# Real-Time 3D Scene Representations for Robotic Systems

Xiangmin Xu

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow


November 2025

# Abstract

Real-time 3D scene representation is a fundamental capability for robotic systems operating in dynamic and resource-constrained environments. In this thesis, "real-time" refers to perception and reconstruction pipelines operating under update latencies ranging from milliseconds to seconds, depending on the level of representation fidelity and system interaction requirements. An agent that cannot perceive the three-dimensional structure of its surroundings in a timely and accurate manner inevitably operates with an incomplete and potentially misleading world model. In teleoperation, insufficient timeliness leads to delayed and unsafe control; in autonomy, insufficient fidelity undermines planning and obstacle avoidance. Despite its central importance, existing 3D scene representation pipelines typically prioritise either responsiveness or visual accuracy, rarely treating timeliness and fidelity as jointly coupled design objectives.

This thesis addresses this gap by developing a communication-aware framework for real-time 3D scene representation that explicitly models and optimises the timeliness–fidelity trade-off. The work begins with the design of an embodied, full-stack reconstruction system deployed on a robotic platform, integrating teleoperation, deterministic pose recovery, and fast neural scene optimisation for interactive scene updates. Building upon this system-level foundation, a multi-camera sensing architecture with distributed edge–cloud computation is introduced to study the temporal structure of networked perception under stochastic communication delays.

A mathematical model is formulated to link Age of Information (AoI) dynamics with 3D reconstruction fidelity, enabling quantitative analysis of how delayed or stale updates affect multi-view fusion and dynamic mapping. The resulting analysis characterises a frontier between freshness and representation quality, demonstrating that both periodic updates and naive AoI minimisation can be suboptimal under realistic communication conditions.

Beyond this analysis, a task-oriented communication framework is proposed on the edge, where the 3D scene representation scheduler is formulated as a reinforcement learning problem. The scheduler optimises task performance by jointly considering update timeliness, 3D scene representation fidelity, and bandwidth constraints, while incorporating image semantic information, camera extrinsics, and the age of information into the scheduling decision. To better capture task relevance, a semantic-aware scheduling mechanism is introduced, enabling the system to prioritise observations based on both visual content and temporal freshness. The framework further investigates two scheduling strategies, namely $\omega$-threshold and $\omega$-waiting

policies, revealing their different impacts on multi-view consistency and information timeliness. Experimental evaluations on multi-camera datasets and an outward-facing scene dataset demonstrate consistent improvements over periodic and AoI-driven baselines, particularly in dynamic environment scenarios where stale updates degrade mapping stability and excessive transmissions waste resources. Saliency-based analysis further provides interpretability, revealing the spatial and semantic cues influencing scheduling decisions.

Overall, this thesis establishes timeliness-aware 3D scene representation as a first-order system design problem rather than a secondary implementation concern. By integrating embodied acquisition, communication modelling, and task-driven scheduling, the work contributes a principled foundation for scalable, adaptive, and communication-aware robotic perception systems.

# Contents

# List of Figures

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my primary supervisor, Dr. Emma Li, for her continuous guidance, support, and patience throughout my doctoral study.

I am equally grateful to Dr. Philip G. Zhao for his valuable advice and technical discussions, which have significantly influenced the direction of my research. My sincere thanks also go to Professor David Flynn for his constructive feedback and broader perspective on the practical and industrial aspects of this work.

I would like to acknowledge Dr. Zhen Meng for his mentorship and generous support during my early research stages. His experience and guidance helped me navigate both academic and technical challenges.

I am deeply thankful to my parents for their unwavering support, trust, and encouragement throughout my academic journey. Their belief in me has been a constant source of strength.

Finally, I would like to thank my girlfriend, Miaorui, for her understanding, patience, and companionship. Her support has meant more than words can fully express.

This thesis would not have been possible without all of them.

# Declaration

I declare that this thesis is my own work and has not been submitted for any other degree at this or any other institution.

Except where explicitly stated otherwise, the research reported in this thesis was carried out by the author. All sources of information have been acknowledged by means of references.

Parts of this thesis are based on joint work with collaborators, and the specific contributions of the author are clearly indicated in the relevant chapters.

# Chapter 1

# Introduction

## 1.1 Research Motivation

### 1.1.1 Challenges in robotic 3D scene understanding

Real-time 3D scene representation is not merely a perception component but a structural prerequisite for stable robotic operation in dynamic environments. Throughout this thesis, robotic teleoperation and embodied inspection are used primarily as representative application scenarios that motivate the timeliness requirements of real-time 3D scene representation systems. The core analysis and evaluation focus on communication-aware perception and scheduling behaviour under controlled datasets and simulated transmission dynamics. In both teleoperation and autonomous systems, decisions are executed against an internal world model whose validity depends on two coupled properties: its geometric fidelity and its temporal freshness. When either property deteriorates, the robot operates on an incomplete or outdated representation of reality, leading to degraded control, unsafe interaction, or unstable planning.

These limitations become especially pronounced in dynamic environments, where the scene evolves faster than traditional reconstruction algorithms can update. Delayed or stale information can degrade human–robot interaction in teleoperated systems and destabilize planning and control for autonomous platforms. As a result, understanding and quantifying the tradeoff between timeliness and fidelity is not just an academic interest, but a practical necessity for reliable robotic operation.

The increasing availability of multi-camera setups, edge computing resources, and robot-mounted sensing platforms creates an opportunity to revisit this tradeoff with a more integrated perspective. Instead of treating perception, communication, and control as isolated components, there is a need for a unified framework that captures their interdependencies. By addressing the fundamental tension between reconstruction accuracy and update latency, this research aims to push robotic 3D scene understanding toward systems that are both responsive and reliable under real-world constraints.

### 1.1.2 The timeliness-fidelity tradeoff in dynamic environments

The real-time performance of a 3D scene representation ensures that the robot acts on information that reflects the current state of its surroundings, which is particularly critical for teleoperated systems where delayed feedback directly diminishes operator control, stability, and safety. High-fidelity 3D scene representation, on the other hand, provides the geometric accuracy and structural completeness required for tasks such as precise manipulation, collision-free navigation, and meaningful scene interpretation. Without sufficient detail, the robot's understanding of its environment becomes coarse or misleading; without timeliness, even a high-quality reconstruction becomes obsolete before it can be used.

The primary downstream motivation of the proposed framework is robotic telepresence and teleoperation, where remote operators rely on continuously updated 3D scene representations to maintain situational awareness and make effective decisions under communication and computation constraints. In such systems, stale or temporally inconsistent reconstructions may directly degrade operator perception, inspection reliability, and remote interaction quality. Consequently, the framework is particularly designed for perception-oriented robotic scenarios where maintaining temporally relevant scene understanding is more critical than achieving ultra-low-latency control-loop response.

However, achieving both speed and accuracy simultaneously is fundamentally challenging, especially in dynamic environments. When objects, humans, or the robot itself are moving, the scene evolves more rapidly than traditional reconstruction pipelines can update. As a result, the system is constantly forced to choose between updating quickly with less processed data or delaying updates to compute a more detailed and consistent model. This tension is not incidental but structural: any real-time 3D scene representation pipeline operating under bounded computation and communication resources must allocate limited budget between update frequency and representation refinement.

In practice, this tradeoff affects every level of robotic perception. For teleoperation, slow or highly processed reconstructions break the sense of real-time situational awareness, resulting in poor human control and delayed corrective actions. For autonomous operation, latency-induced inconsistencies in the world model propagate into planning and control algorithms, creating navigation errors, unstable trajectories, or incorrect decisions in rapidly changing scenes. Even small delays can accumulate into significant deviations when robots must make high-frequency decisions or operate in close proximity to obstacles. In teleoperation scenarios, delays on the order of tens to hundreds of milliseconds may already degrade operator responsiveness and situational awareness. In the communication-aware multi-camera settings studied in this thesis, temporal inconsistency primarily arises from stochastic transmission delays and asynchronous sensor updates, which lead to stale observations across distributed camera streams.

## 1.2 Problem Definition and Research Questions

The central hypothesis of this thesis is that timeliness and fidelity should not be treated as secondary implementation concerns, but as first-order design variables in robotic perception systems. The core research problem is therefore to determine how sensing, communication, and reconstruction policies can be jointly designed such that the resulting world model remains both temporally relevant and structurally reliable under resource constraints.

Formally, the problem can be expressed as determining when and how sensor data should be transmitted and integrated into the reconstruction pipeline to maintain an accurate and up-to-date scene representation under limited resources. This includes modeling the relationship between update latency, scene dynamics, and reconstruction error, as well as designing mechanisms that enable the system to adapt its behavior according to task demands and environmental conditions.

Based on this formulation, the research is guided by the following questions:

- How can timeliness and fidelity be jointly characterized and quantified for real-time 3D scene representations in robotic systems?
- How can sensing, communication, and real-time 3D scene representation be coordinated to achieve task-relevant performance rather than raw visual fidelity alone?
- Can adaptive, learning-based scheduling improve the efficiency and reliability of multi-camera 3D scene representation compared to periodic or AoI-driven baselines?

Addressing these questions provides a structured path toward a system capable of navigating the practical constraints of real-time robotic perception while maintaining the scene quality required for stable teleoperation and autonomous decision-making.

## 1.3 Research Scope and Framework Overview

This thesis focuses on the design and analysis of a robotic-based real-time 3D scene representation framework that operates across both teleoperation and autonomous modes. The scope is deliberately centered on the end-to-end pipeline—from sensing and communication to reconstruction and decision-making—because the timeliness–fidelity tradeoff emerges from the interaction of these components rather than any single module in isolation.

Throughout this thesis, several related but distinct terms are used to describe different aspects of 3D scene representation performance. “Fidelity” refers to the visual and perceptual quality of the reconstructed scene representation relative to the underlying environment, typically measured using metrics such as PSNR, SSIM, and LPIPS, which are formally introduced in Section 5.3.4. In contrast, “accuracy” refers more specifically to geometric or photometric correctness with respect to ground-truth observations or scene structure. Similarly, “temporal

freshness" describes how recently an observation was generated relative to its usage time, and is primarily quantified using Age of Information (AoI). This differs from "latency," which refers to the communication or processing delay experienced during transmission or reconstruction. Finally, "structural reliability" refers to the consistency and stability of reconstructed geometric or perceptual structures across viewpoints and time, particularly under asynchronous or stale multi-view observations.

The overall framework considered throughout this thesis consists of three tightly coupled components: embodied sensing and 3D scene acquisition, communication-aware observation transmission, and timeliness-aware scene representation and scheduling. While these components are investigated under different assumptions and levels of abstraction in different chapters, they should be interpreted as progressively connected parts of a unified robotic perception framework rather than independent systems. In particular, Chapter 4 focuses on an embodied robotic implementation for interactive 3D scene acquisition and reconstruction, where teleoperation and active viewpoint selection are studied within a practical robotic inspection setting. Building upon this embodied context, Chapters 5 and 6 abstract the system toward communication-aware multi-sensor perception models in order to analyse the timeliness–fidelity tradeoff under controlled and reproducible communication dynamics. Chapter 5 studies the fundamental relationship between observation freshness and reconstruction fidelity under stochastic sensing and transmission delays, while Chapter 6 further extends the framework toward burst-aware and task-oriented scheduling strategies under more heterogeneous sensing conditions. Consequently, the robotic teleoperation and embodied inspection scenarios considered throughout this thesis should primarily be interpreted as motivating application contexts for timeliness-aware 3D perception and communication-aware scene representation, rather than fully deployment-optimised end-to-end robotic products.

The framework integrates a camera-based perception system, an edge–cloud computational architecture, and a mobile robotic platform capable of switching between human-in-the-loop and autonomous control. In teleoperation mode, the system prioritizes immediate situational awareness to ensure responsive and stable human control. In autonomous mode, the robot learns from human preferences and executes tasks without direct human oversight. This dual-mode design allows the same reconstruction pipeline to be examined under two fundamentally different operational requirements, exposing complementary constraints on timeliness and fidelity.

A major focus of the framework is its distributed architecture: visual data are captured onboard, processed at the edge, and optionally transmitted to remote operators or cloud resources depending on task demands. This edge–cloud collaboration enables scalability but introduces communication delays that directly affect reconstruction timeliness. By embedding these constraints into the system design, the framework provides a realistic platform for analyzing how sensing, communication, and computation jointly determine the achievable performance.

Overall, the scope of this research is to establish a unified perspective that connects robotic

perception, real-time 3D reconstruction, teleoperation, autonomous navigation, and networked edge-compute systems. This integrated viewpoint forms the basis for subsequent theoretical modeling and algorithmic development targeting the timeliness–fidelity balance.

## 1.4 Key Contributions

This thesis makes the following contributions:

- **An embodied robotic platform for real-time 3D scene representation and active viewpoint planning.** A full-stack monocam robotic perception system that integrates data acquisition, deterministic pose recovery, and real-time neural scene reconstruction within an edge–cloud architecture. The platform enables closed-loop integration between robotic sensing and scene reconstruction, and further supports active viewpoint planning strategies such as next-best-view optimization based on reinforcement learning and human feedback.
- **Formalisation and analysis of the timeliness–fidelity tradeoff.** A mathematical model linking Age of Information dynamics with reconstruction quality metrics, enabling quantitative characterisation of how stochastic communication delays influence multi-view fusion and dynamic mapping performance.
- **A task-oriented adaptive communication strategy.** A reinforcement learning-based scheduler that selects when and what sensor data to transmit by optimising downstream task performance rather than reconstruction fidelity alone, demonstrating measurable gains over periodic and AoI-driven baselines.

## 1.5 Thesis Structure

This thesis is organised into six chapters, progressing from fundamental concepts in 3D scene representation to task-oriented, timeliness-aware perception and communication systems for robotic applications.

Chapter 1 introduces the research background, motivation, and problem setting. It discusses the limitations of conventional perception pipelines in networked and resource-constrained environments, and highlights the need for representations and communication strategies that jointly consider timeliness and fidelity. The main research questions and contributions of the thesis are also summarised.

Chapter 2 reviews the related literature in three main areas: 3D scene representations, timeliness-aware and task-oriented communication systems, teleoperation systems, and next-best-view planning. It surveys both explicit and implicit scene representations, as well as recent neural

rendering approaches. The chapter also introduces the concept of the Age of Information and its role in modelling timeliness in networked sensing systems. This review establishes the theoretical and practical gaps that motivate the research contributions of this thesis.

Chapter 3 introduces fundamental mathematical concepts for implicit/explicit 3D scene representations and image similarity metrics, which serve as the foundation for the subsequent technical chapters.

Chapter 4 presents the mathematical formulation of timeliness-aware 3D scene representation systems. It first introduces the fundamental models of explicit and implicit 3D representations, and provides a unified mathematical view of scene reconstruction. The chapter then formulates the notion of timeliness using the Age of Information and extends it to the context of 3D perception. A decision-theoretic framework is introduced to describe how sensing, communication, and reconstruction actions affect both information freshness and reconstruction fidelity. This chapter serves as the theoretical foundation for the methods developed in the subsequent chapters.

The core technical contributions of the thesis are presented in Chapters 5 to 7, each corresponding to a major research work along the theme of task-oriented, timeliness-aware 3D perception.

Chapter 5 presents a teleoperation-based embodied 3D scene representation framework for robotic inspection in nuclear decommissioning environments. It introduces a real-time reconstruction pipeline that integrates haptic teleoperation, deterministic camera pose recovery, and fast neural radiance field optimisation for interactive scene updating. Built upon this embodied acquisition and 3D scene representation system, the chapter further demonstrates how next-best-view (NBV) planning can be incorporated into the perception loop to prioritise task-relevant regions. By leveraging expert operator preferences, the extended framework enables active viewpoint selection that improves inspection efficiency and reconstruction quality under limited sensing budgets. This chapter is partially based on a UKAEA RAICo funded project, and also partially based on: *Preference-Driven Active 3D Scene Representation for Robotic Inspection in Nuclear Decommissioning*, submitted to IEEE/RSJ IROS 2026.

Chapter 6 investigates the fundamental timeliness–fidelity tradeoff in real-time 3D scene representation systems. It establishes a communication-aware reconstruction framework, where multiple cameras transmit images to an edge server over stochastic wireless channels. The transmission delays are explicitly modelled using probabilistic channel models, allowing the system to analyse how communication latency affects reconstruction quality. Within this framework, the scheduling problem is formulated based on the Age of Information (AoI) of each camera stream. A single-step reinforcement learning method is introduced, where the system state is defined purely by the AoI values of the cameras, and the agent decides whether each received image should be used for reconstruction. The objective is to directly optimise reconstruction quality metrics such as PSNR, SSIM, and LPIPS under stochastic communication delays. In

addition to the reinforcement learning solution, the chapter also studies threshold-based waiting strategies and characterises how different maximum-AoI thresholds affect reconstruction performance. The results reveal that the optimal waiting threshold depends on the underlying channel conditions, demonstrating a non-trivial relationship between communication delay and reconstruction fidelity. This chapter is based on *Timeliness-Fidelity Tradeoff in 3D Scene Representations*, IEEE INFOCOM 2024 Workshops.

Chapter 7 builds upon the tradeoff analysis developed in Chapter 6 and extends it to a task-oriented communication framework for real-time 3D scene representation. Instead of focusing solely on communication delay and AoI dynamics, this chapter considers a more realistic cross-system setting, where sensing, communication, and task objectives jointly influence the scheduling decisions. The system is modelled under bursty or task-dependent communication conditions, and the scheduling problem is formulated as a contextual decision-making process. Compared with the simplified formulation in Chapter 5, the reinforcement learning state incorporates not only timeliness indicators but also task-related or semantic information extracted from the incoming data streams. This enables the scheduler to adapt its waiting horizon according to both communication dynamics and task requirements. The proposed approach is evaluated across multiple datasets and representation models, including neural radiance fields and 3D Gaussian Splatting, under more complex communication scenarios. Different adaptive waiting patterns are learned by the policy, demonstrating how task-oriented scheduling can outperform fixed-threshold or single-step strategies. This chapter is based on: *Task-Oriented Communications for 3D Scene Representation: Balancing Timeliness and Fidelity*, submitted to IEEE Transactions on Mobile Computing.

Finally, Chapter 8 concludes the thesis by summarising the main contributions and discussing directions for future research, including extensions to large-scale robotic systems, multi-agent perception, and embodied intelligence applications.

## 1.6 List of Publications

**Xu, X.**, Meng, Z., Zhang, Y., She, C., & Zhao, P. G. (2024, May). Timeliness-Fidelity Tradeoff in 3D Scene Representations. In *IEEE INFOCOM 2024 Workshops* (pp. 1–7). IEEE.

**Xu, X.**, Meng, Z., Chen, K., Yang, J., Li, E., Zhao, P. G., & Flynn, D. (2025). Task-Oriented Communications for 3D Scene Representation: Balancing Timeliness and Fidelity. arXiv preprint arXiv:2509.17282. Submitted to *IEEE Transactions on Mobile Computing*. (Under Review)

**Xu, X.**, Meng, Z., Li, E., Khamis, M., Zhao, P. G., & Bretin, R. (2025, May). Understanding dynamic human-robot proxemics in the case of four-legged canine-inspired robots. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 7808–7814). IEEE.

Chen, K.*, Meng, Z.*, **Xu, X.***, She, C., & Zhao, P. G. (2024, October). Real-Time Inter-

actions Between Human Controllers and Remote Devices in Metaverse. In *IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)* (pp. 353–358). IEEE. (*Equal contribution)

Meng, Z., Chen, K., **Xu, X.**†, Pulgarin, E. J. L., Li, E., Zhao, P. G., & Flynn, D. Preference-Driven Active 3D Scene Representation for Robotic Inspection in Nuclear Decommissioning. Submitted to the *2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (Under Review) (†Corresponding author)

Diao, Y., Meng, Z., **Xu, X.**, She, C., & Zhao, P. G. (2024, May). Task-oriented source-channel coding enabled autonomous driving based on edge computing. In *IEEE INFOCOM 2024 Workshops* (pp. 1–6). IEEE.

Avogaro, A., Toaiari, A., Cunico, F., **Xu, X.**, Dafas, H., Vinciarelli, A., & Cristani, M. (2024, October). Exploring 3D Human Pose Estimation and Forecasting from the Robot's Perspective: The HARPER Dataset. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5828–5835). IEEE.

Yang, J., Meng, Z., **Xu, X.**, Chen, K., Li, E. L., & Zhao, P. G. (2025, January). Task-Oriented Edge-Assisted Cooperative Data Compression, Communications and Computing for UGV-Enhanced Warehouse Logistics. In *IEEE Consumer Communications & Networking Conference (CCNC)* (pp. 1–8). IEEE.

Mitchell, D., Emor Baniqued, P. D., Zahid, A., West, A., Nouri Rahmat Abadi, B., Lennox, B., **Xu, X.**, & Jiang, Z. (2023). Lessons learned: Symbiotic autonomous robot ecosystem for nuclear environments. *IET Cyber-Systems and Robotics*, 5(4), e12103.

Ma, M., Lou, C., **Xu, X.**, Yang, J., Cunningham, J., & Zhang, L. (2024). Distributionally robust decarbonizing scheduling considering data-driven ambiguity sets for multi-temporal multi-energy microgrid operation. *Sustainable Energy, Grids and Networks*, 38, 101323.

# Chapter 2

# Literature Review

## 2.1 Robotic Vision Systems

Early robotic vision systems were predominantly based on single-camera Simultaneous Localization and Mapping (SLAM), focusing on real-time ego-motion estimation and sparse environment mapping. MonoSLAM [8] by Davison et al. demonstrated that a single moving camera can function as a real-time pose sensor by jointly estimating camera motion and a sparse map of natural landmarks within a Bayesian filtering framework. This work established visual SLAM as a viable alternative to range-based sensing modalities such as Light Detection and Ranging (LiDAR). Parallel Tracking and Mapping (PTAM) [9] introduced a key architectural insight by decoupling tracking and mapping into parallel threads. This separation enabled accurate keyframe-based bundle adjustment while maintaining real-time performance in small-scale augmented reality scenarios. Building upon these ideas, ORB-SLAM [10] provided a versatile and highly accurate monocular SLAM framework integrating robust feature extraction, keyframe management, loop closure detection, and relocalization. Due to its reliability and extensibility, ORB-SLAM has become a de facto baseline in robotic perception and augmented reality applications.

While early systems emphasized sparse feature-based representations, subsequent work extended visual SLAM toward dense scene reconstruction. KinectFusion [46] demonstrated that a moving RGB-D camera can reconstruct dense volumetric geometry in real time by integrating depth measurements into a Truncated Signed Distance Function (TSDF) representation while simultaneously estimating camera pose using iterative closest point (ICP). This work showed that real-time dense mapping was feasible on commodity hardware and significantly expanded the scope of robotic scene understanding. ElasticFusion [11] further advanced dense visual SLAM by introducing a surfel-based map representation combined with a deformation graph to maintain global consistency without relying on an explicit pose graph. Such dense mapping systems highlighted the importance of maintaining both geometric fidelity and temporal coherence in reconstructed environment models.

Beyond purely visual pipelines, robotic perception has increasingly adopted visual–inertial fusion to improve robustness under aggressive motion and challenging illumination conditions. Leutenegger et al. proposed OKVIS [12], which formulates visual–inertial odometry as a tightly coupled nonlinear optimization over landmark reprojection and inertial residuals. Similarly, VINS-Mono [13] introduced a sliding-window optimization framework with online extrinsic calibration and loop closure, and has been widely adopted in aerial and mobile robotics. Related systems such as MSCKF [101] and ROVIO [102] further demonstrate how tightly integrated sensor fusion and continuous optimization have become standard design principles in modern robotic vision systems.

While classical SLAM focuses primarily on geometric mapping, recent research has increasingly explored richer environment representations that jointly encode geometry, appearance, and semantic information. For example, semantic SLAM approaches incorporate object-level information into the mapping process, enabling robots to reason about the environment at a higher level of abstraction. Systems such as Kimera [28] combine metric reconstruction with semantic scene understanding, allowing robots to build dense maps enriched with object labels and structural information.

More recently, neural implicit representations have emerged as a powerful paradigm for robotic scene reconstruction. Neural Radiance Fields (NeRF) [22] represent scenes as continuous volumetric functions that map spatial coordinates and viewing directions to density and color, enabling photorealistic novel view synthesis from sparse multi-view observations. Subsequent extensions such as Instant-NGP [52] dramatically accelerated training and rendering by introducing multiresolution hash encoding, making neural scene reconstruction feasible for near real-time applications. These neural representations provide significantly richer scene models than traditional geometric maps, capturing both fine geometric structure and view-dependent appearance. To integrate neural scene representations with robotic perception, several works have explored neural SLAM systems. For example, iMAP [27] proposed a real-time neural implicit mapping framework in which scene representation and camera pose estimation are optimized jointly. NICE-SLAM [31] introduced hierarchical feature grids to improve scalability and enable high-resolution reconstruction in large indoor scenes. Such systems demonstrate that neural implicit representations can potentially replace traditional volumetric maps as the core scene representation in robotic perception pipelines.

More recently, explicit neural representations such as 3D Gaussian Splatting [54] have further advanced real-time neural rendering. In this approach, scenes are modeled as collections of anisotropic Gaussian primitives that can be rasterized efficiently on the GPU, achieving real-time rendering speeds while maintaining high visual fidelity. These developments have led to a new generation of neural mapping systems capable of supporting both real-time reconstruction and interactive visualization.

As robotic perception systems become increasingly capable of building rich environment

models, the concept of a robotic *world model* has gained renewed attention. In robotics and embodied AI, a world model refers to an internal representation that captures the structure and dynamics of the surrounding environment, allowing an agent to predict future observations and plan actions accordingly. Recent research in embodied AI and reinforcement learning has explored neural world models that combine perception, representation learning, and predictive modeling into a unified framework. These models aim to enable robots to reason about their environment beyond immediate sensor observations, supporting tasks such as planning, simulation, and decision making.

Overall, the literature on robotic vision has produced highly robust real-time SLAM and reconstruction pipelines that enable localization, mapping, and dense scene modeling using onboard sensing. However, most of these systems are designed under the assumption of locally available measurements and centralized computation. The role of communication constraints and information timeliness remains largely unaddressed at the perception modeling level. This limitation becomes particularly critical in networked robotic systems, where multiple sensors, cameras, or agents must share information across communication networks. Understanding how scene representations evolve under communication delay and sensing timeliness constraints therefore becomes an essential problem for next-generation robotic perception systems.

## 2.2 Multi-Camera and Multi-Agent Perception

In multi-robot and multi-camera settings, distributed state estimation has been studied extensively in the robotics and control communities. Early work focused on maintaining globally consistent maps while allowing robots to operate under communication constraints. Cunningham et al. [62] proposed the Distributed Delayed-State Information Filter (DDF-SAM), which enables consistent multi-robot SLAM by explicitly modelling delayed measurements and decentralized updates. Similarly, Howard et al. [95] investigated multi-robot mapping through decentralized data fusion, showing that collaborative mapping significantly improves environment coverage and robustness compared to single-robot systems. These distributed formulations aim to preserve global consistency while reducing bandwidth usage, highlighting the growing importance of communication-aware estimation in networked robotic systems.

More recently, centralized collaborative systems have demonstrated real-time multi-agent mapping capabilities. COVINS [63] enables scalable multi-agent visual–inertial mapping by streaming keyframes to a central server that performs global optimization. Similarly, Kimera-Multi [28] extends metric-semantic SLAM to multi-robot scenarios, allowing agents to share dense mesh reconstructions and semantic information across the system. Related systems such as CCM-SLAM [96] and collaborative ORB-SLAM variants further demonstrate how centralized map fusion can maintain global consistency across multiple robots. These systems illustrate that multi-camera and multi-agent perception significantly improve robustness, field-of-view

coverage, and loop closure detection compared to single-agent pipelines, particularly in large-scale environments.

