

Multi-Drone Coordinated Image Sampling for 3D Map Reconstruction through Efficient and Scene-Adaptive Coverage Control

Doctoral Dissertation by

Muhammad Hanif

Department of Systems and Control Engineering
School of Engineering
Institute of Science Tokyo



Written under the supervision of
Professor Takeshi Hatanaka

June, 2025

Copyright © 2025
Muhammad Hanif.
All Rights Reserved.

Abstract

This dissertation investigates coordinated image sampling for 3D map reconstruction using multi-drone systems. To address this, a so-called angle-aware coverage control strategy suitable for 3D reconstruction has been presented, based on the notion of constraint-driven control in the literature, but this strategy has two fundamental problems. First, the conventional method can lead to inefficient operations, especially in larger fields, where drones travel long distances to visit small unsampled areas. More importantly, the measurement granularity must be specified in advance, although the required image density for high-precision 3D reconstruction depends on the unknown scene. To overcome these limitations, this dissertation presents novel control frameworks that improve both efficiency and scene-adaptivity.

We first propose a method to enhance the efficiency of multi-drone image sampling for 3D reconstruction. The key idea is to integrate a Voronoi-based coverage control law as the nominal input into the quadratic programming (QP) problem used to solve the original angle-aware control. This approach retains the viewpoint diversity promoted by angle-aware coverage while improving mission completion efficiency through effective coverage of distant and unobserved areas, particularly in large-scale scenarios.

Moreover, with recent advances in real-time 3D reconstruction algorithms, it becomes possible to adapt drone motion using feedback from the evolving scene. We extend the coverage framework to integrate real-time outputs from NeuralRecon, which processes captured images to generate an evolving 3D mesh. Mesh changes are quantified and used to adaptively update an importance index, serving as a coverage priority metric to guide drones toward regions needing further observation. To handle this time-varying objective while ensuring safety and coordination, we design a QP-based controller that constrains the decay rate of the objective and enforces safety constraints such as collision avoidance and workspace limits.

The proposed methods are validated through simulations in Unity and ROS2, and indoor and outdoor experiments using real UAVs. For the efficient coverage method, comparative analysis demonstrates improved monitoring efficiency for large-scale 3D reconstruction. Additionally, the scene-adaptive method outperforms the conventional approach without feedback, yielding more complete and accurate 3D maps. These results confirm the practicality of both methods and their potential to support real-time, scalable 3D reconstruction using multi-drone systems.

Acknowledgment

All praise is due to Allah Ta'ala for His countless blessings that enabled me to complete this study and dissertation. Peace and blessings be upon the Prophet Muhammad, whose teachings continue to inspire us to seek knowledge and bring benefit to our surroundings and humankind. I sincerely hope that this work will bring benefit to myself, those around me, and humankind in general. May it serve as a small contribution in the pursuit of knowledge and positive impact.

I would like to express my deepest gratitude to my beloved parents, Naimah and Farhat Umar, who have raised me with care, love, and unwavering support. Their values, prayers, and lifelong dedication have been the foundation of everything I am. This Ph.D. is especially inspired by my father's own academic journey—despite completing his dissertation, he was unable to receive his doctoral degree. In this regard, my pursuit of the Ph.D. is not only a personal milestone but a continuation of his noble effort and legacy. I am also truly grateful to my dear sister, Hanan Talida, who has been a constant source of warmth, encouragement, and companionship throughout my life. To my life partner, Salsabilla, thank you for your patience, sacrifice, and enduring strength during the ups and downs of this journey. Your presence has been a source of balance and resilience. To my beloved daughter, Humaira, I dedicate this milestone with the hope that it may one day serve as a source of benefit and inspiration. I also extend my heartfelt thanks to all my family members for their love, prayers, and unwavering belief in me.

I wish to express my sincere appreciation to my supervisor, Professor Takeshi Hatanaka, for his patient guidance, insightful mentorship, and constant encouragement. You have been like a father figure to me in the world of control theory. Your commitment to academic excellence, openness to ideas, and humility have left a profound impression on me. I have learned not only about research, but also about integrity and perseverance under your guidance. I deeply hope for continued blessings, good health, and success in your life and career.

I would also like to thank all professors and faculty members at Tokyo Tech who have taught, challenged, and supported me throughout my studies. Special thanks to Nishiyama-san and Suzuki-san for their tireless administrative support and kind assistance

in both daily matters and critical moments—I am sincerely grateful for your presence.

My deepest thanks go to the WISE Program for Super Smart Society (SSS), which provided an intellectually stimulating and multidisciplinary platform that greatly enriched my doctoral journey. I am especially grateful to Professor Kei Sakaguchi, Professor Nishihara, and the entire WISE SSS team, including Sumie Maruyama, Kuniko Yasumi, Yukiko Tomita, and many others, for their dedication and inspiration throughout the program.

I am fortunate to have had the support and friendship of my laboratory colleagues—both seniors and juniors—whose discussions, feedback, and companionship have been invaluable throughout this journey. Special thanks to Dan-san, Yamashita-san, and Wildan-san, who were sources of inspiration and guidance when I first joined the lab. I am also grateful to Mengmou, Daniel, and Javi for their insightful discussions and encouragement, which helped sharpen my research skills.

I would also like to thank my other labmates—Okada-san, Lu-san, Shimizu-san, Sumino-san, Suenaga-san, Oshima-san, Toyomoto-san, Mochizuki-san, Tanaka-san, Takizawa-san—and many others for the moments we shared, both in research and beyond.

Lastly, I would like to express my heartfelt appreciation to everyone who has supported and encouraged me, directly or indirectly, during this endeavor. Your presence—whether through words, actions, or simply by being there—has meant more to me than I can express. Thank you.

Table of Content

Ch. 1	Introduction	1
1.1	Motivation	1
1.2	Related Work	4
1.3	Contributions	7
1.4	Summary of Key Assumptions	9
1.5	Outline of the Dissertation	11
1.6	Notations	12
Ch. 2	Preliminaries: Coverage Control & 3D Reconstruction	14
2.1	Introduction	14
2.2	Control Barrier Function (CBF)-Based Control	14
2.3	Coverage Control	16
2.3.1	Classical Coverage Control	16
2.3.2	Persistent Coverage Control	18
2.3.3	Angle-Aware Coverage Control	20
2.3.4	Rotatable Angle-Aware Coverage Control	23
2.4	3D Reconstruction	25
2.4.1	Offline Reconstruction Techniques	27
2.4.2	Online Reconstruction Techniques	29
2.5	Chapter Summary	32
Ch. 3	Efficient Angle-Aware Coverage Control	35
3.1	Introduction	35
3.2	Problem Formulation	35
3.3	Baseline Controller Method & Drawback	37
3.3.1	Baseline Angle-Aware Coverage Controller	37
3.3.2	Drawback in Large-Scale Map Reconstruction	38
3.4	Controller Design for Large-Scale Map Reconstruction	41
3.5	Demonstration on a ROS Simulation	44

3.5.1	Evaluation for a Single-Drone Case	44
3.5.2	Evaluation for a Multi-Drone Case	45
3.6	Experiments: Indoor and Outdoor Validation	48
3.6.1	Indoor Experiment	48
3.6.2	Outdoor Experiment	51
3.7	Chapter Summary	52
Ch. 4	Scene-Adaptive Angle-Aware Coverage Control	56
4.1	Introduction	56
4.2	Problem Formulation	57
4.2.1	Drones, Virtual Field, and Geometry	57
4.2.2	Sensing Performance and Importance Index	59
4.3	Coordinated Image Sampling with Real-Time Map Feedback	60
4.3.1	Quantification of Mesh Changes	60
4.3.2	Importance Index Update Based on Mesh Feedback	61
4.3.3	QP-Based Controller Design	62
4.3.4	Control Logic with Feedback Trigger	66
4.4	Simulation-Based Evaluation	67
4.4.1	Single-Drone Simulation	68
4.4.2	Multi-Drone Simulation	70
4.5	Experiments: Online Reconstruction Validation	70
4.5.1	Experimental Setup	70
4.5.2	Experimental Results	72
4.6	Chapter Summary	73
Ch. 5	Conclusion	77
5.1	Summary of Contributions	77
5.2	Limitations and Future Work	78
	References	80
	Publications Related to This Research	92

List of Figures

1.1	Collaborative 3D Map Reconstruction using Drone Networks Scene	5
2.1	Control Barrier Function (CBF) Illustration	15
2.2	Comparison of Coverage Control Strategies	16
2.3	Classical Coverage Control Illustration	17
2.4	Persistent Coverage Control Illustration	20
2.5	Angle-Aware Coverage Control Illustration	22
2.6	Rotatable Angle-Aware Coverage Control Illustration	23
2.7	Illustration of SfM-MVS Pipeline	27
2.8	NeuralRecon-based Online 3D Reconstruction Pipeline	29
2.9	Comparison between NeRF and 3D Gaussian Splatting	31
3.1	Illustration of the Angle-Aware Coverage Problem	36
3.2	Control Architecture for Baseline Angle-Aware Coverage Controller	39
3.3	ROS Simulation of Baseline Controller	40
3.4	Plot for Cost function J using Baseline Controller	42
3.5	Control Architecture for Efficient Angle-Aware Coverage Controller	43
3.6	Comparison of Importance Index ψ Evolution (Single-Drone Simulation) .	46
3.7	Comparison of Cost Function J Evolution (Single-Drone Simulation) . . .	46
3.8	Comparison of Importance Index ψ Evolution (Multi-Drone Simulation) .	47
3.9	Comparison of Cost Function J Evolution (Multi-Drone Simulation) . . .	47
3.10	Overview and Schematic Diagram of the Experimental Testbed	49
3.11	Comparison of Importance Index ψ Evolution (Indoor Experiment)	50
3.12	Comparison of Cost Function J Evolution (Indoor Experiment)	50
3.13	Overview of the Experimental Farmland (Smart Agriculture R&E Field) .	51
3.14	System Architecture for the Outdoor Experiment	52
3.15	Comparison of Importance Index ψ Evolution (Outdoor Experiment) . . .	53
3.16	Comparison of Cost Function J Evolution (Outdoor Experiment)	53
3.17	3D Map Reconstructed using Proposed Method	55

3.18	3D Map Reconstructed using Baseline Method	55
4.1	Illustration of the Angle-Aware Coverage Problem with Camera Rotation	57
4.2	Geometric Relation between a Drone p_i and Point q_j	58
4.3	Controller Architecture of Scene-Adaptive Coverage Control	63
4.4	Snapshot of the Unity Scene Targeted for 3D Reconstruction	66
4.5	Simulation Snapshot of Scene-Adaptive Coverage Control using Single Drone	69
4.6	Plot of J for the Single-Drone Performance Evaluation	69
4.7	Comparison between Reconstructed Map for Single-Drone Evaluation . .	70
4.8	Simulation Snapshot of Scene-Adaptive Coverage Control using Multi-Drone	71
4.9	Plot of J for the Multi-Drone Performance Evaluation	71
4.10	Comparison between Reconstructed Map for Multi-Drone Evaluation . . .	72
4.11	Indoor Experimental System Architecture	73
4.12	Indoor Experimental Setup for Scene-Adaptive Coverage Control	75
4.13	Snapshot of the Experiment with Single Drone	76

List of Tables

1.1	List of Frequently Used Notations	13
3.1	Simulation Parameters for Baseline Angle-Aware Evaluation	41
3.2	Simulation Parameters for Single-Drone Evaluation	44
3.3	Simulation Parameters for Multi-Drone Evaluation	45
3.4	Experimental Parameters for Indoor Validation	51
3.5	Experimental Parameters for Outdoor Evaluation	54
4.1	Simulation Parameters for 5D Coverage with Real-Time Map Feedback .	68
4.2	Indoor Simulation Parameters for Scene-Adaptive Coverage	74

Chapter 1

Introduction

1.1 Motivation

Multi-robot systems (MRS) have emerged as a pivotal concept in robotics, enabling multiple autonomous agents to collaborate effectively on complex tasks while enhancing efficiency, scalability, and robustness across various applications [1]. One prominent application area is precision agriculture [2–4], where multi-robot teams enable coordinated tasks such as weed detection [5], crop health monitoring [6], and harvesting [7, 8]. MRS are also widely adopted in environmental monitoring and exploration, supporting tasks like gas sensing in hazardous zones [9] and solar radiation measurement in thermosolar facilities [10]. In disaster response [11, 12], these systems prove valuable for applications such as wildfire tracking, where drone teams can rapidly scan forested areas to detect and monitor fire outbreaks [13, 14], as well as search and rescue missions in collapsed or dangerous environments [15]. Furthermore, in the context of infrastructure inspection, coordinated robots can perform efficient and thorough assessments of bridges, buildings, and utility lines, reducing the risks and limitations associated with single-agent or manual inspections [16, 17].

Another impactful application of MRS is three-dimensional (3D) map reconstruction, where unmanned aerial vehicles (UAVs) can collaboratively capture images from multiple viewpoints to construct high-quality 3D models for scene analysis and planning [18–20]. Particularly in this context, the coordination of UAVs during the image sampling process plays a critical role in achieving accurate and reliable reconstruction outcomes. To achieve high-quality reconstruction, at least two key factors must be considered. First, each part of the scene must be observed from multiple viewpoints to enhance both accuracy and robustness of the reconstructed model [21]. This requires sufficient overlap between camera views, which in turn demands coordinated planning within a five-dimensional space that captures both the drone’s position and camera orientation [22, 23]. Second, the

one-time visit constraint must be addressed [24], which involves systematically acquiring images such that each point within the target environment is effectively covered during a single mission.

To address these challenges, the existing literature has explored various motion planning strategies for image sampling. Approaches such as viewpoint selection [25, 26] and trajectory optimization [27] aim to maximize information gain or minimize reconstruction uncertainties by selecting informative viewpoints or refining drone trajectories. However, these methods typically rely on partial or prior map knowledge and are often implemented through centralized planning, limiting their scalability. As a distributed alternative, coverage control has been extensively studied for coordinating mobile agents in continuous spaces. Classical coverage control based on centroidal Voronoi partitioning [28, 29] offers a decentralized framework with theoretical guarantees on convergence, making it particularly attractive for collaborative sensing tasks under limited prior knowledge.

Beyond classical formulations, coverage control has also been extended to handle environments with irregular or complex shapes [30–35]. However, these methods typically operate in lower-dimensional spaces and do not consider coverage in a 5D search space that includes both position and camera orientation. In addition, many of these strategies tend to converge to static configurations, making them less effective for achieving complete coverage across the target field. Persistent coverage control offers an alternative by enabling agents to patrol the environment over time [36–39], but these approaches are generally limited to two-dimensional coverage and do not account for viewpoint angles, which are critical for high-quality 3D reconstruction.

To better accommodate the requirements of 3D map reconstruction, Shimizu et al. [24] introduced an angle-aware coverage control strategy that explicitly takes into account both spatial position and camera orientation in the coverage formulation. By modeling the problem in a five-dimensional domain, consisting of a three-dimensional distribution of target points and a two-dimensional space of viewing angles, this method enables drones to capture images from a wider range of informative perspectives. Subsequent experiments by Suenaga et al. [40] validated the effectiveness of the method, demonstrating improved reconstruction quality compared to conventional persistent coverage control approaches [39]. However, in large-scale map reconstruction scenarios, the previous controller often struggles to quickly reach distant unobserved areas, leading to inefficiencies in overall monitoring performance. This limitation becomes increasingly pronounced as the size of the target field increases.

Building on this foundation, subsequent work extended the angle-aware coverage strategy [24] to incorporate a gimbal-mounted camera, enabling explicit control of camera

orientation independently from the drone’s body frame [41, 42]. While the original formulation assumed that the drone’s camera orientation was fixed relative to its heading direction, the extended system allows each drone to dynamically adjust its viewing direction to better capture informative angles. In this formulation, the target is represented in a 5D space, consisting of spatial location and desired viewing angle, while the drone’s control input lies in a 4D space: two dimensions for its planar movement and two for the gimbal’s orientation. The drone is assumed to fly at a fixed altitude. This added degree of angular flexibility allows the system to observe a broader set of target viewpoints compared to the previous fixed-orientation method [24, 43].

Despite recent progress in coverage control strategies, the effectiveness of 3D reconstruction missions remains constrained by the limitations of the underlying reconstruction pipeline. Traditional methods such as Structure from Motion (SfM) and Multi-View Stereo (MVS) [44, 45] are inherently offline, requiring a full set of images before reconstruction can begin. As a result, the system cannot assess map quality or adapt its sensing strategy during flight. Previous work, such as Plan3D [46] and CEO-MLCPP [47], address this limitation through multi-stage planning strategies. In these approaches, UAVs perform an initial mission to generate a sparse reconstruction, followed by trajectory optimization and a second mission to capture additional images. While this improves the resulting model quality, it requires multiple deployments and significantly increases operational time and cost.

Recent advancements in learning-based 3D reconstruction algorithms have introduced a new paradigm that enables online and iterative map generation. Techniques such as NeuralRecon [48], VisFusion [49], Instant-NGP [50], and Gaussian Splatting [51] leverage deep neural networks and streaming visual data to incrementally update 3D maps in near real time. These methods allow the reconstruction process to evolve dynamically as new observations are collected, making it possible to monitor the quality of the map and adjust the sensing strategy accordingly. Compared to classical SfM-MVS pipelines, this offers a significant advantage for applications requiring in-flight adaptability, particularly in time-constrained or resource-limited missions.

Several recent studies have explored the use of real-time 3D reconstruction as online feedback to improve robot motion planning. SplatNav [52] and SAFER-Splat [53] utilize reconstructed map information as online feedback to enable safe and informed navigation during mapping missions. Other frameworks, including NeurAR [54], RT-GuIDE [55], GS-Planner [56], and HGS-Planner [57], incorporate reconstruction feedback to guide view selection and improve trajectory planning. While these approaches highlight the growing potential of integrating online mapping with planning, extending such feedback-

driven mechanisms to multi-robot coverage control frameworks remains a promising yet underexplored direction.

In summary, using multi-drone systems for coordinated image sampling in 3D reconstruction is an impactful application that can facilitate more effective and scalable data collection (see Illustration 1.1). However, this task presents two key challenges: achieving efficient mission completion, particularly in large-scale environments, and enabling scene-adaptivity that responds to reconstruction progress. In particular, existing coverage strategies, such as the angle-aware method by Shimizu et al. [24], often struggle to efficiently reach distant unobserved areas in large-scale reconstruction scenarios. This inefficiency in overall monitoring performance becomes more pronounced as the size of the target environment increases. Furthermore, most existing strategies operate without feedback from the reconstruction process, making them unable to adjust their sensing based on map quality. Recent developments in real-time 3D reconstruction algorithms, such as NeuralRecon [48] and related methods, offer a promising opportunity to address this gap by enabling systems to adapt their coverage based on online reconstruction quality. These challenges and opportunities form the foundation for the problem explored in this dissertation.

1.2 Related Work

The research presented in this dissertation builds upon three key areas in robotics and computer vision: coverage control for multi-robot systems, classical and learning-based 3D reconstruction pipelines, and recent efforts integrating real-time map feedback into robotic planning. This section reviews the evolution of each field and their convergence in addressing adaptive and coordinated image sampling for multi-drone systems.

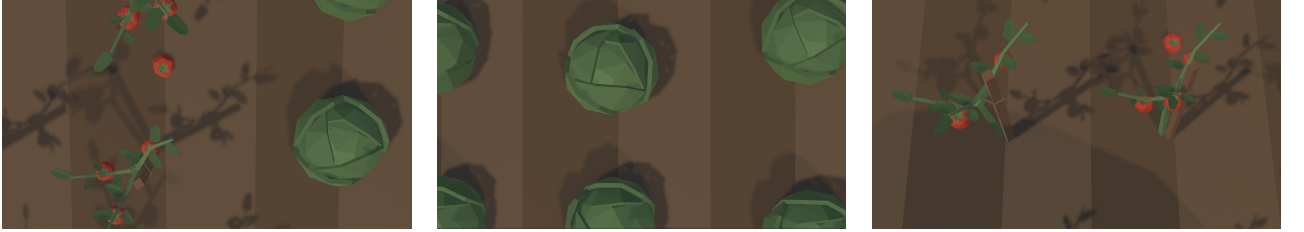
Coverage Control in Multi-Robot Systems

Coverage control is a foundational topic in cooperative robotics, concerned with deploying multiple agents to efficiently monitor spatial domains. A seminal contribution by Cortés et al. [28] introduced a distributed Voronoi-based approach that laid the groundwork for numerous extensions, including methods that account for dynamic environments [33], anisotropic sensing [58], and aerial applications [29, 59].

Persistent coverage control emerged to address scenarios requiring repeated observation over time. Notable works such as [37, 39, 60, 61] incorporate spatiotemporal dynamics through importance decay models and constraint-based formulations to ensure long-term monitoring. These strategies have been applied in surveillance, environmental monitoring,



(a) 3D Map Reconstruction Scene



(b) Image captured by each drone

Fig. 1.1: Collaborative 3D map reconstruction using drone networks scene. This image is adapted from [41].

and search tasks [62, 63].

To improve data quality for vision-based applications, angle-aware coverage control has been proposed [24, 40]. This formulation considers both spatial location and camera orientation. It has been further extended into rotatable angle-aware strategies [41], where drones actively control their viewpoints using gimballed cameras to achieve more comprehensive coverage for 3D reconstruction tasks.

These three categories of coverage control methods, namely classical, persistent, and angle-aware, form the core theoretical foundation of this dissertation. A more detailed explanation, including their mathematical formulations and representative simulations, is provided in Chapter 2.

Offline 3D Reconstruction and Multi-Drone Image Sampling

Classical 3D reconstruction pipelines are primarily based on Structure-from-Motion (SfM) and Multi-View Stereo (MVS) techniques [44, 45, 64]. These approaches have been implemented in widely used open-source tools such as COLMAP [65], openMVG [66], and Meshroom [67]. Early efforts also leveraged feature-based methods such as SIFT [68] and bundle adjustment optimization [69] to estimate camera poses and sparse geometry.

To improve the effectiveness of image collection for these offline pipelines, several works have explored multi-robot coordination and viewpoint planning. For instance, Plan3D [46] introduced a two-stage planning framework, while CEO-MLCPP [47] proposed an optimization-based controller for viewpoint selection across multiple flights. These methods have shown improvements in reconstruction quality, although they remain limited by their reliance on batch processing.

A detailed discussion of these classical reconstruction pipelines is provided in Chapter 2.

Real-Time 3D Reconstruction and Neural Approaches

More recently, learning-based approaches have enabled real-time or near real-time 3D reconstruction using streaming input data. Notable examples include NeuralRecon [48], TransformerFusion [70], and VisFusion [49], which employ neural architectures to incrementally fuse visual features into volumetric or mesh representations. These methods support continuous map updates and are increasingly integrated with robotics systems for feedback-based planning.

In parallel, neural rendering methods such as NeRF [71], Instant-NGP [50], and Gaussian Splatting [51] have advanced scene modeling by representing geometry as radiance fields or point-based primitives. While initially too slow for real-time use, recent variants have made online integration more practical. Additional technical details and a broader review of neural reconstruction methods are presented in Chapter 2.

Scene-Aware Planning and Real-Time Feedback Integration

The emergence of real-time neural reconstruction has enabled new planning strategies that incorporate scene feedback during operation. Several recent works have leveraged this capability to adaptively guide agent behavior based on evolving 3D maps, enabling more efficient and safe exploration.

For instance, SplatNav [52] utilizes Gaussian Splatting representations to inform drone

navigation by dynamically updating obstacle boundaries and traversability fields. Its follow-up work, SAFER-Splat [53], integrates online control barrier functions (CBFs) with obstacle detection directly from streaming splat updates. This allows drones to perform reactive, safety-aware planning and obstacle avoidance in previously unseen or dynamically changing environments.

Other approaches focus on uncertainty-aware exploration. NeurAR [54] introduces a planning framework that uses scene geometry and photometric uncertainty from neural fields to guide a UAV toward regions where the map is incomplete or uncertain. Similarly, RT-GuIDE [55] combines depth uncertainty and scene dynamics to compute a gradient field that encourages the robot to sample from novel or changing viewpoints, thereby improving map completeness over time.

Complementing these uncertainty-driven methods, GS-Planner [56] and HGS-Planner [57] propose strategies that directly optimize next-best-view selection based on the reconstructed scene structure. GS-Planner leverages the Gaussian Splatting representation to estimate the marginal value of candidate viewpoints in terms of reducing map uncertainty, while HGS-Planner enhances this with a hierarchical sampling approach to improve scalability and planning efficiency in larger environments.

Despite their strengths, most of these methods are designed for single-agent systems and are primarily focused on navigation or viewpoint planning tasks. Integration of real-time map feedback into coverage control frameworks remains underexplored, and scaling such strategies to multi-agent settings with distributed coordination presents a promising direction for future research.

1.3 Contributions

This dissertation addresses the problem of coordinated image sampling for 3D reconstruction using multi-drone systems, with a focus on improving mission efficiency and scene-adaptivity. To this end, two complementary contributions are presented.

The first contribution, detailed in Chapter 3, introduces an efficient angle-aware coverage control strategy to overcome the limited efficiency of the existing method [24]. The baseline strategy by Shimizu et al. [24] is first analyzed through simulation, revealing reduced efficiency in large-scale environments due to limited coordination capabilities. To address this issue, a new quadratic programming (QP)-based controller is proposed, which incorporates classical Voronoi-based coverage control [29, 72] as a nominal input within the original angle-aware framework. This approach enables drones to coordinate their positions more effectively while maintaining the viewpoint diversity required for high-quality image sampling. The proposed method is validated through simulations and

real-world experiments in indoor and outdoor settings, and demonstrates improved mission completion efficiency compared to the baseline.

The second contribution, presented in Chapter 4, extends this framework to enable scene-adaptive behavior using real-time feedback from online 3D reconstruction. The method builds on angle-aware coverage with camera rotational motion control [41] and introduces two key extensions. First, the control input space is expanded from four to five dimensions by including altitude control alongside planar movement and gimbal-based orientation. Second, the sensing performance model is revised to more accurately reflect the contribution of each viewpoint. The reconstruction process is powered by NeuralRecon [48], which generates a dynamically evolving 3D mesh of the environment. Feedback from changes in the mesh is used to adjust the coverage density by updating an importance index, prioritizing regions with significant updates for additional observation. Since the environment is initially unknown, the system uses the magnitude of mesh updates as a proxy for reconstruction uncertainty, assuming that highly changing regions are insufficiently reconstructed and require further sensing. A QP-based controller is then designed to adapt the drones' positions and orientations accordingly, while ensuring performance decay constraints and maintaining safety. The method is validated through high-fidelity simulation using Unity with ROS2 and real-world experiments, demonstrating improved reconstruction quality compared to a baseline without map feedback.

The contributions are summarized as follows:

1. **Efficient Angle-Aware Coverage Control for Multi-Drone Image Sampling using Offline Reconstruction Pipelines (Chapter 3)**
 - Performance analysis of the baseline angle-aware strategy [24], revealing efficiency limitations in large-scale deployments.
 - A novel QP-based controller that integrates Voronoi-based coverage as nominal input to enhance spatial exploration while preserving viewpoint diversity.
 - Use of control barrier functions (CBFs) to ensure collision avoidance and boundary safety.
 - Simulation-based evaluation in ROS demonstrating improved mission completion time and scalability in both single- and multi-drone cases.
 - Real-world indoor and outdoor experiments validating the practical effectiveness of the proposed controller.
2. **Scene-Adaptive Angle-Aware Coverage with Camera Rotational Motion Control for Multi-Drone Image Sampling using Online Reconstruction Pipelines (Chapter 4)**
 - Extension of angle-aware coverage to include altitude control, resulting in a

five-dimensional control input space and updated sensing performance model.

- Integration with NeuralRecon [48] for real-time 3D reconstruction and mesh-based feedback.
- Feedback mechanism that quantifies mesh updates and adjusts coverage density via an importance index.
- QP-based control design that adapts to reconstruction progress while enforcing safety and performance decay constraints.
- Simulation-based validation in Unity and ROS2 showing improved reconstruction completeness and quality.
- Experimental validation using a multi-drone testbed confirming system feasibility and improved performance.

Together, these contributions advance the capabilities of multi-drone systems for 3D reconstruction by introducing scalable, efficient, and scene-adaptive coverage strategies grounded in principled control design and validated through extensive simulation and experimentation.

1.4 Summary of Key Assumptions

This dissertation is developed under several assumptions regarding both the robots and the environment in which they operate. These assumptions are intended to clarify the scope of the work and highlight the conditions under which the proposed methods are designed and evaluated. While the following provides a general overview of these assumptions, more specific and detailed conditions are described in the relevant chapters where each algorithm, simulation, or experimental validation is presented.

Robot Assumptions

- **Accurate Localization Available:** The proposed methods assume that each drone can obtain accurate and reliable position estimates throughout the mission. In indoor experiments, this capability is provided by a motion capture system that delivers high-frequency, high-precision position data. For outdoor scenarios, Real-Time Kinematic (RTK) Global Navigation Satellite System (GNSS) measurements are assumed to be available, ensuring centimeter-level positioning accuracy.
- **Awareness of Other Drones' Positions:** Each drone is assumed to have access to the positions of other drones in the team. In the simulation and experimental setup used in this dissertation, this is implemented using a semi-distributed archi-

texture in which a central node aggregates the position information from all agents and redistributes it, enabling each agent to be aware of the others' positions for coordination purposes.

