for wilderness search and rescue," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 709–716.

[5] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[6] L. Lin and M. A. Goodrich, "Re-routing uavs in the wild: Preemptive path planning for efficient wilderness search," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3138–3143.

[7] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multirotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2135–2142.

[8] C. I. Connolly, "The determination of next best views," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1985, pp. 432–435.

[9] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 417–433, 1993.

[10] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[11] B. Tovar, L. Muñoz-Gómez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy, and S. Hutchinson, "Planning exploration strategies for simultaneous localization

and mapping," *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 314–331, 2006.

[12] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*. IEEE, 1997, pp. 146–151.

[13] ——, "Frontier-based exploration using multiple robots," in *Proceedings of the Second International Conference on Autonomous Agents*. ACM, 1998, pp. 47–53.

[14] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[15] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the efficiency of frontier-based exploration strategies," in *Proceedings of the ISR/ROBOTIK Conference*. VDE Verlag, 2010, pp. 36–43.

[16] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012.

[17] M. Juliá, □. Reinoso, A. Gil, M. Ballesta, and L. Payá, "A hybrid solution to the multi-robot integrated exploration problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 3, pp. 473–486, 2010.

[18] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4557–4564.

[19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[20] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an mav," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9570–9576.

[21] E. Vespa, N. Funk, P. H. J. Kelly, and S. Leutenegger, "Adaptive-resolution octree-based volumetric slam," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 654–662.

[22] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang, and Y. W. Yun, "An improved frontier-based approach for autonomous exploration," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 292–297.

[23] M. Faria, I. Maza, and A. Viguria, "Applying frontier cells based exploration and lazy theta* path planning over single grid-based world representation for autonomous inspection of large 3d structures with an uas," *Journal of Intelligent & Robotic Systems*, vol. 93, pp. 113–133, 2019.

[24] M. Scott and K. Jerath, "Multi-robot exploration and coverage: Entropy-based adaptive maps with adjacency control laws," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4403–4408.

[25] E. Palazzolo and C. Stachniss, "Effective exploration for mavs based on the expected information gain," *Drones*, vol. 2, no. 1, p. 9, 2018.

[26] E. Kaufman, K. Takami, Z. Ai, and T. Lee, "Autonomous quadrotor 3d mapping and exploration using exact occupancy probabilities," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 49–56.

[27] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3d object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[28] M. Faria, A. S. Ferreira, H. Pérez-Leon, I. Maza, and A. Viguria, "Autonomous 3d exploration of large structures using an uav equipped with a 2d lidar," *Sensors*, vol. 19, no. 22, p. 4849, 2019.

[29] A. Ribeiro and M. Basiri, "Efficient 3d exploration with distributed multi-uav teams: Integrating frontier-based and next-best-view planning," *Drones*, vol. 8, no. 11, p. 630, 2024.

[30] L. Bartolomei, L. Teixeira, and M. Chli, "Fast multi-uav decentralized exploration of forests," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5576–5583, 2023.

[31] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Automated real-time inspection in indoor and outdoor 3d environments with cooperative aerial robots," *Applied Sciences*, vol. 10, no. 24, p. 8836, 2020.