Beyond classical feature-based SLAM, multi-view geometry has long provided the theoretical foundation for multi-camera reconstruction. The formulation of multiple view geometry by Hartley and Zisserman [48] formalized epipolar constraints, projective geometry, and bundle adjustment for multi-view estimation. These geometric principles underpin modern structure-from-motion (SfM) pipelines, such as those implemented in systems like COLMAP [91]. Building on these foundations, multi-view stereo (MVS) methods reconstruct dense surface geometry from calibrated image sets. Large-scale benchmarks introduced by Seitz et al. [92] established quantitative standards for evaluating reconstruction accuracy. Subsequent MVS methods, including PatchMatch-based stereo approaches [97, 98], significantly improved scalability and reconstruction density. While highly accurate, traditional MVS pipelines are typically offline and computationally intensive, limiting their applicability in real-time robotic perception scenarios.

The emergence of neural implicit representations has significantly reshaped multi-view reconstruction. Neural Radiance Fields (NeRF) [22] model scenes as continuous volumetric functions that map spatial coordinates and viewing directions to density and radiance, allowing photorealistic novel view synthesis from sparse multi-view observations. Subsequent extensions such as NeRF++ [30], Mip-NeRF [26], and Instant-NGP [52] improved scalability, anti-aliasing, and training efficiency. More recent variants such as NeRF-W [99] and TensoRF [56] further addressed challenges related to dynamic scenes, appearance variations, and efficient neural representation. These models inherently rely on multi-camera supervision, as scene geometry emerges from photometric consistency across multiple viewpoints.

To bridge neural fields and robotic perception, several works have integrated NeRF into SLAM systems. iMAP [27] introduced a neural implicit map that is jointly optimized with camera poses in real time. NICE-SLAM [31] further improved scalability by incorporating hierarchical feature grids, enabling higher resolution scene reconstruction. NeRF-SLAM [29] combined NeRF-based mapping with classical pose tracking, demonstrating improved reconstruction fidelity in indoor environments. These approaches suggest that neural representations can potentially replace traditional volumetric maps in multi-camera robotic systems, providing richer geometry and appearance modelling.

More recently, 3D Gaussian Splatting (3DGS) [54] introduced an explicit neural scene representation based on a set of anisotropic Gaussian primitives. Unlike volumetric ray integration used in NeRF, 3DGS performs differentiable rasterization of projected Gaussians, achieving real-time rendering speeds while maintaining high visual fidelity. This representation has quickly attracted interest in robotics and real-time mapping. Extensions such as Gaussian-SLAM [55] and Splat-SLAM [55] integrate Gaussian-based scene representations into real-time SLAM pipelines. Compared to classical multi-camera SLAM, neural field-based systems exhibit several distinct properties. First, geometry and appearance are jointly represented within

a continuous neural function. Second, they rely on dense photometric supervision rather than sparse feature correspondences. Third, they introduce significantly higher computational and memory demands, particularly when multiple high-resolution image streams are processed simultaneously.

Despite these advances, most existing multi-camera and neural 3D scene representation systems assume that all camera views are temporally synchronized and immediately available for optimization. In practical robotic deployments, however, visual observations are often transmitted over networks with limited bandwidth and non-negligible latency. The influence of communication delay, packet loss, or view staleness on neural multi-view consistency remains largely unexplored. As multi-camera perception increasingly moves toward distributed and edge–cloud architectures, understanding how information timeliness interacts with neural scene representation becomes essential. This observation motivates the timeliness-aware modelling and communication scheduling strategies developed in the subsequent chapters of this thesis.

## 2.3 Timeliness-Aware Perception

In classical computer vision pipelines, temporal information is primarily treated as a modeling cue for motion, dynamics, or temporal coherence rather than as a constrained system resource. Temporal modeling is typically embedded through optical flow estimation [64], recurrent architectures, temporal attention, or multi-frame consistency regularization in neural rendering [22, 67]. In these formulations, time is assumed to be dense, synchronized, and locally accessible. Temporal misalignment arises from scene motion or non-rigid geometry, not from communication latency or update staleness.

Networked perception systems introduce a fundamentally different temporal regime. Observations may arrive late, out of order, or be intermittently dropped due to bandwidth constraints and channel dynamics. This problem has long been studied in control and estimation theory under delay-aware and packet-loss-aware filtering. Sinopoli et al. [66] showed that Kalman filtering with intermittent observations exhibits a critical packet loss threshold beyond which estimation error diverges. Schenato et al. [65] and Hespanha et al. [68] characterized stability and performance degradation in networked control systems under stochastic delays. These works formally establish that timeliness is not merely an implementation issue but a variable that directly governs estimation error dynamics.

The concept of Age of Information (AoI) was later introduced by Kaul et al. [15] as a metric that captures the freshness of information at a receiver. Unlike latency, which measures transport delay of a specific packet, AoI quantifies the time elapsed since the generation of the most recently received update. Subsequent surveys and theoretical developments [17, 71] demonstrated that minimizing transmission frequency does not necessarily minimize estimation error. Optimal policies must jointly consider update generation rate, queueing dynamics, and service

discipline. Importantly, AoI introduces a system-level coupling between communication and estimation performance.

In remote state estimation, AoI-aware scheduling has been shown to outperform naive periodic updates. Sun et al. [16] derived optimal sampling policies that explicitly balance freshness and resource constraints. Kadota et al. [71] characterized optimal broadcast scheduling strategies under stochastic arrivals. More recently, Chen et al. [36] explicitly investigated the timeliness–fidelity tradeoff in wireless sensor networks, analyzing whether a fusion center should wait for all sensor updates or proceed with only a subset of available measurements. Their results show that waiting for additional observations can reduce estimation variance but simultaneously increase Age of Information, thereby introducing a non-trivial optimal stopping structure. These findings highlight that timeliness and fidelity are fundamentally coupled objectives rather than independent performance metrics. Despite these advances, the majority of AoI-aware estimation frameworks consider low-dimensional linear dynamical systems. The system state is typically modeled as a vector in $\mathbb{R}^n$, and fidelity is quantified through mean-square error or covariance trace. In contrast, modern visual perception operates on high-dimensional image observations and learns implicit scene representations whose parameters may number in the millions. In multi-view 3D sensing, geometry and appearance emerge from jointly optimizing photometric consistency across views. When some views are stale relative to the current scene state, temporal misalignment can violate geometric consistency assumptions, leading to artifacts such as ghosting, blurred surfaces, or inconsistent geometry.

Only limited work has attempted to connect timeliness metrics with visual perception. Communication efficient inference systems compress features or selectively transmit frames [69, 70], while video streaming systems adapt bitrate to network conditions. However, these approaches primarily optimize bandwidth or latency, and do not explicitly model information freshness as a variable affecting reconstruction fidelity.

Timeliness-aware perception therefore extends delay-aware estimation into the high dimensional regime. Rather than treating freshness as an external communication metric, it treats AoI as an intrinsic variable that shapes the effective dataset available to the perception module. The fidelity of the learned representation becomes jointly determined by sensing noise, model capacity, and the freshness profile of observations. This shift reframes perception as a coupled communication–estimation–learning problem, where scheduling policies and representation learning must be co-designed.

## 2.4 Real-Time Teleoperation and Autonomous Control

Teleoperation has long been a central paradigm in robotics, enabling human operators to control remote manipulators in hazardous or inaccessible environments such as nuclear facilities, deep-sea exploration, and space operations. Early foundational work primarily focused on achieving

stability and transparency when communication delays are present. Anderson et al. [85] introduced passivity-based bilateral control frameworks to guarantee stability in delayed teleoperation loops. Similarly, Niemeyer and Slotine [86] proposed the wave-variable transformation, which reformulates exchanged control signals into energy-consistent variables to maintain stability under constant communication delay. These studies established the theoretical foundations for delay-robust teleoperation systems and inspired a large body of subsequent work on time-delay compensation and stability analysis.

As teleoperation systems evolved toward distributed and networked architectures, research expanded into the broader domain of networked control systems (NCS). In such systems, sensing, control computation, and actuation may be separated by communication networks with limited bandwidth and variable latency. Hespanha et al. [87] provided a comprehensive survey of networked control systems, demonstrating how communication constraints directly affect closed-loop stability and performance. Similarly, Sinopoli et al. [66] showed that intermittent observations in Kalman filtering lead to a critical packet loss threshold, beyond which estimation error diverges. These findings reveal that communication reliability and sensing availability are fundamental factors influencing control performance in networked robotic systems.

Beyond stability considerations, modern teleoperation increasingly emphasizes the role of high-dimensional perceptual feedback. Sheridan [88] investigated human supervisory control in teleoperation, highlighting how perception, cognition, and operator workload influence task efficiency. Drascic and Milgram [89] studied the perceptual and performance effects of video delay in teleoperation tasks, demonstrating that even moderate latency can significantly degrade human operator efficiency and task accuracy. Subsequent studies in telepresence robotics and human–robot interaction further showed that high-fidelity visual feedback is essential for maintaining operator situational awareness [88, 100].

In contemporary robotic systems, remote environments are often represented through continuously updated 3D models or digital twins. Such representations enable operators to observe and interact with the remote environment through reconstructed virtual scenes rather than raw camera streams alone. Tao et al. [90] surveyed digital twin technologies for cyber–physical systems, highlighting their role in real-time integration between physical processes and virtual models. In robotics, similar concepts appear in map-based teleoperation and model-mediated control, where reconstructed environment models support planning, visualization, and interaction. Under this paradigm, perception, modeling, and communication become tightly coupled, and reconstruction latency directly affects the quality of operator decisions.

Recent research has also explored the integration of teleoperation with varying levels of robotic autonomy. Shared-control frameworks allow human operators to provide high-level guidance while autonomous modules handle low-level motion planning or collision avoidance. For example, autonomy-assisted teleoperation systems have been widely studied in space robotics and surgical robotics, where maintaining safety under communication delay is critical. These

systems illustrate a continuum between fully manual teleoperation and fully autonomous control, in which perception quality and communication reliability directly influence the degree of autonomy that can be safely deployed.

In haptic communication and teleoperation systems, stringent Ultra Reliable Low Latency Communications (URLLC) constraints further highlight the tight coupling between communication and control. She et al. [3] provided a comprehensive tutorial on URLLC in 6G systems, showing that reliability and latency guarantees must be co-designed with application-level requirements such as control stability and real-time interaction. The tutorial emphasizes cross-layer design principles, where domain knowledge from robotics and control theory is incorporated into communication system design, rather than optimizing physical-layer metrics in isolation. Such perspectives reinforce the view that communication performance should be evaluated in terms of its impact on closed-loop system behavior, rather than solely by throughput or bit error rate.

Overall, research on teleoperation and networked control demonstrates that communication delay, perception freshness, and control stability are deeply intertwined. As robotic platforms increasingly rely on continuous 3D scene representations to support remote monitoring and interaction, timeliness becomes not only a network-level metric but also a control-relevant variable. Understanding how perception timeliness affects system-level decision making is therefore essential for future robotic perception systems. This observation motivates the timeliness-aware and task-oriented communication framework developed in the subsequent chapters of this thesis.

## 2.5 Task-Oriented Communication Paradigms

Classical communication systems are fundamentally fidelity-oriented. Under the Shannon information theoretic framework [1], a transmitter is designed to deliver a bitstream such that the receiver can reconstruct the original source with minimal distortion under capacity and rate constraints. This paradigm has proven extremely successful for human-oriented multimedia services such as voice, video, and image delivery, where faithful source reconstruction is directly aligned with user experience. However, this objective becomes increasingly mismatched with modern robotic perception and edge-intelligence pipelines, where the downstream task—for example control, inspection, object detection, segmentation, or 3D scene representation—is the true consumer of information. In such systems, faithfully reproducing all source details is neither necessary nor always beneficial. Instead, it can waste bandwidth and latency budget on content that is irrelevant to the downstream decision-making process.

To address this mismatch, recent research has explored task-oriented communication, in which communication system design explicitly incorporates the objective of the downstream task. In contrast to classical source–channel separation, task-oriented approaches typically involve joint optimization across sensing, communication, and inference modules. Early theoret-

ical perspectives on this idea can be traced to semantic information theory [35], which argued that communication systems should consider not only symbol transmission but also meaning and utility. More recently, task-oriented communication has been formalized within modern machine learning pipelines, where communication encoders are optimized to preserve features that are most relevant for the inference task. A key difficulty is that strict task-oriented methods often imply cross-system co-design: the communication encoder and decoder must be trained jointly with the inference or control model. While this can significantly improve task performance, it reduces system modularity and complicates deployment in large-scale communication infrastructures. Meng et al. [2] contributed to positioning task-oriented methodology as a design principle for next-generation networks and embodied digital systems. Their task-oriented metaverse framework highlights the need to treat sensing, communication, and computing as a co-designed pipeline whose objective is task completion rather than faithful media transport. Similarly, Diao et al. [4] proposed a framework that explicitly addresses the tension between task-oriented and reconstruction-oriented communication. Their approach introduces an Information-Bottleneck-inspired formulation together with an information reshaper that preserves compatibility with conventional reconstruction pipelines while prioritising task-relevant features. In addition, a Joint Source-Channel Coding (JSCC) modulation mechanism is employed to maintain compatibility with classical modulation structures, thereby enabling gradual integration with existing communication systems.

Task-oriented communication has also been studied extensively in wireless edge intelligence scenarios. In these systems, sensory data collected by edge devices are transmitted to edge servers for machine learning inference. Several works have demonstrated that transmitting compressed feature representations rather than raw data can dramatically reduce communication load while maintaining inference accuracy. For instance, deep feature compression techniques allow edge devices to transmit intermediate neural features instead of full images, enabling more efficient vision-based analytics in bandwidth-limited environments. These approaches illustrate that communication efficiency can be significantly improved when the transmitted information is directly aligned with the requirements of the downstream task.

In haptic and teleoperation communication systems, the end objective is neither waveform reproduction nor raw packet delivery, but stable and reliable human-in-the-loop control under stringent URLLC constraints. Kizilkaya et al. [5] proposed a task-oriented prediction communication co-design framework in which wireless resource allocation is optimized subject to task-dependent reliability constraints. Such formulations illustrate a recurring theme: task-oriented design makes the performance metric application-native, turning communication quality into an operational requirement tied directly to the human–robot interaction loop.

Semantic communication is often discussed as a complementary direction to task-oriented communication. Instead of guaranteeing symbol-level correctness, semantic communication aims to deliver the meaning required by the receiver for a specific task. Recent research has

attempted to formalize semantic information beyond Shannon's bit-level abstraction. Qin et al. [6] proposed a generalized semantic communication framework that explicitly considers both source semantics and channel/environment semantics, positioning semantic processing as an extension beyond classical information theory.

In vision-centric settings, semantic communication has been instantiated through learned semantic coding pipelines. Xie et al. [7] studied deep-learning-based image semantic coding, where the design objective is semantic exchange rather than strict pixel-level reconstruction. Their results demonstrate that transmitting semantic representations can significantly reduce transmission overhead while maintaining task performance. Subsequent work in this area has further explored semantic feature extraction and representation learning for communication-efficient visual perception systems, highlighting the close connection between communication efficiency and machine learning representation quality.

Taken together, these developments indicate a broader paradigm shift in communication system design: from source reconstruction toward task-aware information delivery. Whether framed through cross-system co-design, URLLC-constrained control, semantic feature transmission, or edge intelligence architectures, the unifying principle is that communication resources should be allocated according to their impact on downstream objectives.

For robotic perception systems that rely on continuous 3D scene representations, this perspective has important implications. In such systems, communication decisions directly influence how quickly and accurately the scene representation can be updated. Consequently, timeliness, task relevance, and reconstruction fidelity must be jointly considered rather than optimized in isolation. This perspective motivates the task-oriented and timeliness-aware communication framework developed in the subsequent chapters of this thesis.

# Chapter 3

# Background

## 3.1 Formulation of 3D Scene Representation

Spatial intelligence refers to a system's ability to perceive and reason about the three-dimensional structure of its surrounding environment [38]. For embodied agents operating in the physical world, such as robots, this capability underlies a broad range of perception and decision-making tasks. In practice, effective interaction with the environment requires not only instantaneous sensory measurements, but also an internal representation that captures scene geometry and appearance in a consistent spatial frame. From this perspective, 3D scene representation constitutes a fundamental component of spatial intelligence. It provides a structured description of the environment that supports the integration of observations collected from different viewpoints and sensing modalities. Without an explicit or implicit 3D representation, sensory data remain fragmented and cannot be reliably used for tasks that depend on spatial reasoning, such as localisation, mapping, or motion planning. Accordingly, 3D scene reconstruction can be viewed as the process by which a system estimates and maintains such a representation directly from sensor data. Rather than being an isolated geometric problem, reconstruction serves as a foundational step in enabling spatial understanding for embodied intelligent systems.

Three-dimensional scene representations is a central problem in robotic perception systems. An agent observes the physical world through a collection of sensors that provide partial, noisy, and viewpoint-dependent measurements. The objective of 3D scene representation is to infer a representation of the environment that is geometrically consistent and suitable for further processing. Such a representation is commonly required by robotic autonomous downstream tasks such as localisation, mapping, navigation, and also by physical interaction, where decisions depend on an explicit understanding of spatial structure. Although different 3D scene representation methods vary substantially in sensing modality, algorithmic design, and computational complexity, they share a common underlying objective. A 3D scene representation can be viewed as an inference problem over a continuous spatial domain, where information from discrete observations is aggregated to estimate properties of the scene at arbitrary spatial

locations. We consider a static scene that can be characterized by an unknown latent function

$$\mathscr{F} : \mathbb{R}^3 \to \mathbb{R}^C, \tag{3.1}$$

where the input $\mathbf{d} = [x, y, z] \in \mathbb{R}^3$ denotes a location in 3D space, and the output encodes intrinsic properties of the scene at that location. The dimensionality and semantics of the output space are determined by the chosen 3D scene representation and reflect the aspects of the scene that the 3D scene representation method seeks to model, i.e. it may encode binary occupancy of space [39], the signed distance to the nearest surface [40], or volumetric density [41] and radiance as used in volume-based rendering approaches. More generally, $\mathscr{F}(\mathbf{d})$ may combine geometric information with appearance-related attributes such as color or reflectance [42]. This functional formulation does not prescribe how the scene is discretised or rendered, but instead provides a common description that encompasses both explicit and implicit reconstruction methods.

By modeling 3D scene representation as the estimation of a continuous or discrete scene function, differences among existing approaches can be interpreted primarily in terms of how $\mathscr{F}$ is parameterised and how observations constrain its values. This abstraction serves as the basis for the reconstruction models discussed in the following sections. This functional abstraction provides a unifying perspective on a wide range of reconstruction methods. Rather than explicitly modeling surfaces or volumes using discrete primitives, modern approaches seek to estimate a parametric approximation $\mathscr{F}_\theta$ of the latent scene function $\mathscr{F}$ from sensor observations. Under this view, differences among reconstruction paradigms arise primarily from how the function $\mathscr{F}_\theta$ is parameterized and how it is related to measurements through a rendering or projection process.

Let the perception system acquire observations from one or multiple sensors. The complete observation set is denoted as

$$\mathbf{O} = \{(\mathbf{o}_n, \xi_n)\}_{n=1}^{N}, \tag{3.2}$$

where $\mathbf{o}_n$ denotes the measurement from the $n$-th sensor, and $\xi_n$ represents the corresponding sensing model, including the intrinsic and extrinsic parameters of the coresponding sensor. The 3D scene representation task then amounts to estimating a parametric scene representation $\mathscr{F}_\theta$ that is consistent with the available observations $\mathbf{O}$. This can be expressed in the generic optimization form

$$\hat{\theta} = \arg\min_{\theta} \mathscr{L}(\mathscr{F}_\theta, \mathbf{O}), \tag{3.3}$$

where $\mathscr{L}(\cdot)$ denotes a reconstruction loss defined through a rendering or projection operator that relates the 3D scene representation to ground-truth sensor measurements.

This formulation intentionally abstracts away method-specific assumptions and implementation details, and serves as a common foundation for the reconstruction approaches discussed in the following sections.

### 3.1.1 Explicit 3D Scene Representations

Explicit 3D scene representations describe the environment using a finite set of discrete geometric primitives. Common choices include point clouds, polygonal meshes, and voxel grids, which have long been used in robotic mapping and 3D reconstruction pipelines due to their direct geometric interpretability [44–47].

#### Point clouds

A point cloud represents a scene as an unordered set of discrete 3D samples

$$\mathbf{P} = \{(\mathbf{p}_j, \mathbf{a}_j)\}_{j=1}^{M}, \quad \mathbf{p}_j \in \mathbb{R}^3, \tag{3.4}$$

where $\mathbf{p}_j$ denotes the spatial location and $\mathbf{a}_j$ optionally encodes attributes such as color, surface normal, or confidence. This representation is commonly produced by range sensing (e.g., RGB-D or LiDAR) and by multi-view geometry pipelines, and is therefore widely adopted as an intermediate output in classical 3D reconstruction systems [47, 48]. From an estimation viewpoint, $\mathbb{P}$ can be interpreted as a finite sampling of an underlying continuous surface or volumetric structure; consequently, the geometric fidelity is governed by both sampling density and measurement noise.

Local neighbourhood relationships, surface connectivity, or topology are not explicitly encoded in point clouds. As a result, subsequent processing typically requires the construction of a neighbourhood graph, for example via $k$-nearest neighbours [49] or radius-based search, in order to estimate local surface properties or support surface reconstruction:

$$\mathbf{N}(\mathbf{p}_j) = \mathrm{kNN}(\mathbf{p}_j; \mathbf{P}) \quad \text{or} \quad \mathbf{N}(\mathbf{p}_j) = \{\mathbf{p}_k \in \mathbf{P} : \|\mathbf{p}_k - \mathbf{p}_j\| \leq r\}. \tag{3.5}$$

This additional step introduces sensitivity to non-uniform sampling and outliers, and makes the representation less stable under sparse or incomplete observations. Therefore, while point clouds are simple, compact, and directly interpretable, they are often insufficient as a final representation when tasks require explicit surfaces, watertight geometry, or high-quality rendering [44].

#### Meshes

A polygonal mesh explicitly models scene surfaces by defining a set of vertices and their connectivity, typically written as

$$\mathbf{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F}), \tag{3.6}$$

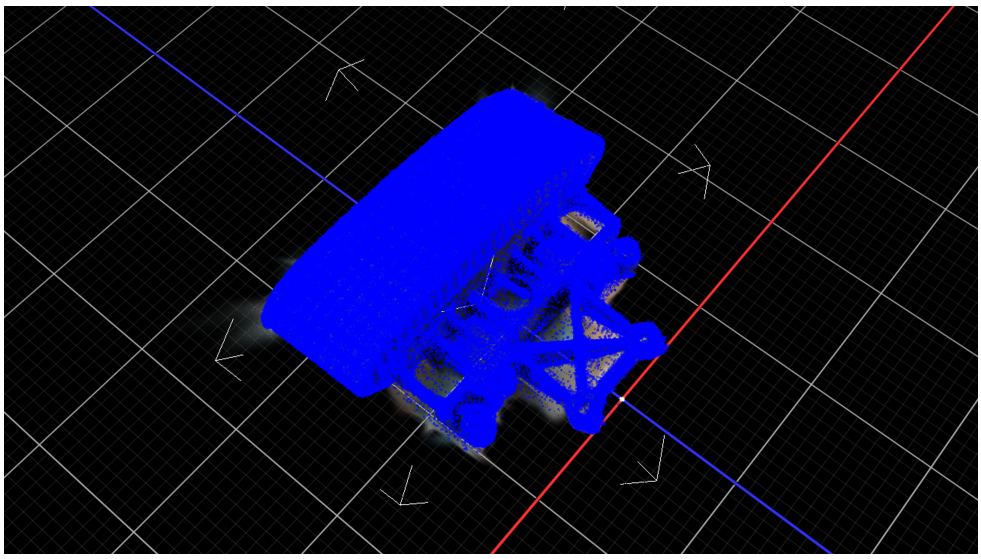


Figure 3.1: A Point Cloud (as blue dots) Scene Representation of the JET tile.

where

$$\begin{aligned}
&\mathbf{V} = \{\mathbf{v}_i \in \mathbb{R}^3\}_{i=1}^{N_v}, \\
&\mathbf{E} \subseteq \{(i,j) \mid i,j \in \{1,\ldots,N_v\}\}, \\
&\mathbf{F} \subseteq \{(i,j,k) \mid i,j,k \in \{1,\ldots,N_v\}\}.
\end{aligned}$$

where $\mathbf{V}$ denotes a set of vertices, $\mathbf{E}$ denotes a set of edges, and $\mathbf{F}$ denotes a set of faces. Compared to point-based representations, meshes provide an explicit encoding of surface topology and local neighbourhood structure, enabling surface-level reasoning, efficient rasterization, and direct compatibility with graphics pipelines.

Meshes are commonly obtained by reconstructing surfaces from discrete geometric samples, such as point clouds or volumetric fields. Given a sufficiently dense set of oriented points, classical surface reconstruction methods seek to estimate a continuous indicator or implicit function whose zero level set defines the surface. A representative example is Poisson surface reconstruction, which formulates surface recovery as the solution to a Poisson equation defined over space and extracts the resulting surface as an isosurface [44]. Alternatively, when a volumetric representation is available, mesh surfaces can be extracted from a discretised scalar field using isosurface extraction algorithms such as Marching Cubes [45].

Despite their expressive power, mesh-based representations exhibit several limitations in practical robotic reconstruction systems. Constructing watertight and topologically consistent meshes from noisy or sparsely sampled data is computationally demanding and often requires global optimisation or iterative refinement. As a consequence, mesh reconstruction pipelines are typically executed offline or at relatively low update rates, which limits their applicability in time-sensitive or real-time perception settings. Moreover, mesh connectivity introduces dis-

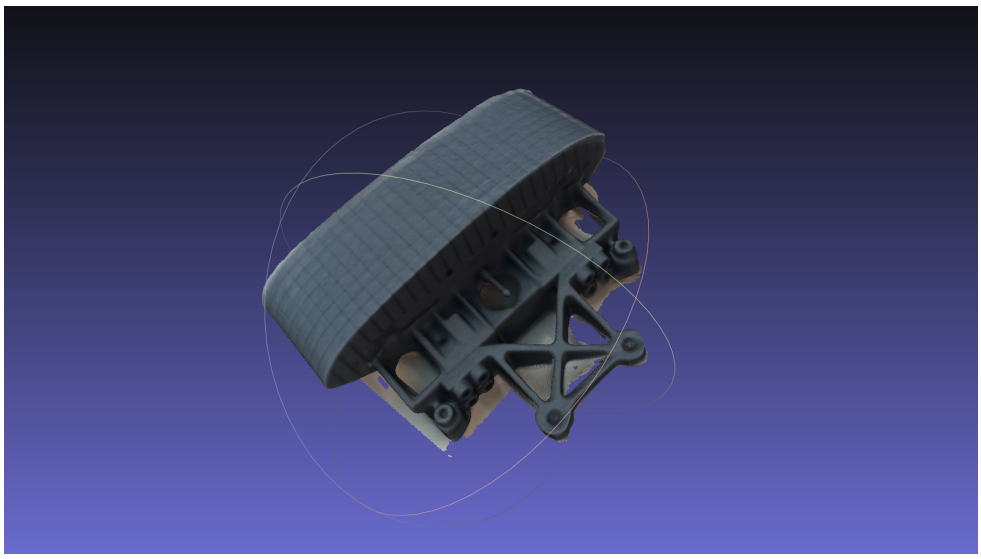


Figure 3.2: A Mesh Scene Representation of the JET tile.

crete combinatorial structures that are difficult to optimise jointly with continuous parameters using gradient-based methods. Even incremental mesh update strategies require careful handling of topology changes, making stable and efficient online optimisation challenging. As a result, meshes are more commonly used as a final output or evaluation format, rather than as an intermediate representation during continuous optimisation.

These limitations motivate alternative scene representations that retain geometric fidelity while supporting continuous querying, differentiable rendering, and efficient updates, as discussed in the following sections.

### Voxel grids and volumetric fusion

Voxel-based representations discretise 3D space into a regular grid and associate a value with each spatial cell. Formally, the scene volume is partitioned into a lattice of voxels $\mathbf{v}$ indexed by $(i, j, k) \in \{1, \ldots, N_x\} \times \{1, \ldots, N_y\} \times \{1, \ldots, N_z\}$, where geometric information is stored at each grid cell. Depending on the quantity stored in each voxel, such representations can exhibit either explicit or implicit geometric properties, effectively forming a bridge between the two paradigms.