- **Calibrated Sensor Model:** The cameras mounted on each drone are assumed to have known intrinsic parameters (e.g., focal length, principal point) and well-characterized field-of-view specifications. This calibration is performed before deployment and is assumed to remain fixed during operation, ensuring that image measurements can be reliably projected into the reconstruction framework.
- **Reliable Wireless Communication:** The proposed approach assumes the availability of a reliable wireless communication link with bounded latency. This ensures that shared state information and control commands are transmitted in real time, which is essential for coordinated multi-drone behavior.
- **Single Integrator Dynamics Model:** The high-level controller in this dissertation models each drone using a single integrator dynamics representation. The actual nonlinear flight dynamics are handled by an onboard autopilot (e.g., PX4 or equivalent), which tracks the commanded velocity or position references issued by the proposed coverage control algorithm. This abstraction is widely adopted in multi-agent control research as it simplifies the theoretical formulation while remaining practical for implementation.

Environment Assumptions

- **Static Scene during Reconstruction:** The scene to be reconstructed is assumed to remain static during the sampling and mapping process, i.e., there are no significant time-varying changes such as moving obstacles or rapidly deforming surfaces. This assumption is commonly adopted in 3D reconstruction research and ensures that collected images represent a consistent world state during processing.
- **Bounded Operating Region:** The overall field boundary in which the drones operate is assumed to be known and fixed in advance. However, the detailed content inside this region (e.g., object distribution, scene geometry) is assumed to be unknown and without prior knowledge. This allows safe trajectory planning within well-defined spatial limits while still addressing unknown and potentially complex scenes inside the boundary.
- **Reliable Reconstructed Map:** For methods that incorporate online map feedback, the reconstruction is assumed to be sufficiently accurate and reliable for the purpose of coverage control. In this dissertation, the NeuralRecon-based online

reconstruction framework is used, which incrementally updates the 3D scene representation. While reconstruction errors may exist, it is assumed that these errors do not fundamentally compromise the feedback used to adapt the coverage behavior.

Overall, these assumptions define the scope and boundary conditions of this research. They allow focusing on the core challenges of coverage control and adaptive image sampling without being confounded by low-level hardware limitations or extreme environmental uncertainty. The detailed implications and context of each assumption are revisited in later chapters alongside the specific methods and experimental validations where they apply.

1.5 Outline of the Dissertation

This dissertation is structured as follows:

- **Chapter 2 — Preliminaries: Coverage Control and 3D Reconstruction**

This chapter reviews key theoretical and algorithmic foundations that support the main contributions of the dissertation. It begins with a discussion of control barrier function (CBF)-based control frameworks, followed by a comprehensive overview of coverage control strategies, ranging from classical methods to more recent formulations such as persistent and angle-aware coverage. The chapter also surveys 3D reconstruction algorithms, including both classical offline pipelines based on Structure-from-Motion (SfM) and Multi-View Stereo (MVS), as well as recent neural reconstruction techniques that support online and real-time mapping.

- **Chapter 3 — Efficient Angle-Aware Coverage Control**

This chapter introduces the first main contribution: an efficient angle-aware coverage control strategy for coordinated image sampling in multi-drone systems using offline 3D reconstruction pipelines. The proposed method extends the existing angle-aware coverage formulation [24] by incorporating Voronoi-based spatial coordination as a nominal input. A QP-based controller integrates this input with angle-aware objectives and control barrier functions to ensure safe and effective deployment. The method is validated through both simulation and real-world experiments, conducted in both indoor and outdoor settings, demonstrating improved mission completion efficiency particularly in large-scale map reconstruction scenarios.

- **Chapter 4 — Scene-Adaptive Angle-Aware Coverage Control**

This chapter presents the second main contribution: a scene-adaptive coverage control framework that integrates real-time feedback from an online 3D reconstruction

pipeline, NeuralRecon [48]. By quantifying changes in the evolving mesh, the system dynamically updates coverage priorities to focus sensing on under-observed or unstable regions. The method is validated through Unity-ROS2 simulations and multi-drone experiments, demonstrating improved reconstruction completeness and quality compared to baseline methods without map feedback.

- **Chapter 5 — Conclusion and Future Work**

This chapter summarizes the key findings and contributions of the dissertation. It also discusses current limitations and outlines promising directions for future research, including extensions to radiance field-based active reconstruction using multi-drone systems.

1.6 Notations

The key notations used throughout this dissertation are summarized in Table 1.1. These symbols will be further explained as necessary in each chapter.

Table. 1.1: List of Frequently Used Notations

Symbol	Description
<i>Drone State and Control</i>	
i	Index of the drone
n	Total number of drones in the multi-robot system
$\mathcal{I}$	$\mathcal{I}$ is the set of all drones
p_i	State of drone i in space
(x_i, y_i, z_i)	3D position coordinates of drone i
θ_i^h, θ_i^v	Horizontal and vertical camera angles of drone i
u_i	Control input for drone i (velocity)
$\mathcal{V}_i$	Voronoi region assigned to drone i
<i>Virtual Field and Coverage</i>	
q_j	Discrete point in the 5D virtual sensing field
m	Total number of discrete points q_j in the virtual field
$\mathcal{Q}$	Set of all sensing points in the virtual field
ϕ_j	Importance index associated with point q_j
Φ	Global importance distribution across $\mathcal{Q}$
J_i	Local objective function of drone i
J	Global coverage objective (sum of J_i)
$h(p_i, q_j)$	Sensing performance of drone i at point q_j
<i>Reconstruction Feedback</i>	
r_k	Evaluation point in the mesh space
$\Delta V(r_k)$	Local mesh change magnitude at r_k
ψ_j	Feedback-informed importance modifier at point q_j
$\mathcal{T}$	Time instance of the mesh update from NeuralRecon
<i>Controller Parameters</i>	
δ_1, δ_2	Weights in the importance index formulation
γ	Exponential decay rate bound in CBF formulation
ϵ	Control slack threshold in QP controller
$D, \sigma_1\text{--}\sigma_4$	Parameters for sensing model distribution kernels
$\alpha(\cdot)$	Class- $\mathcal{K}$ function used in the CBF condition
J_{th}	Lower bound threshold for the objective function J

Chapter 2

Preliminaries: Coverage Control & 3D Reconstruction

2.1 Introduction

This chapter provides the essential theoretical background for understanding the rest of the dissertation. We begin by introducing Control Barrier Function (CBF)-based control, which serves as a foundation for constraint-driven control used throughout this work. We then review coverage control methods, from classical formulations to recent developments such as persistent and angle-aware approaches, which are related to the application of coordinated image sampling in 3D reconstruction. Finally, we introduce core 3D reconstruction techniques, covering both traditional offline pipelines and emerging online methods that enable real-time mesh updates.

2.2 Control Barrier Function (CBF)-Based Control

Control Barrier Functions (CBFs) are a class of control-theoretic tools used to enforce safety constraints in dynamical systems [73,74]. A CBF specifies a condition under which the system state remains within a designated safe set for all future time, thereby ensuring forward invariance. CBFs have been applied in a range of safety-critical tasks such as autonomous driving [75–77], robotic manipulation [78–80], robotic locomotion [81–83], and aerial exploration [39, 84, 85]. In the context of multi-agent systems, CBFs have also been employed to enforce a variety of safety-related constraints, including collision avoidance [86–88], inter-agent connectivity [89,90], field boundary compliance [85,91], and energy sustainability [39,42,62], while simultaneously enabling coordinated task execution.

To formalize the use of CBFs, we consider a control-affine system of the form:

$$\dot{x} = f(x) + g(x)u,$$

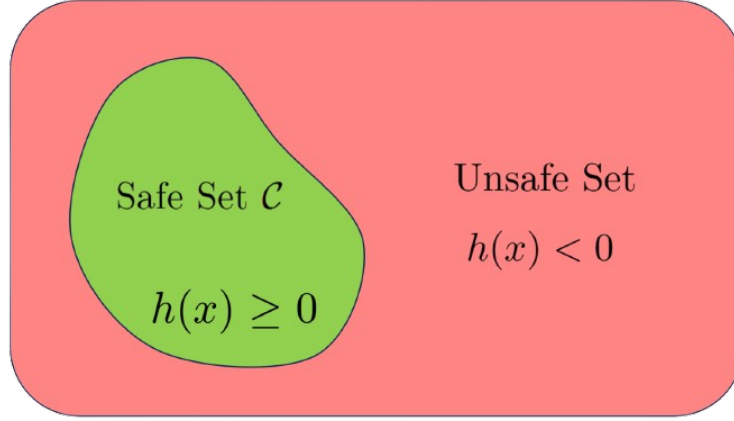


Fig. 2.1: Illustration of a control barrier function (CBF), which defines a safe set $\{x \mid h(x) \geq 0\}$ and an unsafe set $\{x \mid h(x) < 0\}$, used to enforce safety constraints in control systems.

where $x \in \mathbb{R}^n$ is the state, $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz continuous vector fields. We assume that $x(t)$ has a unique solution on a time interval $[t_0, t_1]$.

We define a continuously differentiable function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ and a safe set $\mathcal{C} := \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$ (see Fig. 2.1). The function $h(x)$ is said to be a control barrier function (CBF) for the set $\mathcal{C}$ if there exists a set $\mathcal{D}$ with $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^n$ such that:

$$\sup_{u \in \mathcal{U}} [L_f h(x) + L_g h(x)u + \alpha(h(x))] \geq 0, \quad \forall x \in \mathcal{D},$$

where $L_f h(x)$ and $L_g h(x)$ denote the Lie derivatives of $h(x)$ along $f(x)$ and $g(x)$, and $\alpha(\cdot)$ is an extended class- $\mathcal{K}$ function (commonly $\alpha(h) = \gamma h$ with $\gamma > 0$).

A set $\mathcal{S} \subseteq \mathbb{R}^n$ is said to be forward invariant if $x(t) \in \mathcal{S}$ holds for all $t \in [t_0, t_1]$ whenever $x(t_0) \in \mathcal{S}$. The forward invariance of $\mathcal{C}$ is ensured under any locally Lipschitz continuous controller u that satisfies:

$$L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0.$$

To implement this in practice, a Quadratic Program (QP) is solved to obtain a control input u^* that remains close to a nominal control input u_{nom} while satisfying the CBF constraint [73]:

$$\begin{aligned} u^*(x) = \arg \min_{u \in \mathcal{U}} \quad & \|u - u_{\text{nom}}\|^2 \\ \text{s.t.} \quad & L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0. \end{aligned}$$

In this dissertation, CBF-based QP control serves as the foundation for constraint-driven control used throughout the proposed framework. It is employed to enforce safety

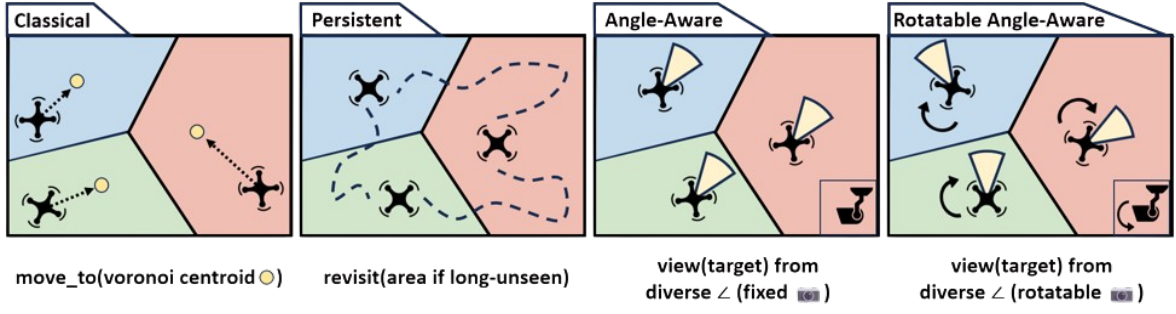


Fig. 2.2: Comparison of four coverage control strategies for multi-drone systems. Classical coverage control moves each drone toward the centroid of its assigned Voronoi region. Persistent coverage aims to revisit areas that have not been observed for an extended period of time. Angle-aware coverage considers not only position but also the viewing angle, promoting diverse perspectives on the target using fixed cameras. Rotatable angle-aware coverage further enhances this by allowing the drone's camera orientation to be actively adjusted.

constraints such as collision avoidance and workspace boundaries, as well as to certify sampling performance by constraining the decay rate of the objective function. These principles form the core of the angle-aware coverage strategy and the map-feedback-based adaptation mechanism developed in subsequent chapters.

2.3 Coverage Control

In this section, we review several strategies for multi-robot coverage control, ranging from the classical Voronoi-based approach [28] to more advanced methods such as persistent [60, 84] and angle-aware coverage [24, 41]. These methods form the basis for later developments in coordinated image sampling for 3D reconstruction using drone networks.

A visual comparison of the representative strategies is provided in Fig. 2.2, highlighting their key differences in sensing behavior, motion dynamics, and control mechanisms.

2.3.1 Classical Coverage Control

Classical coverage control provides a foundational framework for coordinating multiple mobile agents to monitor a spatial domain efficiently. The key objective is to position agents in such a way that redundant sensing is minimized and the overall sensing effectiveness is maximized. This strategy was initially developed for applications involving static or slowly changing environments, where agents are expected to converge to a stable configuration that ensures optimal sensing distribution.

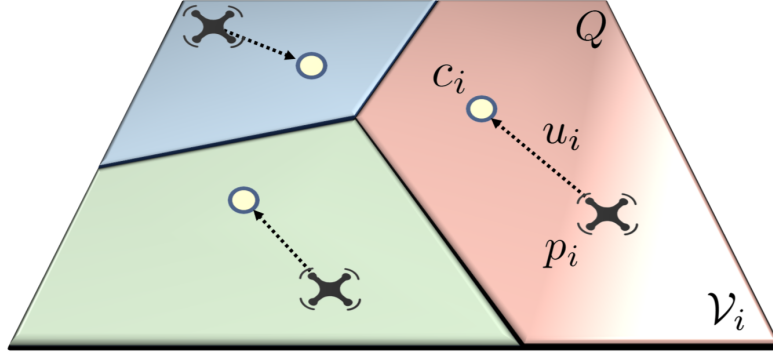


Fig. 2.3: Illustration of classical coverage control using a Voronoi-based partitioning approach, which allocates regions to drones based on spatial proximity to ensure even coverage.

At the heart of classical coverage control is the use of Voronoi diagrams to partition the environment based on geometric proximity, assigning each agent a region comprising all points closer to it than to any other. This locality-based decomposition enables agents to operate within clearly defined spaces. To optimize coverage, each agent moves toward the centroid of its region, weighted by a spatial importance function. This motion drives the system toward a Centroidal Voronoi Tessellation (CVT) [92], a configuration that balances sensing effort across the domain. The underlying mechanism follows Lloyd's algorithm [93], which was originally proposed for vector quantization and later adapted to multi-agent systems. In this iterative process, agents update their Voronoi cells, compute the corresponding centroids, and reposition themselves based on this information to improve overall coverage quality.

This work was initially popularized by a seminal contribution from Cortés et al. [28], who proposed a distributed gradient-based method that drives agents toward Voronoi centroids, resulting in convergence to a Centroidal Voronoi Tessellation (CVT). This framework has since been extended to non-convex environments [34], online adaptation [33], limited-range interactions [72], as well as anisotropic sensing [58, 94]. These developments also later paved the way for aerial applications [29, 59, 95], where this principle was adapted or served as inspiration for vision-based multi-drone systems.

Let $Q \subset \mathbb{R}^2$ denote the compact mission space, and let $I = \{1, \dots, n\}$ be the index set for n drones, each located at position $p_i \in Q$. The domain is partitioned into Voronoi regions based on Euclidean distance:

$$\mathcal{V}_i(p) = \{q \in Q \mid \|q - p_i\|^2 \leq \|q - p_j\|^2, \forall j \in I\}, \quad (2.1)$$

where $p = [p_1^\top, \dots, p_n^\top]^\top$ is the stacked vector of all drone positions. Each drone is assumed

to sense only within its own Voronoi region. To control its position, each drone is modeled by first-order dynamics:

$$\dot{p}_i = u_i, \quad u_i \in \mathbb{R}^2, \quad (2.2)$$

where u_i is the control input assigned to drone i .

The coverage objective is defined by a locational cost function as introduced in [28]:

$$J(p) = \sum_{i=1}^n \int_{\mathcal{V}_i(p)} \|p_i - q\|^2 \phi(q) dq,$$

where $\phi(q)$ is a spatial density function indicating the sensing importance of location q .

The gradient of $J(p)$ with respect to p_i is computed as:

$$\nabla_{p_i} J(p) = 2 \int_{\mathcal{V}_i(p)} (p_i - q) \phi(q) dq = 2M_{\mathcal{V}_i}(p_i - c_i),$$

where $M_{\mathcal{V}_i} = \int_{\mathcal{V}_i(p)} \phi(q) dq$ is the mass of region $\mathcal{V}_i$ under the density $\phi(q)$, and c_i is the centroid:

$$c_i = \frac{1}{M_{\mathcal{V}_i}} \int_{\mathcal{V}_i(p)} q \phi(q) dq.$$

This reveals that the gradient of the cost function points from the centroid toward the current position. Substituting the negative gradient as the control input yields:

$$\dot{p}_i = u_i = -k(p_i - c_i),$$

where $k > 0$ is a control gain. This control law drives each drone toward the centroid of its Voronoi cell, leading the system to converge to a Centroidal Voronoi Tessellation (CVT), minimizing the global coverage cost as illustrated in Fig. 2.3.

2.3.2 Persistent Coverage Control

Persistent coverage control extends classical approaches to scenarios in which the sensing requirement varies over time. Unlike classical strategies that drive agents to static optimal configurations, persistent coverage seeks to ensure that all regions of the environment are revisited continuously. Early schemes were proposed in [37, 60], introducing persistent monitoring behaviors through continuous agent movement. A key idea that enables this persistence is the use of a time-varying spatial importance function, as formulated in [61], which employs a so-called time-varying control barrier function [96] to represent information decay and recovery. This framework was later extended in [39, 84] to provide formal performance guarantees for multi-drone persistent coverage. These strategies are particularly effective in applications requiring continuous situational awareness,

including long-term environmental monitoring [62, 97, 98], as well as object search and tracking [39, 63].

As formulated in [39], let $Q \subset \mathbb{R}^2$ denote the ground field of interest, and let $p_i \in Q$ represent the position of drone $i \in I := \{1, \dots, n\}$. In this formulation, the authors mainly consider the 2D motion of each drone, i.e., $p_i = [x_i, y_i]^\top \in Q$. Each drone is governed by the same first-order dynamics described in (2.2), where $u_i \in \mathbb{R}^2$ denotes the control input. The sensing region of each drone is modeled as a circular disk $B_i(p_i) := \{q \in Q \mid \|q - p_i\| \leq R\}$ of fixed radius R . The effective sensing region is then defined as the intersection of this disk with the Voronoi region $\mathcal{V}_i(p)$ introduced in (2.1):

$$S_i(p) := B_i(p_i) \cap \mathcal{V}_i(p).$$

The coverage model assumes that all drone motions and measurements are described in a shared global frame Σ_w , with axes $\{x, y, z\}$ representing the world coordinate system. The drone positions p_i evolve in the xy -plane, while the sensing footprints project downward onto Q , the ground surface (See Fig. 2.4).

To model the need for repeated sensing, a time-varying importance function $\phi(q, t)$ is introduced and dynamically updated as follows:

$$\dot{\phi}(q, t) = \begin{cases} -\delta\phi(q, t), & \text{if } q \in \cup_{i=1}^n S_i(p), \\ \bar{\delta}(1 - \phi(q, t)), & \text{otherwise,} \end{cases}$$

where $\delta, \bar{\delta} > 0$ are decay and recovery rates, respectively. This update rule causes the importance in covered regions to decay while increasing in uncovered regions, driving drones to revisit areas persistently.

The persistent coverage objective is expressed through the following time-varying cost function:

$$J(p, t) = - \sum_{i=1}^n \int_{S_i(p)} \|q - p_i\|^2 \phi(q, t) dq + b \int_{Q \setminus \cup_i S_i(p)} \phi(q, t) dq,$$

where $b < 0$ is a weighting parameter that balances the importance between covered and uncovered regions. A performance requirement is imposed by enforcing $J(p(t), t) \geq \gamma$ for some constant threshold $\gamma < 0$.

To ensure this performance level, a constraint-based controller is formulated using the control barrier function (CBF) framework. Defining $\tilde{J}(p, t) := J(p, t) - \gamma$, the control objective is to maintain the forward invariance of the set $\{p \in Q^n \mid \tilde{J}(p, t) \geq 0\}$. Each drone i solves the following quadratic program:

$$\begin{aligned} u_i^* &= \arg \min_{u_i \in \mathbb{R}^2} \|u_i\|^2 \\ \text{s.t.} \quad & \nabla_{p_i} J(p, t)^\top u_i + \xi_i(p, t) \geq 0, \end{aligned}$$

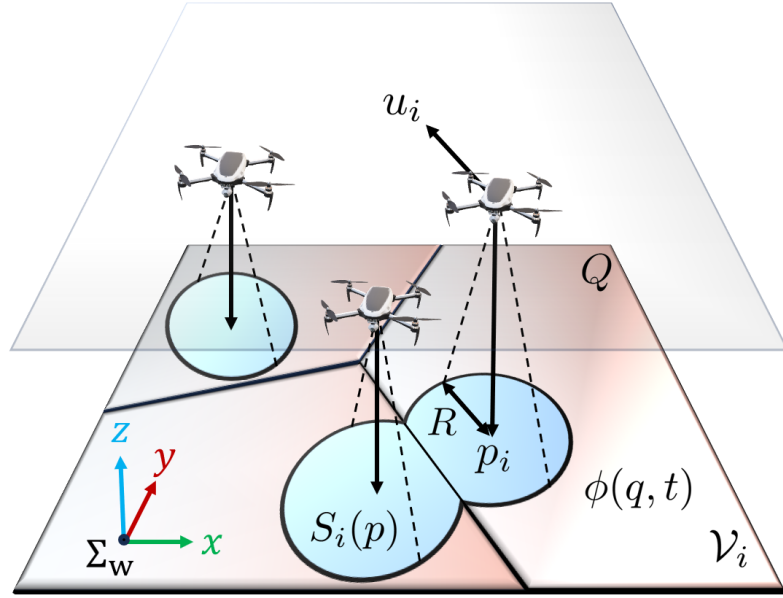


Fig. 2.4: Illustration of persistent coverage control, where drones continuously revisit areas that has not been observed for an extended period of time.

where $\xi_i(p, t)$ is a compensating term designed to satisfy $\dot{\tilde{J}}(p, t) \geq -\alpha(\tilde{J}(p, t))$ for a chosen class- $\mathcal{K}$ function $\alpha(\cdot) = k\tilde{J}$.

This formulation guarantees that the objective $J(p, t)$ remains above the specified threshold γ , thereby achieving persistent coverage with certified performance. The structure of $J(p, t)$ and its gradient $\nabla_{p_i} J(p, t)$ enables a partially distributed implementation based on local information and interactions, making the method scalable for large-scale multi-drone systems.

2.3.3 Angle-Aware Coverage Control

Angle-aware coverage control extends classical and persistent strategies by explicitly incorporating the orientation of vision sensors into the coverage formulation. This extension is particularly motivated by drone-based 3D reconstruction applications [16,19,20,99], where capturing scenes from a variety of viewpoints is essential for achieving high reconstruction accuracy [21]. Unlike earlier methods that focus primarily on spatial coverage, angle-aware strategies aim to ensure that each point in space is observed from complementary viewpoints. This is critical, as many structure-from-motion [44] and multi-view stereo [100] algorithms depend on multi-angle observations to reconstruct scene depth and geometry reliably.

To address this requirement, Shimizu et al. [24] proposed a novel angle-aware coverage control framework that considers both the spatial position and viewing direction of on-

board cameras. In their formulation, the drone's camera is assumed to be fixed to the body, meaning its orientation could not be independently actuated. As a result, the five-dimensional coverage requirement, which consists of three-dimensional spatial positions and two-dimensional viewing angles, is projected onto a two-dimensional control space by mapping each target viewpoint onto the ground plane. This allows drones to optimize angular diversity solely through position control. Suenaga et al. [40] demonstrated experimentally that this angle-aware approach significantly improves 3D reconstruction quality over traditional coverage methods.

As formulated in [24], a team of n drones, indexed by $\mathcal{I} := \{1, \dots, n\}$, operates on a 2D plane $\mathcal{P} \subset \mathbb{R}^2$ at a fixed altitude $z = z_c$. The world frame Σ_w is defined such that its z -axis is perpendicular to this plane (See Fig. 2.5). Each drone $i \in \mathcal{I}$ has a planar position $p_i = [x_i, y_i]^\top \in \mathcal{P}$ and is equipped with a body-fixed camera whose orientation cannot be actuated independently. All drones share the same altitude and attitude, and their motion is restricted to the plane $\mathcal{P}$. The motion of each drone is governed by first-order dynamics as previously defined in (2.2), where $u_i \in \mathbb{R}^2$ denotes the control velocity input.

To characterize the observation requirements for 3D reconstruction, the target field is defined as a compact set $\mathcal{B} \subset \mathbb{R}^3$, representing the 3D points of interest in the environment. Each point $[x, y, z]^\top \in \mathcal{B}$ is associated with desirable viewing angles: a horizontal angle $\theta_h \in [0, 2\pi)$ and a vertical angle $\theta_v \in [0, \pi]$. Based on this, a virtual field $\mathcal{Q}_c \subset \mathbb{R}^5$ is introduced to capture both spatial and directional requirements, where each target is represented as

$$q = [x, y, z, \theta_h, \theta_v]^\top \in \mathcal{Q}_c.$$

Since the drone's camera is fixed and could not be actuated independently, control is limited to 2D motion on the plane. Thus, each virtual point $q \in \mathcal{Q}_c$ is projected onto the plane $\mathcal{P}$ using a mapping $\zeta : \mathcal{Q}_c \rightarrow \mathcal{P}$:

$$\zeta(q) = \begin{bmatrix} x - (z_c - z) \tan\left(\frac{\pi}{2} - \theta_v\right) \cos \theta_h \\ y - (z_c - z) \tan\left(\frac{\pi}{2} - \theta_v\right) \sin \theta_h \end{bmatrix},$$

which represents the position on $\mathcal{P}$ from which a drone should observe q to align with the desired viewing direction.

The virtual field $\mathcal{Q}_c$ is then discretized into a finite set of target points indexed by $\mathcal{M} := \{1, \dots, m\}$, where each $q_j = [x_j, y_j, z_j, \theta_{h,j}, \theta_{v,j}]^\top \in \mathcal{Q}_c$ encodes a 3D target point with associated viewing angles. The coverage performance of drone i with respect to virtual point q_j is quantified by:

$$h(p_i, q_j) = \exp\left(-\frac{\|p_i - \zeta(q_j)\|^2}{2\sigma^2}\right),$$

where $\sigma > 0$ controls the effective sensing range.

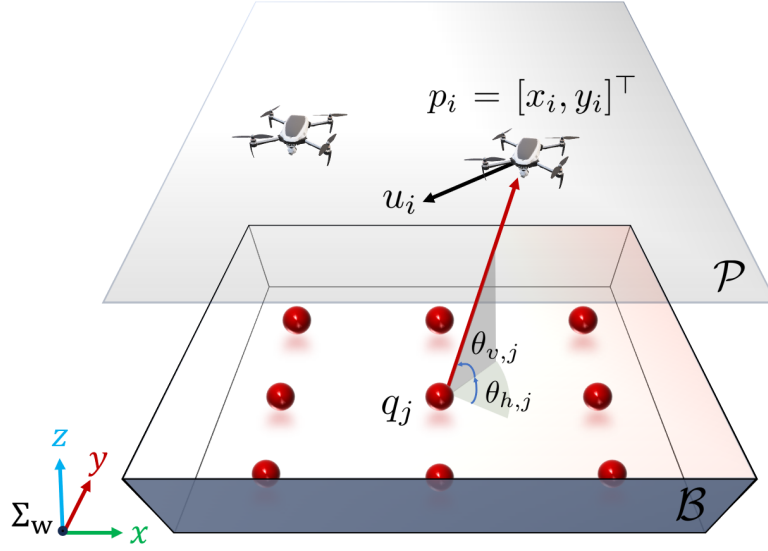


Fig. 2.5: Illustration of angle-aware coverage control. The coverage objective considers not only the spatial distribution of drones but also the viewing angles toward the target point. However, the camera orientation is fixed and cannot be independently controlled, being aligned with the drone's heading direction.

Each virtual point q_j is associated with an importance index $\phi_j(t)$, which evolves over time as:

$$\dot{\phi}_j = -\delta \max_{i \in \mathcal{I}} h(p_i, q_j) \phi_j, \quad \delta > 0,$$

ensuring that well-observed points decay in importance, while poorly covered regions retain priority.

The global objective is to reduce the cumulative importance across all virtual points:

$$J(p) = \sum_{j \in \mathcal{M}} \phi_j A,$$

where A is the volume of each discretized cell in $\mathcal{Q}_c$.

To promote efficient collaboration, each drone computes its coverage contribution:

$$I_i = \sum_{j \in \mathcal{V}_i(p)} \delta h(p_i, q_j) \phi_j A,$$

where $\mathcal{V}_i(p)$ denotes the Voronoi partition assigned to drone i in the 2D projected space.

To ensure that each drone contributes sufficiently to reducing the overall importance index, we define the candidate control barrier function $b_i(p, \phi) := I_i - \gamma$, where $\gamma > 0$ is a predefined performance threshold. This function enforces the condition $I_i \geq \gamma$, guiding drones to avoid low-importance regions and maintain meaningful observations. The corresponding constraint is incorporated into a quadratic program (QP) to compute

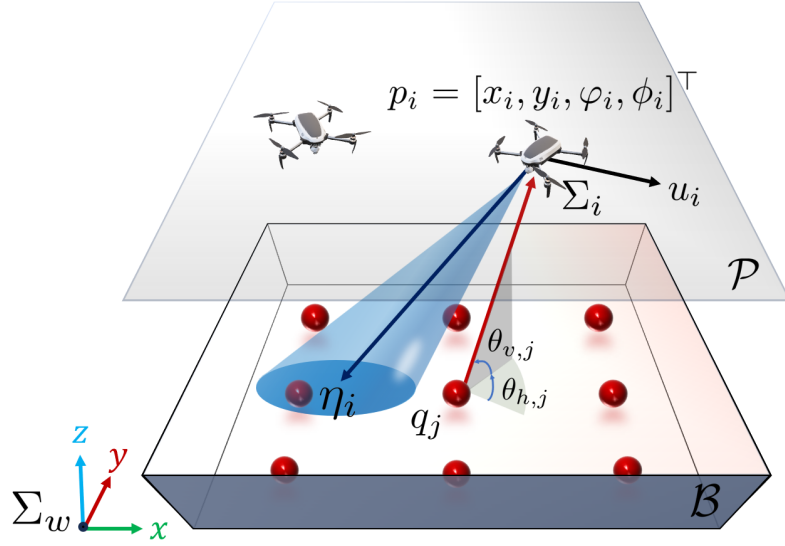


Fig. 2.6: Illustration of rotatable angle-aware coverage control. Similar to the angle-aware strategy, the coverage objective includes both spatial positioning and viewing direction. In this case, however, the camera orientation can be actively adjusted, allowing the drone to optimize its viewpoint independently of its flight heading.

the final control input:

$$\begin{aligned} (u_i^*, w_i^*) &= \arg \min_{u_i, w_i} \epsilon \|u_i\|^2 + |w_i|^2 \\ \text{s.t. } & \dot{b}_i + \alpha(b_i) \geq w_i, \end{aligned}$$

where $u_i \in \mathbb{R}^2$ is the position control input, w_i is a slack variable, $\alpha(\cdot)$ is a class- $\mathcal{K}$ function, and $\epsilon > 0$ is a weighting parameter. This formulation allows drones to dynamically adapt to the evolving importance landscape and maintain effective angle-aware coverage.

2.3.4 Rotatable Angle-Aware Coverage Control

While the previous angle-aware coverage control assumes that the camera orientation is fixed, this assumption limits the ability to dynamically adjust viewpoints, especially in increasingly complex environments. To address this limitation, Lu et al. [41] proposed an extension known as *angle-aware coverage with camera rotational control*. In this dissertation, we refer to this approach as *rotatable angle-aware coverage control*. The method enables each drone to independently actuate its yaw and pitch angles through a gimbal mechanism, in addition to performing planar movement. This added degree of freedom allows for more comprehensive coverage in the five-dimensional sensing space. Although a full six-degree-of-freedom configuration including both three-dimensional position and orientation would offer complete viewpoint control, the roll angle is often excluded in

practice [22, 23]. This is because it has limited impact on reconstruction quality and removing it helps reduce system complexity. As a result, the search space is simplified to five-dimensional camera poses. Furthermore, by introducing camera orientation control, this method overcomes a major drawback of the fixed-angle formulation, which was limited in its ability to observe all desired target viewpoints. As a result, the rotatable angle-aware strategy significantly enhances the ability of drones to fulfill diverse viewpoint requirements, enabling more thorough and effective coverage of the reconstruction scene.

In this formulation [41], each drone $i \in \mathcal{I} := \{1, \dots, n\}$ is equipped with a gimbal-mounted camera and assumed to operate at a fixed altitude $z = z_c$. Its state is defined as $p_i = [x_i, y_i, \varphi_i, \phi_i]^\top \in \mathbb{R}^4$, where $(x_i, y_i) \in \mathcal{P}$ represents the planar position, and (φ_i, ϕ_i) are the yaw and pitch angles of the gimbal camera. The yaw angle φ_i is defined on $[0, 2\pi)$, and the pitch angle ϕ_i on $(0, \pi/2]$. The world frame Σ_w is oriented such that its z -axis is orthogonal to the plane $\mathcal{P}$, and a local frame Σ_i is fixed on the camera of drone i . With this setup, each drone is controlled in a four-dimensional space to satisfy a five-dimensional sensing requirement. The motion follows extended first-order dynamics:

$$\dot{p}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\varphi}_i \\ \dot{\phi}_i \end{bmatrix} = \begin{bmatrix} u_{x,i} \\ u_{y,i} \\ u_{\varphi,i} \\ u_{\phi,i} \end{bmatrix} = u_i, \quad u_i \in \mathbb{R}^4.$$

Each drone's optical axis, denoted η_i , is determined by the camera's yaw and pitch angles and is used to evaluate directional alignment with target viewpoints (See Fig. 2.6). The virtual target field $\mathcal{Q}_c$ and its discretization remain as previously defined in Subsection 2.3.3, where each point $q_j = [x_j, y_j, z_j, \theta_{h,j}, \theta_{v,j}]^\top \in \mathcal{Q}_c$ specifies a desired viewpoint.

The coverage performance function $h(p_i, q_j)$ is modeled as the product of two Gaussian functions:

$$h(p_i, q_j) = \exp\left(-\frac{h_1(p_i, q_j)}{2\sigma_1^2}\right) \exp\left(-\frac{h_2(p_i, q_j)}{2\sigma_2^2}\right),$$

where h_1 penalizes deviation from the field-of-view region and h_2 measures the alignment between the optical axis η_i and the desired viewing direction $(\theta_{h,j}, \theta_{v,j})$. For a detailed definition and derivation of h_1 and h_2 , refer to [41].

The importance index $\psi_j(t)$ associated with each target evolves according to:

$$\dot{\psi}_j = -\delta \max_{i \in \mathcal{I}} h(p_i, q_j) \psi_j, \quad \delta > 0,$$

ensuring persistent observation of poorly covered regions.

The global objective is to minimize the total residual importance:

$$J(p) = \sum_{j \in \mathcal{M}} \psi_j A,$$

where $\mathcal{M} := \{1, \dots, m\}$ indexes the discretized target points and A denotes the volume of each discretized cell in $\mathcal{Q}_c$. The contribution of drone i is defined as:

$$I_i = \sum_{j \in \mathcal{V}_i(p)} \delta h(p_i, q_j) \psi_j,$$

where $\mathcal{V}_i(p)$ assigns each target to the drone with the highest $h(p_i, q_j)$.

To ensure that each drone contributes meaningfully to the overall sensing objective, we define a candidate control barrier function $b_{i,I}(p, \psi) := I_i - \gamma$, where $\gamma > 0$ specifies the desired performance threshold. This constraint is incorporated into a QP as:

$$\dot{b}_{i,I} + \alpha(b_{i,I}) \geq w_i,$$

where $\alpha(\cdot)$ is a class- $\mathcal{K}$ function and w_i is a slack variable.

The final control input is computed by solving:

$$\begin{aligned} (u_i^*, w_i^*) &= \arg \min_{u_i, w_i} \epsilon \|u_i\|^2 + |w_i|^2 \\ \text{s.t. } &\dot{b}_{i,I} + \alpha(b_{i,I}) \geq w_i, \end{aligned}$$

where $\epsilon > 0$ weighs energy cost against performance guarantee. This formulation enables drones to actively adjust both camera pose and position, providing more comprehensive coverage for high-quality 3D mapping.

2.4 3D Reconstruction

Three-dimensional (3D) reconstruction refers to the process of estimating the geometric structure and shape of real-world objects or environments using sensor data such as images, depth maps, or point clouds. It is a foundational technique for spatial understanding and interpretation, supporting a broad spectrum of applications in computer vision, robotics, and digital modeling [101]. In robotics and autonomous navigation, 3D reconstruction enhances environmental awareness and contributes to safety-critical tasks such as path planning [53, 102]. In augmented and virtual reality (AR/VR), it enables immersive scene representation and interactive rendering [103]. In cultural heritage preservation, it offers non-invasive tools for digitizing, documenting, and virtually restoring historical artifacts [104]. In medical imaging, 3D reconstruction is used to create patient-specific anatomical models that assist in preoperative surgical planning [105, 106]. Finally, in industrial inspection, it supports visual evaluation tasks such as weld inspection and contributes to high-precision modeling efforts [107].

Based on the taxonomy discussed in [101], 3D reconstruction methods can be categorized along several dimensions, including the type of sensing modality, the form of data

representation, and the temporal capability. This classification is adopted in this dissertation to help contextualize and position the reconstruction techniques that are most relevant to our problem setting.

First, in terms of sensing modality, 3D reconstruction approaches are generally classified into *active* and *passive* methods. Active methods interact with the environment by emitting structured signals or energy to capture depth information. Examples include LiDAR [108, 109], structured light [110], and time-of-flight cameras [111], which provide accurate range measurements but often require specialized and costly hardware. In contrast, passive methods observe the environment without emitting signals, relying solely on measurements of the light reflected or emitted from surfaces. These approaches typically utilize standard optical sensors, such as RGB [48, 65, 112–115] or RGB-D cameras [116, 117], and infer depth through geometric analysis or learning-based estimation. In this dissertation, we focus specifically on passive methods using monocular RGB images, as they offer a lightweight and easily deployable solution that is particularly suitable for aerial platforms like drones.

Second, reconstruction techniques can be broadly classified based on their form of data representation, with a fundamental distinction between *explicit* and *implicit* models. Explicit representations describe 3D geometry using discrete primitives such as point clouds, voxel grids, or polygon meshes. These can be obtained through traditional multi-view geometry methods or recent learning-based approaches that produce geometric outputs. In contrast, implicit representations define geometry as continuous functions, such as signed distance fields, occupancy field, or radiance fields, which are usually learned directly from image data. Recent advances in this area include Neural Radiance Fields (NeRF) [71] and 3D Gaussian Splatting (3DGS) [51], which represent scenes in a compact and view-dependent form. In this dissertation, we primarily adopt reconstruction methods that rely on explicit representations, such as polygon meshes or point clouds.

Finally, although [101] introduces a classification based on scene dynamics, distinguishing between static and dynamic reconstruction depending on whether the environment changes over time, this dissertation adopts a different notion of temporal capability. Here, we refer to temporal capability in terms of whether the 3D reconstruction is performed in an *offline* or *online* manner. Offline reconstruction typically involves collecting a complete set of images before processing, allowing high-fidelity models to be generated using pipelines such as Structure-from-Motion (SfM) [44, 118] followed by Multi-View Stereo (MVS) [100]. However, such approaches provide no intermediate feedback during operation. In contrast, online reconstruction incrementally updates the 3D model as new data becomes available, offering real-time or near real-time feedback. This functionality

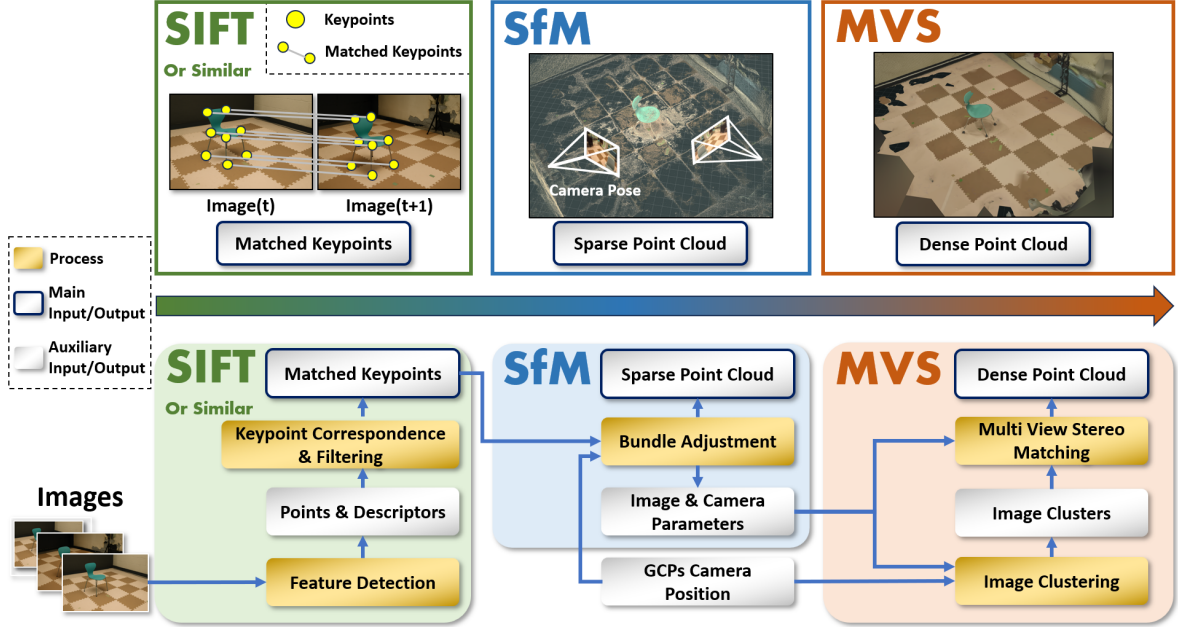


Fig. 2.7: Illustration of the classical SfM-MVS reconstruction pipeline, which typically consists of three main steps: keypoint detection and matching (e.g., SIFT), camera pose and sparse point cloud estimation via Structure-from-Motion (SfM), and dense reconstruction using Multi-View Stereo (MVS). This illustration is inspired by the illustration provided in [119].

is especially valuable in robotics, where it supports adaptive planning, reactive control, and exploration. In particular, our method in Chapter 4 leverages online reconstruction to dynamically inform drone control during flight, which is not feasible with traditional offline techniques.

In the remainder of this section, we review representative 3D reconstruction techniques that are particularly relevant to the drone-based image sampling methods developed in Chapter 3 and Chapter 4. Specifically, we examine approaches commonly adopted in offline and online settings, corresponding to the temporal classification discussed above. The offline techniques, grounded in classical multi-view geometry, relate to the reconstruction workflow used to generate 3D maps for visualizing results in Chapter 3. In contrast, the online techniques support real-time or near real-time map updates and play a key role in enabling the scene-adaptive control framework presented in Chapter 4.

2.4.1 Offline Reconstruction Techniques

In this dissertation, we utilize offline reconstruction techniques based on monocular image inputs using a standard multi-stage pipeline. This pipeline has been widely implemented in popular open-source tools such as COLMAP [44, 65], openMVG [66], and

Meshroom (AliceVision) [67], each offering slight variations and configurations. Here, we summarize the general workflow commonly referred to as the Structure from Motion and Multi-View Stereo (SfM-MVS) approach, illustrated in Fig. 2.7, and described in detail in [119].

The reconstruction process begins with the extraction of keypoints from each image. Keypoints are image regions with distinct appearance, typically found in textured or high-contrast areas. Algorithms such as Scale Invariant Feature Transform (SIFT) [68] are commonly used for this task, as they generate descriptors that are robust to changes in scale, rotation, and illumination. These descriptors are matched across multiple images to establish 2D correspondences between keypoints observed from different views.

Following feature matching, a coarse geometric consistency check is performed to verify the matches. The Structure from Motion (SfM) stage then estimates the camera intrinsic parameters, extrinsic parameters (camera poses), and a sparse three-dimensional point cloud of the scene. This is accomplished through bundle adjustment, a nonlinear optimization process that minimizes the reprojection error of the matched keypoints [44, 69]. The resulting output includes both the sparse geometry and the calibrated camera poses. Optionally, the reconstruction can be scaled and georeferenced using Ground Control Points (GCPs), which are known reference points in the environment, or by leveraging data from onboard navigation systems such as GPS and IMUs.

Once the sparse model and camera calibration are obtained, the Multi-View Stereo (MVS) stage produces a dense reconstruction of the scene. MVS algorithms compute per-pixel depth maps by analyzing photometric consistency across multiple overlapping images [64, 65]. For computational efficiency, images are typically clustered based on camera pose proximity, and dense reconstruction is performed within these clusters. The resulting depth maps are then fused into a dense point cloud or a surface mesh, using volumetric fusion techniques such as Truncated Signed Distance Fields (TSDF) or surface reconstruction methods like Poisson reconstruction or Delaunay triangulation.

The final product of the SfM-MVS pipeline is a dense three-dimensional model of the scene, typically in the form of a colored point cloud or polygon mesh. In some applications, the reconstruction may be post-processed to generate additional outputs such as digital surface models (DSMs), orthomosaics, or canopy height models (CHMs). While this approach offers high accuracy and completeness, it is inherently offline and requires the entire set of images to be acquired prior to processing, making it unsuitable for applications requiring real-time feedback.

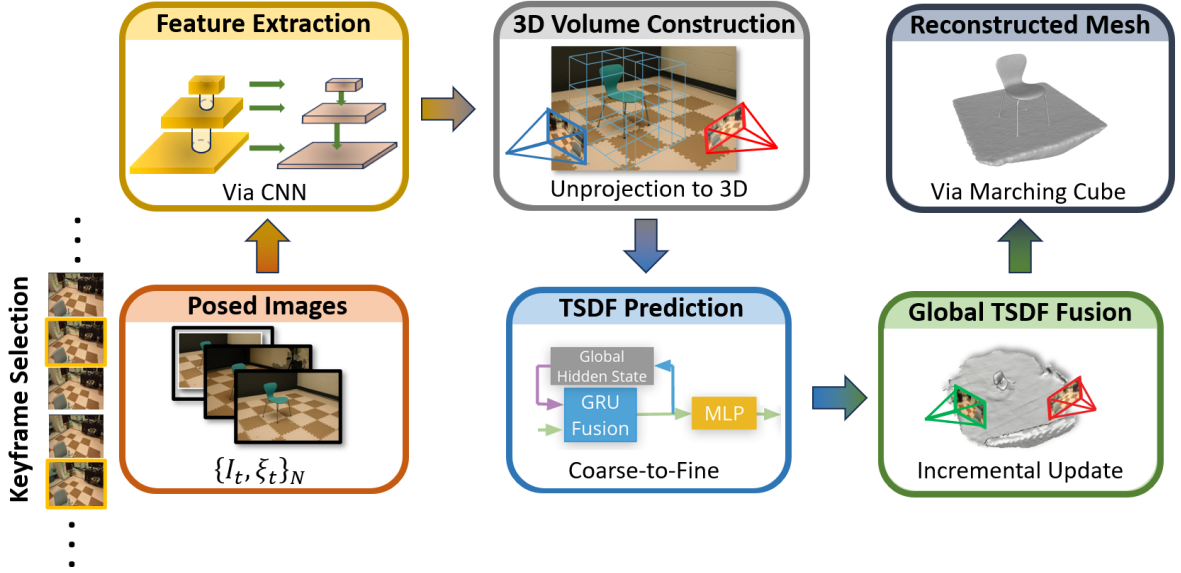


Fig. 2.8: Illustration of an online 3D reconstruction pipeline based on NeuralRecon. The process begins with **keyframe selection** and the use of **posed images**, which are passed through a neural network for **feature extraction**. These features are integrated into a **global TSDF volume** using differentiable volume fusion techniques. Finally, the accumulated volume is converted into a **reconstructed mesh** that can be rendered or used for downstream tasks. This figure is adapted based on NeuralRecon’s system design from [48].

2.4.2 Online Reconstruction Techniques

Online 3D reconstruction refers to the process of incrementally generating and updating a 3D map as new sensor data arrives. Unlike offline pipelines, which require all input images to be collected prior to processing, online methods operate iteratively and can produce partial reconstructions in real time or near real time. This capability enables immediate feedback during data acquisition and supports dynamic decision-making, which is especially critical for robotics and drone applications.

Recent developments in deep learning have played a key role in enabling online reconstruction by accelerating components such as feature extraction, depth estimation, and surface fusion [120]. These methods differ in how they represent 3D geometry, typically categorized as depth map-based, voxel-based, point cloud-based, mesh-based, or implicit surface-based [121]. For example, some techniques, such as NeuralRecon [48], TransformerFusion [70], and VisFusion [49], employ volumetric fusion of learned features to produce mesh- or voxel-based representations. Meanwhile, implicit representations have recently gained significant popularity, especially with the introduction of neural radiance

fields. These approaches, such as Neural Radiance Fields (NeRF) [71] and 3D Gaussian Splatting (3DGS) [51], model scenes as continuous volumetric functions or point-based primitives, enabling photorealistic novel view synthesis and compact scene encoding. Although classical NeRF models are computationally intensive and thus impractical for real-time use, follow-up works like Instant-NGP [50] propose efficient encoding strategies that reduce training time and support near real-time rendering, making online integration increasingly feasible.

In this dissertation, we adopt the NeuralRecon [48] framework for real-time reconstruction, which is particularly well-suited for feedback-driven drone control as developed in Chapter 4. To provide broader context and identify future directions, we also briefly review other emerging techniques such as NeRF [71] and 3DGS [51], which represent promising advances in fast 3D scene modeling and rendering.

NeuralRecon

NeuralRecon [48] is a learning-based framework for real-time 3D reconstruction from monocular RGB image streams. It plays a central role in this dissertation as the underlying algorithm for enabling real-time map feedback, which is critical to the scene-adaptive coverage control strategy proposed in Chapter 4. By combining deep convolutional networks with geometric reconstruction principles, NeuralRecon incrementally produces mesh-based representations of the environment that evolve as new images arrive.

The overall pipeline (illustrated in Fig. 2.8) consists of five main stages: feature extraction, 3D feature volume construction, coarse-to-fine TSDF prediction, global TSDF fusion, and mesh reconstruction. First, a deep convolutional neural network (CNN) extracts multi-scale spatial features from each input RGB image I_t . These features are unprojected into a 3D voxel grid using known camera poses $\xi_t \in SE(3)$, which are typically obtained from an external SLAM (Simultaneous Localization and Mapping) system such as ORB-SLAM [122]. This produces a 3D feature volume for each frame, positioned consistently within a global coordinate system.

Next, a recurrent fusion module based on gated recurrent units (GRUs) updates the global TSDF volume by integrating feature information across frames. This module maintains a global hidden state that enables the system to accumulate temporal context over time, enhancing the coherence of predictions. The fused features are then passed through a multi-layer perceptron (MLP), which predicts the signed distances at each voxel location in a coarse-to-fine manner. These predicted values form a volumetric representation of the scene in the form of a TSDF, which encodes the signed distance between each voxel

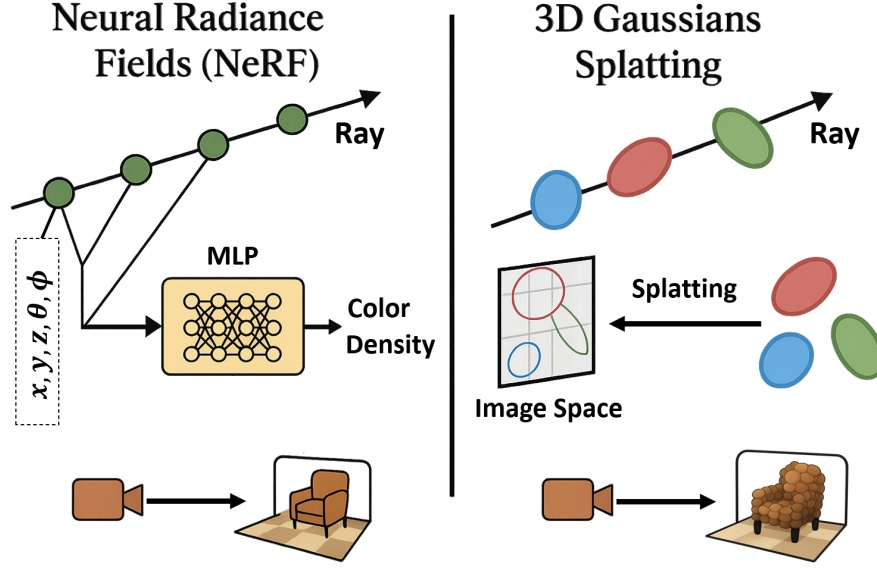


Fig. 2.9: Comparison between Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS). **Left:** NeRF samples points along camera rays, and uses a multi-layer perceptron (MLP) to map spatial coordinates and viewing directions to color and density values, enabling photorealistic view synthesis through volume rendering. **Right:** 3DGS represents the scene as a set of anisotropic 3D Gaussians, which are projected and blended directly in image space using alpha compositing, supporting real-time rendering via splatting. This illustration is inspired by the illustration provided in [124].

center and the nearest surface boundary.

Once the TSDF is sufficiently updated, surface extraction is performed using the Marching Cubes algorithm [123], which converts the volumetric grid into a triangle mesh. The resulting mesh consists of 3D points (vertices) connected by triangle faces to approximate the shape of the scene. This format is suitable for computing geometric properties like point positions and surface normals, which are useful for subsequent processing steps.

In this dissertation, the mesh generated by NeuralRecon plays a central role in the real-time feedback mechanism presented in Chapter 4. At each frame, the mesh is analyzed to detect changes in the reconstructed surface, and this information is used to update the importance index that guides multi-drone coordination. By continuously reflecting the current state of the environment, the mesh enables adaptive and responsive control throughout the reconstruction process.

Other Emerging Techniques

In addition to volumetric fusion-based methods like NeuralRecon, several emerging techniques have recently gained traction for online 3D reconstruction due to their use of

neural rendering and novel scene representations. These approaches often employ implicit or point-based formulations that support compact and continuous modeling of geometry and appearance (see Fig. 2.9).

One prominent method is Neural Radiance Fields (NeRF) [71], which represents a scene as a continuous volumetric radiance field. For each 3D spatial location and viewing direction, typically encoded as (x, y, z, θ, ϕ) , NeRF employs a multi-layer perceptron (MLP) to predict the corresponding color and density. By evaluating the MLP at many sampled locations along each camera ray and applying volume rendering, NeRF generates photorealistic images from novel viewpoints. Despite its high rendering quality, the original formulation of NeRF requires substantial computational time for training and rendering, limiting its suitability for real-time applications.

To overcome these limitations, Instant Neural Graphics Primitives (Instant-NGP) [50] introduces a multiresolution hash encoding combined with a lightweight MLP, significantly accelerating both training and inference. This enables near real-time performance while preserving visual quality, making Instant-NGP practical for interactive visualization and fast scene capture.

Another notable advancement is 3D Gaussian Splatting (3DGS) [51], which replaces dense volumetric or voxel-based representations with a set of anisotropic 3D Gaussian primitives. Each Gaussian encodes properties such as position, color, opacity, and shape (via a covariance matrix). Instead of querying a neural network, 3DGS directly projects these Gaussians onto the image space along the camera rays. Through a differentiable process known as splatting, the Gaussians are blended using alpha compositing to form the rendered image. This formulation offers high rendering speed and quality, supporting real-time use cases such as dynamic scene visualization, mapping, and exploration.

Although these techniques are not directly used in the proposed system, they represent promising future directions. Their compact and continuous scene representations, fast rendering pipelines, and differentiable structure offer potential benefits for robotic perception, adaptive control, and closed-loop active exploration in dynamic environments.

2.5 Chapter Summary

This chapter has reviewed the theoretical foundations underlying the methodologies proposed in this dissertation. We began by introducing Control Barrier Functions (CBFs) as a key framework for constraint-driven control, which enables safety-guaranteed motion planning through quadratic programming. This concept serves as a building block for the control formulations used in later chapters.

Next, we explored a progression of coverage control strategies. Starting from classi-

cal formulations based on Voronoi partitioning and Lloyd’s algorithm, we extended the discussion to persistent coverage control, which accounts for time-varying sensing requirements. We then introduced angle-aware coverage control, which incorporates viewpoint diversity to enhance the quality of vision-based tasks such as 3D reconstruction. Finally, we examined a more advanced formulation that integrates camera rotation control, allowing agents to simultaneously optimize their position and viewing orientation.

On the 3D reconstruction side, we reviewed both offline and online techniques. The offline pipeline, comprising Structure-from-Motion and Multi-View Stereo, provides high-accuracy reconstructions but lacks real-time capability. In contrast, recent learning-based approaches such as NeuralRecon, Instant NeRF, and Gaussian Splatting support online or near real-time reconstruction, enabling dynamic updates and interaction with robotic agents.

The tools and methods introduced in this chapter establish the conceptual and algorithmic basis for the efficient angle-aware and scene-adaptive coverage control frameworks proposed in the following chapters.

Chapter 3

Efficient Angle-Aware Coverage Control

3.1 Introduction

This chapter presents the efficient angle-aware coverage control method for coordinated image sampling using multi-drone systems, as introduced in our prior work [43, 125]. The proposed method addresses key limitations in conventional angle-aware coverage control [24], particularly for large-scale 3D map reconstruction tasks. To this end, we extend the baseline angle-aware controller by incorporating a Voronoi-based nominal input into the QP formulation, enabling improved mission completion efficiency. The effectiveness of the proposed method is then demonstrated through simulation conducted in a ROS-based environment. Subsequently, experimental validation is also performed both indoors at the Science Tokyo Robot Zoo Sky testbed and outdoors at the Smart Agriculture R&E Field of Science Tokyo.

3.2 Problem Formulation

As briefly introduced in Chapter 2 (see Section 2.3.3), we consider a team of n drones deployed to monitor a field by capturing images from diverse viewing angles. This task aims to support 3D map reconstruction using techniques such as Structure from Motion (SfM), where angular diversity in the observed images significantly enhances the reconstruction quality.