The simplest formulation is a binary occupancy grid,

$$\mathbf{V}_{\text{occ}} : \{1, \ldots, N_x\} \times \{1, \ldots, N_y\} \times \{1, \ldots, N_z\} \to \{0, 1\}, \tag{3.7}$$

where each voxel encodes whether the corresponding spatial region is occupied by the scene geometry. In this case, the representation is explicit: the object volume is directly stored as a collection of occupied cells, and the surface is defined by the boundary between occupied and empty voxels.

More expressive volumetric representations store continuous geometric quantities within each voxel. A widely adopted example is the TSDF [39, 46], which stores a discretised approximation of a signed distance field on a voxel grid. Let $\mathbf{v}$ denote the spatial location of a voxel center. The TSDF value is defined as

$$D(\mathbf{v}) = \text{clip}(d(\mathbf{v}, \partial\mathscr{S}), [-d_{\max}, d_{\max}]), \tag{3.8}$$

where $d(\mathbf{v}, \partial\mathscr{S})$ is the signed distance to the closest surface $\partial\mathscr{S}$ and $d_{\max} > 0$ is a truncation threshold. Here, $\mathscr{S} \subset \mathbb{R}^3$ denotes the solid region occupied by the scene, and $\partial\mathscr{S}$ its boundary surface [50]. By convention, the distance is negative inside the surface and positive outside. Distances outside the truncation band are clamped, improving robustness to noise and outliers.

Unlike occupancy grids, TSDFs define geometry implicitly: the surface is not stored directly but is instead recovered as the zero level set of the distance field. However, the field itself is stored explicitly on a discrete voxel lattice. As a result, TSDF-based representations combine characteristics of both explicit and implicit models, forming a hybrid volumetric formulation.

Volumetric fusion integrates multiple depth observations by incrementally updating voxel values, following the weighted averaging scheme in [39]. Given a new observation that induces a distance estimate $D_{\text{new}}(\mathbf{v})$ with weight $w_{\text{new}}$, the TSDF value is updated via weighted averaging:

$$D^{(t+1)}(\mathbf{v}) = \frac{w^{(t)}(\mathbf{v})\, D^{(t)}(\mathbf{v}) + w_{\text{new}}\, D_{\text{new}}(\mathbf{v})}{w^{(t)}(\mathbf{v}) + w_{\text{new}}}, \tag{3.9}$$

with the corresponding weight updated as

$$w^{(t+1)}(\mathbf{v}) = w^{(t)}(\mathbf{v}) + w_{\text{new}}. \tag{3.10}$$

This explicit accumulation process enables measurements from multiple viewpoints to be fused into a single, consistent volumetric representation and provides inherent noise smoothing through spatial aggregation.

Once a volumetric field has been constructed, an explicit surface representation is typically extracted by computing an isosurface of the TSDF, commonly using the Marching Cubes algorithm [45]. This two-stage pipeline—volumetric fusion followed by surface extraction—has been widely adopted in dense mapping and real-time reconstruction systems.

Despite their effectiveness, voxel-based representations exhibit fundamental scalability limitations. Both memory usage and computational cost scale cubically with spatial resolution, which makes high-resolution modeling expensive over large environments. Although sparse data structures and spatial hashing schemes have been proposed to alleviate this issue, voxel grids remain constrained by resolution–extent trade-offs and update latency. As a result, volumetric fusion methods are typically restricted to bounded workspaces or coarse resolution when real-time performance is required, motivating the development of more scalable implicit scene

representations.

### 3.1.2 Implicit 3D Scene Representations

Explicit geometric representations, as mentioned above, are established in earlier 3D scene representation studies. These representations describe a scene using discrete primitives and are often straightforward to interpret. However, they are inherently resolution-dependent, sensitive to sampling density, and difficult to optimize in an end-to-end manner. Moreover, explicit representations are typically not defined over continuous space, which limits their suitability for high-quality rendering and for reasoning at arbitrary spatial locations.

In contrast, modern reconstruction methods increasingly adopt implicit scene representations, where geometry and appearance are encoded as continuous functions over space. Under this paradigm, the scene is represented by a function

$$\mathscr{F}_\theta(\mathbf{x}) \approx \mathscr{F}(\mathbf{x}), \tag{3.11}$$

which approaximately maps a spatial location $\mathbf{x} \in \mathbb{R}^3$ to a set of scene properties. This formulation decouples the representation from any fixed discretisation and allows the scene to be queried at arbitrary resolution.

The continuity of implicit 3d scene representation in space enables smooth surface modeling and avoids artifacts caused by discrete sampling. And parameterising $\mathscr{F}_\theta$ with differentiable function approximators, such as neural networks or 3D/2D gaussians, allows the representation to be optimised directly from image-based supervision using gradient-based methods. Finally, implicit formulations naturally integrate with differentiable rendering operators, making them well-suited for inverse graphics and multi-view reconstruction.

A number of representative reconstruction paradigms can be interpreted within this implicit framework. Signed Distance Function (SDF) based methods model geometry by regressing the distance to the nearest surface at each spatial location, enabling accurate surface reconstruction [39, 40]. Occupancy-based approaches represent the scene using continuous occupancy probabilities, providing a compact description of shape without explicitly modeling surfaces [43]. More recently, volumetric radiance field methods encode both geometry and appearance within a single implicit function, enabling high-fidelity view synthesis from sparse multi-view observations [22].

Despite differences in parameterization and supervision, these methods share a common abstraction: the scene is represented implicitly by a continuous function, and reconstruction amounts to estimating this function such that its rendered observations are consistent with sensor measurements. Consequently, different reconstruction paradigms can be characterized primarily by the semantic meaning assigned to the implicit function $\mathscr{F}_\theta$ and by the rendering or projection operator used to relate it to image-space supervision.

### Signed Distance Functions

A widely used implicit geometric representation is the SDF [39, 40], which represents scene geometry as a continuous scalar field defined over three-dimensional space. Given a solid region $\mathscr{S} \subset \mathbb{R}^3$, the SDF is defined as

$$\mathscr{F}_{\text{SDF}}(\mathbf{x}) = \begin{cases} +d(\mathbf{x}, \partial\mathscr{S}), & \mathbf{x} \notin \mathscr{S}, \\ 0, & \mathbf{x} \in \partial\mathscr{S}, \\ -d(\mathbf{x}, \partial\mathscr{S}), & \mathbf{x} \in \mathscr{S}, \end{cases} \tag{3.12}$$

where $d(\mathbf{x}, \partial\mathscr{S})$ denotes the Euclidean (L2) distance from the query point $\mathbf{x}$ to the boundary surface $\partial\mathscr{S}$. Under this formulation, the scene surface is implicitly represented as the zero level set of $\{\mathscr{F}_{\text{SDF}} = 0\}$, and no explicit surface connectivity or discretisation is required.

An ideal SDF has to satisfy the Eikonal constraint [50]

$$\|\nabla \mathscr{F}_{\text{SDF}}(\mathbf{x})\| = 1, \tag{3.13}$$

almost everywhere, as the gradient magnitude of a true distance function is unity. This property provides a strong geometric prior that regularises the learned field and promotes geometrically consistent surface reconstructions. In practice, deviations from this condition may arise due to noise, incomplete observations, or approximate parameterisations, and the Eikonal term is therefore commonly imposed as a soft regularisation during optimisation.

To enable learning-based reconstruction from image observations, SDFs are commonly parameterised as implicit functions, for example, using neural networks $\mathscr{F}_\theta(\mathbf{x})$, where $\theta$ denotes learnable parameters. This representation allows the SDF to be queried at arbitrary spatial locations and decouples the geometric resolution from any fixed spatial discretisation, in contrast to voxel-based distance fields. For image-based supervision, SDF-based models are typically extended to jointly encode appearance information:

$$\mathscr{F}_\theta(\mathbf{x}) = \big(d_\theta(\mathbf{x}), \mathbf{c}_\theta(\mathbf{x})\big), \tag{3.14}$$

where $d_\theta(\mathbf{x})$ represents the signed distance and $\mathbf{c}_\theta(\mathbf{x})$ encodes color or reflectance attributes. Rendering is then performed by explicitly computing ray–surface intersections, often via sphere tracing, and evaluating photometric consistency between rendered and observed pixels [51]. This surface-based rendering process enables accurate geometry recovery and sharp surface reconstruction compared to volumetric integration methods.

In practice, SDFs are learned from discrete observations by enforcing consistency between the implicit distance field and the available measurements. When point cloud or depth observations are available, supervision can be applied directly at sampled surface locations by constraining the signed distance to be zero. Given a set of surface samples $\{\mathbf{o}_i\}_{i=1}^N$, a commonly

used objective takes the form

$$\mathscr{L}_{\text{SDF}} = \sum_{i=1}^{N} \left| \mathscr{F}_\theta(\mathbf{o}_i) \right| + \lambda \int_\Omega \left( \|\nabla \mathscr{F}_\theta(\mathbf{o})\| - 1 \right)^2 d\mathbf{o}, \tag{3.15}$$

where the first term enforces zero level-set consistency at observed surface points and the second term regularises the field to satisfy the Eikonal constraint over the spatial domain $\Omega$ [40]. Here, $\Omega \subset \mathbb{R}^3$ denotes a bounded spatial domain over which the implicit field is defined and regularised, typically chosen as a region enclosing the observed scene geometry.

When only image observations are available, supervision becomes indirect. Rather than observing signed distances explicitly, the SDF must be trained by enforcing consistency between rendered images and observed pixels. This is typically achieved using surface-based differentiable rendering, in which the implicit surface is intersected with camera rays, for example via sphere tracing, and photometric discrepancies are minimised in the image plane [51]. While this formulation enables SDF-based reconstruction from images alone, it relies on accurate ray–surface intersection and tightly couples geometric estimation with rendering. As a result, optimisation is computationally demanding and sensitive to geometric inaccuracies, which limits the robustness and scalability of purely SDF-based approaches under image-only supervision.

### Neural Radiance Fields

While signed distance functions implicitly encode scene geometry via a zero-level surface, they do not directly model appearance or view-dependent effects under sparse image supervision. Neural Radiance Fields (NeRF) methods address this limitation by formulating scene reconstruction as a volumetric rendering problem defined along camera rays [22].

Let the $t$-th RGB image be a discrete sampling of radiance on a pixel grid,

$$I_t : \{1,\dots,W\} \times \{1,\dots,H\} \to [0,1]^3, \qquad I_t[u,v] \in \mathbb{R}^3, \tag{3.16}$$

where $(u,v)$ denotes pixel coordinates and $I_t[u,v]$ is the observed RGB color. For notational convenience, we may also flatten the pixel index and write $I_{i,t} := I_t[u_i, v_i]$. Each pixel $(u,v)$ uniquely induces a camera ray $\mathbf{r}_t(u,v)$ through the calibrated camera model, establishing the correspondence

$$(u,v) \;\longleftrightarrow\; \mathbf{r}_t(u,v) \;\longleftrightarrow\; I_t[u,v].$$

Consider a pinhole camera with intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ and extrinsic parameters $(\mathbf{R},\mathbf{t}) \in \mathrm{SE}(3)$ mapping world coordinates to the camera frame. For a pixel location $\mathbf{p} = (u,v,1)^\top$ in homogeneous image coordinates, the corresponding camera ray in world space is given by

$$\mathbf{r}(s) = \mathbf{r}_o + s\,\mathbf{d}, \qquad s \in [s_n, s_f], \tag{3.17}$$

where the ray origin is $\mathbf{r}_o = -\mathbf{R}^\top \mathbf{t}$ and the ray direction is

$$\mathbf{d} = \frac{\mathbf{R}^\top \mathbf{K}^{-1} \mathbf{p}}{\left\| \mathbf{R}^\top \mathbf{K}^{-1} \mathbf{p} \right\|}. \tag{3.18}$$

This explicit construction establishes a one-to-one correspondence between each image pixel and a continuous set of 3D query points sampled along its associated viewing ray.

NeRF represents the scene as a continuous volumetric field parameterised by a neural network

$$\mathscr{F}_\theta(\mathbf{x}, \mathbf{d}) = \big(\sigma_\theta(\mathbf{x}), \mathbf{c}_\theta(\mathbf{x}, \mathbf{d})\big), \tag{3.19}$$

where $\mathbf{x} \in \mathbb{R}^3$ denotes spatial location, $\mathbf{d} \in \mathbb{S}^2$ denotes viewing direction, $\sigma_\theta(\mathbf{x}) \geq 0$ is the volume density, and $\mathbf{c}_\theta(\mathbf{x}, \mathbf{d}) \in \mathbb{R}^3$ is the emitted radiance. Unlike SDF-based representations, geometry is not encoded via an explicit surface, but rather emerges implicitly through regions of high accumulated density along rays. While the functional form of $\mathscr{F}_\theta$ differs across representations, it consistently parameterises a continuous implicit field queried at arbitrary spatial locations, with the output semantics determined by the chosen scene model.

Given a ray $\mathbf{r}(s)$, the pixel color is computed via the volume rendering integral

$$\mathbf{C}(\mathbf{r}) = \int_{s_n}^{s_f} T(s)\, \sigma_\theta(\mathbf{r}(s))\, \mathbf{c}_\theta(\mathbf{r}(s), \mathbf{d})\, ds, \tag{3.20}$$

where the transmittance function

$$T(s) = \exp\left( - \int_{s_n}^{s} \sigma_\theta(\mathbf{r}(u))\, du \right) \tag{3.21}$$

models occlusion and visibility along the ray. In practice, the integral is approximated using stratified or hierarchical sampling, resulting in a fully differentiable image formation process.

Let $I_t[u, v] \in \mathbb{R}^3$ denote the observed RGB color at pixel $(u, v)$ in image $t$, and let $I_{i,t}$ be its flattened representation. Each pixel induces a corresponding camera ray $\mathbf{r}_{i,t}$. NeRF is trained by minimizing the photometric reconstruction loss

$$\min_\theta \sum_{t,i} \| \mathbf{C}_\theta(\mathbf{r}_{i,t}) - I_{i,t} \|_2^2, \tag{3.22}$$

which couples multi-view geometry, appearance, and camera calibration through the shared volumetric field.

A major computational bottleneck of NeRF arises from the need to evaluate $\mathscr{F}_\theta$ at hundreds of sample points per ray. Instant Neural Graphics Primitives (Instant-NGP) [52] alleviates this issue by replacing positional encodings with a multi-resolution hash grid encoding [52]. Specif-

ically, a 3D location $\mathbf{x}$ is mapped to a set of feature vectors

$$\phi(\mathbf{x}) = \left[\phi^{(1)}(\mathbf{x}), \ldots, \phi^{(L)}(\mathbf{x})\right], \tag{3.23}$$

where each level $\ell$ corresponds to a voxel grid of increasing resolution, stored in a compact hash table and accessed via trilinear interpolation. This encoding enables logarithmic memory growth with respect to resolution, while preserving the spatial locality required for high-frequency detail reconstruction. As a result, Instant-NGP achieves orders-of-magnitude speedups in training and inference, making NeRF-style volumetric representations viable for interactive and robotic settings.

From a representational perspective, NeRF generalises implicit surface models by lifting geometry from a zero-level set to a volumetric density distribution. This shift enables faithful view synthesis under sparse supervision, but introduces substantial computational overhead due to dense ray sampling and volumetric integration. Subsequent hybrid approaches aim to recover explicit or surface-aligned structures from NeRF-like fields, bridging volumetric and surface-based implicit representations.

Beyond the original NeRF formulation, a number of neural scene representation frameworks have been proposed to improve reconstruction fidelity, rendering efficiency, scalability, and optimisation speed. These methods form the reconstruction backbone used throughout the later chapters of this thesis. Instant-NGP [52] improves the training efficiency of NeRF through the use of multi-resolution hash-grid encoding and lightweight neural networks. Compared with the original NeRF formulation, Instant-NGP significantly accelerates scene optimisation and rendering, enabling near-real-time reconstruction updates that are particularly suitable for communication-aware robotic perception systems. TensorRF [56] further improves neural scene representation efficiency through tensor decomposition techniques, where the volumetric scene representation is factorised into compact low-rank tensor components. Compared with conventional NeRF approaches, TensorRF substantially reduces memory consumption and optimisation complexity while maintaining competitive rendering quality. These characteristics make TensorRF particularly suitable for large-scale scene representation and resource-constrained reconstruction settings. Nerfacto [57] extends NeRF-based scene representation through improved ray sampling, proposal networks, scene contraction, and training stabilisation strategies. These improvements enable more robust reconstruction performance across unbounded and large-scale environments while maintaining relatively high rendering fidelity. Nerfacto is adopted in later experiments as a representative modern NeRF-style reconstruction framework. These reconstruction frameworks are revisited throughout the later chapters of this thesis under different timeliness-aware sensing and communication settings. Rather than proposing a new reconstruction backbone itself, this thesis primarily investigates how communication dynamics, observation freshness, and scheduling behaviour interact with different neural scene representation paradigms in distributed robotic perception systems.

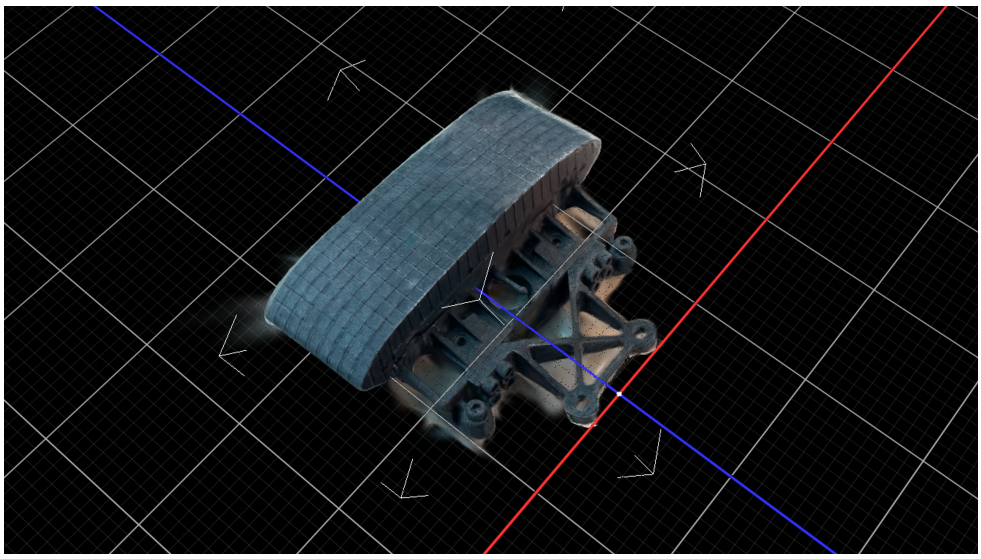


Figure 3.3: A 3D Gaussian Scene Representation of the JET tile.

### 3D Gaussian Splatting

While NeRF parameterises a continuous volumetric radiance field queried along rays, 3D Gaussian Splatting (3DGS) [54] represents the scene using a finite set of continuous, explicit primitives optimised directly from multi-view image supervision. This representation can be expressed under a unified functional abstraction

$$\mathscr{F}_\theta : \mathbb{R}^3 \longrightarrow \mathscr{P}, \tag{3.24}$$

where $\mathscr{P}$ denotes the parameter space of Gaussian primitives and $\theta$ collects all learnable parameters. In 3DGS, $\mathscr{F}_\theta$ induces a finite set of $N$ primitives

$$\mathscr{F}_\theta \equiv \{\mathbf{G}_k\}_{k=1}^{N}, \qquad \mathbf{G}_k := \left(\mu_k, \Sigma_k, \alpha_k, \mathbf{a}_k\right) \in \mathscr{P}, \tag{3.25}$$

where $\mu_k \in \mathbb{R}^3$ is the 3D mean, $\Sigma_k \in \mathbb{R}^{3\times 3}$ is a symmetric positive definite covariance, $\alpha_k \in [0,1]$ is an opacity coefficient, and $\mathbf{a}_k$ parameterises appearance (potentially view-dependent).

Each primitive defines a continuous spatial kernel

$$g_k(\mathbf{x}) = \exp\left(-\tfrac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right), \qquad \mathbf{x} \in \mathbb{R}^3, \tag{3.26}$$

which assigns a smoothly decaying influence to any spatial query point $\mathbf{x}$ based on its displacement from the Gaussian centre $\mu_k$. The quadratic form $(\mathbf{x}-\mu_k)^\top \Sigma_k^{-1}(\mathbf{x}-\mu_k)$ defines a Mahalanobis distance under the metric induced by $\Sigma_k^{-1}$, thereby generalising the isotropic Euclidean distance to an orientation-aware, anisotropic measure.

For any constant value of the exponent, the corresponding level set

$$(\mathbf{x}-\mu_k)^\top \Sigma_k^{-1}(\mathbf{x}-\mu_k) = c$$

forms an ellipsoidal surface in $\mathbb{R}^3$. The principal axes of this ellipsoid are given by the eigenvectors of $\Sigma_k$, while the eigenvalues determine the squared lengths of the semi-axes. As a result, $\Sigma_k$ jointly encodes both the spatial extent and the orientation of the Gaussian primitive, enabling elongated, surface-aligned support regions that are well suited for approximating locally planar geometry.

To ensure that the covariance matrix remains symmetric positive definite ($\Sigma_k \succ \mathbf{0}$) throughout optimisation, it is commonly parameterised as

$$\Sigma_k = \mathbf{R}_k \,\mathrm{diag}\big(\mathbf{s}_k^2\big)\, \mathbf{R}_k^\top, \tag{3.27}$$

where $\mathbf{s}_k \in \mathbb{R}^3_+$ specifies the per-axis scales of the ellipsoid and $\mathbf{R}_k \in \mathrm{SO}(3)$ is a rotation matrix, often represented internally via a unit quaternion. This decomposition explicitly separates scale and orientation, guarantees positive definiteness by construction, and avoids the numerical instabilities associated with directly optimising the entries of $\Sigma_k$. In practice, the learned Gaussians tend to become highly anisotropic, with two large tangential axes and a short normal axis, leading to an emergent alignment with underlying scene surfaces.

To model specularities and other view-dependent effects, 3DGS often represents the emitted radiance as a low-order spherical harmonics (SH) expansion of the viewing direction $\mathbf{d} \in \mathbb{S}^2$:

$$\mathbf{c}_k(\mathbf{d}) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \mathbf{a}_{k,\ell m} Y_{\ell m}(\mathbf{d}), \qquad \mathbf{a}_{k,\ell m} \in \mathbb{R}^3, \tag{3.28}$$

where $Y_{\ell m}$ are SH basis functions and $\{\mathbf{a}_{k,\ell m}\}$ are learnable RGB coefficients (collectively denoted by $\mathbf{a}_k$). The special case $L = 0$ reduces to view-independent colour. In practice, $\alpha_k$ and $\mathbf{c}_k(\mathbf{d})$ are composed in a differentiable rasterisation pipeline after projecting the 3D Gaussian into an elliptical 2D footprint in screen space.

Unlike NeRF and SDF models where $\mathscr{F}_\theta(\mathbf{x})$ is evaluated as a continuous scalar/vector field at arbitrary locations, 3DGS uses $\mathscr{F}_\theta$ to parameterise a *set-valued* representation: a discrete collection of continuous primitives whose union approximates the scene. Geometry is therefore encoded by the spatial distribution and anisotropy of $\{g_k\}$, rather than by a zero-level surface or a volumetric density integrated along rays.

Given a calibrated pinhole camera with intrinsic matrix $\mathbf{K}$ and extrinsic parameters $(\mathbf{R}, \mathbf{t})$, the image formation process maps $\mathscr{F}_\theta$ to the image plane via differentiable Gaussian splatting. Each 3D Gaussian $\mathscr{G}_k$ is projected into a 2D elliptical Gaussian footprint through first-order approximation of the perspective projection, yielding a screen-space mean and covariance.

For a pixel $(u, v)$, the rendered colour is obtained by alpha compositing the contributions of

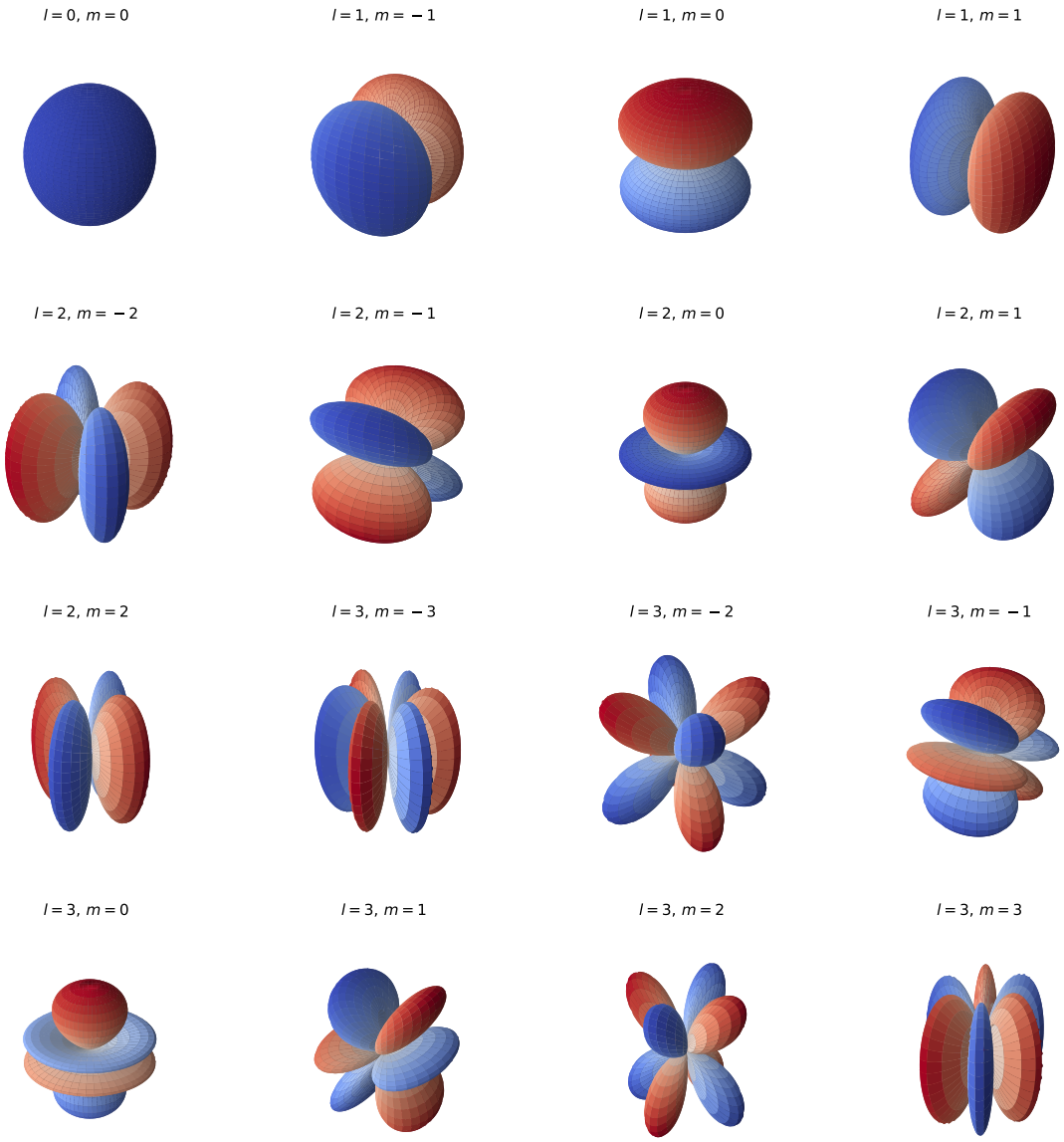



Figure 3.4: Real spherical harmonics visualised.