Let $\mathcal{I} := \{1, 2, \dots, n\}$ denote the index set of drones, each equipped with a fixed downward-facing camera. Each drone's position in the world frame Σ_w is represented as $[x_i \ y_i \ z_i]^T \in \mathbb{R}^3$. For simplicity, all drones maintain a constant altitude z_c , reducing the control problem to the 2D position $p_i := [x_i \ y_i]^T \in \mathcal{P} \subseteq \mathbb{R}^2$. The position p_i must

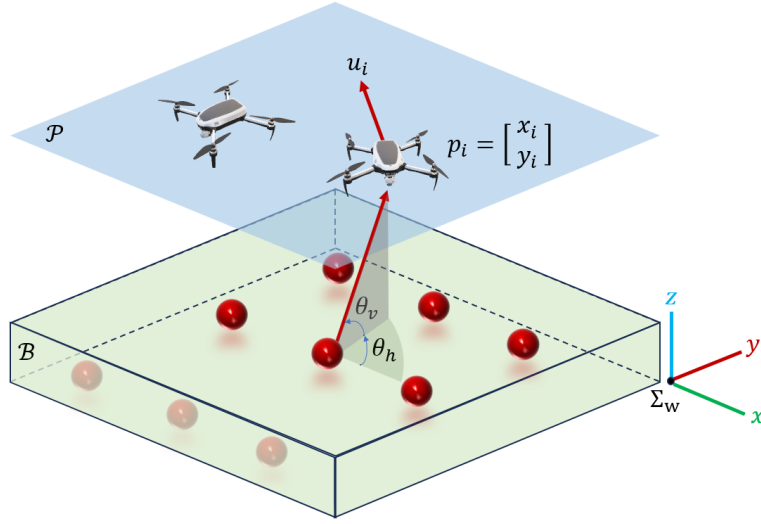


Fig. 3.1: Illustration of the angle-aware coverage problem.

remain within the operational region $\mathcal{P}$, and the drone dynamics are given by:

$$\dot{p}_i = u_i, \quad u_i \in \mathcal{U} \subseteq \mathbb{R}^2,$$

where u_i is the velocity control input to be designed.

Let $\mathcal{B} \subset \mathbb{R}^3$ denote the target field, a compact surface region that needs to be observed. Each point $[x \ y \ z]^T \in \mathcal{B}$ is ideally to be observed from multiple directions, characterized by a horizontal angle $\theta_h \in \Theta_h \subseteq [-\pi, \pi)$ and a vertical angle $\theta_v \in \Theta_v \subseteq (0, \pi/2]$ (See Figure 3.1). To represent both spatial and angular dimensions, we define a virtual 5D field $\mathcal{Q}_c \subset \mathbb{R}^5$, where each point is given by $(x, y, z, \theta_h, \theta_v)$.

The corresponding drone position p_i required to view a target $[x, y, z]^T$ from the angles θ_h and θ_v is uniquely determined via the mapping:

$$\zeta : [x \ y \ z \ \theta_h \ \theta_v]^T \mapsto \begin{bmatrix} x - (z_c - z) \tan\left(\frac{\pi}{2} - \theta_v\right) \cos(\theta_h) \\ y - (z_c - z) \tan\left(\frac{\pi}{2} - \theta_v\right) \sin(\theta_h) \end{bmatrix}.$$

We define the monitoring performance function $h : \mathcal{P} \times \mathcal{Q}_c \rightarrow [0, 1]$ that quantifies how well a drone at p_i observes a target point $q \in \mathcal{Q}_c$, using a design parameter $\sigma > 0$:

$$h(p_i, q) := \exp\left(-\frac{\|p_i - \zeta(q)\|^2}{2\sigma^2}\right).$$

Next, we discretize the virtual field $\mathcal{Q}_c$ into m cells, each with a representative center $q_j \in \mathcal{Q}_c$ for $j \in \mathcal{M} := \{1, 2, \dots, m\}$, and assume all cells share equal volume A . Each cell is assigned an importance index $\phi_j \in [0, \infty)$, which quantifies the remaining need for observation. This index evolves dynamically according to:

$$\dot{\phi}_j = -\delta \max_{i \in \mathcal{I}} h(p_i, q_j) \phi_j,$$

where $\delta > 0$ is a constant rate of decay.

The control objective is to minimize the total weighted importance index across all target cells:

$$J := \sum_{j=1}^m \phi_j A. \quad (3.1)$$

Driving J toward zero implies that all target points have been sufficiently observed from various viewing directions. A secondary control goal is to encourage drones to vacate areas that are already well-covered (low ϕ_j) and to stay longer in regions that require further observation (high ϕ_j), thus improving overall sampling efficiency.

3.3 Baseline Controller Method & Drawback

3.3.1 Baseline Angle-Aware Coverage Controller

To achieve the objectives defined in the previous section, the authors of [24] presented a controller that enforces the constraint $\dot{J} \leq -\gamma$ based on the concept of control barrier functions. The time derivative of J follows:

$$\begin{aligned} \dot{J} &= \sum_{j=1}^m \dot{\phi}_j A = - \sum_{j=1}^m \delta \max_{i \in \mathcal{I}} h(p_i, q_j) \phi_j A \\ &= - \sum_{i=1}^n \sum_{j \in \mathcal{V}_i(p)} \delta h(p_i, q_j) \phi_j A = - \sum_{i=1}^n I_i, \end{aligned}$$

where

$$I_i := \sum_{j \in \mathcal{V}_i(p)} \delta h(p_i, q_j) \phi_j A$$

corresponds to the contribution by drone i to reduce J . The set $\mathcal{V}_i(p)$, which depends on $p := (p_i)_{i \in \mathcal{I}}$, is a Voronoi-like partition of the set $\mathcal{M}$ defined as:

$$\mathcal{V}_i(p) := \{j \in \mathcal{M} \mid \|p_i - \zeta(q_j)\| \leq \|p_k - \zeta(q_j)\| \ \forall k \in \mathcal{I}\}.$$

Now, the number of cells m tends to be huge due to the higher dimension of the field $\mathcal{Q}_c$ than in standard coverage control [29, 72]. This makes implementation of the controller computationally very expensive. To address this issue, the drone field $\mathcal{P}$ is discretized into l polygons, denoted as $\{\mathcal{P}_k \mid k \in \mathcal{L}\}$, where $\mathcal{L} := \{1, \dots, l\}$. We also denote the coordinates of the center of the k -th polygon by $\chi_k \in \mathcal{P}_k \subset \mathcal{P}$. The coverage, represented by q_j , is then assessed by projecting it onto one of these polygons, specifically $\mathcal{P}_k$ where $\zeta(q_j)$ is situated.

This process leads to the transformation of $\phi := (\phi_j)_{j \in \mathcal{M}}$ into $\psi := (\psi_k)_{k \in \mathcal{L}}$, $\psi_k \in [0, \infty)$, as governed by the following equation:

$$\psi_k = \sum_{j \in \mathcal{M} \text{ s.t. } \zeta(q_j) \in \mathcal{P}_k} \phi_j.$$

Under the approximation $\zeta(q_j) \approx \chi_k$, the update of ψ_k is approximated as:

$$\begin{aligned} \dot{\psi}_k &= \sum_{j \in \mathcal{M} \text{ s.t. } \zeta(q_j) \in \mathcal{P}_k} \dot{\phi}_j \\ &= \sum_{j \in \mathcal{M} \text{ s.t. } \zeta(q_j) \in \mathcal{P}_k} -\delta \max_{i \in \mathcal{I}} h(p_i, q_j) \phi_j \approx -\delta \max_{i \in \mathcal{I}} \bar{h}(p_i, \chi_k) \psi_k, \end{aligned} \quad (3.2)$$

where $\bar{h}$ is the approximate performance function using χ_k :

$$\bar{h}(p_i, \chi_k) := \exp \left(-\frac{\|p_i - \chi_k\|^2}{2\sigma^2} \right).$$

Consequently, in view of $J = \sum_k \psi_k A$, the time derivative of J becomes:

$$\dot{J} = \sum_{k=1}^l \dot{\psi}_k A \approx - \sum_{i=1}^n \sum_{k \in \bar{\mathcal{V}}_i(p)} \delta \bar{h}(p_i, \chi_k) \psi_k A,$$

where

$$\bar{\mathcal{V}}_i(p) := \{k \in \mathcal{L} \mid \|p_i - \chi_k\| \leq \|p_i - \chi_k\| \ \forall i \in \mathcal{I}\}.$$

Finally, the controller in [24] is formulated as the following QP:

$$\begin{aligned} (u_i^*, w_i^*) &= \arg \min_{(u_i, w_i) \in \mathcal{U} \times \mathbb{R}} \epsilon \|u_i\|^2 + |w_i|^2, \\ \text{s.t.} \quad &\bar{\xi}_{1i}^T u_i + \bar{\xi}_{2i} \geq w_i \end{aligned}$$

where w_i is a slack variable ($w_i \geq 0$), $\epsilon > 0$ is a weight for penalizing control effort, and:

$$\begin{aligned} \bar{\xi}_{1i} &:= \sum_{k \in \bar{\mathcal{V}}_i(p)} -\delta \frac{p_i - \chi_k}{\sigma^2} \bar{h}(p_i, \chi_k) \psi_k A, \\ \bar{\xi}_{2i} &:= -a\gamma + \sum_{k \in \bar{\mathcal{V}}_i(p)} (-\delta^2 \bar{h}^2(p_i, \chi_k) + a\delta \bar{h}(p_i, \chi_k)) \psi_k A. \end{aligned}$$

3.3.2 Drawback in Large-Scale Map Reconstruction

To evaluate the performance of the baseline angle-aware coverage controller defined in equation (3.3) within a large-scale setting, we conducted a simulation using the Robot Operating System (ROS) Noetic [126], solving the associated quadratic programming

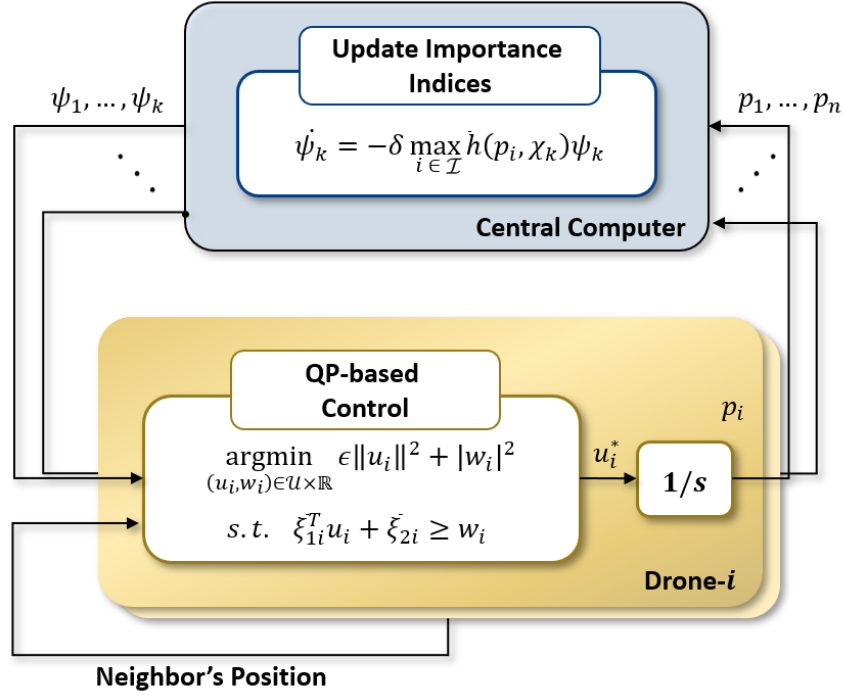


Fig. 3.2: Control architecture for baseline angle-aware coverage controller.

problem using CVXOPT [127]. The simulation models a 3D monitoring task over an expansive area, with a focus on the controller's ability to efficiently reduce the importance index ψ under realistic constraints. The detailed parameters used in this simulation are summarized in Table 3.1.

Figure 3.3a shows the layout of the large-scale monitoring field used in the simulation. Figure 3.3b presents the time evolution of the importance index ψ under the baseline controller (3.3). While the drone gradually reduces unobserved regions, high- ψ patches remain scattered across the field. These sparse regions are often far from the drone's current position, resulting in long travel distances for minimal coverage gains.

Figure 3.4 shows the evolution of the cost function J . Although the function generally decreases, its rate of decay slows significantly after $t = 600$ s. This slowdown is due to the increasing activation of slack variables as the drone must travel greater distances to observe remaining unmonitored patches. These patches are not explicitly prioritized in the original objective function, resulting in inefficient exploratory behavior.

This drawback is particularly problematic for large-scale monitoring missions, where efficient global coverage is essential. While this limitation was not clearly visible in earlier small-scale experiments such as those in Science Tokyo Robot Zoo Sky [40], it becomes significantly pronounced in expanded environments.

The next section addresses this issue by proposing a controller that encourages more

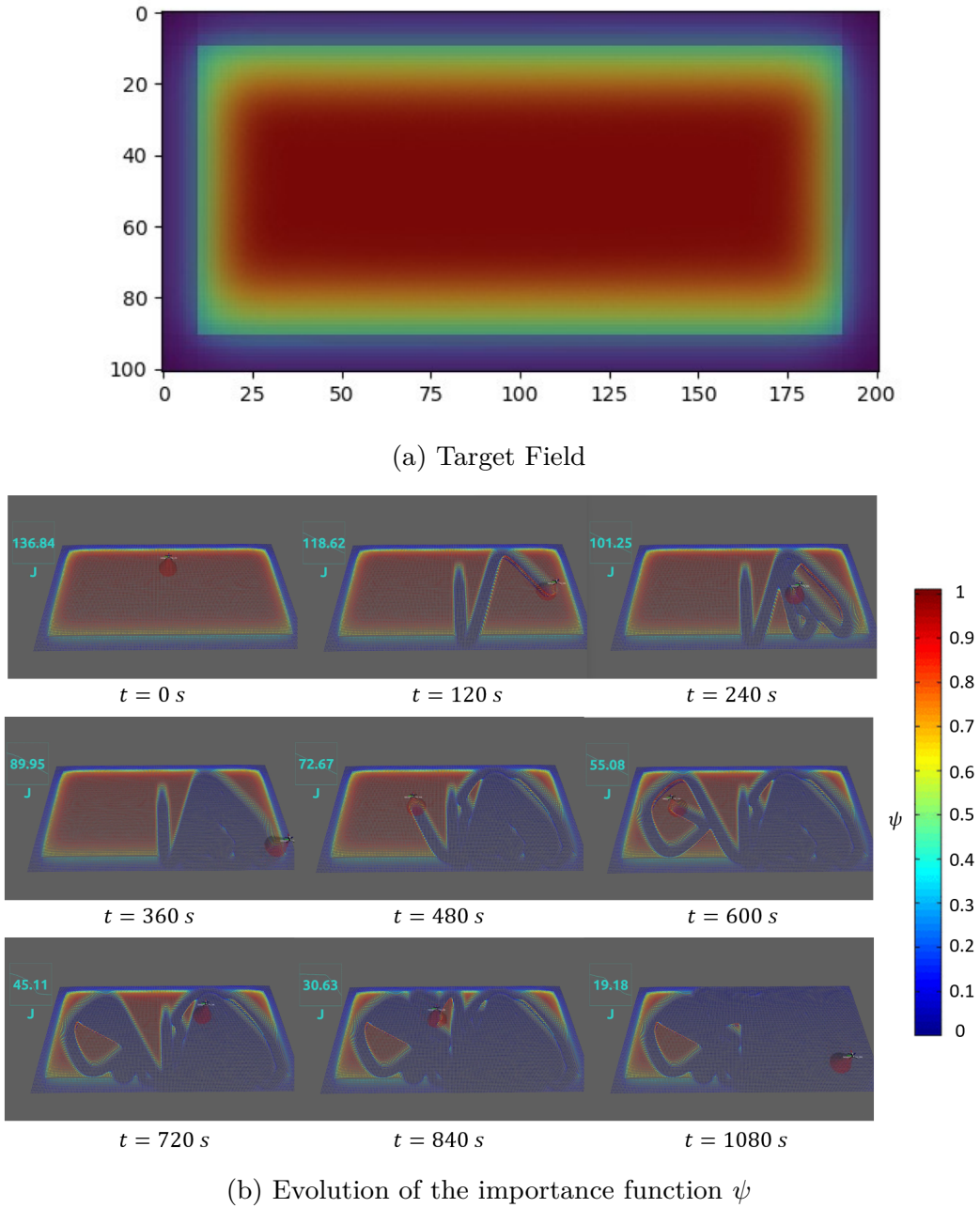


Fig. 3.3: ROS simulation of baseline angle-aware coverage controller (3.3). This simulation can be viewed [here](#).

Table. 3.1: Simulation parameters for baseline angle-aware evaluation

Parameter	Value
Drone altitude z_c	1.0 m (fixed)
Velocity bounds $\mathcal{U}$	$[-0.6, 0.6]$ m/s per axis
Horizontal viewing angle range Θ_h	$[-\pi, \pi)$
Vertical viewing angle range Θ_v	$[\pi/3, \pi/2]$
Sensing scale σ	0.1
Importance decay rate δ	5.0
Virtual field size $ \mathcal{Q}_c $	1.5×10^7 cells
Virtual cell size	$0.02 \times 0.02 \times 0.1 \times \frac{\pi}{30} \times \frac{\pi}{30}$
Cell volume A	$\frac{4\pi^2}{9} \times 10^{-7} \text{ m}^3 \text{ rad}^2$
Monitoring area $\mathcal{P}$	$6.0 \times 12.0 \text{ m}^2$
$\mathcal{P}$ discretization	10^4 cells ($0.03 \times 0.03 \text{ m}^2$)
Penalty parameter ϵ	0.0001
Barrier constraint margin γ	0.15
Coefficient a	1.0

efficient exploration, particularly in large-scale 3D map reconstruction scenarios.

3.4 Controller Design for Large-Scale Map Reconstruction

This section presents a novel controller suitable for the large-scale map reconstruction. In the previous section, we hypothesized that the small pieces of the unobserved area were due to the lack of evaluating the distance between $\chi_k, k = 1, 2, \dots, l$ and p_i . We thus reflect this factor in the controller design in this section. A trivial idea could be to modify the cost function J so that it explicitly involves the distance. However, after conducting extensive research, the results did not meet our expectations for achieving efficient coverage. Specifically, the drones were stuck to specific positions without continuing exploratory actions. We thus approach the problem differently, namely we employ the nominal control input u_{nom} and reflect the distance in the design of u_{nom} . The primary motivation behind the introduction of u_{nom} is to enable the drones not only to reduce the objective function J in (3.1) but also to be attracted to locations with higher and denser importance values, even when these locations are distant from the drone's current position. Consequently, in the context of large-scale monitoring, the drones are expected to

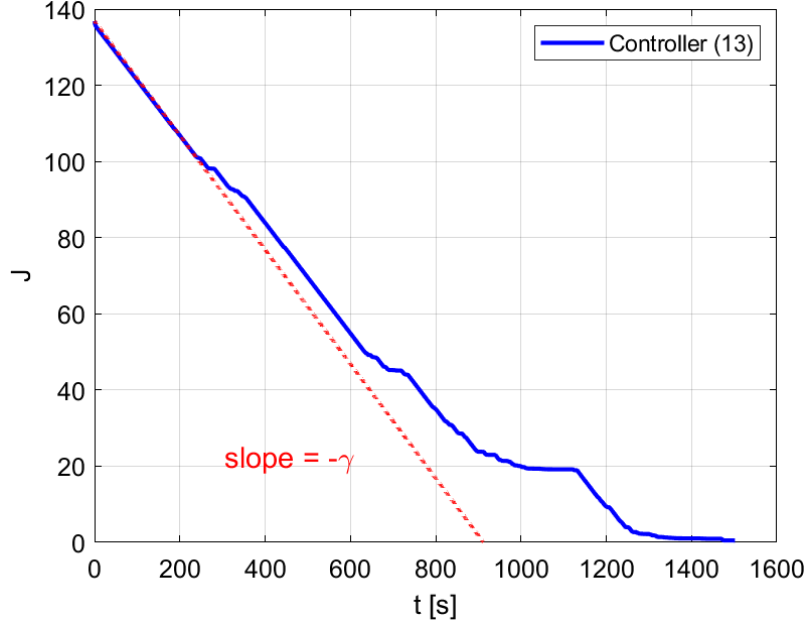


Fig. 3.4: Cost function J for monitoring using the baseline controller (3.3). The red dashed line indicates the slope $-\gamma$.

behave more efficiently, while also avoiding unnecessary extensive travel to inspect small unobserved areas.

We construct the nominal control based on the coverage cost function introduced in [28]:

$$J_c = \int_{q \in \mathcal{P}} \left(\min_{i \in \mathcal{I}} \|p_i - q\|^2 \right) \psi_c(q) dq = \sum_{i=1}^n \int_{q \in \mathcal{V}_i^c(p)} \|p_i - q\|^2 \psi_c(q) dq,$$

where $\mathcal{V}_i^c(p)$ is the Voronoi region of drone i over $\mathcal{P}$, defined as

$$\mathcal{V}_i^c(p) := \{q \in \mathcal{P} \mid \|p_i - q\| \leq \|p_k - q\| \quad \forall k \in \mathcal{I}\},$$

and $\psi_c : \mathcal{P} \rightarrow [0, \infty)$ is defined by $\psi_c(q) = \psi_k$ if $q \in \mathcal{P}_k$.

Define the centroid of each Voronoi region as:

$$\text{cent}(\mathcal{V}_i^c(p)) = \frac{\int_{\mathcal{V}_i^c(p)} q \psi_c(q) dq}{\int_{\mathcal{V}_i^c(p)} \psi_c(q) dq}.$$

The gradient of J_c with respect to p_i is given by [28]:

$$\frac{\partial J_c}{\partial p_i} = \left(\int_{\mathcal{V}_i^c(p)} \psi_c(q) dq \right) (p_i - \text{cent}(\mathcal{V}_i^c(p))).$$

This motivates a move-to-centroid control strategy based on the Lloyd descent algorithm [93], in which each drone moves toward the centroid of its importance-weighted Voronoi region.

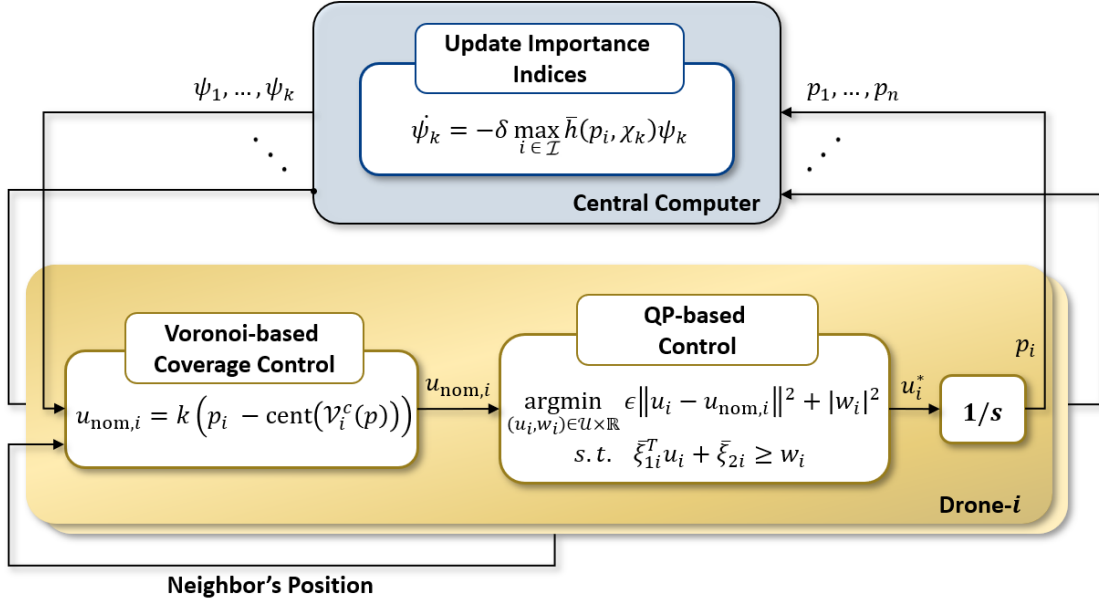


Fig. 3.5: Overall control architecture integrating voronoi-based nominal input into QP-based control.

We define the nominal input as:

$$u_{\text{nom},i} = k(p_i - \text{cent}(\mathcal{V}_i^c(p))), \quad i \in \mathcal{I},$$

where $k > 0$ is a control gain. In practice, we approximate the centroid as:

$$\text{cent}(\mathcal{V}_i^c(p)) \approx \sum_{k: \chi_k \in \mathcal{V}_i^c(p)} \psi_k \chi_k.$$

Here, ψ_k are the updated importance indices as defined by (3.2).

We now integrate the nominal input into the QP-based control framework. The original QP controller in (3.3) is reformulated as follows:

$$\begin{aligned} (u_i^*, w_i^*) = & \arg \min_{(u_i, w_i) \in \mathcal{U} \times \mathbb{R}} \epsilon \|u_i - u_{\text{nom},i}\|^2 + |w_i|^2, \\ \text{s.t. } & \bar{\xi}_{1i}^T u_i + \bar{\xi}_{2i} \geq w_i. \end{aligned}$$

This controller maintains the certified performance constraint $\dot{J} \leq -\gamma$ while guiding the system with the aid of the distance-aware nominal control input $u_{\text{nom},i}$.

The full control pipeline is illustrated in Fig. 3.5. The centralized computation updates ψ based on coverage history, while the Voronoi regions and QP solutions are executed in a distributed fashion by each drone. As discussed in [39], the centralized update step is computationally feasible for many applications.

Table. 3.2: Simulation parameters for single-drone evaluation

Parameter	Value
Number of drones	$n = 1$
Initial position	$p_1 = [1.0 \ 0.2]^T$ m
Altitude	1.0 m (fixed)
Velocity bounds	$[-0.6, 0.6]$ m/s per axis
Horizontal view angle (Θ_h)	$[-\pi, \pi)$
Vertical view angle (Θ_v)	$[\pi/3, \pi/2]$
Virtual field resolution	1.5×10^7 cells
Polygonal map resolution ($\mathcal{P}$)	10^4 polygons of 0.03×0.03 m ²
Other parameters	$\sigma = 0.1, \epsilon = 10^{-4}, \gamma = 0.15, \delta = 5, a = 1$

3.5 Demonstration on a ROS Simulation

This section presents a validation of the proposed controller (3.5) through simulation experiments conducted in a ROS Noetic environment using the same setup described in Section 3.3.2. We evaluate the performance of the controller in both single-drone and multi-drone scenarios to demonstrate its scalability and efficiency.

Remark 1. In the ROS simulation, the drone dynamics are modeled using the single integrator assumption, which aligns with the controller design.

3.5.1 Evaluation for a Single-Drone Case

We first examine the performance of the proposed controller using a single drone ($n = 1$). The simulation parameters used are summarized in Table 3.2. These settings are consistent with the earlier evaluation in Section 3.3.2.

In Figure 3.6, snapshots of the simulation for both of (3.3) and (3.5) are presented, featuring a color map depicting the importance index ψ . On closer inspection, it becomes apparent that the proposed controller (3.5) leads the drone to cover the region in a more organized fashion, prioritizing areas nearer to the Voronoi region’s centroid and gradually progressing towards other denser regions. In contrast, the original controller (3.3) leads the drone to a more random coverage pattern, resulting in scattered unobserved patches across the field after a while. Consequently, the proposed controller achieves more efficient

Table. 3.3: Simulation parameters for multi-drone evaluation

Parameter	Value
Number of drones	$n = 3$
Initial positions	$p_1 = [1.0 \ 0.2]^T$ m, $p_2 = [-1.0 \ 2.0]^T$ m, $p_3 = [-1.0 \ -2.0]^T$ m
Altitude	1.0 m (fixed)
Velocity bounds	$[-0.6, 0.6]$ m/s per axis
Horizontal view angle (Θ_h)	$[-\pi, \pi)$
Vertical view angle (Θ_v)	$[\pi/3, \pi/2]$

area coverage and quicker task completion compared to the previous controller (3.3).

Furthermore, Figure 3.7 also confirms that the proposed controller (3.5) achieves a faster overall reduction rate of the cost function J while avoiding frequent constraint violations in the transient, compared to the previous controller (3.3). Specifically, the controller (3.5) reaches $J \approx 0$ at $t \approx 900s$, whereas the controller (3.3) needs about $t \approx 1300s$ for completing $J \approx 0$. Consequently, based on the above simulation, we can infer that the proposed controller not only covers the area more efficiently but also exhibits a faster reduction in J than the previous one.

Note that some readers may think that just using enough number of drones can overcome the drawback of the controller (3.3) as long as the system cost would be ignorable. Actually, when we use multiple drones, the total area to be monitored by each drone gets small. It is thus beneficial to confirm that the above advantage of the proposed controller is valid for the multi-drone scenario also, which will be investigated in the next subsection.

3.5.2 Evaluation for a Multi-Drone Case

We now evaluate the proposed controller using three drones ($n = 3$), initialized as listed in Table 3.3. This setting tests the scalability and coordination performance of the proposed method.