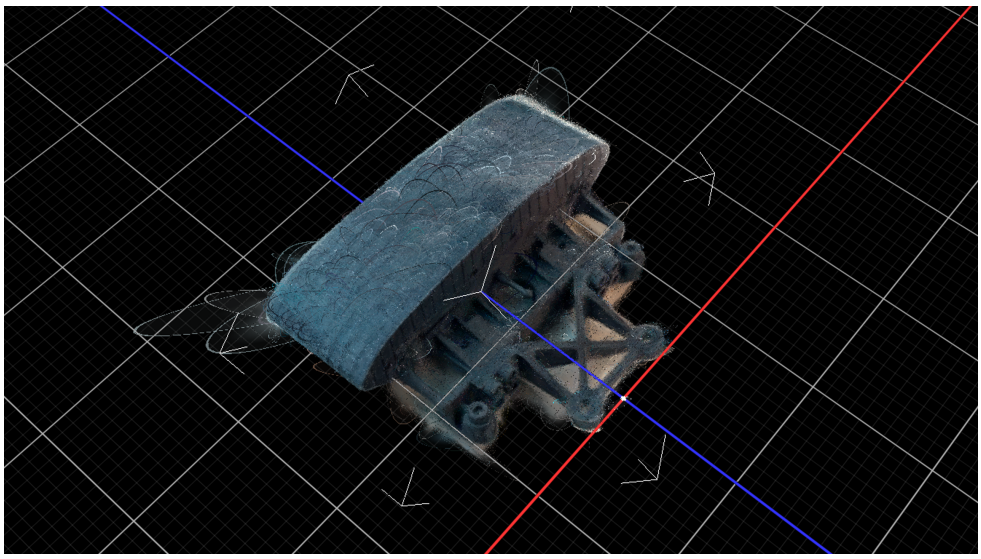


Figure 3.5: A 3D Gaussian Scene Representation of the JET tile, with Gaussians Visualized.

all Gaussians whose projected support overlaps the pixel,

$$\mathbf{C}(u,v) = \sum_k T_k(u,v)\,\alpha_k\,\mathbf{c}_k, \tag{3.29}$$

where $T_k(u,v)$ denotes the accumulated transmittance computed via depth-ordered front-to-back compositing. This forward rasterisation process is fully differentiable with respect to $\theta$.

The parameters $\theta$ of $\mathscr{F}_\theta$ are optimised by minimising a photometric reconstruction loss over all views and pixels,

$$\min_\theta \sum_{t,i} \|\mathbf{C}_\theta(\mathbf{r}_{i,t}) - I_{i,t}\|_2^2, \tag{3.30}$$

where $\mathbf{C}_\theta(\mathbf{r}_{i,t})$ denotes the colour obtained by splatting the Gaussian set $\mathscr{F}_\theta$ under camera $t$.

From a representational perspective, 3DGS occupies an intermediate position between implicit volumetric fields and explicit surface models. Although no explicit surface is defined, the learned Gaussian centres tend to concentrate around scene boundaries, forming an emergent surface-aligned structure. This behaviour is illustrated in Fig. 3.3, where the reconstructed synthetic JET tile exhibits high-fidelity geometry, while the corresponding Gaussian visualisation reveals a sparse distribution of anisotropic primitives tightly clustered around object surfaces.

By replacing volumetric integration with explicit primitive-based rasterisation, 3DGS achieves real-time rendering and efficient optimisation, making it well suited for interactive reconstruction and embodied robotic systems. However, the absence of explicit topology motivates subsequent hybrid approaches that recover surface representations, such as meshes or signed distance functions, from the learned Gaussian parameters.

## 3.2 Image Similarity

### 3.2.1 Peak Signal-to-Noise Ratio (PSNR)

We first adopt the peak signal-to-noise ratio (PSNR) to measure pixel-wise reconstruction accuracy. PSNR similarity is computed between the ground-truth image $I$ and the synthesized image $\hat{I}$ rendered from the reconstructed scene:

$$\mathrm{PSNR}\big(I,\hat{I}\big) = 10\log_{10}\left(\frac{R_I^2}{\mathrm{MSE}\big(I,\hat{I}\big)}\right), \tag{3.31}$$

where $R_I = 2^k - 1$ denotes the maximum possible pixel value for a $k$-bit image, and the Mean Squared Error (MSE) is defined as

$$\mathrm{MSE}\big(I,\hat{I}\big) = \frac{1}{L_H L_W}\sum_{m=1}^{L_H}\sum_{n=1}^{L_W}\big(i_{m,n} - \hat{i}_{m,n}\big)^2. \tag{3.32}$$

Here, $L_H$ and $L_W$ denote the image height and width, respectively, and $i_{m,n}$ and $\hat{i}_{m,n}$ represent the pixel intensities at location $(m,n)$ in $I$ and $\hat{I}$. The logarithmic form compresses a wide range of MSE values into a convenient scale and aligns the metric with the decibel (dB) representation commonly used in signal processing. A higher PSNR indicates smaller pixel-wise reconstruction error, but it does not explicitly capture perceptual or structural artefacts; therefore, PSNR is reported together with SSIM and LPIPS.

PSNR provides a direct and interpretable measure of reconstruction fidelity, making it particularly suitable for evaluating the convergence behaviour and numerical stability of neural scene representation methods. In the context of novel view synthesis, a higher PSNR indicates that the reconstructed 3D scene representation can more accurately reproduce unseen views under the assumed imaging model. Nevertheless, PSNR evaluates 3D scene representation quality purely based on per-pixel squared error, and therefore does not explicitly account for human perceptual sensitivity or structural consistency. In particular, spatially localised geometric artefacts, such as blurred edges or ghosting effects caused by temporally misaligned observations, may lead to limited changes in PSNR despite being visually salient. For this reason, PSNR is used in conjunction with perceptual and structure-aware metrics in the subsequent evaluation.

### 3.2.2 Structural Similarity Index (SSIM)

To evaluate perceptual quality beyond pixel-wise differences, we employ the structural similarity index (SSIM), which measures similarity in terms of luminance, contrast, and structure. The

SSIM between $I$ and $\hat{I}$ is defined as

$$\mathrm{SSIM}(I,\hat{I}) = \frac{(2\mu_I\mu_{\hat{I}} + k_1L_d^2)(2\sigma_{I\hat{I}} + k_2L_d^2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + k_1L_d^2)(\sigma_I^2 + \sigma_{\hat{I}}^2 + k_2L_d^2)}, \tag{3.33}$$

where $\mu_I$ and $\mu_{\hat{I}}$ denote the mean pixel intensities, $\sigma_I^2$ and $\sigma_{\hat{I}}^2$ denote the variances, and $\sigma_{I\hat{I}}$ denotes the covariance between $I$ and $\hat{I}$. The constants $k_1$ and $k_2$ are stabilisation parameters, and $L_d$ denotes the dynamic range of the image. SSIM correlates better with human visual perception than PSNR and is particularly sensitive to geometric distortions such as blurred edges or inconsistent surface structures, which commonly arise when multi-view observations are temporally misaligned.

Unlike pixel-wise metrics such as PSNR, the SSIM is designed to capture perceptual differences by decomposing image similarity into luminance, contrast, and structural components. This decomposition is motivated by the observation that the human visual system is more sensitive to changes in local structure than to uniform shifts in pixel intensity. By explicitly modelling the correlation between local image patches, SSIM places greater emphasis on preserving spatial relationships and edge structures, which are closely tied to the underlying scene geometry. As a result, SSIM is particularly effective in revealing geometric inconsistencies, such as blurred surfaces, duplicated edges, or ghosting artefacts that may arise when multi-view observations are temporally misaligned. From a system perspective, this property makes SSIM well suited for evaluating networked 3D perception systems. Temporal inconsistency in the observation set $\Omega(t)$ can violate the assumption of a shared scene configuration across views, leading to structural distortions in the reconstructed representation. While such distortions may only induce modest changes in pixel-wise error, they often result in a noticeable degradation of SSIM, thereby providing a more sensitive indicator of geometry-related failure modes.

Nevertheless, SSIM remains a hand-crafted metric operating on local statistics and does not explicitly encode higher-level semantic or perceptual priors. For this reason, SSIM is employed alongside learned perceptual metrics in the subsequent evaluation to provide a more comprehensive assessment of reconstruction quality.

### 3.2.3 Learned Perceptual Image Patch Similarity (LPIPS)

To further assess perceptual similarity, we adopt the learned perceptual image patch similarity (LPIPS) metric [53]. Unlike hand-crafted metrics, LPIPS measures image similarity in the feature space of a deep convolutional neural network that has been empirically shown to correlate well with human perceptual judgments. The underlying premise is that distances between deep feature representations capture perceptually meaningful differences that are not well reflected by raw pixel-wise comparisons.

In the $t$-th time slot, the LPIPS score between the ground-truth image $I$ and the synthesized

image $\hat{I}$ is computed as

$$\mathrm{LPIPS}\left(I,\hat{I}\right) = g\left(d\left(I,\hat{I}\right)\right), \tag{3.34}$$

where $d(\cdot,\cdot)$ denotes the feature-space distance and $g(\cdot)$ maps this distance to a scalar perceptual similarity score. Specifically, the feature-space distance is defined as

$$d\left(I,\hat{I}\right) = \sum_{l=1}^{L} \frac{1}{L_H L_W} \left\| \mathbf{w}_l \odot \left(I^{(l)} - \hat{I}^{(l)}\right) \right\|_2^2, \tag{3.35}$$

where $I^{(l)}(t)$ and $\hat{I}^{(l)}(t)$ denote the feature maps extracted from the $l$-th layer of a pretrained convolutional neural network. Different layers capture image characteristics at different semantic levels, ranging from local edges and textures in early layers to higher-level structures and semantic patterns in deeper layers. The learned channel-wise weights $\mathbf{w}_l$ balance the relative contributions of different feature dimensions. By aggregating discrepancies across multiple feature layers, LPIPS provides a perceptual distance that is sensitive to high-level visual attributes such as texture consistency, structural integrity, and semantic coherence. As a result, LPIPS is particularly effective in revealing perceptual artefacts that may arise from temporally inconsistent supervision. In the context of networked 3D perception, stale observations in the fusion set $\Omega(t)$ can introduce misaligned geometry or appearance inconsistencies that are weakly penalised by pixel-wise metrics but manifest clearly as discrepancies in deep feature space.

### Comparisons

Fig. 3.6 provides a qualitative comparison that illustrates the complementary behaviour of the three evaluation metrics considered in this work. For a global brightness shift, which primarily alters pixel intensities while preserving spatial structure, PSNR exhibits a noticeable degradation, whereas SSIM and LPIPS remain relatively stable. This reflects the fact that pixel-wise metrics are sensitive to uniform intensity deviations, even when the underlying image structure is largely unchanged. In contrast, structural distortions such as Gaussian blur lead to pronounced reductions in SSIM and LPIPS, despite more moderate changes in PSNR. Blurring attenuates high-frequency details and weakens edge consistency, which directly impacts structural and perceptual similarity but may not substantially increase pixel-wise error. The most severe degradation is observed under ghosting artefacts, where duplicated or misaligned edges are clearly visible in the synthesized image. Such artefacts, which are characteristic of temporally inconsistent or misaligned observations, are strongly penalised by SSIM and LPIPS, while PSNR alone may underestimate their perceptual severity. This comparison highlights that no single metric fully captures all aspects of reconstruction quality. Taken together, the results in Fig. 3.6 motivate the joint use of pixel-level and perceptual metrics for evaluating neural 3D scene representations. In particular, in timeliness-aware perception settings, temporal misalignment can manifest as structurally and perceptually salient artefacts that are insufficiently reflected by pixel-wise fi-


| Case | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| Brightness shift | 18.45 | 0.8046 | 0.0276 |
| Gaussian blur | 24.99 | 0.8214 | 0.3234 |
| Local ghosting | 23.95 | 0.9329 | 0.0660 |

Figure 3.6: Examples illustrating the complementary behaviour of PSNR, SSIM, and LPIPS under different reconstruction artefacts.

delity alone.

# Chapter 4

# Monocular-Camera-Based 3D Scene Representation on Robotic Arm Platforms

## 4.1 System Overview

This chapter presents an embodied robotic framework for real-time 3D scene representation. This robotic 3D scene representation platform was developed and deployed within the context of the UK Atomic Energy Authority (UKAEA) RAICo1 project on real-time 3D reconstruction and modelling for robotics applications.

Nuclear decommissioning environments provide a particularly relevant application context for timeliness-aware 3D scene representation due to their hazardous, remote, and visually constrained operating conditions. In such scenarios, robotic teleoperation and remote inspection often rely heavily on continuously updated scene representations to maintain operator situational awareness while minimising repeated exposure, traversal, or sensing operations. Consequently, the contribution of this work is not the nuclear application itself, but the investigation of how embodied sensing, communication-aware perception, and interactive 3D reconstruction can support

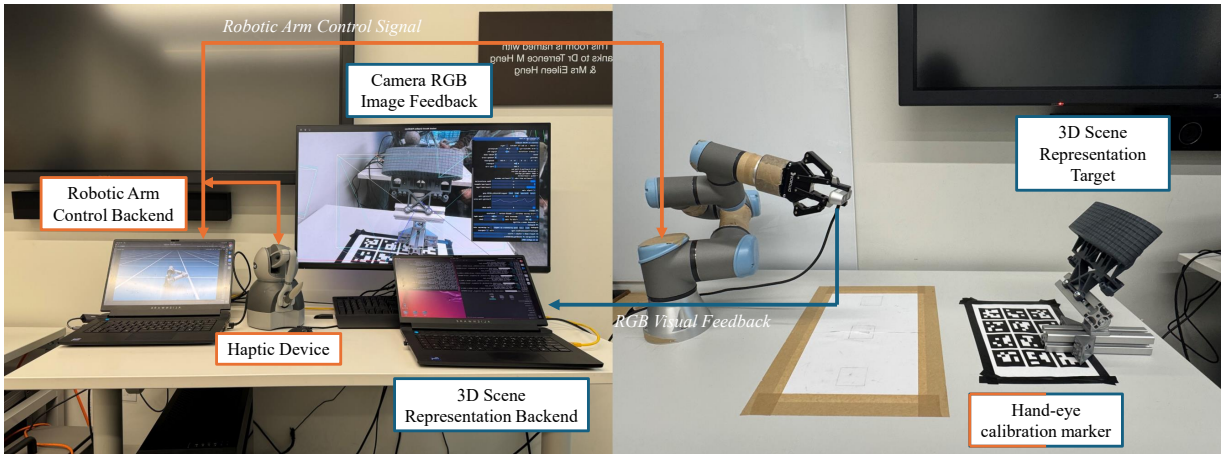



Figure 4.1: Overview of the embodied real-time 3D scene representation framework.

remote inspection tasks under practical operational constraints. The embodied robotic platform considered in this chapter was developed and evaluated primarily within a controlled laboratory environment in collaboration with the UKAEA RAICo1 project, rather than being directly deployed inside active nuclear decommissioning facilities. The nuclear inspection context should therefore be interpreted primarily as a motivating remote-inspection application scenario used to guide the system design and operational requirements.

As illustrated in Fig. 4.1, a human operator teleoperates an industrial robotic arm using a haptic device to acquire multi-view RGB observations. An eye-in-hand RGB camera rigidly mounted on the end effector captures images while the system simultaneously recovers the camera pose in real time via robot kinematics and hand–eye calibration. The image–pose stream is then used to incrementally train a fast neural radiance field. The system performs online view integration and generates intermediate 3D scene reconstruction updates within several seconds, enabling temporally continuous scene feedback during robotic teleoperation. While low-level robot control and image streaming operate at frame-rate timescales, the sampling interval of images for the 3D scene representation is at a second-level due to the operator and the movement of the robotic arm; the 3D neural scene optimisation itself is updated at second-level latency depending on scene complexity and GPU resources. The notion of "real-time" in this work is task-dependent and interaction-oriented. The objective is not instantaneous convergence of the neural representation, but maintaining sufficiently fresh scene updates for robotic perception and teleoperation.

## 4.2 Hardware Design

### 4.2.1 Teleoperation Interface

The teleoperation interface of the proposed system is based on the *3D Systems Touch* haptic device. This grounded haptic device provides six-degree-of-freedom (6-DoF) positional input through a stylus mechanism, allowing the operator to intuitively manipulate the viewpoint of the robotic arm in Cartesian space.

The 3D Systems Touch device offers high-resolution position sensing within a compact workspace. Although its physical range of motion is limited compared to the robot's reachable volume, its spatial precision enables fine-grained control during close-range inspection. The internal servo loop operates at a high update rate, ensuring smooth and responsive interaction between operator input and robot motion. Unlike discrete input devices such as keyboards or joysticks, the haptic interface provides continuous spatial control. This continuous interaction is particularly important for multi-view data acquisition, where smooth camera trajectories help maintain temporal consistency between consecutive frames and reduce motion-induced artifacts.

Since the workspace of the haptic device is significantly smaller than that of the UR3e manip-

ulator, a scaling transformation is applied between the stylus motion and the robot end-effector displacement. Let $\mathbf{p}_h(t) \in \mathbb{R}^3$ denote the stylus position in the haptic coordinate frame, and $\mathbf{p}_{ee}(t) \in \mathbb{R}^3$ denote the commanded end-effector position in the robot base frame. The mapping can be abstracted as a linear scaling transformation:

$$\mathbf{p}_{ee}(t) = \mathbf{p}_{ee}^0 + S\left(\mathbf{p}_h(t) - \mathbf{p}_h^0\right), S = \mathrm{diag}(s_x, s_y, s_z), \tag{4.1}$$

where $\mathbf{p}_{ee}^0$ is the reference end-effector position, $\mathbf{p}_h^0$ is the neutral stylus position, $S$ is a diagonal scaling matrix. The scaling coefficients $(s_x, s_y, s_z)$ are selected to balance workspace coverage and fine manipulation precision. Small stylus movements are mapped to proportional Cartesian motions of the robot, allowing both coarse exploration and fine local adjustment.

A key advantage of using a haptic interface lies in the natural perception–action coupling it enables. The operator can visually observe both real-time image feedback and intermediate 3D scene representations and immediately adjust the viewpoint to compensate for under-sampled regions. This interactive refinement process leads to a more diverse spatial distribution of viewpoints compared to fixed pre-programmed trajectories. In practice, operators tend to focus on visually salient or geometrically complex regions, which implicitly improves coverage of challenging surfaces. Therefore, the haptic interface does not merely provide motion input, but actively shapes the data distribution used for neural 3D scene representations.

A clutching mechanism is implemented to allow repositioning of the stylus without commanding robot motion, preventing unintended large pose jumps. Additionally, safety constraints such as workspace limits and joint boundaries are enforced at the control layer to ensure safe teleoperation. In summary, the 3D Systems Touch device provides an intuitive and high-resolution human–machine interface for embodied multi-view data acquisition. Its integration with the robotic manipulator enables interactive exploration, fine viewpoint control, and real-time perception-driven adjustment during reconstruction.

### 4.2.2 Robotic Manipulator and Sensor Configuration

A Universal Robotics UR3e robotic arm is adopted as the viewpoint actuation platform for multi-view data acquisition. The manipulator provides six revolute joints, enabling full spatial positioning and orientation control of the end effector. Compared to heavier industrial manipulators, the UR3e offers a compact footprint and sufficient positioning repeatability for tabletop-scale reconstruction tasks. For real-time neural 3D scene representations, absolute positioning accuracy is less critical than repeatability and smooth motion execution. Since camera poses are derived from joint encoder readings through forward kinematics, consistent joint behavior and low backlash are essential. The UR3e provides stable kinematic performance within its operational envelope, allowing reproducible viewpoint trajectories across repeated experiments. Additionally, the manipulator supports configurable velocity and acceleration limits. During

image acquisition, motion speed is intentionally constrained to reduce motion blur and ensure photometric consistency between consecutive frames. This design reflects a tradeoff between acquisition speed and reconstruction fidelity.

The end-effector assembly consists of two primary components: (i) a Robotiq 2F-85 parallel gripper, and (ii) a rigidly mounted RGB camera.

The Robotiq 2F-85 gripper is attached directly to the UR3e flange. The RGB camera is mounted on a custom bracket integrated with the gripper assembly. This stacked configuration ensures that the camera optical center remains mechanically fixed relative to the end-effector structure. Any micro-flexibility between the gripper and camera mount could introduce subtle pose inconsistencies, which would propagate into the image–pose stream used for neural field optimization. The addition of the Robotiq 2F-85 and camera module alters the mass distribution of the end effector. This effect could be neutralized by adding payload weight in the UR3e robotic arm manipulator settings. Sudden accelerations are avoided, as they may introduce vibration that degrades image sharpness. The combined kinematic reach of the UR3e and the compact experimental setup enables near half-hemispherical coverage around the target object. By varying end-effector elevation and azimuth angles, the system can progressively reduce unobserved regions. The presence of the gripper does not obstruct the primary field of view, as the camera is positioned to avoid occlusion by the gripper fingers.

In summary, the UR3e manipulator, the Robotiq 2F-85 gripper, and the rigidly mounted RGB camera together form an integrated sensing–actuation module. The manipulator provides controlled and repeatable spatial motion, the gripper extends functional interaction capabilities, and the eye-in-hand camera captures synchronized multi-view observations. This physical configuration establishes a stable and versatile foundation for embodied real-time 3D scene representation.

## 4.3 Hand–Eye Calibration and Real-Time Pose Recovery

To recover camera poses in the world frame during operation, the fixed hand-eye transformation between the end effector and the camera is estimated via a ROS1-based hand–eye calibration procedure. A planar fiducial target (QR code) placed on the worktable provides a stable visual reference during calibration.

### 4.3.1 Coordinate Frames and Problem Formulation

The objective of hand-eye calibration is to estimate the constant rigid transformation between the end-effector frame and the camera frame:

$$T_{EE}^{C}, \tag{4.2}$$

where $EE$ denotes the end effector frame of the robotic arm, $W$ denotes the world frame, $C$ denotes the camera frame, and $B$ is the robot base frame. Here $T_{EE}^{C} \in \mathbb{R}^{[4\times4]}$ is the transformation matrix expressing camera frame w.r.t. end-effector frame.

Once this transform is known, the camera pose in the world frame can be recovered online through:

$$T_W^C(t) = T_W^{EE}(t)\, T_{EE}^C, \tag{4.3}$$

where $T_W^{EE}(t)$ is obtained from robot forward kinematics at time $t$.

This formulation eliminates the need for external tracking systems or visual odometry during runtime. The camera pose is determined purely from joint encoder readings and the calibrated rigid transform. As a result, pose recovery is deterministic and free from drift accumulation.

### 4.3.2 Hand–Eye Calibration Procedure

The hand–eye calibration problem can be conceptually described as estimating the fixed rigid transformation between the end effector and the camera, given a set of robot motions and corresponding observed target motions.

During calibration, the robot is moved to multiple distinct poses. For each pose $i$, two quantities are recorded: (i) the end-effector pose $T_W^{EE}(i)$ obtained from forward kinematics, and (ii) the observed pose of the fiducial target relative to the camera, denoted as $T_C^F(i)$.

To eliminate dependence on the unknown world frame, relative motions between pose pairs are constructed. For two robot configurations $i$ and $j$, the relative end-effector motion is

$$A_{ij} = \left(T_W^{EE}(i)\right)^{-1} T_W^{EE}(j), \tag{4.4}$$

and the corresponding relative target motion observed by the camera is

$$B_{ij} = T_C^F(i)\left(T_C^F(j)\right)^{-1}. \tag{4.5}$$

The hand–eye relationship can then be expressed in the classical form

$$A_{ij}X = XB_{ij}, \tag{4.6}$$

where $X = T_{EE}^C$ is the unknown rigid transformation between the end-effector and camera frames.

In this work, the Tsai–Lenz method [84], implemented in the ROS1 Noetic package, is used to solve this problem. The solution proceeds in two stages. First, the rotational component of $X$ is estimated by solving a linear system derived from the rotational parts of $A_{ij}$ and $B_{ij}$. Once the rotation is determined, the translational component is obtained via linear least-squares using the previously estimated rotation.

Calibration accuracy directly affects reconstruction convergence. Even small rotational errors in $T_{EE}^{C}$ can propagate into systematic viewpoint misalignment, leading to blurred or distorted neural reconstructions.

Once $T_{EE}^{C}$ is estimated, runtime pose recovery becomes computationally lightweight. In the $t$-th time slot, an image $I(t)$ is sampled from the camera. The corresponding camera pose in homogeneous form is computed as

$$T_{W}^{C}(t) = T_{W}^{EE}(t)\, T_{EE}^{C}. \tag{4.7}$$

For implementation, the minimal pose representation $\mathbf{p}_c(t)$, consisting of translation and quaternion rotation, is extracted from $T_{W}^{C}(t)$. This computation involves only matrix multiplication and is negligible compared to neural rendering costs. Therefore, pose recovery does not introduce additional computational bottlenecks in the pipeline.

### 4.3.3 Time Synchronization

To ensure consistency between image capture and pose estimation, timestamps from the camera driver and the robot controller are synchronized within the ROS1 middleware framework. Since camera frames and joint states are published as independent ROS topics, their arrival times are not inherently identical. Without synchronization, even small temporal offsets may lead to incorrect pose association, particularly when the manipulator is in motion.

In the proposed system, each RGB image frame $I(t)$ is assigned the closest corresponding joint state message based on timestamp matching. Specifically, the joint configuration whose timestamp minimizes the temporal difference with the image frame is selected for forward kinematics computation. This strategy ensures that the computed end-effector pose $T_{W}^{EE}(t)$ corresponds as closely as possible to the actual camera exposure time.

Although the UR3e controller and the camera operate at different frequencies, their clocks are aligned within the same ROS1 runtime environment, which limits synchronization error to the millisecond scale. For moderate robot velocities used during data acquisition, such timing deviations result in negligible spatial error.

In summary, the time synchronization mechanism ensures reliable image–pose pairing, which is essential for stable and accurate online neural scene optimization.

## 4.4 Software Framework

### 4.4.1 Teleoperation Control

The teleoperation control pipeline maps haptic inputs to end-effector motion commands. Motion generation is based on the RMPflow controller in Isaac Sim. In this thesis, RMPflow is used as a

practical policy for generating smooth, constraint-aware motions (e.g., joint limits and collision avoidance) while tracking operator-specified end-effector targets. The resulting commands are executed on the physical UR3e via ROS1 interfaces.

The control loop operates continuously during teleoperation. Filtered pose commands from the haptic interface are transmitted to the motion policy layer, which generates feasible joint-space trajectories. These trajectories are streamed to the UR3e controller at a fixed update frequency, ensuring smooth and stable camera motion during image acquisition. Velocity and acceleration limits are explicitly enforced to reduce motion-induced image blur and maintain photometric consistency across consecutive frames.

### 4.4.2 Data Acquisition Pipeline

During operation, the framework continuously collects RGB frames from the eye-in-hand camera and associates each frame with a synchronized camera pose computed from $T_W^C(t)$. These paired observations $\{I_t, T_W^C(t)\}$ are streamed to the reconstruction module in an online fashion, enabling incremental updates as new views arrive.

In practice, the data acquisition process follows a fixed runtime sequence. When an RGB frame is received from the camera driver, the most recent synchronized joint state message is retrieved. Forward kinematics is then applied to compute $T_W^{EE}(t)$, and the calibrated hand–eye transform is used to obtain $T_W^C(t)$.

The recovered homogeneous transformation matrix is subsequently converted into the camera-to-world convention required by the reconstruction backend. Specifically, the transform is reformatted to match the COLMAP-style [91] extrinsic representation used by our 3D scene representation backend. This conversion is deterministic and corresponds to a fixed coordinate transformation between the robot base frame and the reconstruction world frame.

Once converted, the image–pose pair is appended to a reconstruction buffer. The system maintains a dynamically growing dataset during acquisition, which allows the reconstruction module to access all previously observed views.

### 4.4.3 Preprocessing via Object Detection and Segmentation

Before being passed to the neural reconstruction module, the captured RGB frames undergo a lightweight preprocessing stage. The primary objective of this step is to suppress background clutter and focus the reconstruction process on the target object.

Specifically, a YOLO [32]-based object detector is applied to each incoming frame to obtain a bounding box enclosing the main object of interest. A YOLO model has to be pretrained with task-specific fine-tuning, as the bounding box provides a spatial prior for subsequent segmentation. Within the detected region of interest, a segmentation algorithm is applied to extract the foreground mask. Unlike the detector, the segmentation stage does not require additional pre-

training for the specific experimental setup. The segmentation mask isolates the object from the background, reducing irrelevant visual information in the training data. The resulting masked image retains the original camera pose $T_W^C(t)$ computed from robot kinematics. Only pixel intensities are modified, while geometric supervision remains unchanged. Specifically, the task-irrelevant pixels (background) are replaced with transparent pixels in the alpha channel of PNG. The masked image–pose pair is then forwarded to the reconstruction buffer.

This preprocessing stage improves reconstruction stability by reducing background interference, especially in indoor laboratory environments where static background structures may otherwise dominate the neural optimization process. Furthermore, by concentrating training rays on foreground regions, the effective utilization of GPU computation is improved, leading to faster convergence for object-centric scenes.

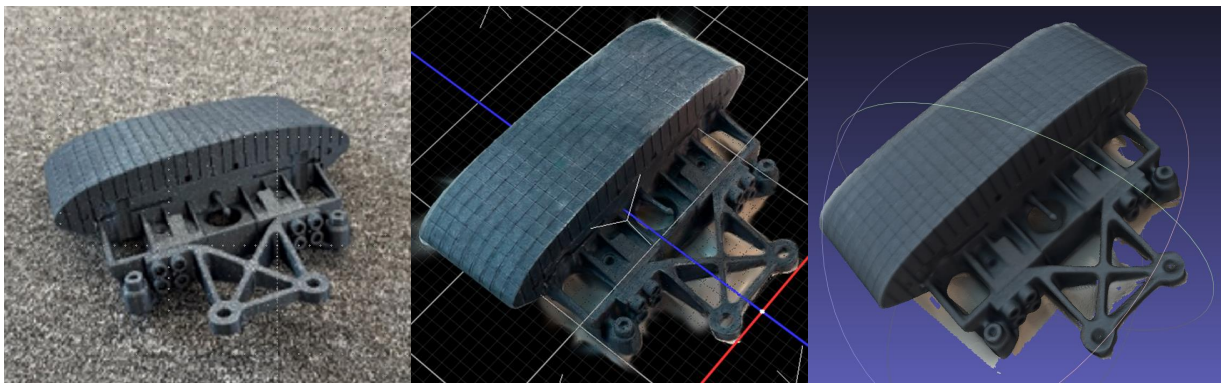
Importantly, this step does not alter the deterministic pose recovery pipeline. It acts purely on image content prior to neural training, and can be enabled or disabled depending on experimental requirements.

### 4.4.4 Real-Time 3D Scene Representation

For real-time 3D scene representation, the system integrates a fast training neural radiance field based on Instant-NGP. The 3D scene representation module incrementally incorporates newly acquired views into the training set, enabling online refinement of the neural field as data is collected. When the number of accumulated images exceeds a minimal initialization threshold (typically greater than three views), an initial batch processing step is performed. During this step, camera poses are normalized such that the geometric center of all camera positions is re-centered as the origin of the reconstruction world frame. This normalization improves numerical conditioning and stabilizes subsequent neural optimization.

After initialization, neural training proceeds incrementally. In the $t$-th time slot, the current network parameters $\theta_t$ are updated using all available image–pose pairs up to that moment. When a new image–pose pair arrives, it is appended to the dataset, and optimization continues from the previous parameter state rather than restarting from scratch. This warm-start strategy enables smooth temporal refinement of the reconstructed scene. To further compensate for small residual pose inaccuracies, an extrinsic refinement stage is integrated into the training loop. Camera extrinsic parameters are jointly optimized with scene representation, allowing minor pose corrections within bounded limits. This refinement mitigates small calibration biases or timestamp-induced deviations without introducing global drift.

The operator retains the option to manually reset the training process in cases where the scene configuration changes significantly or when experimental conditions require reinitialization. Otherwise, reconstruction evolves continuously as new viewpoints are acquired. All reconstruction experiments in this chapter are executed on an RTX 4090 GPU. This hardware–algorithm combination enables: (i) real-time addition of new viewpoints, (ii) incremental parameter up-



(a) (b) (c)

Figure 4.2: (a) The tile in the real world; (b) 3DGS-based 3D scene representation with segmentation; (c) Surface-continuous mesh reconstructed via PGSR.

dates within seconds, and (iii) interactive monitoring of reconstruction progress. Consequently, the system transforms neural 3D scene representation from an offline post-processing task into a real-time embodied perception process.

### 4.4.5 Post-Processing to Surface-Continuous Mesh

While Instant-NGP enables real-time neural scene representation during data acquisition, certain robotics applications require explicit geometric models rather than implicit neural fields. For this purpose, the collected image-pose dataset is further post-processed to generate a surface-continuous 3D mesh.

After data acquisition is completed, the full dataset $\{I_t, T_W^C(t)\}_{t=1}^N$ is exported in COLMAP-compatible format. This dataset is then processed using Planar-based Gaussian Splatting (PGSR) [83], which can be interpreted as a surface-regularized extension of 3D Gaussian Splatting. Unlike purely volumetric neural fields, PGSR enforces local surface continuity constraints, leading to geometrically coherent and watertight 3D scene representations.

The post-processing pipeline consists of: 1)Training a Gaussian-based scene representation using the captured views, 2) Applying surface regularization to enforce local planarity and continuity, and 3) Extracting a triangulated mesh from the optimized representation.

In practice, for tabletop-scale scenes captured in this framework, the full post-processing pipeline completes within approximately 30 minutes on an RTX 4090 GPU. The resulting mesh provides explicit geometry, which can be directly used for visualization, inspection analysis, simulation, or downstream robotic planning tasks. The integration of PGSR as an offline post-processing stage complements the real-time NeRF-based visualization. The former provides interactive feedback during acquisition, while the latter yields high-fidelity surface models for deployment-level use. Together, these two stages establish a practical pipeline from embodied data collection to deployable 3D geometric assets.

## 4.5 Extension: Active Viewpoint Planning for Human Preference Driven 3D Scene Representations

Beyond teleoperation-based data acquisition, the embodied reconstruction platform developed in this chapter also provides a practical testbed for studying active viewpoint planning in robotic perception systems.

In teleoperation mode, the operator manually explores the scene while the system continuously records synchronized image–pose pairs and updates the neural scene representation. Although this human-in-the-loop strategy provides strong flexibility, it does not scale well to long-duration inspection tasks, nor does it provide an objective mechanism for optimizing viewpoint selection. To address this limitation, the same perception–action pipeline can be extended to support algorithmic next-best-view (NBV) planning. Specifically, the deterministic pose recovery mechanism, real-time neural reconstruction module, and dataset buffering architecture introduced earlier form a closed-loop environment in which viewpoint policies can be evaluated and optimized in real time.

Building upon this infrastructure, we developed a preference-driven active 3D scene representation framework that integrates expert operator feedback into robotic viewpoint planning. Instead of relying solely on geometric uncertainty or photometric reconstruction metrics, the proposed approach learns a viewpoint selection policy based on Reinforcement Learning from Human Feedback (RLHF) collected from 3D scene representations.

### 4.5.1 System Overview

The overall framework is illustrated in Fig. 4.3. The system integrates robotic data acquisition, neural scene representation, and reinforcement learning from human feedback into a unified closed-loop optimization process.

The RLHF-based viewpoint planning pipeline consists of five main stages.

**1) Robotic exploration and trajectory generation:** At each time step, the robotic arm moves to a candidate camera pose and captures an RGB observation together with the corresponding camera pose. These observations are accumulated to form viewpoint trajectories that serve as the input for the reconstruction pipeline.

**2) Expert preference evaluation:** 3D scene representations reconstructed from different viewpoint trajectories are presented to expert operators through an interactive interface. The operators compare the reconstructed scenes and indicate which result better satisfies task-specific visualization requirements.

**3) Reward model learning:** The collected pairwise preferences are used to train a reward predictor that estimates the quality of viewpoint trajectories. This learned reward model serves as a surrogate objective that approximates human evaluation of reconstruction quality.

**4) Policy optimization:** Using the learned reward model, a viewpoint selection policy is optimized through deep reinforcement learning. In our implementation, the Proximal Policy Optimization (PPO) algorithm is employed to maximize the expected cumulative reward over viewpoint trajectories.

**5) Online iterative refinement:** The optimized policy is deployed on the robotic platform to generate new viewpoint trajectories. The resulting reconstructions can again be evaluated by expert operators, allowing additional preference data to be collected and incorporated into the training process, thereby progressively improving the policy.

During operation, the robotic arm executes viewpoint trajectories generated by a learned policy rather than manual teleoperation. At each viewpoint, the onboard RGB camera captures an image while the robot controller records the corresponding camera pose. These image–pose pairs are immediately integrated into the neural scene representation pipeline, producing an updated 3D reconstruction of the scene.

The reconstructed models are then presented to expert operators through an interactive visualization interface, where different viewpoint trajectories can be compared. The operator selects the preferred reconstruction result, providing human feedback that reflects task-specific visualization requirements.

This feedback is subsequently used to train a reward model, which estimates the quality of viewpoint trajectories. The learned reward is then used to optimize a viewpoint policy through deep reinforcement learning, allowing the robotic system to automatically discover trajectories that generate preferred 3D scene representations.

### 4.5.2 RLHF Formulation for Viewpoint Optimization

The active viewpoint planning problem can be formulated as a sequential decision-making process, where a robotic agent selects camera viewpoints to progressively improve the quality of a 3D scene representation.

**State representation** In the $t$-th time slot, the state $\mathbf{s}_t$ encodes the current observation of the scene, including the captured RGB image and its associated camera pose. In practice, the RGB observation is processed by a visual feature extractor to produce a compact feature representation

$$o_t = \mathscr{F}_e(I_t) \in \mathbb{R}^d, \tag{4.8}$$

where $I_t$ denotes the captured image and $\mathscr{F}_e(\cdot)$ represents the feature extraction network. In practice it is represented by the backbone of a pretrained YOLOv11 model. The state $\mathbf{s}_t$ therefore summarizes the information available to the agent for selecting the next viewpoint.

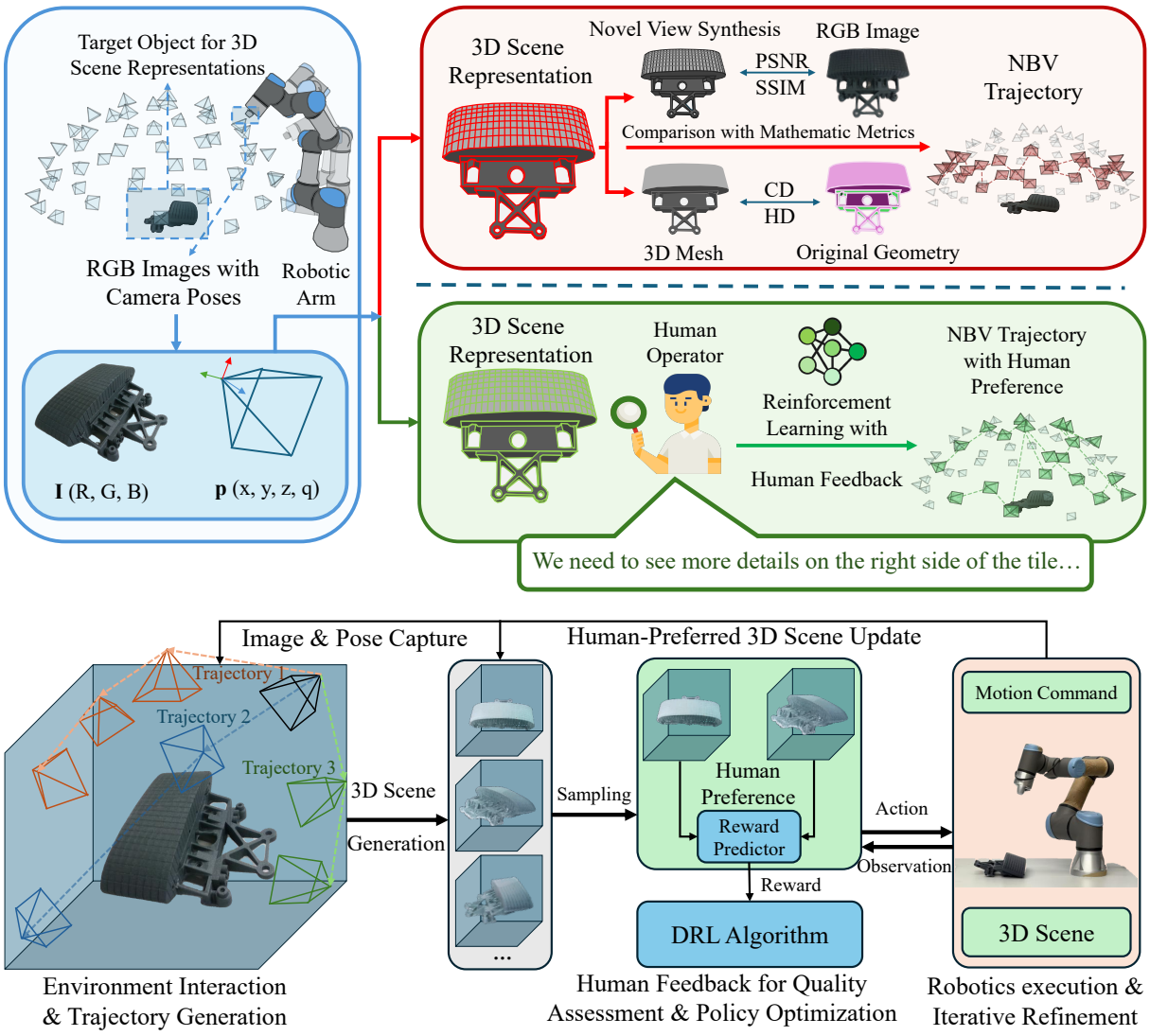



Figure 4.3: Overview of the preference-driven active 3D scene representation framework built upon the embodied reconstruction platform described in this chapter. The figures are reproduced from our related publication.

**Action space** The action $\mathbf{s}_t$ corresponds to selecting a candidate camera pose within the reachable workspace of the robotic manipulator. In the $t$-th time slot, the action is defined as

$$\mathbf{a}_t = \{a_1(t), a_2(t), \ldots, a_A(t)\}, \tag{4.9}$$

where each action represents a predefined viewpoint parameterized by a six-dimensional pose (position and orientation) of the camera. Executing action $\mathbf{a}_t$ moves the robotic end-effector to the corresponding viewpoint and triggers image acquisition. The candidate viewpoints considered in this chapter are sampled from a predefined discrete viewpoint set surrounding the target object, as illustrated in Fig. 4.3, rather than being generated through continuous pose optimisation in the full robot configuration space. Consequently, the reinforcement learning policy operates by selecting viewpoints from a finite action space under visibility and reconstruction-quality considerations.

**Trajectory generation** A sequence of observations and actions generated during exploration forms a trajectory

$$\tau = (\mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2, \mathbf{a}_2, \ldots, \mathbf{s}_T, \mathbf{a}_T), \tag{4.10}$$

which determines the set of captured images used to reconstruct the 3D scene representation.

**Preference-based reward modelling** Defining an explicit reward function for viewpoint planning is challenging because reconstruction quality depends on task-specific and often subjective criteria. Instead of manually specifying the reward, we adopt RLHF to infer the reward function from expert preferences.

Specifically, expert operators are presented with pairs of reconstructed scenes generated from two trajectories $\tau_i$ and $\tau_j$. The operator indicates a preference

$$y_{ij} = \begin{cases} 1, & \text{if } \tau_i \succ \tau_j \\ 0, & \text{otherwise,} \end{cases} \tag{4.11}$$

where $\tau_i \succ \tau_j$ indicates that trajectory $\tau_i$ produces a more desirable scene reconstruction.

A reward predictor $\hat{r}_\phi(s,a)$, parameterized by $\phi$, is trained to approximate the latent human reward signal. Following the standard preference-learning formulation, the probability that trajectory $\tau_i$ is preferred over $\tau_j$ is modeled as

$$P(\tau_i \succ \tau_j) = \frac{\exp\left(\sum_t \hat{r}_\phi(\mathbf{s}_t^i, \mathbf{a}_t^i)\right)}{\exp\left(\sum_t \hat{r}_\phi(\mathbf{s}_t^i, \mathbf{a}_t^i)\right) + \exp\left(\sum_t \hat{r}_\phi(\mathbf{s}_t^j, \mathbf{a}_t^j)\right)}. \tag{4.12}$$

Given a dataset of preference comparisons $\mathscr{D} = \{(\tau_i, \tau_j, y_{ij})\}$, the reward model is trained by minimizing the cross-entropy loss

$$\mathscr{L}(\phi) = -\sum_{(\tau_i,\tau_j,y_{ij})\in\mathscr{D}} \Big[ y_{ij} \log P(\tau_i \succ \tau_j) + (1-y_{ij}) \log P(\tau_j \succ \tau_i) \Big]. \tag{4.13}$$

**Policy optimization** Once the reward function is learned, the viewpoint selection policy $\pi_\theta(\mathbf{a}|\mathbf{s})$ is optimized to maximize the expected cumulative reward

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta} \left[ \sum_{t=1}^{T} \gamma^{t-1} \hat{r}_\phi(\mathbf{s}_t, \mathbf{a}_t) \right]. \tag{4.14}$$

In this work, the policy is optimized using the PPO algorithm, which performs stable policy gradient updates while constraining large deviations between successive policies. This enables efficient learning of viewpoint selection strategies that align with operator preferences while maintaining robust robotic exploration behaviour.

### 4.5.3 Experimental Validation

The proposed framework was experimentally validated on a real robotic inspection platform. The system employs a Universal Robots UR3e robotic manipulator equipped with an Intel RealSense D435i RGB camera mounted on the end-effector. The UR3e provides six degrees of freedom (DOF) and a repeatability of approximately $\pm 0.03$ mm, enabling precise and repeatable viewpoint positioning during the inspection process. The D435i camera is adopted for its easy connectivity with the ROS module and its accurate out-of-the-box intrinsics. In experiment, the depth channel output was never taken into any calculations.

To support both real-time robot control and computationally intensive 3D scene representation and reinforcement learning processes, the experimental testbed adopts a distributed architecture consisting of two computing servers. The robotic control server executes motion trajectories and processes sensor feedback through a ROS-based control framework. Motion generation is implemented using the Riemannian Motion Policy flow (RMPflow) motion planning module in NVIDIA Isaac Sim [93], while joint motion execution is handled through the MoveIt kinematics plugin to ensure smooth and collision-free movements. A second server performs the deep reinforcement learning optimization and neural 3D scene representation tasks. Captured RGB images and corresponding camera poses are transmitted from the control server to the learning server, while optimized target viewpoints are returned to the robot controller, forming a closed-loop perception–action pipeline.

Experiments were conducted in a representative nuclear decommissioning inspection scenario. The task focuses on the 3D visualization and inspection of reactor tiles used in fusion reactor environments. These tiles are typically composed of beryllium-coated Inconel and arranged in a slightly inclined structure to regulate plasma circulation within the reactor core. Damage or degradation of such components can lead to serious operational risks, making accu-

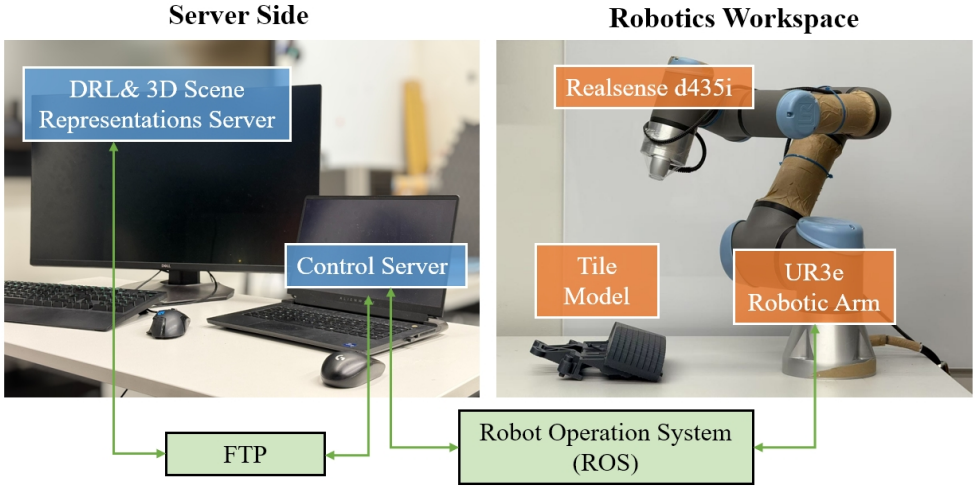



Figure 4.4: Experimental setup of the RLHF-based 3D scene representation system. The setup consists of a UR3e robotic arm with an Intel RealSense D435i camera, controlled via a ROS-based framework. A control server handles motion execution, while a RL server optimizes viewpoint selection based on human feedback. A File Transfer Protocol (FTP) ensures efficient data transfer, enabling real-time policy refinement for 3D scene representation.

rate visual inspection a critical requirement. To emulate this inspection scenario in a controlled laboratory setting, high-fidelity full-scale 3D printed replicas of reactor tiles were used as experimental targets.

The broader embodied framework introduced earlier in this chapter describes the general robotic teleoperation and reconstruction pipeline considered throughout this work. In contrast, the setup shown in Fig. 4.4 corresponds specifically to the experimental platform used for the quantitative evaluations presented in this chapter. A simplified and more controlled hardware configuration was adopted in order to improve experimental reproducibility and reduce platform-specific variability during reconstruction and viewpoint-selection evaluation.

Before data acquisition, a hand–eye calibration procedure was performed to estimate the transformation between the camera coordinate frame and the robotic base frame. This calibration ensures accurate association between captured images and camera poses, which is essential for reliable multi-view 3D scene reconstruction. During the experiments, the robotic arm autonomously explores the inspection object by executing viewpoint trajectories generated by the learned policy. At each selected viewpoint, the RGB camera captures an image and records the corresponding pose, which are subsequently used to update the 3D scene representation.

To incorporate human preferences into the viewpoint optimization process, expert operators from the UK Atomic Energy Authority (UKAEA) participated in the evaluation stage. Five experienced personnel familiar with nuclear inspection tasks were recruited to provide preference feedback on 3D scene representations. During the experiment, a total of 400 reconstructed scenes were generated under different viewpoint trajectories and exploration conditions. These reconstructions were paired into 200 comparison groups, and expert operators were asked to

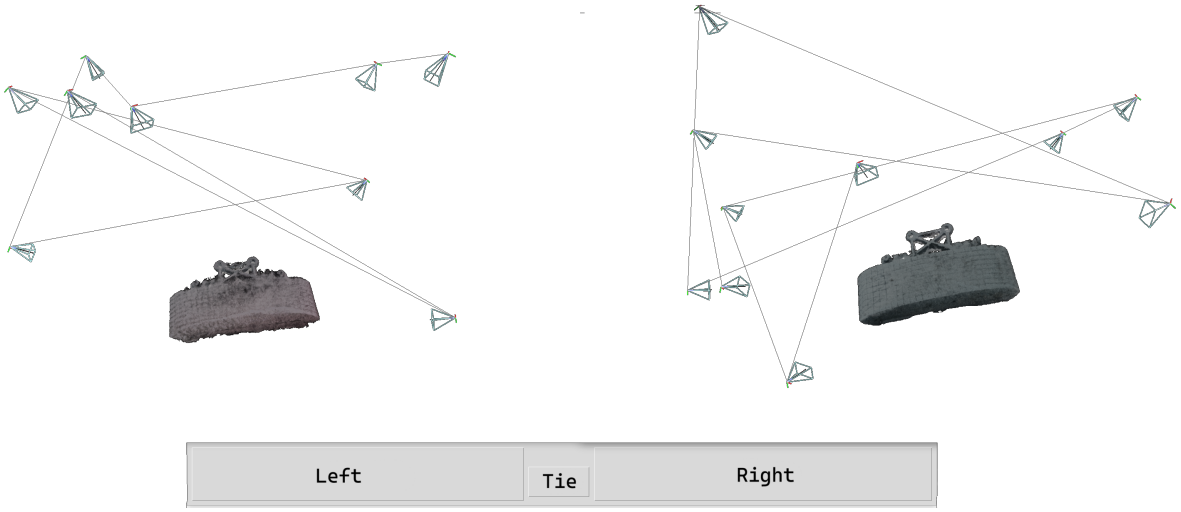



Figure 4.5: User interface for preference-based 3D scene selection and comparison.

indicate which 3D scene representation better satisfied the inspection requirements. The collected pairwise preference data were then used to train the reward predictor within the RLHF framework. Fig. 4.5 illustrates the user interface used for preference-based 3D scene evaluation and viewpoint optimisation. The interface allows users to compare different reconstructed 3D scene representations through interactive inspection, including zooming, rotation, and detailed visual examination. User selections are subsequently used to train a reward predictor for preference-aware viewpoint optimisation. The illustrated line trajectories indicate the viewpoint selection order rather than the actual robotic motion trajectory. In practice, the physical UR3e robot motion is generated separately using the Isaac Sim motion planner to ensure smooth and kinematically feasible execution. In particular, the ordering and policy identity of the compared trajectories were not disclosed to the participants during evaluation.

### 4.5.4 Convergence of Proposed Algorithm with Different 3D Representation Methods

We first analyze the convergence behavior of the proposed RLHF framework when applied to different 3D scene representation pipelines. Specifically, the policy is trained using three representative reconstruction methods: Instant-NGP, 3DGS, and PGSR.

As shown in Fig. 4.6, the learning curves exhibit stable convergence across all three representations. The reward value increases steadily during training and eventually stabilizes, indicating that the policy progressively learns to select viewpoints that better align with expert preference feedback. This result demonstrates that the proposed framework is not tied to a specific reconstruction method and can generalize across different scene representation paradigms.

To ensure statistical reliability, each experiment was repeated five times with different random seeds. The mean reward and corresponding standard deviation are reported to evaluate the

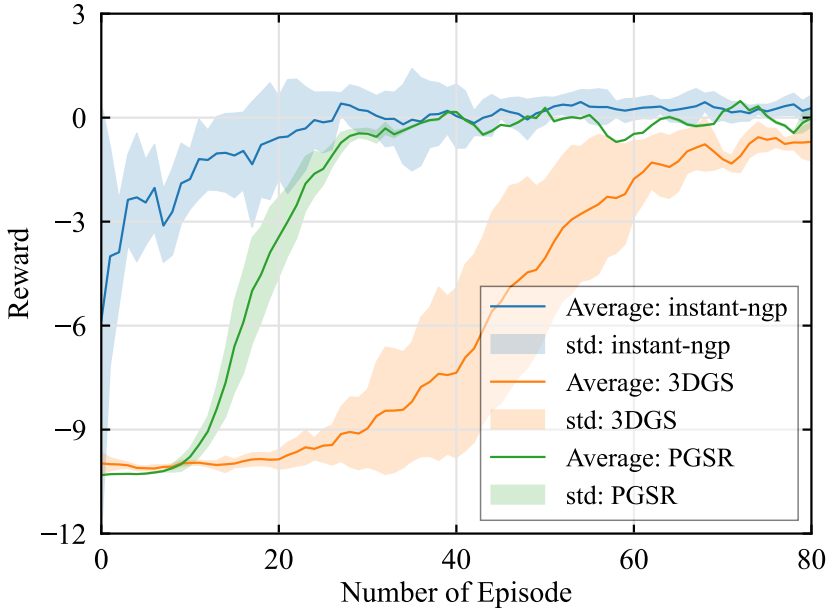



Figure 4.6: Convergence performance of our proposed framework across different 3D scene representation methods.