Figure 3.8 shows the snapshot comparisons between (3.3) and (3.5). Due to the smaller areas to be sampled by each drone, the speed to complete the mission gets faster than the single-drone case. However, we also see that small unobserved areas remain in the left, which slows down the overall coverage in the same way as Figure 3.6. Meanwhile, drones employing our proposed controller systematically cover the region. This observation is further supported by the data in Figure 3.9, which indicates that our controller (3.5) achieves faster area coverage, even faster than the specified decay rate. Consequently, it

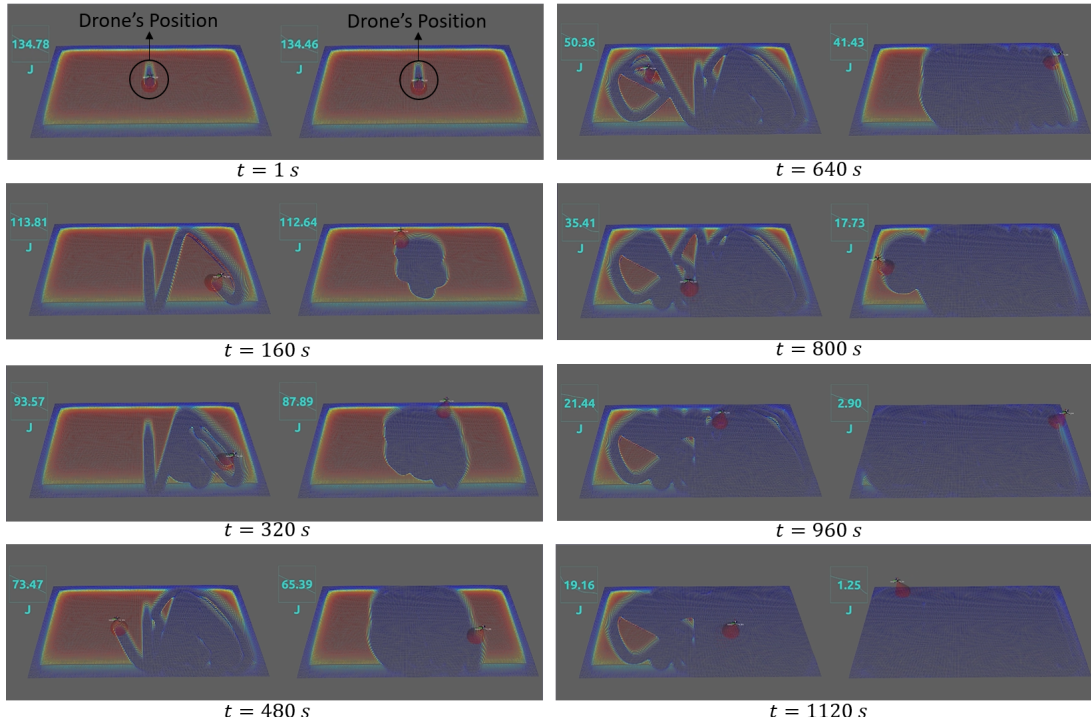


Fig. 3.6: Snapshots of the importance index ψ for the controller (3.3) (left) and the proposed controller (3.5) (right) using a single drone. Full simulation available [here](#).

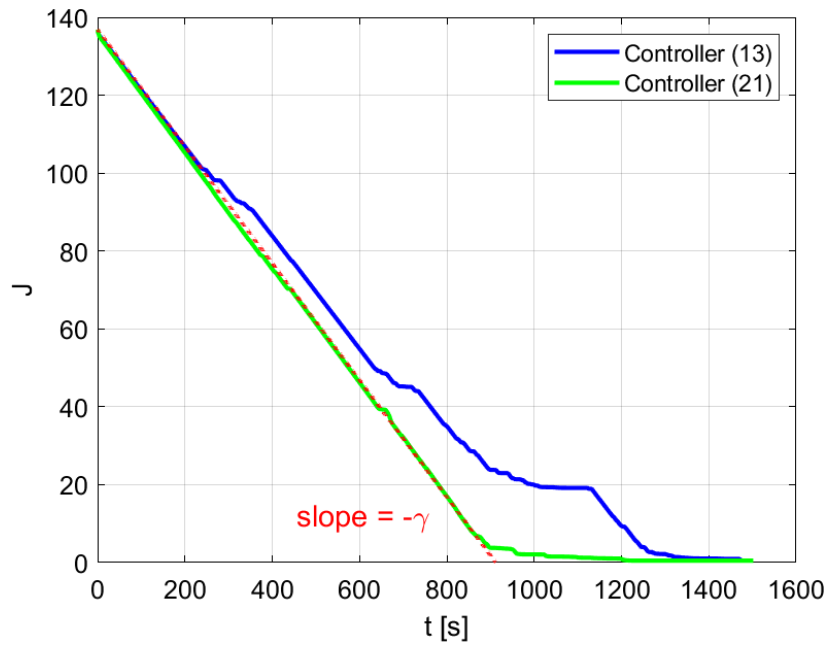


Fig. 3.7: Comparison of cost function J for the controller (3.3) (blue) and proposed controller (3.5) (green). The red dashed line indicates slope $-\gamma$.

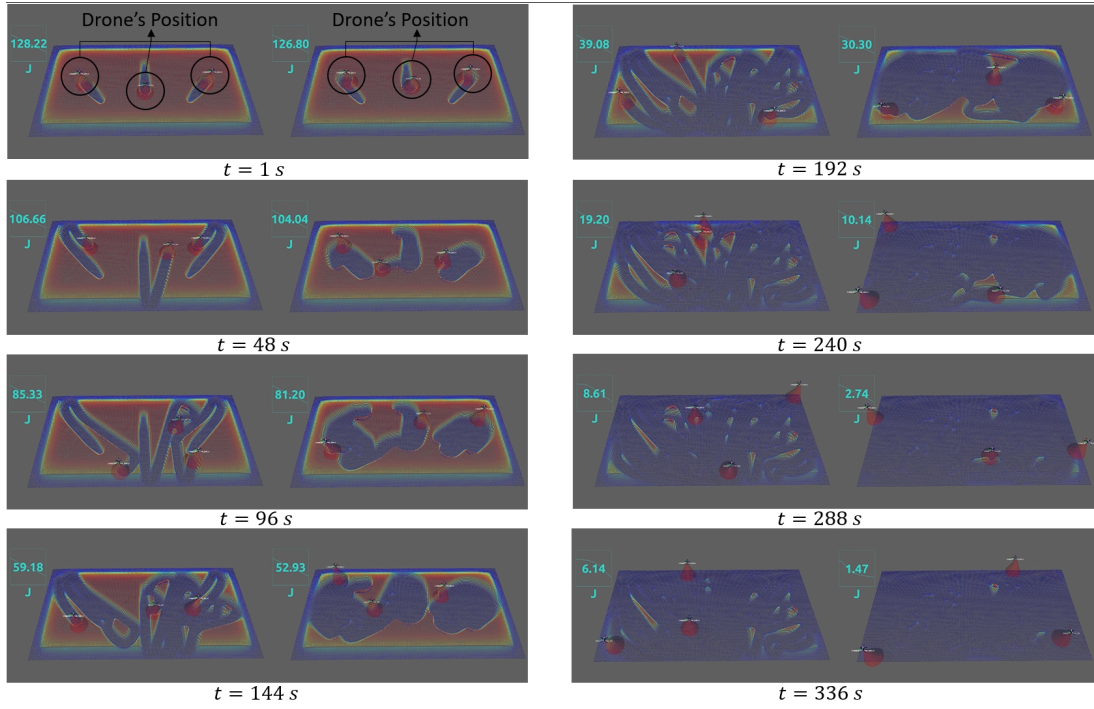


Fig. 3.8: Snapshots of the importance index ψ for the controller (3.3) (left) and the proposed controller (3.5) (right) using multiple drones. Full simulation available here.

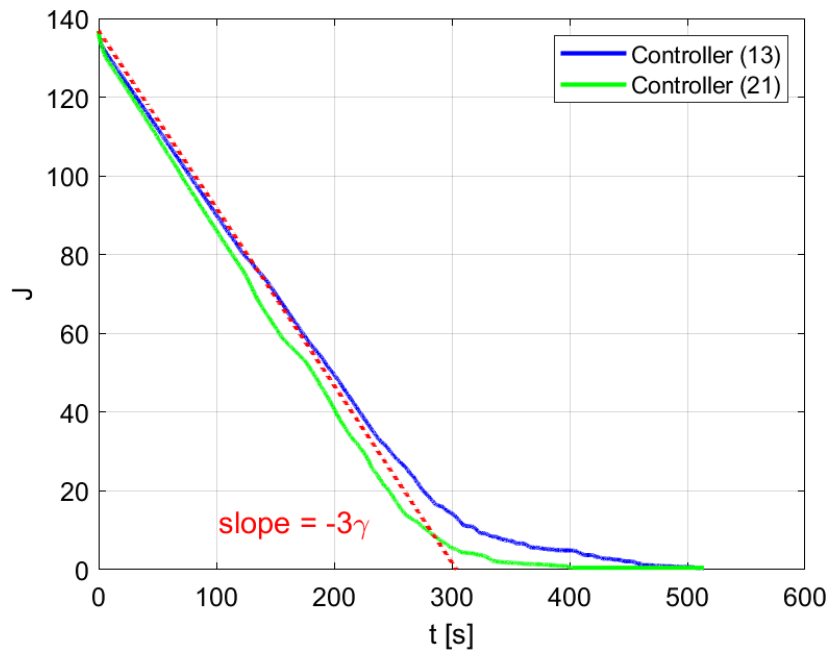


Fig. 3.9: Comparison of cost function J for the controller (3.3) (blue) and the proposed controller (3.5) (green) in a multi-drone setting. Red dashed line indicates slope -3γ .

takes only about $t = 330$ seconds to achieve $J \approx 0$, whereas the previous controller (3.3) requires $t = 470$ seconds.

In summary, we conclude that the proposed controller is beneficial even for the multi-drone case.

3.6 Experiments: Indoor and Outdoor Validation

3.6.1 Indoor Experiment

This section presents real-world experiments to validate the effectiveness of the proposed controller (3.5). The experiments are conducted in the Tokyo Tech Robot Zoo Sky testbed. Figure 3.10a shows the environment, and Figure 3.10b illustrates the overall experimental setup.

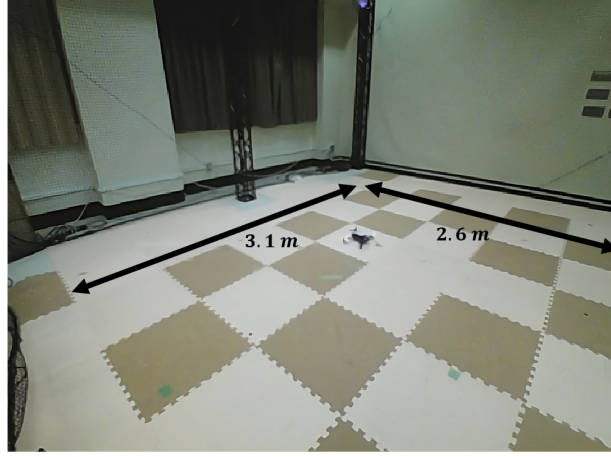
Due to the limited size of the testbed, we virtually simulate a larger target field by reducing the drone's altitude and perception scale. Specifically, the altitude is reduced to 30% of the simulation setting, while the sensing scale σ is adjusted to 0.7 of the original value. These adjustments ensure that the ratio between the drone's field of view and the monitored area remains consistent with the simulation. For safety reasons, the drone's speed is also limited.

The hardware consists of Parrot Bebop 2 drones, a desktop computer (Intel® Core™ i7-8700K, 32GB RAM), and laptops (Intel® Core™ i7-8650U, 8GB RAM). A motion capture system using OptiTrack streams 120 Hz drone position data to the desktop, where the importance function ψ is updated. Each drone receives its control input from a laptop performing distributed computation, since the onboard processor can only handle low-level velocity commands.

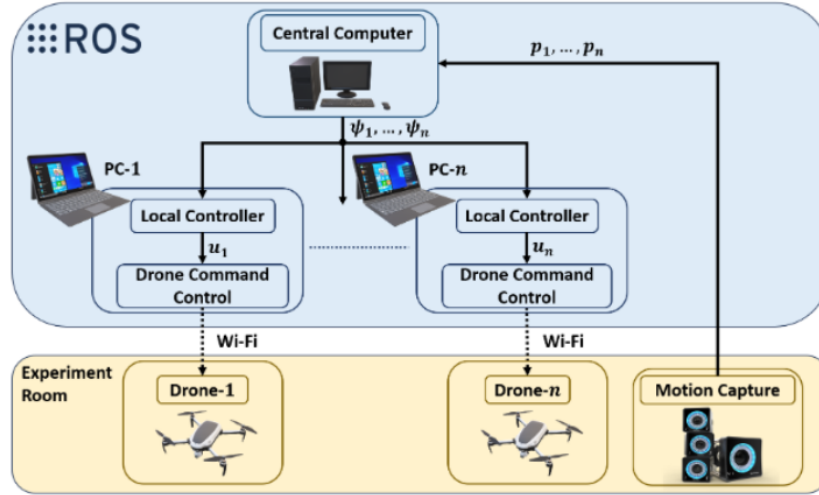
Remark 2. In the experiment, we acknowledge the gap between the single integrator dynamics and the actual drone dynamics. To address this disparity, we designed a local velocity controller for the drone so that the drone velocity follows velocity commands from the high-level controller. Despite this remedy, the actual velocity, of course, does not coincide with the velocity command especially over a high frequency domain. On the other hand, in general, cooperative control, including coverage control, does not intend rapid control as in low-level robot motion control, e.g., in a factory. It is thus expected that the gap between the velocity and its command will not cause major problems.

The parameters used in this experiment are listed in Table 4.2. These settings reflect necessary adaptations from the simulation to the real-world environment.

The experimental results are illustrated in Figure 3.11. The drone controlled by the



(a) Overview of the experimental room



(b) Schematic diagram of the experiment

Fig. 3.10: Overview and schematic diagram of the experimental testbed.

proposed method achieves more systematic coverage compared to the baseline. This is consistent with the results observed in simulation. As shown in Figure 3.12, the proposed controller reaches $J \approx 0$ around $t = 150$ s, whereas the original controller takes $t \approx 180$ s. Although the improvement margin is smaller than in simulation due to the limited field size and uncertainty in the real world, the proposed method still demonstrates a clear advantage.

Although smaller values of σ limit the drone's sensing capability and increase coverage difficulty, the proposed controller still outperforms the baseline. This highlights its effectiveness in realistic constrained settings. Despite the compact testbed (2.2×2.2 m²), these results suggest promising scalability to larger outdoor scenarios, with the added benefit of reduced mission time and energy consumption.

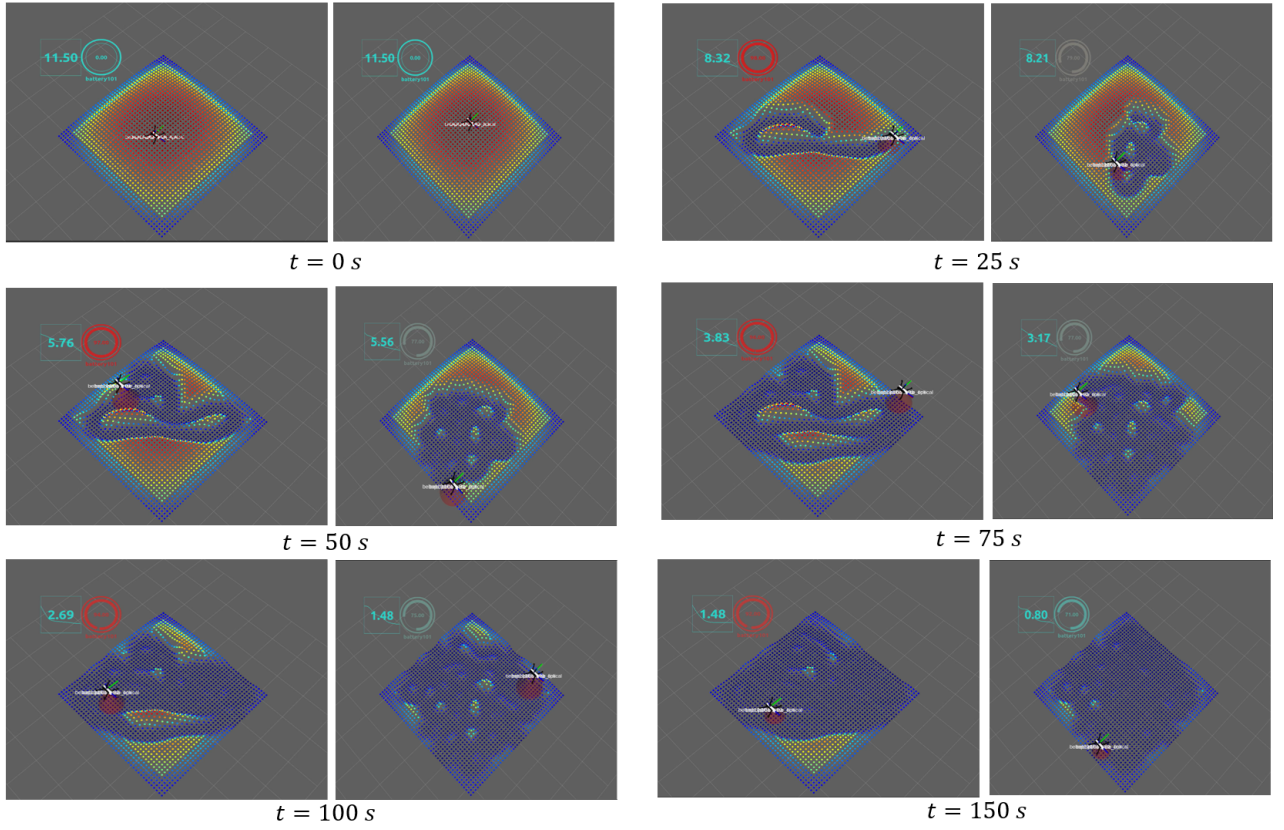


Fig. 3.11: Snapshots of the importance index ψ for the controller (3.3) (left) and proposed controller (3.5) (right) during indoor experiments. Video: [link](#).

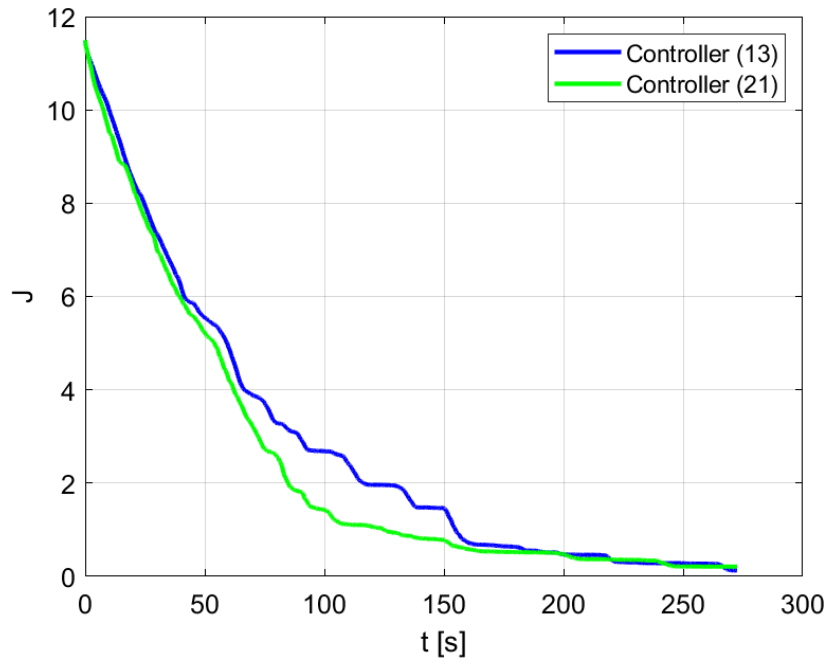


Fig. 3.12: Cost function J comparison between the controller (3.3) (blue) and proposed controller (3.5) (green) in the indoor experiment.

Table. 3.4: Experimental parameters for indoor validation

Parameter	Value
Number of drones	$n = 1$
Initial position	$p_1 = [1.0 \ 0.2]^T$ m
Altitude	0.3 m (fixed)
Velocity bounds	$[-0.2, 0.2]$ m/s per axis
Horizontal view angle (Θ_h)	$[-\pi, \pi)$
Vertical view angle (Θ_v)	$[\pi/3, \pi/2]$
Target field volume	$[-1.1, 1.1] \times [-1.1, 1.1] \times [0.0, 0.5]$ m ³
Virtual field resolution	1.5×10^7 cells
Field discretization ($\mathcal{P}$)	10^4 polygons of 0.03×0.03 m ²
Other parameters	$\sigma = 0.07, \epsilon = 10^{-4}, \gamma = 0.15, \delta = 5, a = 1$

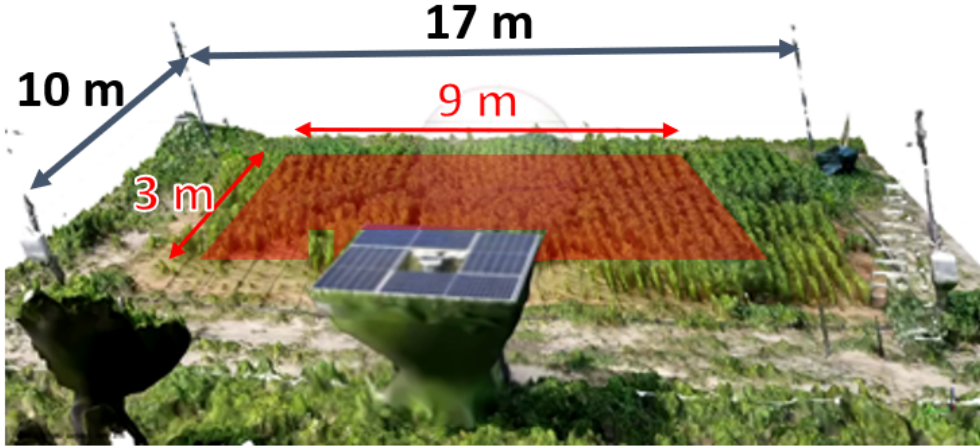


Fig. 3.13: Overview of the experimental farmland (Smart Agriculture R&E Field).

3.6.2 Outdoor Experiment

To evaluate the practical performance of the proposed controller in a real-world outdoor environment, we conducted an experiment at the Smart Agriculture R&E Field on the Suzukakedai campus of Science Tokyo. The field size was approximately $10 \text{ m} \times 17 \text{ m}$, as shown in Fig. 3.13.

The experiment utilized a DJI Mavic 3 Enterprise drone, equipped with an RTK-GNSS module for centimeter-level positioning accuracy. A DJI D-RTK 2 mobile base station was used as a reference, enabling high-precision localization by providing GNSS corrections. The system architecture is illustrated in Fig. 3.14, implemented using ROS2 environment

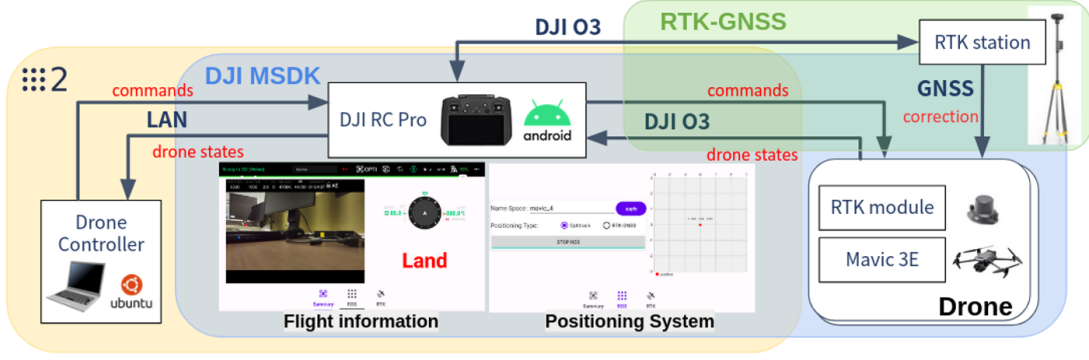


Fig. 3.14: System architecture for the outdoor experiment using DJI Mavic 3 Enterprise and RTK-GNSS.

[128]. Velocity commands were calculated in a central computer and sent to the drone via a DJI RC Pro Enterprise controller over LAN, using custom ROS2-compatible applications interfacing with the DJI Mobile SDK.

The drone's onboard camera captured 1440×1080 resolution images at 30 fps, and all computations were conducted on a workstation equipped with an Intel Core i9-12900H CPU (14 cores, 20 threads), 16 GB RAM, and an NVIDIA RTX 3060 GPU. Optimization was solved using CVXOPT with acceleration via JAX [129] for gradient evaluation.

Table 3.5 lists the key parameters used in this experiment.

We compared the performance of the proposed controller and the baseline controller using a single drone setup. As shown in Fig. 3.15, the proposed controller exhibits a more organized and efficient coverage pattern, with fewer remaining high- ψ regions. This leads to faster overall convergence of the coverage cost J , as plotted in Fig. 3.16. The proposed controller reduced J to near zero in approximately 600 seconds, while the baseline required around 900 seconds.

Finally, images collected during the flight were processed with the commercial SfM software Metashape (Agisoft LLC) to reconstruct a 3D map. The map reconstructed using the proposed method (Fig. 3.17) shows better coverage and scene completeness compared to that from baseline controller (Fig. 3.18), especially in the lower portions of the scene. These results validate the effectiveness of our controller in outdoor conditions for high-quality, efficient 3D mapping.

3.7 Chapter Summary

This chapter presented an efficient angle-aware coverage control framework for coordinated image sampling in large-scale 3D reconstruction using multi-drone systems. Build-

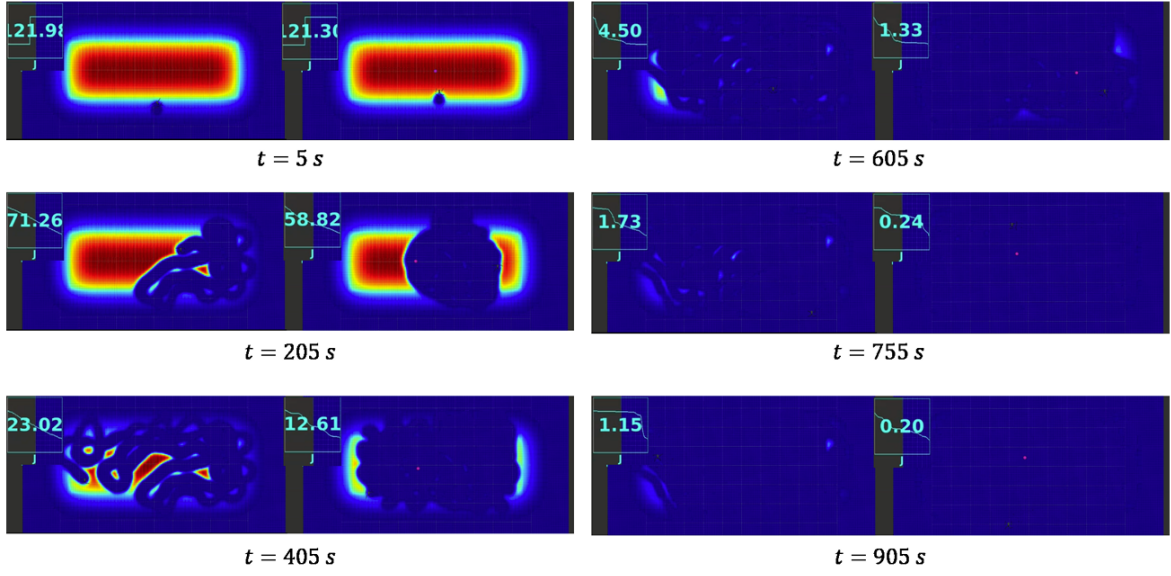


Fig. 3.15: Snapshots of the importance index ψ during the outdoor experiment. Left: baseline controller, Right: proposed controller.

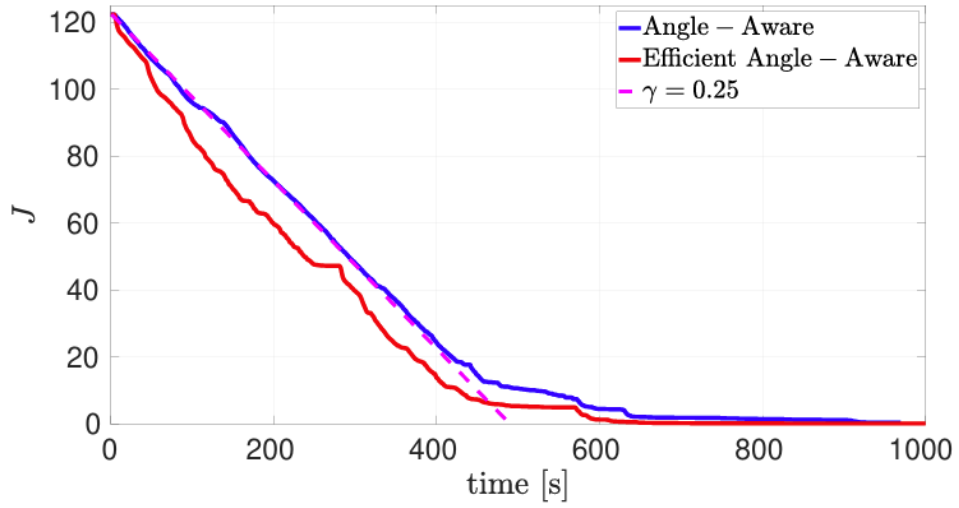


Fig. 3.16: Comparison of cost function J over time. Red: Proposed Controller, Blue: Baseline Controller. Dashed line: $\dot{J} = -\gamma$.

Table. 3.5: Experimental parameters for outdoor evaluation.