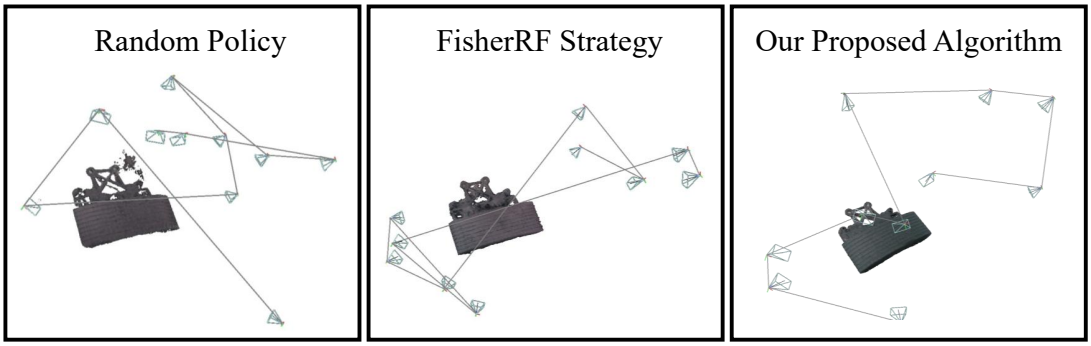



Figure 4.7: Comparative Visualization of Trajectory Efficiency.

robustness of the training process. The relatively small variance across runs indicates that the learned policy consistently converges to similar viewpoint selection strategies. Overall, these results confirm that incorporating human preference feedback enables the system to efficiently guide robotic exploration toward viewpoints that improve the perceptual quality of the reconstructed scenes.

### 4.5.5 Effectiveness Verification of Proposed Framework with Different Viewpoint Numbers

To investigate the impact of viewpoint density on 3D scene representation performance, we evaluated the proposed framework under different numbers of captured images. The evaluation considers both reconstruction fidelity and task execution time. Reconstruction quality is

Table 4.1: Comparative Analysis of Different Viewpoint Numbers with Task Times and Representation Performance

| **Number of Photos** | 5 | 8 | 10 | 13 | 15 | 20 |
|---|---|---|---|---|---|---|
| Average Task Time (s) | 26.34 | 34.72 | 38.91 | 50.18 | 57.44 | 76.21 |
| Average PSNR | 20.19 | 20.87 | 21.83 | 21.46 | 22.79 | 22.73 |
| Average SSIM | 0.8266 | 0.8472 | 0.8760 | 0.8715 | 0.8808 | 0.8841 |
| Average LPIPS | 0.1813 | 0.1654 | 0.1412 | 0.1468 | 0.1215 | 0.1261 |

measured using PSNR, SSIM, and LPIPS, while the total task time reflects the efficiency of the robotic data acquisition process.

As shown in Tab. 4.1, increasing the number of viewpoints generally improves the fidelity, leading to higher PSNR and SSIM values and lower LPIPS scores. However, this improvement comes at the cost of longer robotic exploration time. For instance, using twenty viewpoints achieves the highest SSIM value of 0.8904 and a PSNR of 22.74, but the average task time increases to 76.21 seconds.

The slight performance fluctuations observed around $N = 13$ and $N = 20$ are believed to be associated with training variance and incomplete policy convergence under larger viewpoint-selection spaces. Nevertheless, the overall trend still indicates that increasing the number of viewpoints generally improves reconstruction fidelity at the cost of longer task duration.

In contrast, using ten viewpoints achieves a favorable trade-off between reconstruction quality and acquisition efficiency. With ten images, the system reaches a PSNR of 21.83, an SSIM of 0.8760, and an LPIPS of 0.1412, while maintaining a significantly lower task time of 38.91 seconds. Based on this observation, ten viewpoints were adopted as the default configuration in the subsequent experiments, as it provides a balanced compromise between reconstruction accuracy and robotic motion efficiency.

### 4.5.6 Evaluations on Different Expert Operators

To further analyze the influence of human preferences on viewpoint planning, we conducted experiments involving five expert operators. Each operator independently evaluated reconstructed scenes and selected viewpoints according to their individual inspection preferences. These operator-specific selections were then used to construct five sets of ground-truth viewpoints.

Following the proposed framework, separate policies were trained using preference data from each operator. For evaluation, the reconstruction results were rendered from four preference-specific viewpoints corresponding to each operator and compared with the respective ground-truth images using PSNR. The results summarized in Tab. 4.2 show that when the evaluation is performed using a fixed ground-truth set, the policy trained using the corresponding operator's preference consistently achieves the highest or near-highest PSNR values. This observation indicates that the proposed framework effectively captures operator-specific inspection priorities. Moreover, it demonstrates that RLHF enables the viewpoint policy to adapt to personalized visu-

Table 4.2: Comparative Analysis of representation performance with different expert operators

| **Average PSNR/Operator** | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Ground Truth - 1 | 22.23 | 20.89 | 22.00 | 21.33 | 21.33 |
| Ground Truth - 2 | 21.85 | 25.28 | 19.94 | 20.37 | 21.52 |
| Ground Truth - 3 | 21.48 | 21.38 | 22.18 | 20.37 | 22.34 |
| Ground Truth - 4 | 22.51 | 23.16 | 21.50 | 23.72 | 21.85 |
| Ground Truth - 5 | 21.38 | 22.68 | 22.19 | 21.74 | 23.35 |

alization requirements, which is particularly important in industrial inspection scenarios where operators may focus on different regions of interest.

Interestingly, disagreements between operators may themselves provide useful insight into the proposed framework. Different operators may implicitly prioritise different aspects of the reconstruction process, such as global scene coverage, local geometric detail, trajectory efficiency, or inspection confidence. This suggests that there may not exist a single universally optimal viewpoint policy for all users or operational contexts. Instead, the observed variability further motivates the need for adaptive and task-oriented 3D scene representation frameworks capable of incorporating different human preferences and inspection objectives.

### 4.5.7 Comparison of 3D Scene Representations Quality and Trajectory Efficiency on Different Baselines

We further compare the proposed method with two baseline viewpoint planning strategies: random viewpoint sampling and the FisherRF strategy [94]. The evaluation considers both reconstruction fidelity and robotic trajectory efficiency.

As shown in Tab. 4.3, the proposed algorithm consistently produces shorter exploration trajectories while maintaining competitive or superior reconstruction quality across different scene representation methods. Specifically, the average robotic path length of the proposed method is 2.34 meters, which is shorter than both the random policy (3.28 meters) and FisherRF (2.61 meters), indicating improved motion efficiency.

In terms of reconstruction quality, the proposed method achieves PSNR values of 22.33 for Instant-NGP and 23.54 for 3DGS, together with competitive SSIM and LPIPS scores. These results demonstrate that the RLHF-based viewpoint planning strategy enables the robotic system to prioritize informative viewpoints rather than uniformly sampling the entire workspace, thereby improving both representation quality and exploration efficiency. PGSR is not included in the quantitative comparison because it primarily targets offline surface-consistent refinement rather than incremental online reconstruction during viewpoint selection. Consequently, it is not directly comparable to Instant-NGP or 3DGS within the interactive reconstruction setting considered in this experiment.

Table 4.3: Comparative Analysis of Proposed Algorithm Performance Across Different 3D Representation Methods

| | **Random Policy** | | **FisherRF Strategy** | | **Proposed Algorithm** | |
|---|---|---|---|---|---|---|
| **Approaches** | Instant-ngp | 3DGS | Instant-ngp | 3DGS | Instant-ngp | 3DGS |
| Average Path Length (m) | 3.28 | | 2.61 | | 2.34 | |
| Average PSNR | 16.27 | 18.32 | 23.10 | 23.16 | 22.33 | 23.54 |
| Average SSIM | 0.812 | 0.8341 | 0.9007 | 0.97 | 0.885 | 0.879 |
| Average LPIPS | 0.234 | 0.201 | 0.1107 | 0.039 | 0.109 | 0.0426 |

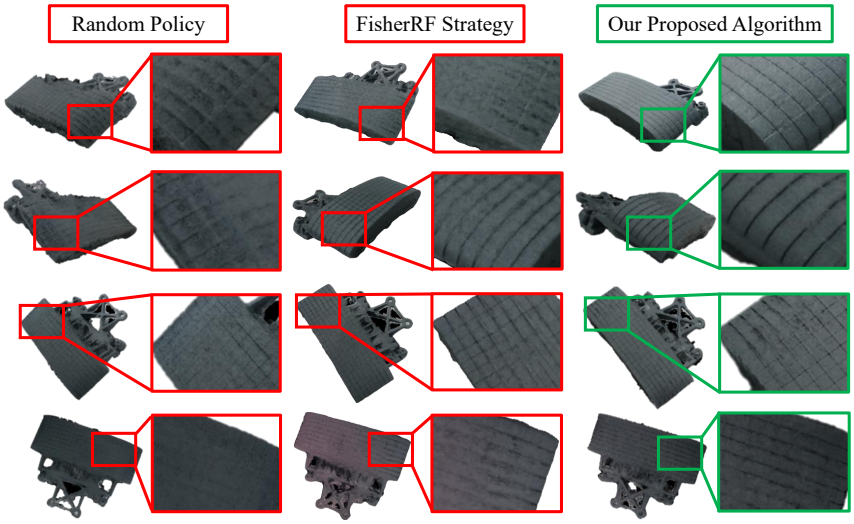



Figure 4.8: Comparative Visualization of Local Fidelity.

### 4.5.8 Comparative Visualization of Trajectory Efficiency and Local Fidelity

To provide further insight into the behavior of the proposed algorithm, we visualize the generated robotic trajectories and the resulting reconstructed scenes. As illustrated in Fig. 4.7, the trajectories produced by the proposed policy exhibit smoother and more structured exploration patterns compared with baseline methods.

In addition to trajectory efficiency, we also examine the visual quality of reconstructed regions. Fig. 4.8 presents representative rendering results from different viewpoint planning strategies. The proposed algorithm tends to focus on geometrically complex or visually important regions of the object, resulting in clearer reconstruction of occluded or detailed areas. In contrast, random viewpoint selection often fails to capture these critical viewpoints, leading to missing details or blurred structures.

These qualitative observations are consistent with the quantitative results presented earlier. Overall, the RLHF-based viewpoint planning strategy improves both robotic exploration efficiency and local reconstruction fidelity, making it particularly suitable for inspection tasks where detailed visualization of specific regions is required.

## 4.6 Discussion

The results presented in this section highlight an important capability of the monocular robotic 3D scene representation system developed in this chapter. The proposed embodied reconstruction framework provides not only a mechanism for collecting image–pose pairs through teleoperated robotic exploration, but also a structured data acquisition pipeline that can support algorithmic viewpoint optimization.

In particular, the deterministic image–pose streaming mechanism introduced earlier in this chapter ensures reliable synchronization between captured RGB observations and robotic poses. Combined with the real-time neural reconstruction module, the system enables incremental scene updates as new viewpoints are acquired. This tight coupling between robotic motion and 3D scene representation creates a closed-loop perception pipeline that is well-suited for active perception algorithms.

Building upon this infrastructure, the RLHF-based viewpoint planning strategy presented in this extension demonstrates how the same reconstruction system can be leveraged for automated viewpoint selection. Instead of relying solely on manual teleoperation, the robotic arm can learn exploration policies that prioritize task-relevant viewpoints based on expert operator feedback.

Although the proposed framework enables interactive second-level reconstruction updates, fully converged high-fidelity neural scene reconstruction during continuous robotic operation remains computationally demanding with current hardware and rendering pipelines. As a result, the embodied robotic platform in this thesis primarily serves as a motivating and representative system context for studying timeliness-aware perception and communication strategies.

These results suggest that the embodied monocular 3D scene representation platform proposed in this chapter can serve as a general experimental testbed for studying active perception in robotic 3D scene representation. Such capability is particularly valuable in inspection scenarios, where robotic systems must balance reconstruction quality, exploration efficiency, and operator-specific visualization requirements.

# Chapter 5

# The Timeliness–Fidelity Tradeoff in Multicam Telepresense

## 5.1 Motivation and Scope

Real-time 3D scene representation is a system-level capability that enables robotic operation in dynamic and networked environments. Its usefulness depends not only on geometric or photometric fidelity, but also on temporal validity: a high-quality reconstruction becomes ineffective if it is based on stale observations. Classical multi-view reconstruction and neural rendering pipelines are typically developed under two assumptions: (i) the scene is static over the acquisition window and (ii) sufficient computation time is available. Robotic perception violates both assumptions. Scenes evolve due to robot motion, object dynamics, and human interaction, while perception outputs are consumed immediately by downstream modules such as teleoperation interfaces, planners, and controllers.

This chapter studies the resulting timeliness-fidelity tension in networked 3D scene representations. Using only the most recent observations improves responsiveness but reduces multi-view coverage and may lead to underconstrained reconstructions. Conversely, waiting for additional views improves geometric constraints but increases temporal misalignment, which can introduce artefacts in dynamic scenes. Rather than proposing a new reconstruction method, we develop a communication-aware analytical framework that isolates how stochastic transmission delays and freshness-based selection policies affect reconstruction quality.

The analysis in this chapter provides a baseline understanding of the timeliness–fidelity tradeoff under simplified but principled assumptions, and serves as the foundation for Chapter 6, where task objectives and semantic relevance are incorporated into adaptive scheduling. The main contributions of this chapter are threefold.

**1) Formalisation of the timeliness–fidelity tradeoff in real-time 3D scene representation.** We identify and explicitly formulate the fundamental tension between information freshness and reconstruction quality in networked robotic perception systems. While classical 3D scene repre-

sentation pipelines focus primarily on fidelity, this chapter introduces timeliness as a first-class objective and establishes a unified perspective for analysing real-time 3D scene representations under communication delays.

**2) AoI-Aware 3D scene representation sensor system modelling.** We develop a principled framework in which multiple cameras transmit observations over stochastic communication channels to an edge server for reconstruction. The scheduling problem is formulated using AoI as a measure of data freshness, enabling a unified description of update delays, waiting strategies, and reconstruction quality. This model captures the essential dynamics of networked real-time 3D scene representation while remaining analytically tractable.

**3) Tradeoff analysis and learning-based scheduling strategies.** Within the proposed framework, we investigate both threshold-based strategies and reinforcement learning-based scheduling, revealing a non-trivial relationship between update frequency, waiting time, and reconstruction quality. The results demonstrate that aggressive updates are not always optimal, and that appropriately designed waiting strategies can significantly improve overall scene representation performance.

## 5.2 From Age of Information to Timeliness-Aware 3D Scene Representation

### 5.2.1 Prerequisite on the Age of Information

Classical communication system design has long relied on delay-centric metrics such as packet latency, throughput, and packet loss rate to characterize network performance. These metrics are well-suited for applications in which each packet is equally valuable and in which outdated information does not fundamentally alter system behavior. However, for systems that operate in dynamic environments—such as remote monitoring, control, and sensing—these traditional metrics are insufficient. In such systems, the timing of information delivery is often more critical than the successful delivery of every packet. An update that arrives too late may be rendered useless, regardless of its accuracy.

The AoI was originally introduced to characterize information freshness in networked monitoring and control systems, where the system state evolves continuously over time, and outdated information may lose its utility even if delivered reliably. It is formulated [15] as a metric that captures the time elapsed since the most recently received update was generated, rather than the transmission delay of individual packets.

Consider a source that generates time-stamped updates, the AoI $\Delta(t)$ this sensor in the $t$-th time slot is defined as:

$$\Delta(t) = t - U(t), \tag{5.1}$$

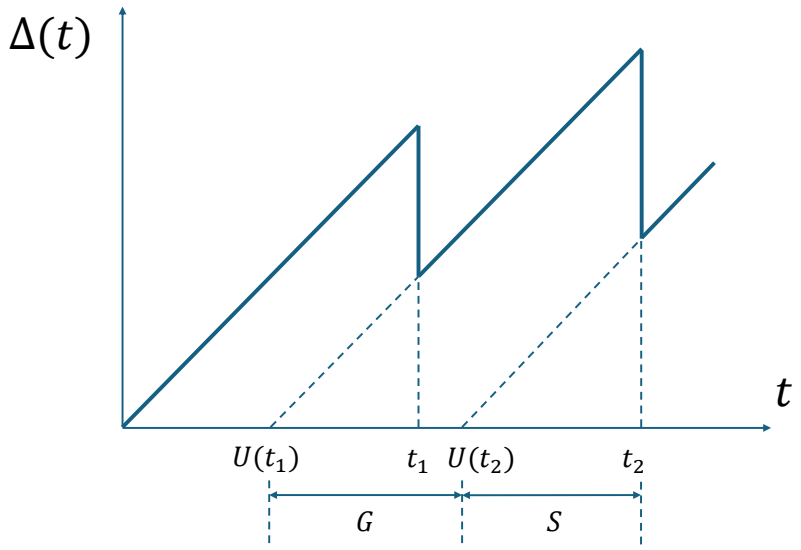



Figure 5.1: Illustration of the Age of Information (AoI) evolution over time.

where $U(t)$ is the time of generation of the latest update that arrived at the $t$-th time slot.

The temporal evolution of AoI over time is illustrated in Fig. 5.1. Between two consecutive update receptions, the AoI increases linearly with unit slope, reflecting the fact that the information available at the receiver becomes progressively older. When a new update generated at time $U(t_i)$ is successfully received at time $t_i$, the AoI drops instantaneously to $t_i - U(t_i)$, which corresponds to the transmission delay of that update. Importantly, the reset value of AoI depends on the generation time of the received update rather than on the reception time itself. As a result, delayed reception of stale updates leads to a higher AoI after the reset, highlighting the distinction between information freshness and packet delay.

### 5.2.2 Discrete-Time AoI Dynamics and Stochastic Update Models

In many practical systems, as shown in Fig. 5.2, including slotted wireless networks and digital sensing pipelines, time is naturally discretized. Specially in the $(t+1)$-th time slot, the AoI process $\{\Delta_n(t+1)\}$ of the $n$-th sensor then evolves according to

$$\Delta_n(t+1) = \begin{cases} t+1-U(t+1), & \text{if an update is received at } t+1, \\ \Delta_n(t)+1, & \text{otherwise.} \end{cases} \tag{5.2}$$

This piecewise-linear evolution highlights that AoI is fundamentally a stochastic process whose dynamics are jointly determined by the update generation mechanism, the service process of the communication system, and the scheduling policy. In contrast to classical delay analysis, where packets are treated independently, AoI couples the temporal statistics of packet generation with the queueing behavior of the network.

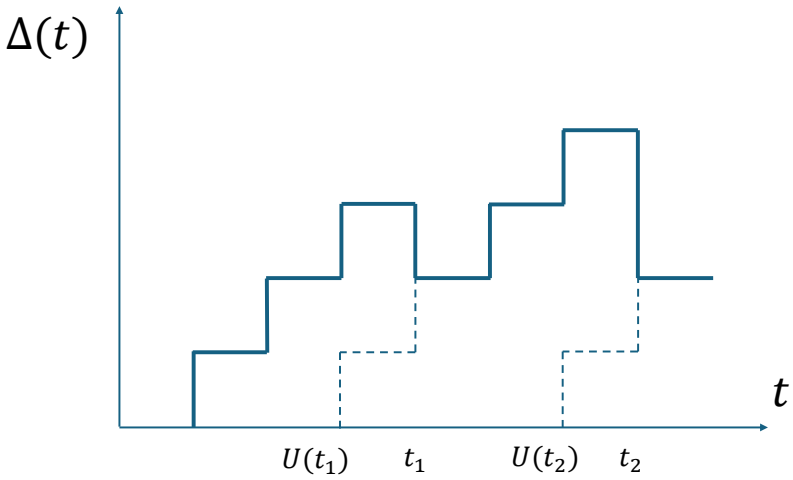



Figure 5.2: The stair-step function of AoI evolution over time.

From a queueing-theoretic perspective, AoI depends on two interacting components: the inter-generation time of updates and the service time distribution of the communication system. Let $G_k$ denote the inter-generation times between consecutive updates and $S_k$ denote the corresponding service times. Even when updates are generated periodically, i.e., $G_k = G$ for all $k$, randomness in the service times $S_k$, caused by channel fading, contention, or retransmissions—induces stochasticity in the AoI process. Conversely, when service times are deterministic, but update generation is stochastic, randomness in $G_k$ alone is sufficient to create variability in AoI trajectories.

Early AoI studies characterized the interaction between update generation and transmission using classical queueing models, most notably variants of the M/M/1 system [15, 20]. It is often beneficial to interpret these models through a two-stage buffering abstraction, which can be formalized as an M/M/1/1 queueing system. Under this formulation, update arrivals at the source are modeled as a Poisson process with rate $\lambda$, while the service process at the receiver is modeled as an exponential random variable with rate $\mu$. The first stage represents buffering at the source side, where generated updates may accumulate when the downstream server is occupied. The second stage represents a single-server system at the receiver side, encompassing transmission, reception, and any subsequent processing required before the update becomes available for decision-making. Within the M/M/1/1 framework, updates generated during periods in which the receiver-side server is busy cannot be immediately served and are therefore delayed at the source. As a consequence, the update that eventually reaches the receiver may have been generated significantly earlier than its reception time. From an AoI perspective, this distinction is critical: although packets are successfully delivered, their generation timestamps may be substantially outdated, leading to large AoI values upon reception.

This two-stage interpretation clarifies the fundamental difference between packet delay and

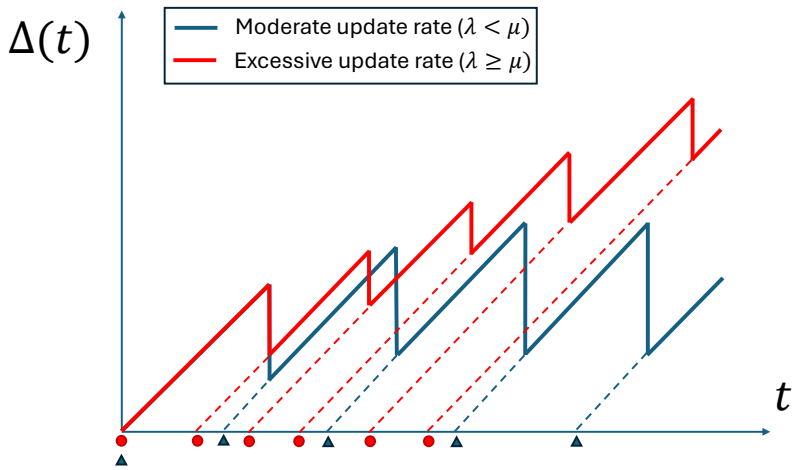



Figure 5.3: The stair-step function of AoI evolution over time.

information freshness. Even when the average system delay remains moderate, queueing at the source can cause stale updates to accumulate. During extended receiver-side busy periods induced by long service times, fresher updates are blocked from entering service, allowing the AoI to increase linearly until the server becomes available again. The AoI reset upon reception therefore reflects the age of the update that survives the queueing process, rather than the most recently generated information. Extensions to M/D/1/1 and D/M/1/1 systems further isolate the effects of update generation and service time variability. In M/D/1/1 systems, randomness arises primarily from the update arrival process, whereas in D/M/1/1 systems, updates are generated periodically but experience stochastic service times. Analytical results under these models demonstrate that deterministic update generation alone is insufficient to guarantee low AoI when service times exhibit significant variability.

The freshness of an update depends not only on the mean service time $\mathbb{E}[S]$, but also on higher-order moments of the service time distribution, such as $\mathbb{E}[S^2]$ [15]. Service time distributions with high variance or heavy-tailed behavior can substantially increase average AoI $\bar{\Delta}_n$, even when average packet delay remains bounded. This effect arises because prolonged service times effectively block the receiver-side server, preventing the timely delivery of fresher updates. This queueing-theoretic perspective motivates AoI-aware queue management strategies, including packet replacement and last-come-first-served disciplines with preemption, which prioritize fresh updates and suppress stale information [15, 20]. Such mechanisms are particularly relevant for perception systems, in which both transmission and downstream processing introduce non-negligible and highly variable service times.

Increasing the update generation rate does not necessarily reduce the resulting AoI. This non-monotonic behavior has been rigorously established in a variety of queueing models, including M/M/1, M/D/1, and related systems [16, 17]. As illustrated in Fig. 5.3, consider a single-source

system in which updates are generated according to a Poisson process with rate $\lambda$ and served by a single server with service rate $\mu$. When $\lambda$ approaches or exceeds $\mu$, the system becomes congested, and newly generated updates must wait behind previously generated ones. As a consequence, the update that is eventually delivered to the receiver may have been generated significantly earlier than its reception time. During such busy periods, the AoI increases linearly until service resumes, resulting in large AoI values despite high transmission activity.

Analytical expressions for the average AoI in such systems reveal that AoI depends not only on the update generation rate $\lambda$, but also on the interaction between $\lambda$ and the service process. In particular, the average AoI of the $n$-th sensor typically admits a form

$$\bar{\Delta}_n = g(\lambda, \mu), \tag{5.3}$$

where $g(\cdot)$ is not monotonically decreasing in $\lambda$. As a result, the AoI-minimizing update rate is often strictly smaller than the maximum feasible generation rate, implying that excessive update generation can be detrimental to information freshness. To mitigate this effect, several works have investigated queue management and service discipline design for AoI optimization. Last-come-first-served (LCFS) queues with preemption have been shown to significantly reduce AoI by prioritizing the service of fresh updates and suppressing stale ones [15, 20]. Similarly, packet dropping and update skipping strategies can improve AoI performance by preventing obsolete updates from occupying the service queue. These results demonstrate that AoI optimization is inherently tied to queueing dynamics and service discipline, rather than being a simple consequence of average delay reduction. In multi-source systems, queueing effects become even more pronounced. When multiple update streams share a common server, the AoI of each source depends not only on its own generation and service processes but also on the scheduling decisions that determine which source is served at each time slot. This coupling leads to complex tradeoffs between fairness and freshness, and has motivated the development of AoI-aware scheduling policies that explicitly account for queue states and service time distributions [18, 19].

Overall, queueing-theoretic analysis reveals that AoI captures a fundamentally different aspect of system performance than classical delay metrics. By explicitly linking update generation, service time variability, and scheduling decisions, AoI provides a principled framework for reasoning about information freshness in stochastic networked systems. These insights are critical when extending AoI-aware modeling to perception pipelines, where visual updates are high-dimensional, costly to transmit, and subject to heterogeneous service times. A commonly studied model assumes that updates are generated according to a renewal process and transmitted over a channel with random service times. Under such models, AoI exhibits a sawtooth pattern whose statistical properties depend on both the update rate and the service time distribution. Closed-form expressions for average AoI have been derived for various queueing systems, revealing non-trivial tradeoffs between update frequency and information freshness [16, 20].

Importantly, these results demonstrate that increasing the update rate does not always reduce

AoI. Beyond a certain point, excessive updates may congest the system and increase waiting times, leading to higher AoI. This phenomenon already suggests that timeliness optimization is fundamentally coupled with resource constraints.

Early studies demonstrated that minimizing average packet delay or maximizing throughput does not necessarily minimize $\bar{\Delta}_n$ [16, 17]. In particular, transmitting outdated packets aggressively may reduce delay while increasing AoI, as fresher updates are delayed or dropped. This mismatch reveals a fundamental limitation of delay-centric metrics when applied to freshness-sensitive systems.

Unlike latency, which measures the time taken by a specific packet to traverse the network, AoI measures the time elapsed since the generation of the most recently received update. As a result, AoI focuses more on the freshness of the latest update and captures both transmission delay and update frequency in a unified manner. Subsequent works have shown that minimizing average delay or maximizing throughput does not necessarily minimize AoI [16, 17]. In particular, aggressively transmitting outdated packets can reduce throughput delay while increasing AoI, revealing a structural mismatch between delay-centric and freshness-centric objectives. The AoI was proposed to address a limitation in traditional communication metrics when applied to decision-centric systems. In remote estimation and control, the system performance depends on the freshness of the state information used by the controller, rather than on how fast individual packets are delivered [18, 20].

Consider a stochastic process $x(t)$ evolving over time, observed remotely via sampled updates. If the estimator at the receiver uses the most recently received sample $x(U(t))$, then the estimation error depends explicitly on $\Delta(t)$. Under mild assumptions on the dynamics of the process, the expected estimation error can be shown to be an increasing function of AoI [16]. This observation motivated a large body of work on AoI-aware scheduling, including:threshold-based update policies [16, 19],pull-based and query-driven sensing architectures [21], AoI–energy and AoI–throughput tradeoffs [20]. However, in almost all of these works, the content of the information is abstracted as a scalar or low-dimensional vector, and fidelity is measured using metrics such as MSE or control cost.

### 5.2.3 AoI-aware sensor fusion

Consider a fusion centre receiving visual observations from $N$ spatially distributed sensors, such as cameras mounted on different robotic platforms or viewpoints. Each sensor $n \in \{1, \ldots, N\}$ acquires observations over time and generates a sequence of measurements $\{\tau_i^n\}_{i=1}^{\infty}$, where $\tau_i^n$ denotes the content of the $i$-th measurement (e.g., an image, feature tensor, or latent representation). The $i$-th measurement from sensor $n$ is generated at its local generation time $S_i^n$ and is received by the fusion centre at time $D_i^n$, with $D_i^n \geq S_i^n$. Due to communication effects such as variable transmission delays, queueing, interference, packet loss, or retransmissions, measurements are not instantaneously available at the fusion centre.

At a given fusion time $t$, the fusion centre may therefore only have access to a subset of previously generated measurements, and the available measurements from different sensors may correspond to different generation times. To characterise this temporal asynchrony, we define, for each sensor $n$, the generation time of the most recently received measurement as

$$u_n(t) := \max\{S_i^n \mid D_i^n \leq t\}. \tag{5.4}$$

The instantaneous Age of Information (AoI) for sensor $n$ at fusion time $t$ is then given by

$$\Delta_n(t) := t - u_n(t), \tag{5.5}$$

which quantifies the staleness of the freshest measurement available from that sensor. The fusion centre operates on a temporally inconsistent dataset

$$\Omega(t) = \left\{(\tilde{\tau}_n(t), u_n(t))\right\}_{n=1}^{N}, \tag{5.6}$$

where $\tilde{\tau}_n(t)$ denotes the most recently received measurement from sensor $n$ available at fusion time $t$, and $u_n(t)$ is its corresponding generation time, defined in eq. 5.4.

The dataset $\Omega(t)$ therefore represents the complete perceptual input available to the fusion centre in the $t$-th time slot, given the underlying sensing, communication, and scheduling processes.

Crucially, while each individual measurement $\tilde{\tau}_n(t)$ may be accurate with respect to the true scene state at its own generation time $u_n(t)$, the collection $\Omega(t)$ is generally temporally heterogeneous. That is, the measurements in $\Omega(t)$ correspond to different physical time instants $\{u_n(t)\}_{n=1}^{N}$, which may be separated by non-negligible and sensor-dependent delays. As a result, $\Omega(t)$ does not, in general, correspond to any single consistent snapshot of the underlying environment.

This temporal inconsistency is not merely a bookkeeping artefact, but a fundamental property of networked perception systems. In the presence of communication constraints, the fusion centre must inevitably reason over a mixture of fresh and stale observations, even when all sensors are individually accurate and well calibrated. Consequently, perception errors may arise not only from sensing noise or modelling inaccuracies, but also from the misalignment of observations in time. From a modelling perspective, $\Omega(t)$ therefore captures both the content of the available measurements and their temporal validity. The generation times $\{u_n(t)\}$ implicitly encode the AoI profile $\{\Delta_n(t)\}$ of the system, which governs how representative the dataset is of the current scene state. Any perception or reconstruction algorithm operating on $\Omega(t)$ must thus contend with the trade-off between exploiting a larger set of observations and ensuring their temporal relevance.

In the context of high-dimensional visual perception, such as 3D scene reconstruction, this

issue becomes particularly pronounced. Geometric consistency across views assumes that observations correspond to a shared underlying scene configuration; however, temporal misalignment in $\Omega(t)$ can violate this assumption, leading to artefacts such as ghosting, blurred geometry, or inconsistent surface estimates. These effects motivate the explicit incorporation of information freshness into the perception and fusion process, rather than treating temporal inconsistency as an incidental implementation detail.

We abstract the perception and fusion process by defining a generic fusion operator parameterised by $\theta$,

$$\hat{\mathbf{s}}(t) = \mathscr{F}_\theta\big(\Omega(t)\big), \tag{5.7}$$

where $\hat{\mathbf{s}}(t)$ denotes the fused estimate produced at fusion time $t$. Depending on the application, $\hat{\mathbf{s}}(t)$ may represent a low-dimensional system state, a dense geometric map, or the parameters of a high-dimensional neural scene representation. This abstraction encompasses both classical estimators and modern neural perception pipelines, with the key distinction that the input to $\mathscr{F}_\theta$ is explicitly indexed by the Age of Information through the temporally inconsistent dataset $\Omega(t)$.

As a consequence, the quality of the fused estimate $\hat{\mathbf{s}}(t)$ depends not only on the sensing noise characteristics of the individual measurements $\tilde{\tau}_n(t)$, but also on the freshness profile $\{\Delta_n(t)\}_{n=1}^N$ induced by the underlying communication and scheduling policies. This perspective elevates timeliness to a first-class system variable in perception, rather than treating it as a secondary implementation detail.

A timeliness-aware fusion objective can then be formulated by jointly optimising the fusion operator $\mathscr{F}_\theta$ and the communication or scheduling policy $\pi$:

$$\min_{\pi,\theta} \mathbb{E}\left[\sum_{t=1}^{T} \underbrace{\ell\big(\mathscr{F}_\theta(\Omega(t)), \mathbf{s}(t)\big)}_{\text{fidelity}} + \lambda \underbrace{\sum_{n=1}^{N} \Delta_n(t)}_{\text{timeliness}}\right], \tag{5.8}$$

where $\mathbf{s}(t)$ denotes the (unknown) ground-truth system or scene state at time $t$, $\ell(\cdot,\cdot)$ is a task-dependent loss (e.g., mean-squared error for state estimation or photometric reconstruction loss for 3D perception), and $\pi$ specifies the sensor scheduling or communication policy subject to resource constraints such as bandwidth, power, or channel contention. The trade-off parameter $\lambda > 0$ controls the relative importance of estimation fidelity versus information freshness.

This objective explicitly exposes the interaction between estimation fidelity and information timeliness. When the timeliness term is removed (i.e., $\lambda = 0$), the optimisation reduces to a conventional fidelity-driven perception problem, in which the fusion operator $\mathscr{F}_\theta$ is trained to minimise reconstruction or estimation error given the available data, without regard to the freshness of the underlying observations. In this regime, performance is governed solely by the statistical properties of the measurements and the expressive capacity of the perception model, implicitly assuming that all observations are equally relevant regardless of their age. How-

ever, in networked sensing scenarios, maximising fidelity alone does not necessarily yield the best real-time performance. Aggressively incorporating all available observations can increase communication load and induce congestion, leading to longer delays and, consequently, higher Age of Information. As a result, the dataset $\Omega(t)$ may contain increasingly stale measurements, which can degrade the effective quality of supervision even if the nominal reconstruction loss continues to decrease.

Introducing the timeliness penalty in the objective fundamentally alters this behaviour. The additional AoI $\Delta_n(t)$ term encourages policies that balance the benefit of incorporating more observations against the cost of operating on outdated information. Rather than favouring maximal update rates, the optimal policy may selectively delay or suppress transmissions to prevent excessive staleness, thereby improving the overall relevance of the data used for fusion. This leads to a non-trivial trade-off: in certain regimes, accepting a modest increase in instantaneous estimation error can yield a substantial reduction in information staleness, resulting in improved downstream perception quality and responsiveness. From an experimental perspective, this formulation naturally motivates a comparative evaluation along two axes. First, by varying $\lambda$, one can characterise the intrinsic trade-off between fidelity and timeliness and identify operating regimes in which each dominates. Second, by comparing the proposed timeliness-aware objective against a fidelity-only baseline, it becomes possible to isolate the impact of explicitly modelling information freshness on both estimation accuracy and temporal consistency. Such comparisons are essential for understanding whether performance gains arise from improved perception modelling alone or from the joint optimisation of perception and communication.

## 5.3 System Model and Temporal Structure

### 5.3.1 Overview

We consider a networked robotic perception system that integrates sensing, wireless communication, and 3D scene representation at the edge. As illustrated in Fig. 5.4, a team of autonomous robots (e.g., UAVs, UGVs, or quadruped platforms) equipped with onboard cameras observes a dynamic environment from multiple viewpoints. Each robot captures 2D visual observations of a scene or object of interest (e.g., a wind turbine or industrial asset), which are required to be fused into a coherent 3D scene representation for downstream tasks such as monitoring, inspection, or teleoperation.

Due to limited onboard computation and energy constraints, raw visual data are transmitted over a wireless uplink to an edge server, where perception and 3D scene representation is performed. The edge server has access to the camera poses associated with each observation, which are assumed to be either transmitted reliably or obtained through a separate localisation and state-estimation module. As a result, the edge server acts as a fusion centre that jointly

processes visual observations and pose information received from multiple robots.

The communication model considered in this chapter follows a relatively simplified stochastic update abstraction commonly adopted in AoI and networked sensing literature. Similar time-slotted and stochastic update formulations have been widely used to analyse information freshness under communication uncertainty [15]. The objective here is not to replicate a specific wireless protocol stack, but rather to isolate and analyse the fundamental relationship between observation freshness and 3D reconstruction fidelity under controlled communication dynamics.

### 5.3.2 Time-Slotted Sensing and Communication Model

As shown in Fig. 5.4, time is discretised into slots indexed by $t \in \{1,2,\dots\}$, with each slot having a fixed duration $T_s$. In the system, $N$ cameras indexed by $n \in \{1,\dots,N\}$ periodically generate visual observations. Specifically, each camera generates one image every $C$ time slots, resulting in a sequence of images indexed by $i = 1,2,\dots$. We denote the $i$-th image captured by camera $n$ as $\tau_i^n$, which represents the measurement content (e.g., an RGB image or a feature tensor). The image $\tau_i^n$ is generated at time slot $S_i^n$ and is transmitted to the edge server over a wireless uplink. Due to stochastic channel conditions, including fading and interference, the transmission duration of each image is random. Let $D_i^n$ denote the time slot at which the transmission of $\tau_i^n$ is completed and the image becomes available at the edge server. The transmission delay is therefore $Y_i^n = D_i^n - S_i^n$. We consider an orthogonal frequency-division multiple access (OFDMA) uplink with $M$ orthogonal subchannels, where $N = M$ cameras are simultaneously served. If the transmission of a previously generated image has not completed, a newly generated image is not transmitted and is discarded.

The communication process considered in this chapter follows standard queueing-theoretic abstractions commonly used in AoI analysis and networked sensing systems. In Kendall's queueing notation, a queue is represented in the form A/S/c/K, where A denotes the packet arrival process, S denotes the service-time distribution, c is the number of servers, and K represents the system capacity. For example, M corresponds to a Markovian (Poisson) arrival process, D denotes deterministic service time, and G denotes a general service-time distribution. In this chapter, the communication process is modelled using a D/G/1/0-style abstraction, where updates are generated periodically and deterministically while the transmission duration follows a general stochastic process. The system contains a single communication server and no waiting buffer, such that stale observations may be discarded when the server is occupied. This abstraction is adopted to isolate the fundamental timeliness–fidelity tradeoff under controlled communication dynamics.

The above model intentionally abstracts several system aspects to isolate the timeliness–fidelity mechanism. We assume periodic sensing with a drop-new update rule (D/G/1/0) to avoid queue build-up and to make AoI evolution explicit, and we model transmission delays as i.i.d. random variables to capture stochastic wireless effects in a tractable manner. Camera poses

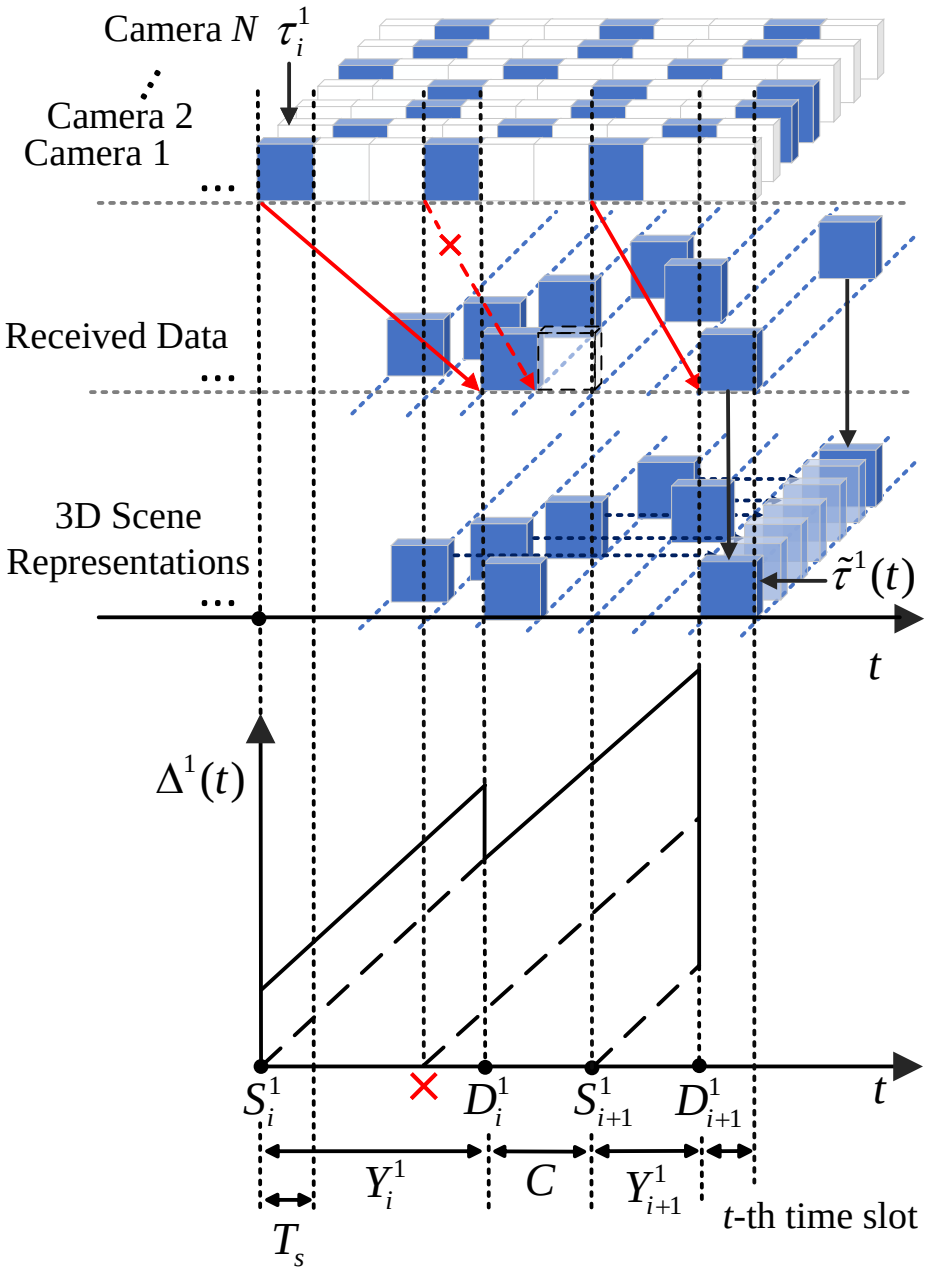



Figure 5.4: Time Sequence Diagram.

are assumed to be available at the fusion centre through a reliable localisation module. These assumptions are used to establish a baseline tradeoff analysis in this chapter; Chapter 6 relaxes the setting by considering more complex and task-dependent communication dynamics.

In the $t$-th time slot, the most recently received image from camera $n$ is the one whose generation time satisfies eq. 5.4. The AoI of camera $n$ at time slot $t$ is defined by eq. 5.5, which quantifies the staleness of the freshest image available from that camera at the edge server. We denote by $\tilde{\tau}_n(t)$ the most recently received image from camera $n$ at time slot $t$, i.e., the image generated at time $u_n(t)$. The set of visual observations available to the edge server at time $t$ is therefore given by eq. 5.6.

Although each individual image $\tilde{\tau}_n(t)$ accurately reflects the scene at its own generation time $u_n(t)$, the dataset $\Omega(t)$ is generally temporally inconsistent. The images in $\Omega(t)$ may correspond to different physical time instants, resulting in a mixture of fresh and stale observations. This temporal heterogeneity is an inherent property of networked sensing systems and cannot be eliminated by perception algorithms alone.

Based on the available dataset $\Omega(t)$, the edge server performs 3D scene representation by fusing a selected subset of the received images, denoted by $\Omega_{\text{sel}}(t)$. We abstract the perception and fusion process by a generic operator parameterised by $\theta$,

$$\hat{\mathbf{S}}(t) = \mathscr{F}_\theta\big(\Omega_{\text{sel}}(t), \mathbf{P}(t)\big), \tag{5.9}$$

where $\hat{\mathbf{S}}(t)$ denotes the estimated scene representation at time $t$, and $\mathbf{P}(t)$ denotes the set of camera poses corresponding to the selected images. Depending on the application, $\hat{\mathbf{S}}(t)$ may represent a dense map, a radiance field, or a Gaussian-based 3D representation.

Not all available images need to be used for 3D scene representation. We introduce a binary decision variable $w_n(t) \in \{0,1\}$, where $w_n(t) = 1$ indicates that the image $\tilde{\tau}_n(t)$ from camera $n$ is used for scene representation at time $t$, and $w_n(t) = 0$ otherwise. The decision vector at time $t$ is denoted by

$$\omega(t) = \big[w_1(t), w_2(t), \ldots, w_N(t)\big]. \tag{5.10}$$

The set of images used for 3D scene representation at time $t$ is then given by

$$\mathscr{D}_s(t) = \{\tilde{\tau}_n(t) \mid w_n(t) = 1, n = 1, \ldots, N\}, \tag{5.11}$$

with the corresponding set of camera poses

$$\mathbf{P}(t) = \{\mathbf{p}^n(t) \mid w_n(t) = 1, n = 1, \ldots, N\}. \tag{5.12}$$

For completeness, we express a generic timeliness-aware objective as in eq. 5.8.

The objective in eq. (**??**) is written in a generic form to indicate that system performance depends jointly on reconstruction fidelity and information freshness. In this chapter, however,

timeliness is treated implicitly: the scheduler observes the AoI profile as the state and optimises 3D scene representation quality measured by PSNR/SSIM/LPIPS. Under temporally inconsistent supervision, stale observations typically manifest as geometric misalignment and perceptual artefacts (e.g., ghosting), which are directly penalised by these fidelity metrics. Therefore, the effect of $\sum_n \Delta_n(t)$ is captured indirectly through its impact on view synthesis quality, without introducing an explicit AoI penalty in the reward. In Chapter 6, we move from this implicit coupling to an explicit task-oriented formulation by incorporating freshness (and semantic/task relevance) as a direct penalty term in the learning objective.

### 5.3.3 3D Scene Representations

At the edge server, the received multi-view visual observations are fused to construct a 3D representation of the underlying scene. In this work, we adopt NeRF [22] as the scene representation backbone, owing to its ability to model continuous geometry and view-dependent appearance from sparse image observations.

In the $t$-th time slot, the fusion centre operates on the temporally inconsistent observation set $\Omega(t)$ defined in this chapter. Based on the subset of images selected for reconstruction, the corresponding camera poses are collected into $\mathbf{P}(t)$. NeRF represents a scene as a continuous volumetric radiance field parameterised by a multilayer perceptron (MLP). Formally, the neural network $\mathscr{F}_\theta$ maps a 3D spatial location and a viewing direction to a volume density and emitted radiance,

$$\mathscr{F}_\theta(\mathbf{x},\mathbf{d}) = \big(\sigma_\theta(\mathbf{x}),\, \mathbf{c}_\theta(\mathbf{x},\mathbf{d})\big), \tag{5.13}$$

where $\mathbf{x} \in \mathbb{R}^3$ denotes a point in world coordinates, $\mathbf{d} \in \mathbb{S}^2$ denotes the viewing direction, $\sigma_\theta(\mathbf{x}) \geq 0$ is the volume density, and $\mathbf{c}_\theta(\mathbf{x},\mathbf{d}) \in \mathbb{R}^3$ is the view-dependent RGB radiance.

Let the $t$-th RGB image be a discrete sampling of scene radiance on a pixel grid,

$$I_t : \{1,\dots,W\} \times \{1,\dots,H\} \to [0,1]^3, \qquad I_t[u,v] \in \mathbb{R}^3, \tag{5.14}$$

where $(u,v)$ denotes pixel coordinates. As discussed in Fig. 3 and Section II, each pixel $(u,v)$ uniquely induces a camera ray $\mathbf{r}_t(u,v)$ through the calibrated camera model, establishing the correspondence

$$(u,v) \;\longleftrightarrow\; \mathbf{r}_t(u,v) \;\longleftrightarrow\; I_t[u,v].$$

Specifically, under the pinhole camera model with intrinsic matrix $\mathbf{K}$ and extrinsic parameters $(\mathbf{R},\mathbf{t}) \in \mathrm{SE}(3)$, the camera ray corresponding to pixel $\mathbf{p} = (u,v,1)^\top$ is given by

$$\mathbf{r}(s) = \mathbf{r}_o + s\,\mathbf{d}, \qquad s \in [s_n, s_f], \tag{5.15}$$

where the ray origin $\mathbf{r}_o$ and direction $\mathbf{d}$ are defined as in Fig. 3. This construction establishes a one-to-one correspondence between each image pixel and a continuous set of 3D query points

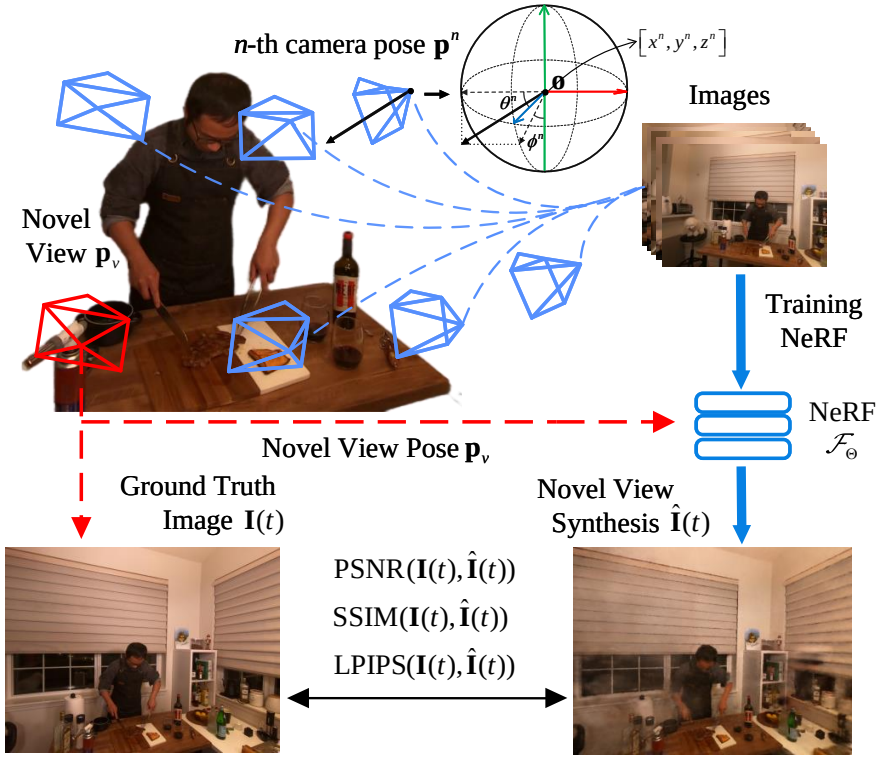



Figure 5.5: Illustration of the novel view synthesis evaluation protocol.

sampled along its associated viewing ray.

Given a camera ray $\mathbf{r}(s)$, the color of the corresponding pixel is obtained via differentiable volume rendering. Specifically, the rendered color is computed as in eq.(3.20). In practice, the rendering integral is approximated using stratified or hierarchical sampling along each ray, resulting in a fully differentiable image formation process. This property enables end-to-end optimisation of the scene representation parameters $\theta$ by minimising a photometric reconstruction loss between rendered and observed images.

In the $t$-th time slot, the scheduler determines the subset of observations that will be incorporated into the reconstruction process. The underlying 3D scene representation module operates in an incremental or periodically updated manner, performing multiple optimisation steps within a reconstruction cycle. The time slot therefore represents a scheduling decision instant rather than a strict wall-clock training step. We denote by $\hat{\mathbf{S}}(t)$ the current scene representation available at time $t$, which corresponds to the latest parameter state $\theta(t)$ of the neural field after the most recent reconstruction update:

$$\hat{\mathbf{S}}(t) \equiv \theta(t), \tag{5.16}$$

which emphasises that the latent scene state is implicitly encoded in the parameters of the neural radiance field. Importantly, the quality and temporal consistency of the reconstructed 3D scene

depend not only on the expressive power of the NeRF model, but also on the freshness of the observations used for supervision. Stale images in $\Omega(t)$ correspond to enforcing photometric consistency with respect to outdated scene states, which can degrade geometric accuracy or introduce artefacts in dynamic or communication-constrained settings. This observation motivates the timeliness-aware optimisation framework introduced in Section 5.4.

### 5.3.4 Performance Metrics

To evaluate the quality of the reconstructed 3D scene representations, we adopt a novel view synthesis protocol, which is widely used in neural scene representation literature [22,54]. Novel view synthesis refers to the process of rendering an image from a viewpoint that was not directly observed during data acquisition. As illustrated in Fig. 5.5, this evaluation paradigm measures how well the learned 3D representation generalises across viewpoints, rather than merely memorising observed images.

Formally, in the $t$-th time slot, given the 3D scene representation denoted by $\hat{\mathbf{S}}(t)$, a novel view image $\hat{I}(t)$ is rendered using the same volumetric rendering process described in Section. 3.1.2. The rendered image corresponds to a virtual camera pose that is withheld from the training or fusion set. By comparing $\hat{I}(t)$ with the corresponding ground-truth image $I(t)$, we quantitatively assess the fidelity of the reconstructed 3D scene. This evaluation protocol is particularly suitable for networked perception systems, as it captures both geometric consistency and view-dependent appearance. Unlike direct image reconstruction metrics on observed views, novel view synthesis penalises inconsistencies arising from temporally misaligned or stale observations in the fusion set $\Omega(t)$. To provide a more comprehensive evaluation of reconstruction fidelity, this work jointly considers PSNR, SSIM, and LPIPS, as these metrics capture complementary aspects of image quality as defined in eq. 3.31-3.34. In particular, PSNR primarily measures pixel-wise reconstruction accuracy, SSIM evaluates structural consistency, while LPIPS reflects perceptual similarity in deep feature space. Using all three metrics together enables a more robust assessment of both geometric consistency and perceptual realism under temporally stale or asynchronously fused observations.

## 5.4 Problem Formulation

We cast freshness-aware view selection as a stochastic decision problem at the edge server. At each time slot, the scheduler observes the current AoI profile of the received multi-view stream and decides which of the available images should be used for 3D scene representation. Importantly, the AoI evolution is primarily driven by exogenous communication processes (random transmission delays and packet arrivals) and is only weakly affected by the selection decision. This structure makes the problem closer to a contextual bandit with stochastic contexts than to a long-horizon control task. We therefore use policy-gradient optimisation (implemented via

a single-step variant of PPO) as a convenient stochastic policy optimiser, rather than as a full sequential control solution.

The scheduling problem considered in this work is difficult to formulate using conventional control-theoretic or optimisation-based approaches due to the high-dimensional coupling between observation freshness, semantic relevance, communication dynamics, and reconstruction quality. In particular, the relationship between transmitted observations and downstream 3D reconstruction fidelity is highly nonlinear and task-dependent, making it difficult to derive tractable analytical control policies. Reinforcement learning therefore provides a flexible framework for learning adaptive scheduling strategies directly from reconstruction-oriented reward signals without requiring explicit modelling of the underlying reconstruction dynamics. In particular, we adopt a policy-gradient-based approach, using PPO as the baseline algorithm due to its simplicity, stability, and sample efficiency. Unlike long-horizon control problems, the environment dynamics in our setting are dominated by one-step stochastic effects induced by communication delays and image availability. As a result, we employ a single-step variant of PPO, which allows the scheduler to react myopically to the current AoI profile while still optimising long-term expected performance.