Parameter	Value
Drone model	DJI Mavic 3 Enterprise
Altitude z_c	2.0 m
Max velocity	0.6 m/s
Horizontal view angle Θ_h	$[-\pi, \pi)$
Vertical view angle Θ_v	$[\pi/3, \pi/2]$
Target field size	$[-1.5, 1.5] \times [-4.5, 4.5] \times [0, 1] \text{ m}^3$
Virtual field cell size	$0.03 \times 0.09 \times 0.1 \times \frac{\pi}{30} \times \frac{\pi}{30}$
Cell volume A	$\frac{2\pi^2}{1000} \text{ m}^3 \text{ rad}^2$
σ	0.15
ϵ	0.1
δ	5.0
γ	0.25
Nominal gain k	1.0
Safety distance d_{ca}	0.5 m

ing upon the baseline angle-aware controller, we introduced a Voronoi-based nominal input integrated into a QP-based formulation to improve spatial efficiency and scalability. Through simulation studies, the proposed method demonstrated faster convergence of the coverage objective and more balanced task allocation across drones. Indoor and outdoor experiments further validated the controller's robustness and effectiveness under real-world conditions. These results establish a solid foundation for the next chapter, where we incorporate real-time feedback from online 3D reconstruction to enable scene-adaptive coverage control.

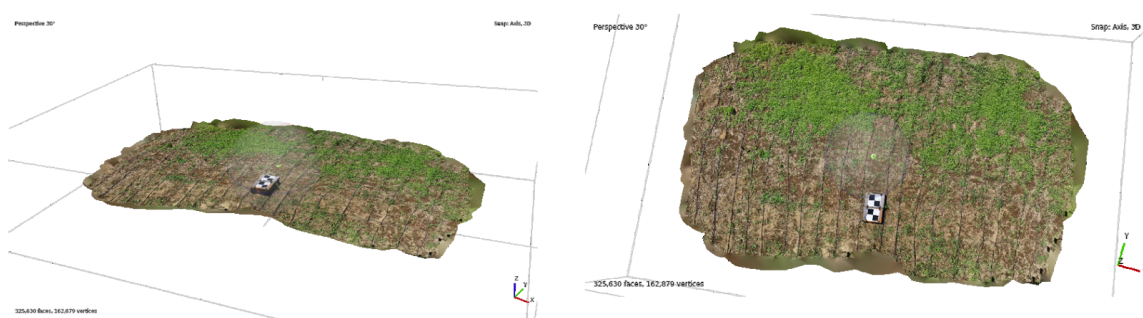


Fig. 3.17: 3D map reconstructed using efficient angle-aware coverage controller method (proposed).



Fig. 3.18: 3D map reconstructed using baseline angle-aware coverage controller method.

Chapter 4

Scene-Adaptive Angle-Aware Coverage Control

4.1 Introduction

This chapter presents a coordinated image sampling framework with real-time map feedback, as introduced in our recent work [130]. In the previous chapter, we considered an image sampling framework for 3D reconstruction based on angle-aware coverage control. In that setting, drones autonomously flew over the target field, captured images based on a pre-defined coverage control objective, and then landed. The collected images were subsequently used for 3D map reconstruction using offline Structure-from-Motion (SfM) techniques. During the flight, however, there was no feedback from the reconstruction process to inform or adapt the drone behavior on the fly.

With the development of online 3D reconstruction algorithms such as NeuralRecon [48], it is now possible to reconstruct the 3D map incrementally during flight. This enables drones to receive real-time feedback from the evolving mesh and adjust their trajectories accordingly. In this work, we refer to this capability as a “scene-adaptive” method, meaning that the drones adapt their control inputs based on online 3D reconstruction feedback. Specifically, the coverage density and sampling priorities are adjusted to revisit regions with significant mesh updates, improving reconstruction completeness. This is in contrast to conventional methods that rely on pre-defined coverage plans and do not change behavior based on reconstruction progress.

The proposed method builds upon the concept of rotatable angle-aware coverage control initially proposed in [41] (see Section 2.3.4), but introduces several key modifications. First, unlike [41], which assumes a fixed flight altitude, our formulation treats the altitude z as a controllable variable, resulting in a full 5D control framework that operates over a 5D virtual field. Second, we redefine the sensing performance function to better capture

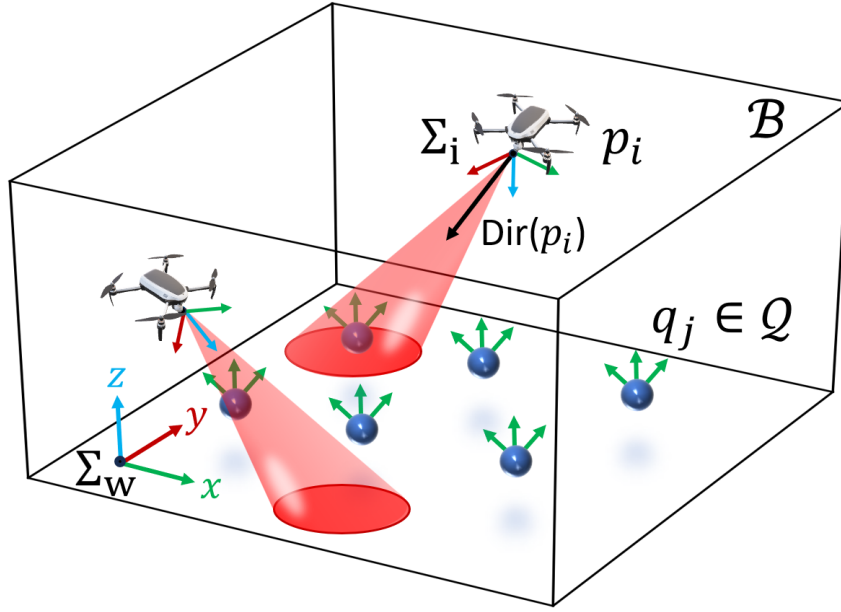


Fig. 4.1: Illustration of the virtual field $\mathcal{B}$ with the world frame Σ_w . The i -th drone observes point $q_j \in \mathcal{Q}$, adjusting its camera's yaw θ_i^h and pitch θ_i^v for optimal coverage.

the geometry of 3D reconstruction tasks. Specifically, our formulation takes into account the alignment between the camera's viewing direction and the surface normal, the distance to the observed point, and the angular resolution, thus differing from the performance metric used in [41].

The effectiveness of the proposed framework is then validated through both simulation in a ROS2 and Unity-based environment and real-world indoor experiments conducted at the Science Tokyo Robot Zoo Sky testbed.

4.2 Problem Formulation

This section introduces the sensing and coverage formulation used in this chapter, which enables each drone to actively control its onboard camera's orientation during flight. This modeling framework is particularly suited for image-based 3D reconstruction tasks, where the diversity of viewpoints plays a critical role in reconstructing accurate and complete meshes. The formulation captures the geometric relationship between drone pose, camera direction, and desired viewing perspectives over a discretized virtual field.

4.2.1 Drones, Virtual Field, and Geometry

We consider a system of n drones, indexed by $\mathcal{I} := \{1, \dots, n\}$, operating in three-dimensional Euclidean space. Each drone is treated as a rigid body, with two frames of

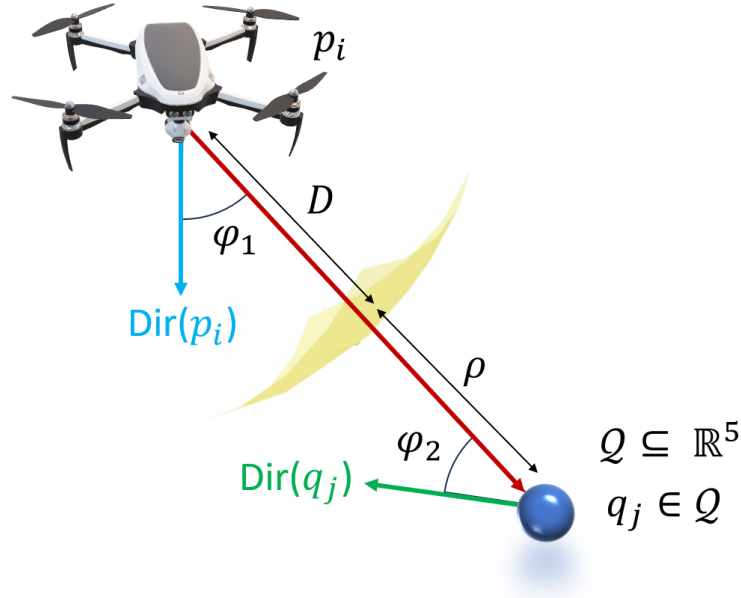


Fig. 4.2: Geometric relation between a drone p_i and point q_j .

reference: the world frame Σ_w and the body frame Σ_i , attached to the camera of the i -th drone. Note that Σ_i is arranged so that its z -axis is equivalent to the camera's optical axis. The drones are also equipped with a gimbal, allowing control of the camera's yaw and pitch angles. The state of each drone i is then represented by $p_i = [x_i \ y_i \ z_i \ \theta_i^h \ \theta_i^v]^T$, where $\theta_i^h \in \Theta^h = [-\pi, \pi]$ and $\theta_i^v \in \Theta^v = (0, \pi/2]$ represent the camera's yaw and pitch angles, respectively. The velocity input $u_i \in \mathcal{U} \subseteq \mathbb{R}^5$ governs the drone's motion, with $\dot{p}_i = u_i$.

The drones cooperate to observe and reconstruct structures within a predefined 3D field $\mathcal{B} \subset \mathbb{R}^3$. Effective 3D reconstruction requires capturing each point in $\mathcal{B}$ from multiple perspectives. To achieve this, the coverage is evaluated over a 5D space $\mathcal{Q}_c = \mathcal{B} \times \Theta^h \times \Theta^v$. We discretize $\mathcal{Q}_c$ into m cells, with a representative point of the j -th cell denoted by $q_j = [x_j \ y_j \ z_j \ \theta_j^h \ \theta_j^v]^T \in \mathcal{Q}_c$. The collection of these representative points, q_j for $j \in \mathcal{M} = \{1, 2, \dots, m\}$, is denoted by $\mathcal{Q}$. The illustration of the problem setup can be seen in Figure 4.1.

Next, we define the geometric components associated with the position, angle, and vector direction of the drones p_i and target points q_j (see Figure 4.2):

- **Pos($\cdot$):** Represents the 3D position component of a 5D coordinate. For a drone p_i , $\text{Pos}(p_i) = [x_i, y_i, z_i]^T$ indicates its spatial location, while for a target point q_j , $\text{Pos}(q_j) = [x_j, y_j, z_j]^T$ denotes its position.
- **Ang($\cdot$):** Represents the angular component. For a drone p_i , $\text{Ang}(p_i) = [\theta_i^h, \theta_i^v]^T$ specifies the camera's yaw and pitch angles, and for a target point q_j , $\text{Ang}(q_j) =$

$[\theta_j^h, \theta_j^v]^T$ indicates its orientation.

- **Dir($\cdot$):** Represents the direction vector derived from angular components, defined as:

$$\text{Dir} = [\cos \theta^v \cos \theta^h, \cos \theta^v \sin \theta^h, \sin \theta^v]^T.$$

For a drone p_i , $\text{Dir}(p_i)$ indicates the camera's optical axis direction in the world coordinate system. For a target point q_j , $\text{Dir}(q_j)$ specifies the expected viewing direction from which the point should be observed.

4.2.2 Sensing Performance and Importance Index

To evaluate the quality of observation by the drones, we consider three key criteria for a drone at state p_i observing a target point q_j in $\mathcal{Q}$:

1. **Alignment of the Drone Camera's Optical Axis:** The drone camera's optical axis should align with the vector pointing from the drone's position to the target point q_j , ensuring the camera is directly facing the target. This is quantified by:

$$\varphi_1(p_i, q_j) = \arccos \left(\frac{\text{Dir}(p_i) \cdot (\text{Pos}(q_j) - \text{Pos}(p_i))}{\|\text{Pos}(q_j) - \text{Pos}(p_i)\|} \right).$$

2. **Orientation of the Target Point Toward the Drone:** The orientation of the target point q_j should face the drone's camera to capture relevant features effectively. This is given by:

$$\varphi_2(p_i, q_j) = \arccos \left(-\frac{\text{Dir}(q_j) \cdot (\text{Pos}(q_j) - \text{Pos}(p_i))}{\|\text{Pos}(q_j) - \text{Pos}(p_i)\|} \right).$$

3. **Appropriate Distance from the Camera to the Target:** The distance between the camera and the target should be optimal for the camera's resolution. This distance is represented by:

$$\rho(p_i, q_j) = \|\text{Pos}(q_j) - \text{Pos}(p_i)\| - D,$$

where D represents the desired observation distance for high-resolution details.

These criteria collectively determine the sensing performance function $h_1(p_i, q_j)$, which evaluates the quality of observation by combining the alignment, orientation, and distance factors:

$$h_1(p_i, q_j) := \exp \left(\frac{-(1 - \cos \varphi_1)^2}{2\sigma_1^2} + \frac{-(1 - \cos \varphi_2)^2}{2\sigma_2^2} + \frac{-\rho^2}{2\sigma_3^2} \right).$$

The parameters σ_1 , σ_2 , and σ_3 adjust the sensitivity to these factors.

We next present the importance index $\phi_j \in [0, \infty)$, assigned to each point $q_j \in \mathcal{Q}$. A high importance value suggests that the point requires further observation. Similarly to [41], ϕ_j is updated based on a differential equation that depends on the sensing performance function h_1 for all drones, the current index ϕ_j , and the real-time map information. The update rule will be presented in the next section.

4.3 Coordinated Image Sampling with Real-Time Map Feedback

In this section, we present an algorithm for coordinated image sampling that leverages real-time map feedback from mesh changes to guide the reconstruction process. The unknown environment is reconstructed using aerial images sampled by drones, with no prior ground truth available.

4.3.1 Quantification of Mesh Changes

As outlined in Section 2, the 3D map generated by NeuralRecon is represented as a mesh of vertices and triangles. This reconstruction is iterative, with updates integrated at discrete time instants $t \in \mathcal{T} = \{1, 2, \dots\}$, where $\mathcal{T}$ denotes the set of discrete times at which the mesh is updated by NeuralRecon. Mathematically, the mesh at time $t \in \mathcal{T}$ is represented as:

$$V^t = \{v_l^t \in \mathbb{R}^3 \mid l = 1, \dots, N^t\} \subseteq \mathbb{R}^3,$$

where v_l^t represents the position of the l -th vertex in 3D space, and N^t is the total number of vertices at time t . As more data is integrated, N^t changes accordingly, reflecting updates to the mesh over time.

To extract meaningful feedback from the ongoing mesh reconstruction, we track changes in the mesh to identify areas that may still require more attention. Regions where the mesh is evolving are expected to continue changing as more images are added, while areas with minimal or no changes are assumed to be sufficiently reconstructed. Significant changes, such as an increase in the number of vertices, indicate areas still in need of further observation.

To efficiently evaluate mesh changes, we define a set of points $\{r_k \in \mathbb{R}^3 \mid k = 1, \dots, s\}$, where each point r_k represents a location in 3D space. The collection of these points forms the set $\mathcal{R}$. The total number of points, s , is typically smaller than the number of elements in $\mathcal{Q}$, reducing computational load while still capturing relevant changes in the mesh. Let $V_{r_k}^t$ represent the collection of mesh vertices at time t that are closest to the

point r_k . Formally,

$$V_{r_k}^t = \{v \in V^t \cap \mathcal{B} \mid r_k = \arg \min_{r \in \mathcal{R}} \|r - v\|\}.$$

Analogously, $V_{r_k}^{t+1}$ represents the closest vertices at time $t+1$. The mesh change evaluated at each point r_k between time t and $t+1$ is then quantified as:

$$\Delta V^{t+1}(r_k) = |V_{r_k}^{t+1}| - |V_{r_k}^t|. \quad (4.1)$$

Here, $|V_{r_k}^t|$ represents the number of vertices in the set $V_{r_k}^t$, and the difference $\Delta V^{t+1}(r_k)$ measures how the mesh has evolved around r_k over time. By calculating $\Delta V^{t+1}(r_k)$ for each point r_k , we can quantify the changes in the mesh, allowing us to identify regions that have been updated or remain incomplete.

4.3.2 Importance Index Update Based on Mesh Feedback

To integrate map feedback into the coverage control framework, we introduce a performance function, $h_2(q_j)$, which measures the influence of mesh changes on a point q_j in the virtual field $\mathcal{Q}$. It is defined as:

$$h_2(q_j) = \sum_{r_k \in \mathcal{R}} \tanh\left(\frac{(\Delta V^{t+1}(r_k))^2}{\kappa^2}\right) \exp\left(\frac{-\|\text{Pos}(q_j) - r_k\|^2}{2\sigma_4^2}\right),$$

where $\Delta V^{t+1}(r_k)$ is defined in (4.1). The squared term $(\Delta V^{t+1}(r_k))^2$ represents the magnitude of mesh changes around r_k , capturing both increases and decreases in the number of vertices. The hyperbolic tangent function, $\tanh\left(\frac{(\Delta V^{t+1}(r_k))^2}{\kappa^2}\right)$, normalizes these changes to the range $[0, 1]$, with κ tuned to adjust the tanh function's transition from 0 to 1, ensuring proper normalization of the mesh changes. The exponential term $\exp\left(\frac{-\|\text{Pos}(q_j) - r_k\|^2}{2\sigma_4^2}\right)$ adjusts the influence of mesh changes based on the distance between r_k and q_j , where σ_4 governs the sensitivity to this distance.

The importance index ϕ_j is updated continuously based on real-time sensing performance, represented by the term

$$\dot{\phi}_j = -\delta_1 \max_{i \in \mathcal{I}} h_1(p_i, q_j) \phi_j, \quad (4.2)$$

which evolves as the drone gathers data. Here, δ_1 acts as a gain parameter, adjusting the responsiveness of ϕ_j to the drone's sensing performance. Then, at discrete instants $t \in \mathcal{T}$, however, we incorporate the influence of mesh feedback as a jump in ϕ_j , reflecting changes in the environment. This jump is expressed as

$$\phi_j(t^+) = \phi_j(t^-) + \delta_2 h_2(q_j), \quad (4.3)$$

where t^- and t^+ denote the continuous time immediately before and after the time corresponding to each discrete time instant $t \in \mathcal{T}$, respectively. Here, δ_2 amplifies the effect of mesh changes, adjusting ϕ_j in regions that may require additional attention in the reconstruction.

In this context, the goal of the coverage control algorithm is to minimize the importance index across the virtual field by optimizing the drones' trajectories. The global objective function J , representing the average importance index across the entire region, is then defined as:

$$J = \sum_{j \in \mathcal{M}} \frac{\phi_j}{|\mathcal{M}|} = \sum_{j=1}^m \frac{\phi_j}{m}. \quad (4.4)$$

Minimizing this average importance index reflects the overall effectiveness of the coverage control, as a lower J indicates better sensing quality and data collection. The algorithm, therefore, seeks to drive J as close to zero as possible by enforcing the constraint

$$\dot{J} \leq -\gamma \quad \forall \tau \notin \mathcal{T}_c$$

for a given $\gamma \geq 0$, where $\mathcal{T}_c$ is the set of continuous times corresponding to the discrete times in $\mathcal{T}$. This inequality ensures that the objective function decreases at a controlled rate to achieve optimal coverage for improved 3D reconstruction. The present objective means that J should decay at least at the rate γ over the time intervals between two adjacent instantaneous jumps of ϕ_j by (4.3).

4.3.3 QP-Based Controller Design

In this section, we propose a solution to achieve the objective outlined previously. To accomplish this, we utilize a quadratic programming (QP)-based controller [73] that enforces the constraint $\dot{J} \leq -\gamma$. We begin by computing the time derivative of J at any time $\tau \notin \mathcal{T}_c$:

$$\begin{aligned} \dot{J} &= \sum_{j \in \mathcal{M}} \frac{\dot{\phi}_j}{|\mathcal{M}|} = - \frac{\sum_{j \in \mathcal{M}} \delta_1 \max_{i \in \mathcal{I}} h_1(p_i, q_j) \phi_j}{|\mathcal{M}|} \\ &= - \frac{\sum_{i=1}^n \sum_{j \in \mathcal{V}_i(p)} \delta_1 h_1(p_i, q_j) \phi_j}{|\mathcal{M}|} = - \frac{\sum_{i=1}^n I_i}{|\mathcal{M}|}, \end{aligned}$$

where $I_i = \sum_{j \in \mathcal{V}_i(p)} \delta_1 h_1(p_i, q_j) \phi_j$, representing the influence of drone i in lowering J in (4.4). The set $\mathcal{V}_i(p)$, which depends on $p := (p_i)_{i \in \mathcal{I}}$, is a Voronoi-like partition of the set $\mathcal{M}$ defined as

$$\mathcal{V}_i(p) := \{j \in \mathcal{M} \mid h_1(p_i, q_j) \leq h_1(p_k, q_j) \quad \forall k \in \mathcal{I}\}.$$

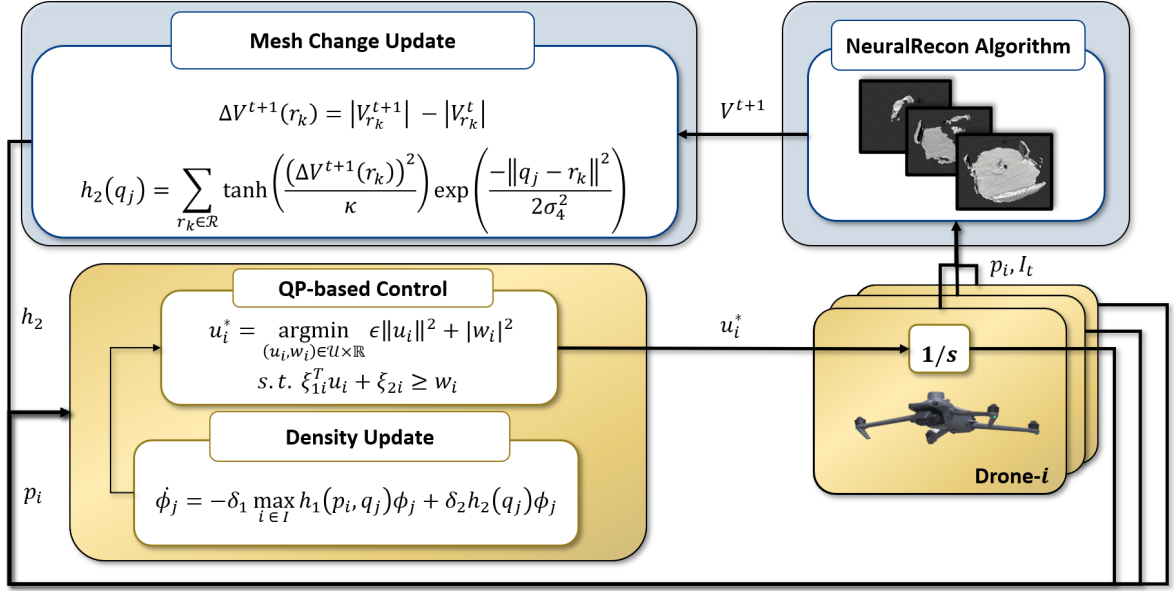


Fig. 4.3: Controller architecture of map-feedback coverage control.

Theorem 1. Let J_i be the local objective function that each drone i should minimize. By enforcing the local constraint on each drone i

$$\dot{J}_i = -\frac{I_i}{|\mathcal{V}_i(p)|} \leq -\gamma, \quad (4.5)$$

we ensure that the global constraint $\dot{J} \leq -\gamma$ is satisfied.

Proof. From the local constraint (4.5), multiplying both sides by $|\mathcal{V}_i(p)|$ gives:

$$-I_i \leq -|\mathcal{V}_i(p)|\gamma.$$

Summing over all drones i :

$$-\sum_{i=1}^n I_i \leq -\sum_{i=1}^n |\mathcal{V}_i(p)|\gamma = -|\mathcal{M}|\gamma.$$

Hence, the global objective function's derivative becomes:

$$\dot{J} = -\frac{\sum_{i=1}^n I_i}{|\mathcal{M}|} \leq -\gamma.$$

This completes the proof. □

Next, we enforce the constraint (4.5) by setting $b_{i,I} \geq 0$, with $b_{i,I}$ defined as

$$b_{i,I} = I_i - |\mathcal{V}_i(p)|\gamma.$$

This condition is satisfied when the velocity input u_i is chosen so that the following inequality holds:

$$\left(\frac{\partial b_{i,I}}{\partial \phi_j}\right) \dot{\phi}_j + \left(\frac{\partial b_{i,I}}{\partial p_i}\right)^T u_i + \alpha_1(b_{i,I}) \geq 0,$$

where α_1 is a locally Lipschitz extended class $\mathcal{K}$ function.

In addition, to address the gimbal pitch limitations, we introduce a constraint following [41]. This constraint restricts the vertical angle θ_i^v of the drone's gimbal within a specified range, ensuring it remains between $\theta_{\min}^v$ and $\theta_{\max}^v$. We define a control barrier function b_{i,θ^v} to enforce this range:

$$b_{i,\theta^v} = (\theta_{\max}^v - \theta_{\min}^v)^2 - \left(\theta_i^v - \frac{\theta_{\min}^v + \theta_{\max}^v}{2}\right)^2 \geq 0.$$

This condition holds when u_i meets the following inequality:

$$\left(\frac{\partial b_{i,\theta^v}}{\partial p_i}\right)^T u_i + \alpha_2(b_{i,\theta^v}) \geq 0,$$

where α_2 is a locally Lipschitz extended class $\mathcal{K}$ function.

Now, we present the QP-based controller as follows:

$$\begin{aligned} (u_i^*, w_i^*) &= \arg \min_{(u_i, w_i) \in \mathcal{U} \times \mathbb{R}} \epsilon \|u_i\|^2 + |w_i|^2 \\ \text{s.t.} \quad &\left(\frac{\partial b_{i,I}}{\partial \phi_j}\right) \dot{\phi}_j + \left(\frac{\partial b_{i,I}}{\partial p_i}\right)^T u_i + \alpha_1(b_{i,I}) \geq w_i, \\ &\left(\frac{\partial b_{i,\theta^v}}{\partial p_i}\right)^T u_i + \alpha_2(b_{i,\theta^v}) \geq 0, \end{aligned}$$

where ϵ is a penalty variable and w_i is a slack variable allowing for relaxation of the constraint.

Theorem 2. Suppose that no q_j (for $j \in \mathcal{M}$) is located on the boundary of $\mathcal{V}_i(p)$. When $\alpha_1(b_{i,I}) : \mathbb{R} \rightarrow \mathbb{R}$ and $\alpha_2(b_{i,\theta^v}) : \mathbb{R} \rightarrow \mathbb{R}$ are set as linear functions such that $\alpha_1(b_{i,I}) = a_1 b_{i,I}$ and $\alpha_2(b_{i,\theta^v}) = a_2 b_{i,\theta^v}$, where $a_1 > 0$ and $a_2 > 0$ are positive scalars, the problem is equivalently reformulated as:

$$\begin{aligned} (u_i^*, w_i^*) &= \arg \min_{(u_i, w_i) \in \mathcal{U} \times \mathbb{R}} \epsilon \|u_i\|^2 + |w_i|^2 \\ \text{s.t.} \quad &\xi_{1i}^T u_i + \xi_{2i} \geq w_i, \\ &\chi_{1i}^T u_i + \chi_{2i} \geq 0, \end{aligned}$$

where

$$\begin{aligned}\xi_{1i} &:= \delta_1 \sum_{j \in \mathcal{V}_i(p)} \frac{\partial h_1(p_i, q_j)}{\partial p_i} \phi_j, \\ \xi_{2i} &:= -a_1 |\mathcal{V}_i(p)| \gamma + \\ &\quad \sum_{j \in \mathcal{V}_i(p)} \delta_1 h_1(p_i, q_j) \phi_j (a_1 - \delta_1 h_1(p_i, q_j)),\end{aligned}$$

and

$$\begin{aligned}\chi_{1i} &:= \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \left(\theta_i^v - \frac{\theta_{\min}^v + \theta_{\max}^v}{2} \right) \end{bmatrix}, \\ \chi_{2i} &:= a_2 \left((\theta_{\max}^v - \theta_{\min}^v)^2 - \left(\theta_i^v - \frac{\theta_{\min}^v + \theta_{\max}^v}{2} \right)^2 \right).\end{aligned}$$

Proof. The constraint in the QP-based controller (4.6) involves the terms $\frac{\partial b_{i,I}}{\partial p_i}$, $\frac{\partial b_{i,I}}{\partial \phi_j}$, $\alpha_1(b_{i,I})$, $\frac{\partial b_{i,\theta^v}}{\partial p_i}$, and $\alpha_2(b_{i,\theta^v})$. Here, the terms ξ_{1i} , ξ_{2i} , χ_{1i} , and χ_{2i} can be computed as follows:

$$\begin{aligned}\xi_{1i} &= \frac{\partial b_{i,I}}{\partial p_i} \\ &= \frac{\partial}{\partial p_i} (I_i - |\mathcal{V}_i(p)| \gamma) = \delta_1 \sum_{j \in \mathcal{V}_i(p)} \frac{\partial h_1(p_i, q_j)}{\partial p_i} \phi_j.\end{aligned}$$

Next, noting that $\alpha_1(b_{i,I}) = a_1 b_{i,I}$, we compute ξ_{2i} as:

$$\begin{aligned}\xi_{2i} &= \frac{\partial b_{i,I}}{\partial \phi_j} + \alpha_1(b_{i,I}) \\ &= \frac{\partial}{\partial \phi_j} (I_i - |\mathcal{V}_i(p)| \gamma) + a_1 (I_i - |\mathcal{V}_i(p)| \gamma) \\ &= -a_1 |\mathcal{V}_i(p)| \gamma + \\ &\quad \sum_{j \in \mathcal{V}_i(p)} \delta_1 h_1(p_i, q_j) \phi_j (a_1 - \delta_1 h_1(p_i, q_j)).\end{aligned}$$

For the gimbal pitch constraint, we compute χ_{1i} as:

$$\chi_{1i} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{\partial b_{i,\theta^v}}{\partial \theta_i^v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \left(\theta_i^v - \frac{\theta_{\min}^v + \theta_{\max}^v}{2} \right) \end{bmatrix}.$$

Finally, noting that $\alpha_2(b_{i,\theta^v}) = a_2 b_{i,\theta^v}$, we compute χ_{2i} as:

$$\begin{aligned}\chi_{2i} &= \alpha_2(b_{i,\theta^v}) \\ &= a_2 \left((\theta_{\max}^v - \theta_{\min}^v)^2 - \left(\theta_i^v - \frac{\theta_{\min}^v + \theta_{\max}^v}{2} \right)^2 \right).\end{aligned}$$

Thus, the optimization problem (4.7) is equivalent to the original controller formulation (4.6). $\square$



Fig. 4.4: Snapshot of the Unity scene targeted for 3D reconstruction by drones.

The overall control architecture is illustrated in Figure 4.3.

Remark finally that the problem (4.7) is well-defined only at time $\tau \notin \mathcal{T}_c$. Accordingly, the control input at each time $\tau \in \mathcal{T}_c$ is not defined. Although the time of solving (4.7) does never exactly synchronize to the time of mesh updates in practice, we define $u_i(\tau)$, $\tau \in \mathcal{T}_c$ by the most recent solution to (4.7).

4.3.4 Control Logic with Feedback Trigger

To implement the previous controller effectively, the system is designed to first allow the drones to scan the entire field and gather basic information about the environment. This initial phase ensures that the drones obtain a rough shape of the 3D map while widely sampling images over the field. We define a threshold J_{th} for the objective function J , below which the mesh change feedback is activated.

If J remains above the threshold J_{th} , indicating that the system is still in the initial scanning phase, the mesh change feedback is not used for updating the importance index, keeping $\delta_2 = 0$. Once J falls below the threshold, the mesh change feedback is activated, and the parameter δ_2 is set to a positive value δ'_2 . This adjustment allows the drones to prioritize specific areas where significant mesh changes have occurred, increasing the importance index in regions that require more attention for improved 3D reconstruction quality:

$$\delta_2 = \begin{cases} 0 & \text{if } J \geq J_{th}, \\ \delta'_2 & \text{if } J < J_{th}. \end{cases}$$

This approach allows the system to first achieve broad coverage, then refine reconstruction in areas where mesh changes indicate a need for closer inspection. The overall control process is summarized in the following pseudocode:

Algorithm 1 Coverage Control Algorithm with Map-Feedback

```

1: Initialize  $\delta_2 = 0$ , gather initial 3D map
2: while True do
3:   Compute  $J$  (objective function) as in (4.4)
4:   if  $J \geq J_{\text{th}}$  then
5:     No map feedback, keep  $\delta_2 = 0$ 
6:   else
7:     Activate map feedback, set  $\delta_2 = \delta'_2$ 
8:   end if
9:   Update  $\phi_j$  using the rule in (4.2)
10:  Compute  $u_i$  using the QP-based controller (4.7)
11:  Update drone positions based on  $u_i$ 
12:  Continue scanning
13: end while

```

4.4 Simulation-Based Evaluation

In this section, we evaluate the proposed algorithm through simulations involving both single and multiple drones, demonstrating the effectiveness of real-time map feedback in enhancing 3D reconstruction quality and increasing sampling efficiency.

The simulation environment is implemented using ROS2 [128] for drone control and Unity as the simulated environment. Communication between the two platforms is managed via the ROS2-Unity bridge. In Unity, the drones simulate real-world operations by capturing images, which are then sent to the ROS2 environment, where the NeuralRecon algorithm processes them to reconstruct a 3D mesh. This mesh is evaluated in real-time, with updates fed back to the coverage controller to guide the drones' movements based on current map changes.

The simulation is conducted in a monitored environment represented as a 5D field, where the spatial region is defined as a cube with dimensions $[-3, 3] \text{ m} \times [-3, 3] \text{ m} \times [0, 2] \text{ m}$ and the viewing angle space is set to $\Theta_h \in [-\pi, \pi)$ and $\Theta_v \in [0, \pi/2]$. This 5D field is discretized into $m = 1.5 \times 10^7$ cells, each measuring $0.02 \text{ m} \times 0.02 \text{ m} \times 0.1 \text{ m} \times \pi/30 \text{ rad} \times \pi/30 \text{ rad}$. The parameter values used in the simulation are summarized in Table 4.1. Safety constraints

for inter-drone collision avoidance [24] and pitch angle limitation [41] are integrated into the QP-based controller (4.7).

Table. 4.1: Simulation parameters for 5D coverage with real-time map feedback

Parameter	Value
Field size	$[-3, 3] \times [-3, 3] \times [0, 2]$ m
Viewing angle range	$\Theta_h = [-\pi, \pi)$, $\Theta_v = [0, \pi/2]$
Virtual cell size	$0.02 \times 0.02 \times 0.1 \times \frac{\pi}{30} \times \frac{\pi}{30}$
Total virtual cells m	1.5×10^7
Max depth D	1.0
Sensing performance parameters	$\sigma_1 = 0.07$, $\sigma_2 = 0.3$, $\sigma_3 = 0.1$, $\sigma_4 = 0.4$
Importance decay	$\delta_1 = 3.0$, $\delta_2 = 1.0$ (or 0.0)
Sensitivity scaling	$\kappa = 0.7$
Barrier and cost parameters	$a = 1.0$, $\epsilon = 0.01$, $\gamma = 0.012$
Cost threshold J_{th}	0.4

4.4.1 Single-Drone Simulation

We first consider $n = 1$ drone with an initial position set to $p_1 = [5.0, 3.0]^T$ m. Two scenarios are compared: one using real-time map feedback and one without. Both scenarios implement the same QP controller (4.7) with $\gamma = 0.012$, but in the no-feedback case, δ_2 is set to zero throughout the simulation.

Figure 4.6 shows the time evolution of J . In the early phase ($t \lesssim 180$ s), both scenarios exhibit similar J reduction as the map feedback is inactive. When J drops below J_{th} , the feedback mechanism activates, resulting in visible fluctuations in J due to mesh changes. This feedback-driven behavior continues until $t \approx 1400$ s, when J stabilizes near zero. The no-feedback scenario, on the other hand, shows a steady decline in J without regard to mesh stability.

A final mesh comparison is presented in Figure 4.7. The reconstruction under map feedback (Fig. 4.7b) more closely resembles the ground truth (Fig. 4.7a) than the version without feedback (Fig. 4.7c), indicating improved map completeness from feedback-guided sensing.

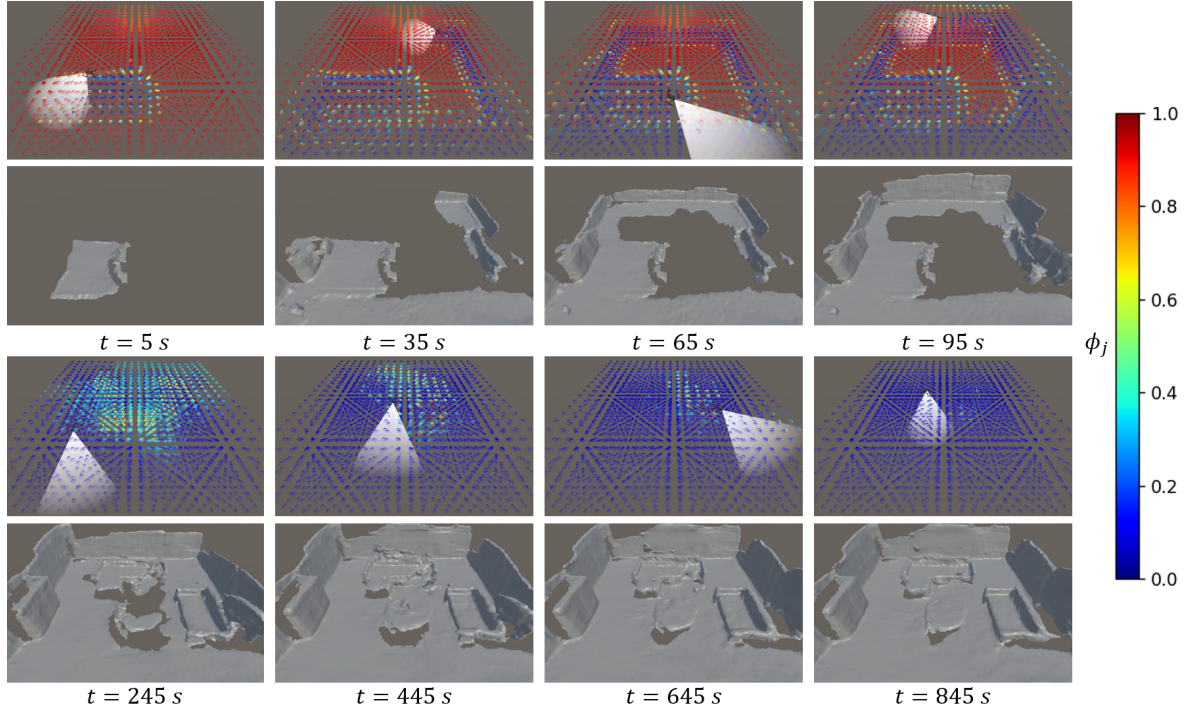


Fig. 4.5: Simulation snapshot of map-feedback coverage control using one drone. The top section shows the distribution of ϕ_j , while the bottom section displays the evolution of the reconstructed mesh. A video of the simulation can be viewed [here](#).

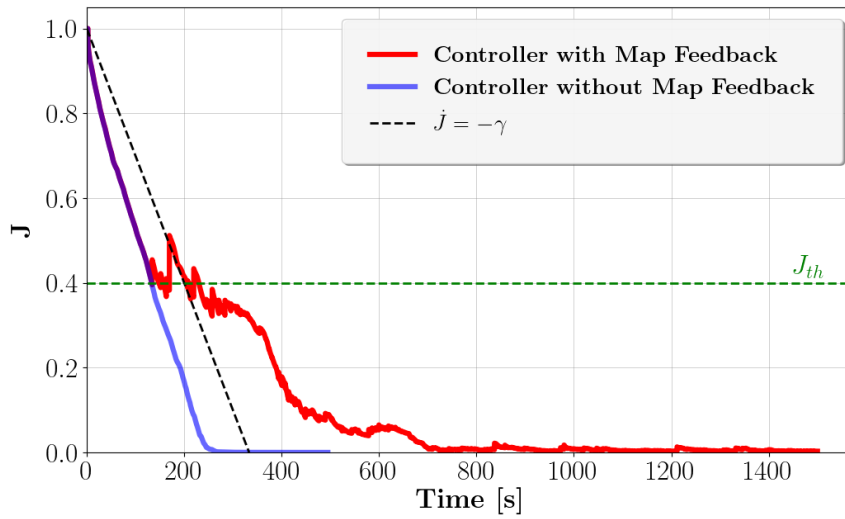


Fig. 4.6: Plot of J for the single-drone performance evaluation.

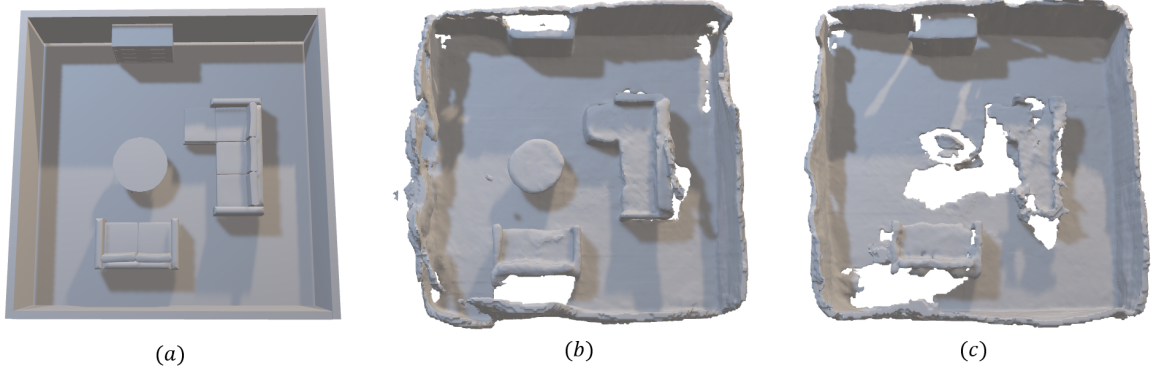


Fig. 4.7: Comparison between reconstructed meshes for the single-drone performance evaluation. (a) Ground truth mesh of the scene in Unity. (b) Final reconstructed mesh using the controller with map feedback. (c) Final reconstructed mesh using the controller without map feedback.

4.4.2 Multi-Drone Simulation

We extend the evaluation to $n = 2$ drones, initially positioned at $p_1 = [5.0, 3.0]^T$ m and $p_2 = [-5.0, -3.0]^T$ m. The two scenarios—feedback and no-feedback—again use the same controller setup with $\delta_2 = 1.0$ and 0.0 respectively.

Figure 4.9 presents the time series of J . As with the single-drone case, early behavior is similar between the two scenarios, followed by map-feedback-induced fluctuations in J until the cost converges. Figure 4.10 compares the final reconstructed meshes. The feedback-enabled scenario achieves greater completeness and visual accuracy relative to the ground truth, confirming the benefit of feedback-driven coordination.

These results demonstrate the scalability and effectiveness of the proposed approach in both single and multi-drone systems, highlighting improved reconstruction quality and adaptability enabled by real-time mesh feedback.

4.5 Experiments: Online Reconstruction Validation

4.5.1 Experimental Setup

To validate the proposed method in real-world settings, we conduct indoor experiments at the Science Tokyo Robot Zoo Sky testbed. The experiments use DJI Mavic Mini 3 drones, equipped with RGB camera. The drone state (position and orientation) is tracked using an OptiTrack motion capture system at 120 Hz, providing accurate localization for real-time feedback control.

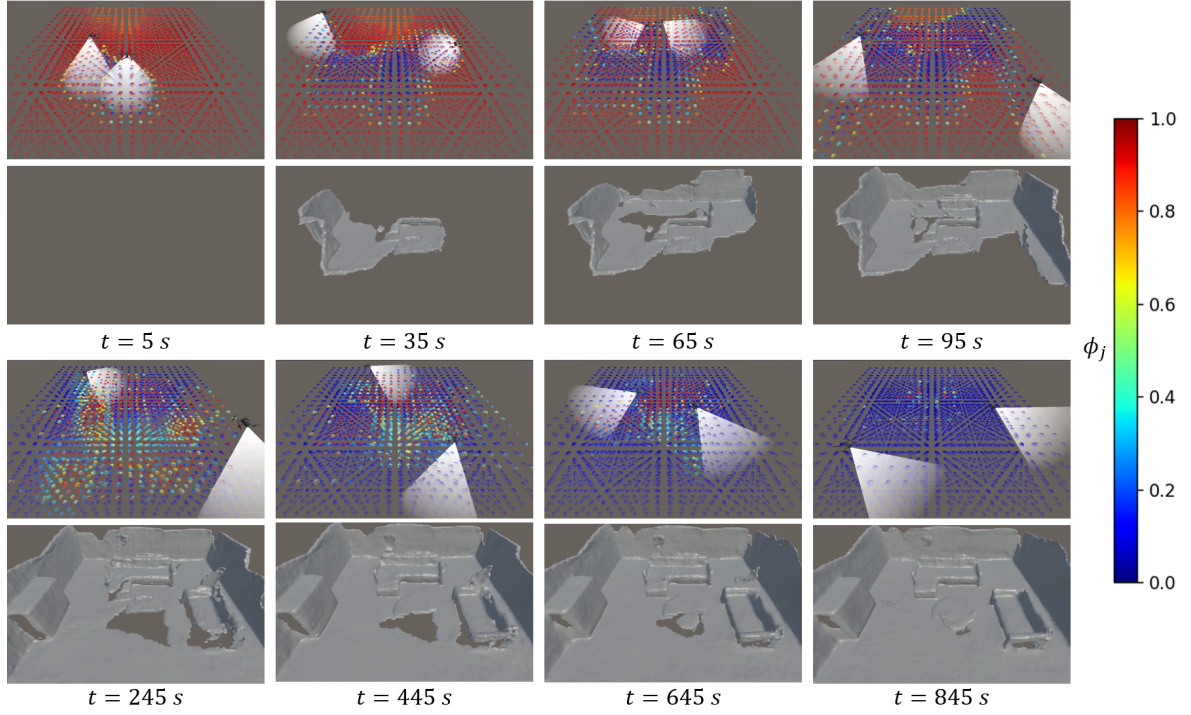


Fig. 4.8: Simulation snapshot of map-feedback coverage control using two drones. The top section shows the distribution of ϕ_j , while the bottom section displays the evolution of the reconstructed mesh. A video of the simulation can be viewed [here](#).

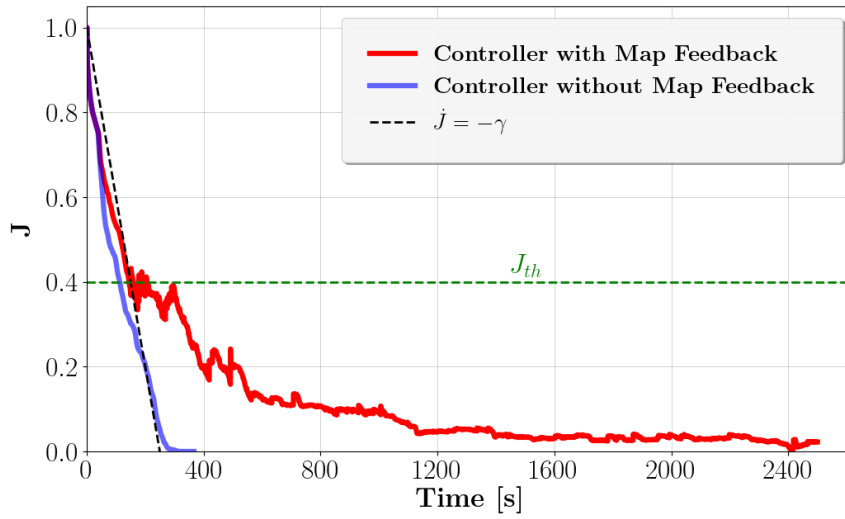


Fig. 4.9: Plot of J for the two-drone performance evaluation.

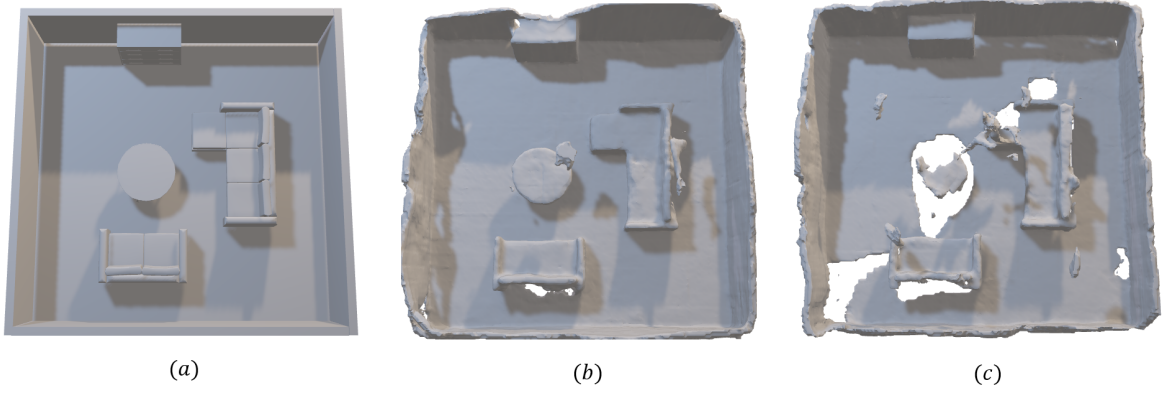


Fig. 4.10: Comparison between reconstructed meshes for the two-drone performance evaluation. (a) Ground truth mesh of the scene in Unity. (b) Final reconstructed mesh using the controller with map feedback. (c) Final reconstructed mesh using the controller without map feedback.

Captured images are transmitted wirelessly to a ground station where the NeuralRecon algorithm processes them to generate a dynamic 3D mesh of the scene. The mesh is continuously updated and evaluated to provide map-change feedback, which is then used to inform the adaptive importance index updates during the mission. The experimental setup architecture can be seen in Figure 4.11.

Two experimental scenarios are considered:

- **Static Importance (No Feedback):** Drones operate using the QP controller without real-time feedback, i.e., $\delta_2 = 0$ throughout the mission.
- **Map Feedback (Adaptive):** Real-time feedback is activated once $J < J_{\text{th}} = 0.4$, allowing $\delta_2 = \delta'_2$ to incorporate mesh-change-based resampling.

For this indoor setup, we configure the physical environment to resemble a miniature structured space. The field of interest spans approximately $2.0\text{ m} \times 2.0\text{ m} \times 2.0\text{ m}$, and each drone operates at altitudes ranging from 0.8 m to 1.5 m. The following parameters are used:

Safety constraints such as pitch angle limits and inter-agent collision avoidance are also enforced in the controller.

4.5.2 Experimental Results

A preliminary indoor experiment was conducted using a single drone equipped with onboard sensing and real-time reconstruction feedback. A snapshot of the flight test is shown in Figure 4.12, and the corresponding reconstruction result with camera viewpoints

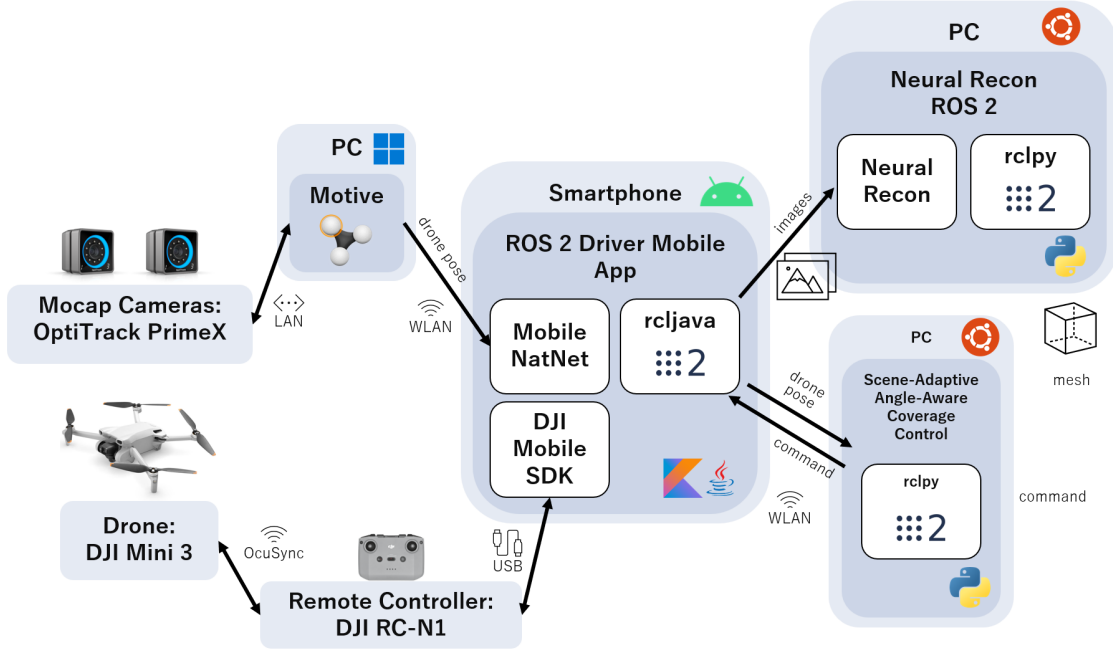


Fig. 4.11: Indoor experimental system architecture for proposed method verification.

is visualized in Figure 4.13.

During the experiment, the drone actively captured viewpoints across the scene while the importance index ϕ evolved in response to coverage. Regions initially marked as high priority (red) gradually turned blue as the drone observed them, reflecting successful reduction in importance. When the mesh feedback module was activated, newly reconstructed regions triggered a reset of importance values, causing relevant areas to turn red again. This illustrates the framework’s ability to adapt coverage behavior based on scene updates.

The NeuralRecon module successfully generated a dense mesh in real time, incrementally fusing observations as new frames were collected. These qualitative results support the feasibility of integrating online 3D reconstruction into adaptive viewpoint planning.

4.6 Chapter Summary

This chapter presented a scene-adaptive framework for coordinated image sampling that integrates real-time map feedback into a coverage control strategy enhanced with camera rotational motion. The proposed method builds upon the angle-aware coverage control formulation by explicitly modeling the control of both drone position and camera orientation, thereby operating in a five-dimensional (5D) control space. This enhanced flexibility enables the drones to align their viewpoints more effectively with the evolving structure of the target scene.

Table. 4.2: Indoor simulation parameters for scene-adaptive coverage

Parameter	Value
Field size	$2.0 \times 2.0 \times 2.0$ m
Drone altitude range	$[0.8, 1.5]$ m
Desired observation distance D	1.2 m
Sensing parameters	$\sigma_1 = 0.07, \sigma_2 = 0.3, \sigma_3 = 0.1, \sigma_4 = 0.4$
Feedback tuning	$\delta_1 = 3.0, \delta'_2 = 1.0, \kappa = 0.7$
QP parameters	$a = 1.0, \epsilon = 0.01$
Feedback threshold J_{th}	0.4

To adaptively guide sampling efforts, the framework leverages an online reconstruction pipeline—NeuralRecon—that continuously updates a volumetric mesh based on input images. A key contribution in this chapter is the feedback mechanism that quantifies changes in the mesh over time and converts them into a dynamically updated importance index. This index informs the controller of which regions remain under-observed or unstable, allowing the system to allocate sensing resources more intelligently.

The control law is formulated as a quadratic program (QP) that minimizes the rate of decrease of a task-specific cost function, while incorporating constraints that ensure safety and feasibility. These include inter-drone collision avoidance, field-of-view limits, and pitch angle bounds for the gimbal-mounted cameras. The QP controller adjusts drone motion and camera orientation in response to both task requirements and reconstruction feedback.

The proposed method was validated through extensive simulation and real-world experiments. Simulations were conducted using a Unity-ROS2 environment, with both single-drone and multi-drone scenarios tested. The results demonstrated that map feedback allows drones to give more attention to the evolving region, resulting in more complete reconstructions. In physical experiments, the approach showed the practicality and feasibility of the proposed method in generating the real-time 3d map iteratively.

Overall, this chapter demonstrated that integrating real-time 3D mesh feedback into a camera-rotation-enabled coverage control framework enables more adaptive and efficient multi-drone 3D reconstruction. The results validate the feasibility and effectiveness of scene-adaptive coordination in improving the quality of the map.



Fig. 4.12: Indoor experimental setup for proposed method verification.

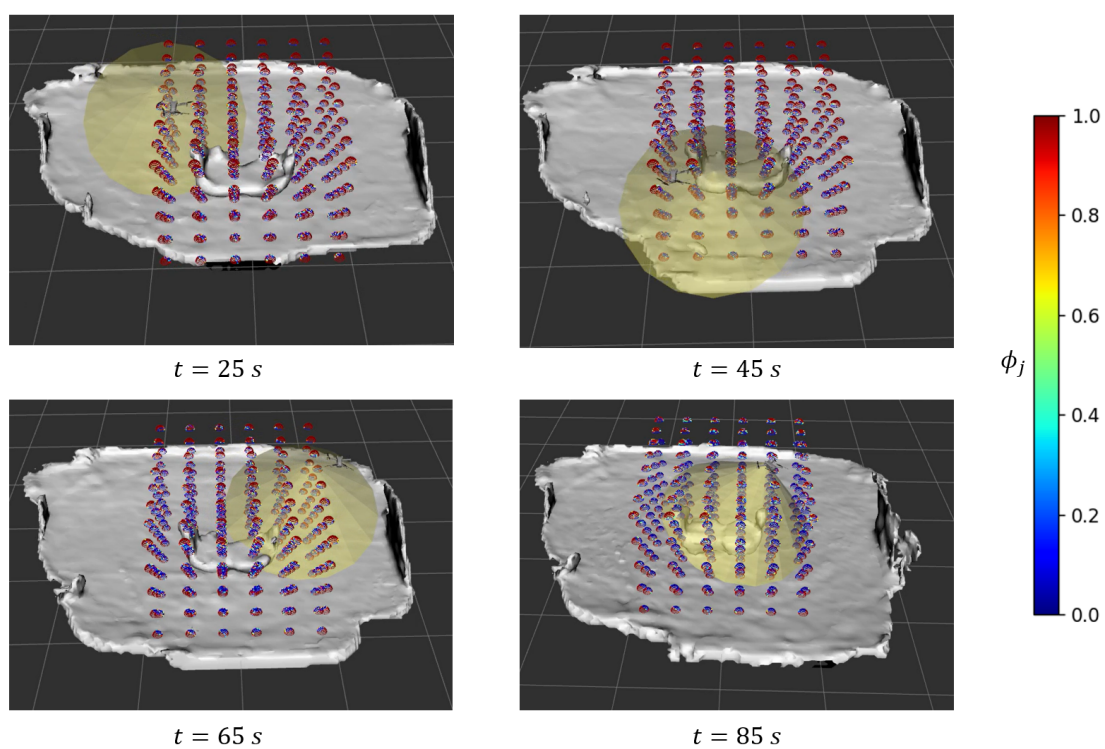


Fig. 4.13: Snapshot of the experiment with single drone.

Chapter 5

Conclusion

5.1 Summary of Contributions

This dissertation addressed the problem of coordinated image sampling for high-quality 3D reconstruction using multi-drone systems. Two main contributions were presented, each corresponding to a distinct reconstruction pipeline: one for classical offline reconstruction and the other for online, real-time reconstruction. The key contributions are summarized below:

- **Efficient Angle-Aware Coverage Control for Multi-Drone Image Sampling using Offline Reconstruction Pipelines**

To address the limited scalability of prior angle-aware coverage strategies in large-scale environments, we proposed an extended formulation that integrates Voronoi-based coverage control as a nominal input within the original angle-aware framework. A QP-based controller was developed to combine this nominal input with the angle-aware gradient field, while incorporating control barrier functions (CBFs) for safety constraints such as collision avoidance and field boundaries. The method was validated in simulation (single and multi-drone settings) and in both indoor and outdoor experiments, demonstrating improved mission completion efficiency and practical applicability compared to baseline methods.

- **Scene-Adaptive Angle-Aware Coverage Control with Camera Rotational Motion using Online Reconstruction Pipelines**

To enable real-time adaptivity, we extended the angle-aware coverage control with gimbal-based camera rotation by modifying the control input space from 4D to 5D (adding altitude) and reformulating the sensing performance function. We introduced a novel feedback mechanism that interprets mesh updates from the Neural-Recon algorithm as a proxy for reconstruction completeness, allowing the system

to adaptively update its coverage density. A QP-based controller was designed to incorporate this dynamic importance index while maintaining safety through CBFs. The approach was validated through Unity – ROS2 simulation and real-world multi-drone experiments, showing improved reconstruction completeness and scene adaptivity compared to baseline methods without map feedback.

5.2 Limitations and Future Work

While this dissertation presents substantial progress in coordinated image sampling for 3D reconstruction using multi-drone systems, several limitations remain and point to future research directions.

One practical limitation concerns the computational overhead introduced by the integration of real-time 3D mesh updates with QP-based coverage control. While the controller is tractable for small- to mid-scale missions, scaling to larger fields or prolonged deployments may require more efficient solvers or hardware acceleration. In particular, parallelizing key components or approximating feedback-driven coverage terms with learned surrogates may help reduce onboard computation loads.

Coordination in the current system is achieved through decentralized coverage objectives without explicit inter-agent communication. While this facilitates scalability, it can limit performance in complex scenarios that require dynamic task reallocation or cooperation over longer horizons. Incorporating lightweight distributed optimization or communication-aware coordination could enhance robustness for larger teams.

Another limitation lies in the scope of experimental validation. While both indoor and outdoor experiments were performed, they were limited to structured environments with manageable scale and visibility. Extending the framework to more unstructured, cluttered, or GPS-denied environments would require integration with SLAM-based localization and more adaptive path planning that accounts for obstacles and environmental uncertainty.

The current framework also does not explicitly model uncertainty in sensing or reconstruction. All decisions are made based on heuristic interpretation of mesh changes, without probabilistic modeling of confidence or completeness. A more rigorous approach may involve incorporating uncertainty-aware perception models or estimating confidence maps to better inform control decisions under occlusion, visual aliasing, or noisy reconstruction outputs.

Looking ahead, one promising direction involves extending the scene-adaptive framework to support neural radiance field representations such as NeRF or Gaussian Splatting. While these techniques offer richer, photorealistic scene representations, they do not yield

explicit mesh changes that can be directly tracked. Developing methods to interpret view-dependent error, density gradients, or neural feature change as proxies for reconstruction progress remains an open challenge—but could allow for feedback-driven coverage control in high-fidelity rendering environments.

Finally, this dissertation highlights an important direction toward quality-driven 3D mapping without reliance on external ground truth or predefined priors. The use of internal reconstruction evolution (e.g., mesh changes) as feedback enables more autonomous and self-adaptive behavior. Further investigation is needed to formalize how such internal metrics correlate with true map quality, and how they can drive sampling strategies that remain effective in unknown or complex scenes.

In conclusion, the methods developed in this dissertation provide a foundation for scalable and adaptive aerial 3D reconstruction. By combining principled coverage control with feedback from online reconstruction pipelines, this work opens the door to more autonomous, efficient, and intelligent multi-robot mapping systems.

References

- [1] J. Cortés and M. Egerstedt, “Coordinated control of multi-robot systems: A survey,” *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.
- [2] M. Mammarella, L. Comba, A. Biglia, F. Dabbene, and P. Gay, “Cooperation of unmanned systems for agricultural applications: A case study in a vineyard,” *Biosystems Engineering*, vol. 223, pp. 81–102, 2022.
- [3] A. C. Tagarakis, D. Kalaitzidis, E. Filippou, L. Benos, and D. Bochtis, “3d scenery construction of agricultural environments for robotics awareness,” in *Information and Communication Technologies for Agriculture— Theme III: Decision*. Springer, 2022, pp. 125–142.
- [4] C. Ju, J. Kim, J. Seol, and H. I. Son, “A review on multirobot systems in agriculture,” *Computers and Electronics in Agriculture*, vol. 202, p. 107336, 2022.
- [5] D. Albani, T. Manoni, A. Arik, D. Nardi, and V. Trianni, “Field coverage for weed mapping: toward experiments with a uav swarm,” in *Bio-inspired Information and Communication Technologies: 11th EAI International Conference, BICT 2019, Pittsburgh, PA, USA, March 13–14, 2019, Proceedings 11*. Springer, 2019, pp. 132–146.
- [6] C. Wu, R. Zeng, J. Pan, C. C. Wang, and Y.-J. Liu, “Plant phenotyping by deep-learning-based planner for multi-robots,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3113–3120, 2019.
- [7] M. Iida, S. Harada, R. Sasaki, Y. Zhang, R. Asada, M. Suguri, and R. Masuda, “Multi-combine robot system for rice harvesting operation,” in *2017 ASABE Annual International Meeting*. American Society of Agricultural and Biological Engineers, 2017, p. 1.
- [8] M. P. Mann, B. Zion, I. Shmulevich, D. Rubinstein, and R. Linker, “Combinatorial optimization and performance analysis of a multi-arm cartesian robotic fruit harvester—extensions of graph coloring,” *Journal of Intelligent & Robotic Systems*, vol. 82, pp. 399–411, 2016.
- [9] M. Rutkauskas, M. Asenov, S. Ramamoorthy, and D. T. Reid, “Autonomous multi-

- species environmental gas sensing using drone-based fourier-transform infrared spectroscopy,” *Optics Express*, vol. 27, no. 7, pp. 9578–9587, 2019.
- [10] J. Martin, M. Hanif, T. Hatanaka, J. Maestre, and E. Camacho, “Predictive receding-horizon multi-robot task allocation applied to the mapping of direct normal irradiance in a thermosolar power plant,” *Solar Energy*, vol. 263, p. 111911, 2023.
- [11] C. Qu, R. Singh, A. E. Morel, F. B. Sorbelli, P. Calyam, and S. K. Das, “Obstacle-aware and energy-efficient multi-drone coordination and networking for disaster response,” in *Proceedings of the 2021 17th International Conference on Network and Service Management*. IEEE, 2021, pp. 446–454.
- [12] C. Qu, F. B. Sorbelli, R. Singh, P. Calyam, and S. K. Das, “Environmentally-aware and energy-efficient multi-drone coordination and networking for disaster response,” *IEEE Transactions on Network and Service Management*, 2023.
- [13] E. Seraj and M. Gombolay, “Coordinated control of uavs for human-centered active sensing of wildfires,” in *Proceedings of the 2020 American Control Conference*. IEEE, 2020, pp. 1845–1852.
- [14] E. Seraj, A. Silva, and M. Gombolay, “Multi-uav planning for cooperative wildfire coverage and tracking with quality-of-service guarantees,” *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 2, p. 39, 2022.
- [15] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, “Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision,” *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.
- [16] S. Daftary, C. Hoppe, and H. Bischof, “Building with drones: Accurate 3d facade reconstruction using mavs,” in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*. IEEE, 2015, pp. 3487–3494.
- [17] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, “Cooperative uavs as a tool for aerial inspection of the aging infrastructure,” in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2017, pp. 177–189.
- [18] M. Mammarella, C. Donati, T. Shimizu, M. Suenaga, L. Comba, A. Biglia, K. Uto, T. Hatanaka, P. Gay, and F. Dabbene, “3d map reconstruction of an orchard using an angle-aware covering control strategy,” *IFAC-PapersOnLine*, vol. 55, no. 32, pp. 271–276, 2022.
- [19] S. K. Gupta and D. P. Shukla, “Application of drone for landslide mapping, dimension estimation and its 3d reconstruction,” *Journal of the Indian Society of Remote*

- Sensing*, vol. 46, pp. 903–914, 2018.
- [20] L. Comba, A. Biglia, D. R. Aimonino, and P. Gay, “Unsupervised detection of vineyards by 3d point-cloud uav photogrammetry for precision agriculture,” *Computers and Electronics in Agriculture*, vol. 155, pp. 84–95, 2018.
 - [21] M. Maboudi, M. Homaei, S. Song, S. Malihi, M. Saadatseresht, and M. Gerke, “A review on viewpoints and path planning for uav-based 3-d reconstruction,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 5026–5048, 2023.
 - [22] Y. Liu, R. Cui, K. Xie, M. Gong, and H. Huang, “Aerial path planning for online real-time exploration and offline high-quality reconstruction of large-scale urban scenes,” *ACM Transactions on Graphics*, vol. 40, no. 6, pp. 1–16, 2021.
 - [23] N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, “Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark,” 2018.
 - [24] T. Shimizu, S. Yamashita, T. Hatanaka, K. Uto, M. Mammarella, and F. Dabbene, “Angle-aware coverage control for 3-d map reconstruction with drone networks,” *IEEE Control Systems Letters*, vol. 6, pp. 1831–1836, 2021.
 - [25] 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.
 - [26] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon" next-best-view" planner for 3d exploration,” in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 1462–1468.
 - [27] S. Song and S. Jo, “Online inspection path planning for autonomous 3d modeling using a micro-aerial vehicle,” in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 6217–6224.
 - [28] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
 - [29] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, “Eyes in the sky: Decentralized control for the deployment of robotic camera networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.
 - [30] Y. Kantaros, M. Thanou, and A. Tzes, “Visibility-oriented coverage control of mobile robotic networks on non-convex regions,” in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 1126–1131.
 - [31] —, “Distributed coverage control for concave areas by a heterogeneous robot—

- swarm with visibility sensing constraints,” *Automatica*, vol. 53, pp. 195–207, 2015.
- [32] M. Schwager, J. McLurkin, and D. Rus, “Distributed coverage control with sensory feedback for networked robots,” in *Robotics: Science and Systems*, 2006, pp. 49–56.
 - [33] M. Schwager, D. Rus, and J.-J. Slotine, “Decentralized, adaptive coverage control for networked robots,” *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.
 - [34] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, “Voronoi coverage of non-convex environments with a group of networked robots,” in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4982–4989.
 - [35] Y. Stergiopoulos, M. Thanou, and A. Tzes, “Distributed collaborative coverage-control schemes for non-convex domains,” *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2422–2427, 2015.
 - [36] J. M. Palacios-Gasós, E. Montijano, C. Sagüés, and S. Llorente, “Distributed coverage estimation and control for multirobot persistent tasks,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1444–1460, 2016.
 - [37] K. Sugimoto, T. Hatanaka, M. Fujita, and N. Huebel, “Experimental study on persistent coverage control with information decay,” in *Proceedings of the 2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan*. IEEE, 2015, pp. 164–169.
 - [38] Y.-W. Wang, M.-J. Zhao, W. Yang, N. Zhou, and C. G. Cassandras, “Collision-free trajectory design for 2-d persistent monitoring using second-order agents,” *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 545–557, 2019.
 - [39] H. Dan, T. Hatanaka, J. Yamauchi, T. Shimizu, and M. Fujita, “Persistent object search and surveillance control with safety certificates for drone networks based on control barrier functions,” *Frontiers in Robotics and AI*, vol. 8, p. 740460, 2021.
 - [40] M. Suenaga, T. Shimizu, T. Hatanaka, K. Uto, M. Mammarella, and F. Dabbene, “Experimental study on angle-aware coverage control with application to 3-d visual map reconstruction,” in *Proceedings of the 2022 IEEE Conference on Control Technology and Applications*. IEEE, 2022, pp. 327–333.
 - [41] Z. Lu, M. Hanif, T. Shimizu, and T. Hatanaka, “Angle-aware coverage with camera rotational motion control,” *SICE Journal of Control, Measurement, and System Integration*, vol. 17, no. 1, pp. 211–221, 2024.
 - [42] Z. Lu, M. Hanif, and T. Hatanaka, “Angle-aware full 3d coverage control with admm-based dynamic assignment of charging stations,” in *Proceedings of the 2024 SICE Festival with Annual Conference*. IEEE, 2024, pp. 1119–1124.

- [43] M. Hanif, T. Shimizu, L. Zhiyuan, M. Suenaga, and T. Hatanaka, “Efficient angle-aware coverage control for large-scale 3d map reconstruction using drone networks,” *SICE Journal of Control, Measurement, and System Integration*, vol. 17, no. 1, pp. 144–155, 2024.
- [44] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.
- [45] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, “Building rome in a day,” *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.
- [46] B. Hepp, M. Nießner, and O. Hilliges, “Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction,” *ACM Transactions on Graphics*, vol. 38, no. 1, pp. 1–17, 2018.
- [47] E. M. Lee, S. Jung, S. Song, D. Choi, D. Lee, S. Lee, S. Kim, and H. Myung, “Ceo-mlcpp: Control-efficient and obstacle-aware multi-layer coverage path planner for 3d reconstruction with uavs,” in *Proceedings of the International Conference on Robot Intelligence Technology and Applications*. Springer, 2022, pp. 27–36.
- [48] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, “Neuralrecon: Real-time coherent 3d reconstruction from monocular video,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 598–15 607.
- [49] H. Gao, W. Mao, and M. Liu, “Visfusion: Visibility-aware online 3d scene reconstruction from videos,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 317–17 326.
- [50] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–15, 2022.
- [51] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [52] T. Chen, A. Swann, J. Yu, O. Shorinwa, R. Murai, M. Kennedy III, and M. Schwager, “Safer-splat: A control barrier function for safe navigation with online gaussian splatting maps,” *arXiv preprint arXiv:2409.09868*, 2024.
- [53] T. Chen, O. Shorinwa, J. Bruno, A. Swann, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager, “Splat-nav: Safe real-time robot navigation in gaussian splatting maps,” *IEEE Transactions on Robotics*, 2025.
- [54] Y. Ran, J. Zeng, S. He, J. Chen, L. Li, Y. Chen, G. Lee, and Q. Ye, “Neurar: Neural

- uncertainty for autonomous 3d reconstruction with implicit neural representations,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1125–1132, 2023.
- [55] Y. Tao, D. Ong, V. Murali, I. Spasojevic, P. Chaudhari, and V. Kumar, “Rt-guide: Real-time gaussian splatting for information-driven exploration,” *arXiv preprint arXiv:2409.18122*, 2024.
- [56] R. Jin, Y. Gao, H. Lu, and F. Gao, “Gs-planner: A gaussian-splatting-based planning framework for active high-fidelity reconstruction,” *arXiv preprint arXiv:2405.10142*, 2024.
- [57] Z. Xu, R. Jin, K. Wu, Y. Zhao, Z. Zhang, J. Zhao, Z. Gan, and W. Ding, “Hgs-planner: Hierarchical planning framework for active scene reconstruction using 3d gaussian splatting,” *arXiv preprint arXiv:2409.17624*, 2024.
- [58] A. Gusrialdi, S. Hirche, T. Hatanaka, and M. Fujita, “Voronoi based coverage control with anisotropic sensors,” in *Proceedings of the 2008 American Control Conference*. IEEE, 2008, pp. 736–741.
- [59] W. Bentz, T. Hoang, E. Bayasgalan, and D. Panagou, “Complete 3-d dynamic coverage in energy-constrained multi-uav sensor networks,” *Autonomous Robots*, vol. 42, pp. 825–851, 2018.
- [60] N. Hübel, S. Hirche, A. Gusrialdi, T. Hatanaka, M. Fujita, and O. Sawodny, “Coverage control with information decay in dynamic environments,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 4180–4185, 2008.
- [61] M. Santos, S. Mayya, G. Notomista, and M. Egerstedt, “Decentralized minimum-energy coverage control for time-varying density functions,” in *Proceedings of the 2019 International Symposium on Multi-robot and Multi-agent Systems*. IEEE, 2019, pp. 155–161.
- [62] M. Egerstedt, J. N. Pauli, G. Notomista, and S. Hutchinson, “Robot ecology: Constraint-based control design for long duration autonomy,” *Annual Reviews in Control*, vol. 46, pp. 1–7, 2018.
- [63] M. Hanif and T. Hatanaka, “Real-time adaptation of drone altitude and object detection model for moving target tracking,” in *Proceedings of the 2024 SICE Festival with Annual Conference*. IEEE, 2024, pp. 533–538.
- [64] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2009.
- [65] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, “Pixelwise view selection for unstructured multi-view stereo,” in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 501–518.

- [66] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, “Openmvg: Open multiple view geometry,” in *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1*. Springer, 2017, pp. 60–74.
- [67] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. De Lillo, and Y. Lanthony, “Alicevision meshroom: An open-source 3d reconstruction pipeline,” in *Proceedings of the 12th ACM Multimedia Systems Conference*, 2021, pp. 241–247.
- [68] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [69] C. Wu, “Towards linear-time incremental structure from motion,” in *Proceedings of the 2013 International Conference on 3D Vision*. IEEE, 2013, pp. 127–134.
- [70] A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Nießner, “Transformerfusion: Monocular rgb scene reconstruction using transformers,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 1403–1414, 2021.
- [71] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [72] J. Cortes, S. Martinez, and F. Bullo, “Spatially-distributed coverage optimization and control with limited-range interactions,” *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.
- [73] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [74] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *Proceedings of the 2019 18th European Control Conference*. Ieee, 2019, pp. 3420–3431.
- [75] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *Proceedings of the 53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6271–6278.
- [76] D. R. Agrawal and D. Panagou, “Safe control synthesis via input constrained control barrier functions,” in *Proceedings of the 2021 60th IEEE Conference on Decision and Control*. IEEE, 2021, pp. 6113–6118.
- [77] A. Alan, A. J. Taylor, C. R. He, A. D. Ames, and G. Orosz, “Control barrier functions and input-to-state safety with application to automated vehicles,” *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2744–2759, 2023.

- [78] W. S. Cortez, D. Oetomo, C. Manzie, and P. Choong, “Control barrier functions for mechanical systems: Theory and application to robotic grasping,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 530–545, 2019.
- [79] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, “Safety-critical manipulation for collision-free food preparation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 954–10 961, 2022.
- [80] H. Wang, J. Peng, F. Zhang, H. Zhang, and Y. Wang, “High-order control barrier functions-based impedance control of a robotic manipulator with time-varying output constraints,” *ISA Transactions*, vol. 129, pp. 361–369, 2022.
- [81] S.-C. Hsu, X. Xu, and A. D. Ames, “Control barrier function based quadratic programs with application to bipedal robotic walking,” in *Proceedings of the 2015 American Control Conference*. IEEE, 2015, pp. 4542–4548.
- [82] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, “Multi-layered safety for legged robots via control barrier functions and model predictive control,” in *Proceedings of the 2021 IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 8352–8358.
- [83] J. Kim, J. Lee, and A. D. Ames, “Safety-critical coordination for cooperative legged locomotion via control barrier functions,” in *Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2023, pp. 2368–2375.
- [84] H. Dan, J. Yamauchi, T. Hatanaka, and M. Fujita, “Control barrier function-based persistent coverage with performance guarantee and application to object search scenario,” in *Proceedings of the 2020 IEEE Conference on Control Technology and Applications*. IEEE, 2020, pp. 640–647.
- [85] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, “Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2897–2904, 2022.
- [86] G. Notomista and M. Egerstedt, “Constraint-driven coordinated control of multi-robot systems,” in *Proceedings of the 2019 American Control Conference*. IEEE, 2019, pp. 1990–1996.
- [87] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [88] Y. Chen, A. Singletary, and A. D. Ames, “Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach,” *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

- [89] B. Capelli and L. Sabattini, “Connectivity maintenance: Global and optimized approach through control barrier functions,” in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 5590–5596.
- [90] N. De Carli, P. Salaris, and P. R. Giordano, “Distributed control barrier functions for global connectivity maintenance,” in *Proceedings of the 2024 IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 12 048–12 054.
- [91] M. Rauscher, M. Kimmel, and S. Hirche, “Constrained robot control using control barrier functions,” in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 279–285.
- [92] Q. Du, V. Faber, and M. Gunzburger, “Centroidal voronoi tessellations: Applications and algorithms,” *SIAM Review*, vol. 41, no. 4, pp. 637–676, 1999.
- [93] S. Lloyd, “Least squares quantization in pcm,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [94] A. Gusrialdi, T. Hatanaka, and M. Fujita, “Coverage control for mobile networks with limited-range anisotropic sensors,” in *Proceedings of the 2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 4263–4268.
- [95] R. Funada, M. Santos, J. Yamauchi, T. Hatanaka, M. Fujita, and M. Egerstedt, “Visual coverage control for teams of quadcopters via control barrier functions,” in *Proceedings of the 2019 International Conference on Robotics and Automation*. IEEE, 2019, pp. 3010–3016.
- [96] G. Notomista and M. Egerstedt, “Persistification of robotic tasks,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 756–767, 2020.
- [97] E. Mendez-Flores, A. Pourshahidi, and M. Egerstedt, “Raccoonbot: An autonomous wire-traversing solar-tracking robot for persistent environmental monitoring,” *arXiv preprint arXiv:2501.14151*, 2025.
- [98] Y. Toyomoto, T. Oshima, K. Oishi, J. M. Maestre, and T. Hatanaka, “Constraint-driven multi-usv coverage path generation for aquatic environmental monitoring,” *arXiv preprint arXiv:2411.00579*, 2024.
- [99] X. Zheng, F. Wang, and Z. Li, “A multi-uav cooperative route planning methodology for 3d fine-resolution building model reconstruction,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 483–494, 2018.
- [100] M. Goesele, B. Curless, and S. M. Seitz, “Multi-view stereo revisited,” in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2006, pp. 2402–2409.
- [101] L. Zhou, G. Wu, Y. Zuo, X. Chen, and H. Hu, “A comprehensive review of vision-based 3d reconstruction methods,” *Sensors*, vol. 24, no. 7, p. 2314, 2024.

- [102] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, “Vision-only robot navigation in a neural radiance world,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.
- [103] S. Li, C. Li, W. Zhu, B. Yu, Y. Zhao, C. Wan, H. You, H. Shi, and Y. Lin, “Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction,” in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–13.
- [104] R. Scopigno, P. Cignoni, N. Pietroni, M. Callieri, and M. Dellepiane, “Digital fabrication techniques for cultural heritage: a survey,” in *Computer Graphics Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 6–21.
- [105] A. R. Widya, Y. Monno, M. Okutomi, S. Suzuki, T. Gotoda, and K. Miki, “Whole stomach 3d reconstruction and frame localization from monocular endoscope video,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, pp. 1–10, 2019.
- [106] —, “Stomach 3d reconstruction using virtual chromoendoscopic images,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, pp. 1–11, 2021.
- [107] P. Rodríguez-Gonzálvez, M. Rodríguez-Martín, L. F. Ramos, and D. González-Aguilera, “3d reconstruction methods and quality assessment for visual inspection of welds,” *Automation in Construction*, vol. 79, pp. 49–58, 2017.
- [108] J. Huang, J. Stoter, R. Peters, and L. Nan, “City3d: Large-scale building reconstruction from airborne lidar point clouds,” *Remote Sensing*, vol. 14, no. 9, p. 2254, 2022.
- [109] K. Qin, J. Li, S. Zlatanova, H. Wu, Y. Gao, Y. Li, S. Shen, X. Qu, Z. Yang, Z. Zhang *et al.*, “Novel uav-based 3d reconstruction using dense lidar point cloud and imagery: A geometry-aware 3d gaussian splatting approach,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 140, p. 104590, 2025.
- [110] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno, “A low cost 3d scanner based on structured light,” in *Computer Graphics Forum*, vol. 20, no. 3. Wiley Online Library, 2001, pp. 299–308.
- [111] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, “High quality depth map upsampling for 3d-tof cameras,” in *Proceedings of the 2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1623–1630.
- [112] Z. Murez, T. Van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, “Atlas: End-to-end 3d scene reconstruction from posed images,” in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 414–431.

- [113] K. Wang and S. Shen, “Mvdepthnet: Real-time multiview depth estimation neural network,” in *Proceedings of the 2018 International Conference on 3d Vision*. IEEE, 2018, pp. 248–257.
- [114] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, “Pixel2mesh: Generating 3d mesh models from single rgb images,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 52–67.
- [115] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, “Deepvoxels: Learning persistent 3d feature embeddings,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2437–2446.
- [116] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, “Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera,” in *Proceedings of the 24th Annual ACM symposium on User Interface Software and Technology*, 2011, pp. 559–568.
- [117] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, “Elasticfusion: Dense slam without a pose graph.” in *Robotics: Science and Systems*, vol. 11. Rome, 2015, p. 3.
- [118] S. Ullman, “The interpretation of structure from motion,” *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979.
- [119] J. Iglhaut, C. Cabo, S. Puliti, L. Piermattei, J. O’ Connor, and J. Rosette, “Structure from motion photogrammetry in forestry: A review,” *Current Forestry Reports*, vol. 5, pp. 155–168, 2019.
- [120] T. Samavati and M. Soryani, “Deep learning-based 3d reconstruction: a survey,” *Artificial Intelligence Review*, vol. 56, no. 9, pp. 9175–9219, 2023.
- [121] J. Wu, O. Wyman, Y. Tang, D. Pasini, and W. Wang, “Multi-view 3d reconstruction based on deep learning: A survey and comparison of methods,” *Neurocomputing*, vol. 582, p. 127553, 2024.
- [122] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [123] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in *Seminal Graphics: Pioneering Efforts that Shaped the Field*, 1998, pp. 347–353.
- [124] G. Chen and W. Wang, “A survey on 3d gaussian splatting,” *arXiv preprint*

- arXiv:2401.03890*, 2024.
- [125] Y. Takizawa, T. Sumino, L. Zhiyuan, K. Uto, and T. Hatanaka, “多角度撮影型被覆制御の農地フィールド実験,” in **自動制御連合講演会講演論文集 第 67 回自動制御連合講演会**. 自動制御連合講演会, 2024, pp. 1419–1426.
- [126] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, E. Berger, and A. Y. Ng, “Ros: An open-source robot operating system,” in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009, p. 5.
- [127] M. S. Andersen, J. Dahl, and L. Vandenberghe, “Cvxopt: A python package for convex optimization,” *Available at cvxopt.org*, vol. 54, 2013.
- [128] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.
- [129] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” 2018. [Online]. Available: <http://github.com/jax-ml/jax>
- [130] M. Hanif, T. Sumino, K. Uto, D. Ichihashi, K. Cheng, and T. Hatanaka, “Impact of real-time map feedback on coordinated image sampling for 3d reconstruction,” in *Proceedings of the 2025 European Control Conference*. IEEE, 2025, pp. 1372–1379.

Publications Related to This Research

Publications

Journal Articles

- [1] Muhammad Hanif, Takumi Shimizu, Zhiyuan Lu, Masaya Suenaga, and Takeshi Hatanaka,
“Efficient Angle-Aware Coverage Control for Large-Scale Map Reconstruction using Drone Networks,”
SICE Journal of Control, Measurement, and System Integration (JCMSI), Vol. 17, No. 1, pp. 144–155, 2024. [Chapters 2, 3]

International Conferences

- [2] Muhammad Hanif, Takumi Sumino, Kuniaki Uto, Daisuke Ichihashi, Kelvin Cheng, and Takeshi Hatanaka,
“Impact of Real-time Map Feedback on Coordinated Image Sampling for 3D Reconstruction,”
European Control Conference (ECC), Stockholm, Sweden, 2025. (Accepted) [Chapter 4]

Other Related Publications

Journal Articles

- [3] Zhiyuan Lu, Muhammad Hanif, Takumi Shimizu, and Takeshi Hatanaka,
“Angle-Aware Coverage with Camera Rotational Motion Control,”
SICE Journal of Control, Measurement, and System Integration (JCMSI), Vol. 17, No. 1, 2024.

International Conferences

- [4] Zhiyuan Lu, Muhammad Hanif, and Takeshi Hatanaka,
“Angle-Aware Full 3D Coverage Control with ADMM-based Dynamic Assignment
of Charging Stations,”
Proceedings of the SICE Annual Conference, pp. 1119–1124, 2024.

Domestic Conferences

- [5] 住野 匠, Muhammad Hanif, 宇都 有昭, 市橋 大介, K. Cheng, 畑中 健志,
「実時間 3 次元復元に基づく復元モデルフィードバック型被覆制御」,
第 12 回 計測自動制御学会 制御部門マルチシンポジウム, 大阪, 2025 年.