### State

In the $t$-th time slot, the state of the RL agent is defined as the AoI vector of the $N$ cameras,

$$\mathbf{s}_t = \big[\Delta_1(t), \Delta_2(t), \ldots, \Delta_N(t)\big] \in \mathbb{R}_+^N. \tag{5.17}$$

This state representation compactly summarises the freshness profile of all available observations at the edge server. By construction, $\mathbf{s}_t$ captures the temporal validity of the observation set $\Omega(t)$, while remaining agnostic to the raw image content and camera poses. This abstraction enables the scheduler to reason about timeliness without directly operating on high-dimensional visual inputs.

### Action

The action taken at time slot $t$ corresponds to selecting which of the most recently received images are used for 3D scene representation. Specifically, the action is defined as

$$\mathbf{a}_t = \boldsymbol{\omega}(t) = \big[w_1(t), w_2(t), \ldots, w_N(t)\big], \tag{5.18}$$

where $w_n(t) \in \{0,1\}$ indicates whether the image $\tilde{\tau}_n(t)$ from camera $n$ is incorporated into the scene reconstruction at time $t$.

Through this binary selection mechanism, the scheduler explicitly trades off between incorporating a larger number of views and prioritising fresh observations. The resulting action

directly determines the subset of the observation set

$$\Omega_{\text{sel}}(t) = \big\{(\tilde{\tau}_n(t), u_n(t)) \mid w_n(t) = 1, n = 1, \ldots, N\big\}.$$

that is passed to the scene representation operator $\mathscr{F}_\theta$.

### Reward

Given the state $\mathbf{s}_t$ and action $\mathbf{a}_t$ in the $t$-th time slot, the instantaneous reward is defined as a weighted combination of novel view synthesis quality metrics,

$$r(\mathbf{s}_t, \mathbf{a}_t) = w_1 \,\text{PSNR}\big(I(t), \hat{I}(t)\big) + w_2 \,\text{SSIM}\big(I(t), \hat{I}(t)\big) + w_3 \,\text{LPIPS}\big(I(t), \hat{I}(t)\big), \tag{5.19}$$

where $\hat{I}(t)$ is the image rendered from the reconstructed 3D scene using the selected observation subset, and $I(t)$ denotes the corresponding ground-truth image. We note that LPIPS is a distance metric where smaller values indicate better perceptual similarity; hence $w_3$ is set to a negative value in our experiments.

The weights $w_1, w_2, w_3$ control the relative importance of pixel-level fidelity, structural consistency, and perceptual similarity. This reward design directly aligns the scheduler's objective with the evaluation metrics introduced in Section 5.3.4, ensuring that scheduling decisions are directly guided by their impact on downstream 3D perception quality rather than including the abstract communication metrics.

### Policy

The policy $\pi_\phi$ maps the state $\mathbf{s}_t$ to a distribution over actions $\mathbf{a}_t$. For each camera $n$, the policy outputs a Bernoulli distribution parameterised by

$$\rho_t^n = \begin{pmatrix} \Pr\big(w_n(t) = 1 \mid \mathbf{s}_t\big) \\ \Pr\big(w_n(t) = 0 \mid \mathbf{s}_t\big) \end{pmatrix}, \tag{5.20}$$

and the joint action distribution is given by the product over all cameras. The policy is represented by a neural network $\pi_\phi(\mathbf{s}_t)$ with parameters $\phi$. Following policy $\pi_\phi$, the expected long-term return is

$$R^{\pi_\phi} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right], \tag{5.21}$$

where $\gamma \in (0, 1]$ is the discount factor.

The optimal policy is obtained by maximising the expected return,

$$\pi^* = \arg\max_{\pi_\phi} R^{\pi_\phi}. \tag{5.22}$$

**Single-Step PPO**

To optimise the policy parameters $\phi$, we employ a single-step variant of Proximal Policy Optimisation. At each time slot, the PPO objective is defined as

$$\mathscr{L}(\mathbf{s}_t, \mathbf{a}_t, \phi) = \min\left( \frac{\pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_{\phi_t}(\mathbf{a}_t \mid \mathbf{s}_t)} A^{\pi_{\phi_t}}(\mathbf{s}_t, \mathbf{a}_t),\ \mathrm{clip}\left( \frac{\pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_{\phi_t}(\mathbf{a}_t \mid \mathbf{s}_t)}, 1-\varepsilon, 1+\varepsilon \right) A^{\pi_{\phi_t}}(\mathbf{s}_t, \mathbf{a}_t) \right), \tag{5.23}$$

where $A^{\pi_{\phi_t}}(\mathbf{s}_t, \mathbf{a}_t)$ denotes the advantage function,

$$A^{\pi_{\phi_t}}(\mathbf{s}_t, \mathbf{a}_t) = Q^{\pi_{\phi_t}}(\mathbf{s}_t, \mathbf{a}_t) - V^{\pi_{\phi_t}}(\mathbf{s}_t), \tag{5.24}$$

and $V^{\pi_{\phi_t}}(\mathbf{s}_t)$ is the state-value function.

At each time slot, the scheduler observes the current AoI state $\mathbf{s}_t$, samples an action $\mathbf{a}_t$ according to the policy, and receives an instantaneous reward based on the resulting 3D reconstruction quality. The policy parameters are then updated via gradient ascent on the PPO objective, yielding an updated scheduling policy that progressively learns to balance observation freshness and reconstruction fidelity.

Although reinforcement learning terminology is adopted throughout this chapter, the proposed formulation is more accurately characterised as a contextual bandit rather than a full Markov Decision Process (MDP). This is because the scheduling decision at each time step primarily depends on the current observation freshness, semantic context, and communication state, while the action itself does not significantly alter the long-term environment dynamics or future state transitions. Consequently, the framework focuses on learning an effective instantaneous scheduling policy under temporally varying sensing conditions rather than solving a long-horizon sequential control problem.

In particular, the state transition in our setting is primarily driven by exogenous communication processes, such as stochastic transmission delays and packet arrivals, rather than by the scheduler's own actions. The AoI state $\mathbf{s}_t$ evolves according to the update and delivery processes of the cameras, and the action $\mathbf{a}_t$ does not directly influence future observation arrivals beyond the current time slot. Consequently, the scheduler's decision at time $t$ mainly affects the immediate quality of the reconstructed scene, with limited impact on the future AoI evolution. This weak coupling between actions and long-term state dynamics motivates a myopic or single-step approximation of the decision process.

From an optimisation perspective, the reward defined in Section 5.3.4 already aggregates the effect of the selected observations through perceptual and structural metrics, which implicitly reflect the consequences of temporal misalignment. As a result, maximising the instantaneous reward aligns well with maximising the expected long-term performance, provided that the communication process is stationary. Under these conditions, a single-step policy-gradient update constitutes a reasonable and efficient approximation to full-horizon optimisation. We

**Algorithm 1** Single-Step PPO for AoI-Based View Selection (Chapter 5)

**Input:** Initial policy parameters $\phi_0$, discount factor $\gamma$, clipping parameter $\varepsilon$, training steps $T_t$, representation parameters $\alpha_\phi$, novel view pose $\mathbf{p}_v$.
**for** $t = 1, 2, \ldots, T_t$ **do**
    Observe AoI state:

$$\mathbf{s}_t = [\Delta_1(t), \ldots, \Delta_N(t)].$$

    Sample binary action vector

$$\mathbf{a}_t = [w_1(t), \ldots, w_N(t)] \sim \pi_{\phi_t}(\mathbf{s}_t).$$

    Construct observation subset

$$\Omega_{\text{sel}}(t) = \left\{ (\tilde{\tau}_n(t), u_n(t)) \mid w_n(t) = 1, n = 1, \ldots, N \right\}.$$

    Train scene representation $\mathscr{F}_\theta$ using $\Omega_{\text{sel}}(t)$.
    Render novel view $\hat{I}(t)$ from pose $\mathbf{p}_v$.
    Compute instantaneous reward:

$$r_t = w_1 \,\text{PSNR}(I(t), \hat{I}(t)) + w_2 \,\text{SSIM}(I(t), \hat{I}(t)) + w_3 \,\text{LPIPS}(I(t), \hat{I}(t)).$$

    Estimate advantage:

$$A_t = Q^{\pi_{\phi_t}}(\mathbf{s}_t, \mathbf{a}_t) - V^{\pi_{\phi_t}}(\mathbf{s}_t).$$

    Update policy parameters by maximising PPO objective:

$$\phi_{t+1} \leftarrow \phi_t + \nabla_\phi \mathscr{L}(\mathbf{s}_t, \mathbf{a}_t, \phi).$$

**end for**
**Output:** Learned policy $\pi^*_\phi$.

emphasise that the use of PPO in this context should be interpreted as a stochastic policy optimisation method rather than a full sequential control solution. The proposed formulation strikes a balance between modelling fidelity and computational tractability, enabling effective learning-based scheduling without introducing unnecessary complexity. Although the formulation resembles a contextual bandit, we retain the policy-gradient framework to accommodate potential extensions where scheduling actions may influence longer-term communication dynamics.

## 5.5 Experiment Setup

To systematically evaluate the proposed timeliness-aware scheduling framework for 3D scene representation, we conduct simulations on the DyNeRF dataset, which is widely adopted in neural scene representation and novel view synthesis research [34]. The dataset provides multi-view dynamic scenes captured by 19 synchronized cameras at 30 FPS over a duration of 10 seconds. For each scene, we select 18 cameras for 3D scene representation (i.e., $N = 18$) and

reserve the remaining camera as a held-out novel view for evaluation. This split ensures that the reconstructed neural representation is assessed under a genuine novel view synthesis protocol, consistent with the evaluation pipeline illustrated in Fig. 3.6.

To emulate networked sensing conditions, the generation interval of each camera is set to $C = 30$ ms, corresponding to the original video frame rate. The transmission delay $Y_i^n$ of each image follows an exponential distribution with mean 60 ms, modelling stochastic wireless channel conditions. The time slot duration is set to $T_s = 1$ ms, enabling fine-grained AoI evolution and scheduling decisions. These parameters are chosen to induce non-trivial temporal misalignment across cameras, resulting in heterogeneous AoI profiles. Under this setting, some cameras may provide fresh observations, while others may experience delayed updates, thereby creating a meaningful trade-off between incorporating more views and prioritising fresh data. The choice of $C = 30$ ms reflects a typical camera sampling rate (approximately 30–33 Hz), while an average transmission delay of 60 ms is representative for wireless uplinks with contention and processing overhead. Importantly, the regime $\mathbb{E}[Y] > C$ deliberately emulates bandwidth-limited conditions in which not every generated frame can be delivered in time. This produces heterogeneous AoI profiles across cameras and yields a meaningful timeliness–fidelity trade-off, which is the focus of our study.

The reward discount factor in the single-step PPO algorithm is set to $\gamma = 1$, reflecting the weak coupling between scheduling actions and future AoI evolution discussed in Section 5.4. Under this formulation, the scheduler primarily optimises instantaneous reconstruction quality while implicitly accounting for timeliness through the AoI-dependent state. The policy network is implemented as a lightweight multilayer perceptron that maps the AoI state vector $\mathbf{s}_t \in \mathbb{R}^N$ to Bernoulli probabilities for each camera. Training is performed over multiple simulated communication episodes to ensure convergence under stochastic delay conditions.

To investigate the trade-off between timeliness and fidelity across different neural scene representation paradigms, we evaluate the proposed scheduler on three representative NeRF-based methods: Instant-NGP, TensoRF, and Nerfacto.

**Instant Neural Graphics Primitives (Instant-NGP)** Instant-NGP [52] leverages a multi-resolution hash encoding and a lightweight neural network to represent volumetric radiance fields. Its highly optimised CUDA implementation enables rapid training and real-time rendering, making it suitable for evaluating timeliness-aware decision-making under near-real-time constraints.

**TensoRF** TensoRF [56] replaces the fully connected neural network in classical NeRF with a low-rank tensor decomposition of the volumetric field. By factorising the feature grid into compact tensor components, TensoRF significantly reduces memory consumption and training time, while maintaining competitive reconstruction quality. This representation provides a com-

plementary trade-off between efficiency and fidelity compared to Instant-NGP.

**Nerfacto** Nerfacto [57] integrates camera pose refinement and per-image appearance conditioning to enhance reconstruction quality. Although implemented in Python and generally slower than CUDA-optimised Instant-NGP, Nerfacto often achieves strong perceptual performance. Including Nerfacto allows us to evaluate whether timeliness-aware scheduling benefits high-quality but computationally heavier representations.

For each representation backbone, we train the neural model using the subset of images selected by the scheduler in each time slot. Novel view images are rendered from the held-out camera pose and compared with the corresponding ground-truth images using PSNR, SSIM, and LPIPS. To characterise the trade-off between timeliness and fidelity, we vary the reward weights associated with the three metrics and analyse the resulting performance under different AoI profiles. This allows us to quantify how scheduling decisions influence both pixel-level accuracy and perceptual consistency across representation paradigms.

## 5.6 Performance Evaluation

In this section, we evaluate the proposed timeliness-aware scheduling framework from two complementary perspectives. First, in Section 5.6.1, we analyse the intrinsic timeliness–fidelity trade-off using a simple threshold-based baseline. This provides an interpretable reference for understanding how information freshness influences reconstruction quality. Second, in Section 5.6.2, we evaluate the proposed single-step PPO scheduler and demonstrate its ability to adaptively optimise 3D scene representation performance beyond fixed heuristics.

### 5.6.1 Timeliness–Fidelity Trade-off with a Threshold-Based Method

To explicitly reveal the trade-off between information freshness and reconstruction fidelity, we first consider a simple threshold-based scheduling strategy. Let $\Gamma$ denote the maximum allowable AoI threshold (MAT). At time slot $t$, the image from camera $n$ is used for 3D scene representation if and only if

$$\Delta_n(t) \leq \Gamma.$$

Otherwise, the image is discarded. This strategy effectively filters observations based on freshness: smaller $\Gamma$ prioritises highly recent views but reduces the number of available cameras, whereas larger $\Gamma$ increases view diversity at the expense of temporal consistency. By varying $\Gamma \in [0, 120]$ ms, we systematically explore how the freshness constraint influences novel view synthesis quality. The upper bound of 120 ms is chosen to cover approximately twice the mean transmission delay, ensuring that both freshness-dominated and staleness-dominated regimes are included. For each threshold value, we conduct 40 independent simulations under stochastic

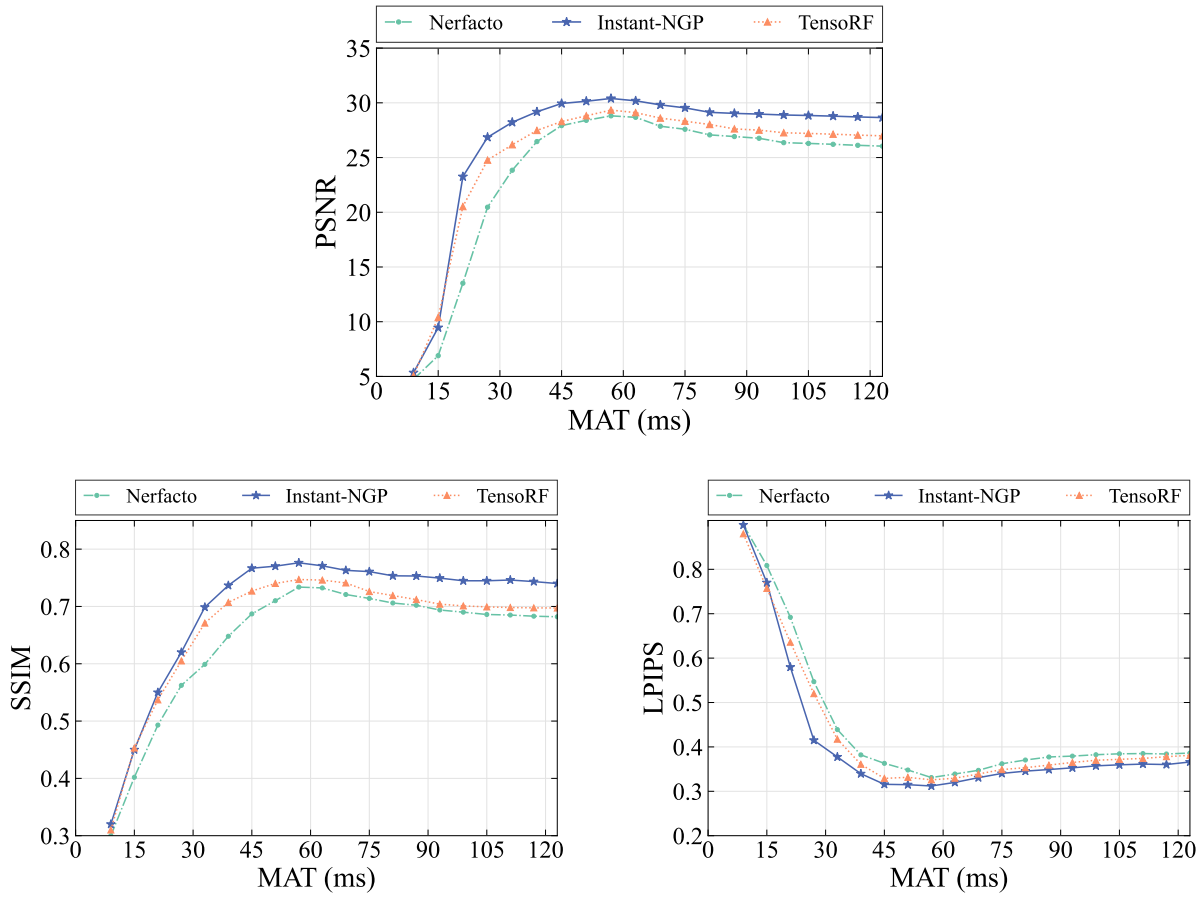



Figure 5.6: Timeliness–fidelity tradeoff under three performance metrics: PSNR, SSIM, and LPIPS.

transmission delays and compute the average PSNR, SSIM, and LPIPS over the rendered novel views. The results, shown in Fig. 5.6, illustrate a clear timeliness–fidelity trade-off.

When $\Gamma$ is too small, the number of usable observations becomes insufficient for stable multi-view reconstruction, leading to degraded geometry consistency and reduced rendering quality. Conversely, when $\Gamma$ is too large, temporally stale images are included in the fusion set, introducing misalignment across views and causing ghosting artefacts or structural inconsistencies. As a result, reconstruction performance first improves and then deteriorates as $\Gamma$ increases, revealing a non-monotonic relationship between timeliness and fidelity. This behaviour confirms that maximising raw update availability (i.e., using all received images) does not necessarily yield optimal 3D scene representation quality. Instead, an appropriate balance between observation freshness and view diversity must be achieved. The threshold-based method provides a simple but interpretable approximation of this balance and serves as a reference baseline for the learning-based scheduler.

Figure 5.7 presents representative novel view synthesis results under different MAT values. Visually, the reconstruction quality varies significantly as the freshness constraint changes. When the MAT is excessively large (e.g., $\Gamma = 120$ ms), a larger number of temporally stale

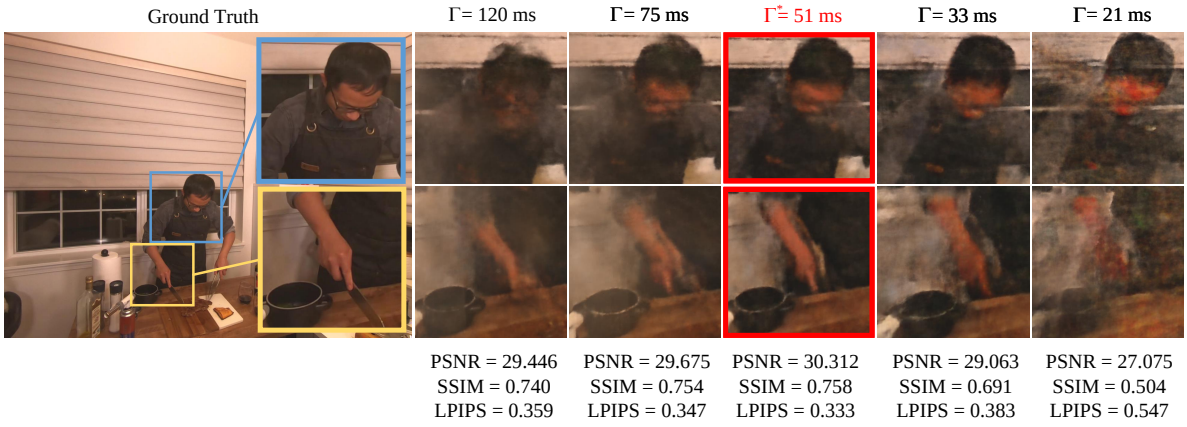



Figure 5.7: Qualitative and quantitative comparison of novel view synthesis results under different maximum AoI thresholds (MATs).

observations are included in the fusion set, leading to noticeable ghosting and structural inconsistencies caused by view misalignment. Although more views are used, the temporal heterogeneity degrades the perceptual quality of the rendered image. In contrast, when the MAT is too small (e.g., $\Gamma = 21$ ms), only highly recent observations are retained. While this improves temporal synchronisation with the ground truth, the limited number of views results in insufficient geometric constraints, producing blurred structures and reduced fidelity. An intermediate threshold, around $\Gamma^* = 51$ ms, achieves the best perceptual and quantitative performance. At this point, the scheduler retains enough views to ensure geometric consistency while filtering out excessively stale observations. This confirms the existence of a non-monotonic relationship between timeliness and reconstruction fidelity.

Importantly, this behaviour demonstrates that neither maximising freshness alone nor maximising view availability alone is optimal. Instead, high-quality 3D scene reconstruction requires balancing temporal consistency and multi-view coverage, which motivates the need for adaptive scheduling strategies.

Figure 5.8 illustrates the relationship between the optimal maximum AoI threshold (MAT) $\Gamma^*$ and the expected transmission delay $\mathbb{E}[Y_i^n]$. For each delay regime, the optimal MAT is obtained by exhaustively searching over threshold values and selecting the one that maximises novel view synthesis performance. The optimal MAT exhibits an empirically monotonic increasing trend with respect to $\mathbb{E}[Y_i^n]$. When the average transmission delay is small, most observations arrive with low staleness, and a strict freshness constraint (i.e., small MAT) is sufficient to maintain both temporal consistency and adequate multi-view coverage. However, as $\mathbb{E}[Y_i^n]$ increases, the AoI distribution shifts towards larger values, and enforcing a small MAT would discard a substantial portion of available views. In such regimes, relaxing the MAT becomes necessary to preserve sufficient geometric constraints for stable real-time 3D scene representations. Consequently, the optimal MAT increases with worsening channel conditions.

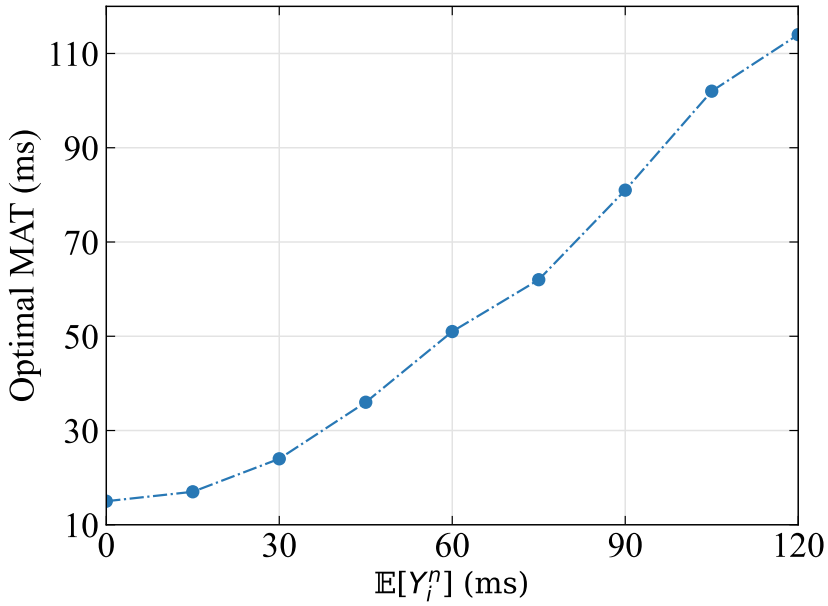



Figure 5.8: The relationship between optimal MAT $\Gamma$ and expected value of transmission duration $\mathbb{E}[Y_i^n]$.

The monotonic trend observed in Fig. 5.8 further motivates the use of an adaptive scheduling strategy. Since the optimal MAT varies with channel conditions, a fixed-threshold method cannot achieve uniformly optimal performance. This observation provides empirical support for the learning-based scheduler proposed in Section 5.6.2.

### 5.6.2 3D Scene Representations with Single-Step PPO Scheduler

In this subsection, we evaluate the proposed single-step PPO scheduler using Instant-NGP as the underlying NeRF backbone. The reward weights are set to $w_1 = 0.02$, $w_2 = 0.5$, and $w_3 = -1$, reflecting a balanced emphasis on pixel-level fidelity, structural consistency, and perceptual similarity. The policy is trained for 600 episodes under stochastic transmission delays. Figure 5.9 shows the evolution of the episode reward during training. Despite the inherent variance induced by the random delay process, the reward exhibits a clear increasing trend and stabilises after approximately 400 episodes, indicating convergence of the scheduling policy. The weighting coefficients in the reward function are primarily introduced to balance the numerical sensitivity and dynamic range differences between PSNR, SSIM, and LPIPS, rather than to encode task-specific preference priorities. Since these metrics operate on substantially different numerical scales and exhibit different variation magnitudes during reconstruction updates, direct unweighted aggregation may cause the optimisation process to become dominated by a single metric. The adopted weighting strategy therefore aims to maintain comparable reward sensitivity across perceptual, structural, and pixel-level reconstruction quality measures. In practice, the weights were selected empirically to produce stable reward variation during training while avoiding excessive

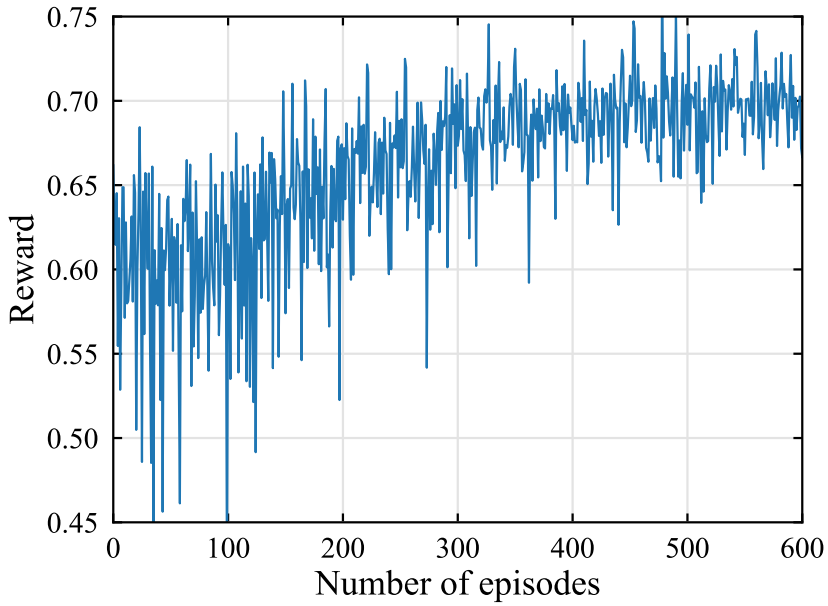



Figure 5.9: Evolution of the reinforcement learning reward during training with the single-step PPO scheduler.

dominance from any individual reconstruction metric. A more comprehensive sensitivity analysis over different weighting configurations may provide additional insight into the robustness and interpretability of the proposed reward formulation, and therefore remains an interesting direction for future investigation.

The reinforcement-learning-based scheduler required approximately 30 minutes of offline training on an NVIDIA RTX 4090 GPU before convergence. At convergence, the learned policy achieves an average training performance of PSNR = 30.54, SSIM = 0.775, and LPIPS = 0.342. On unseen testing episodes, the policy attains PSNR = 30.12, SSIM = 0.779, and LPIPS = 0.350. Although a fully random scheduling baseline was not explicitly implemented, the early-stage training behaviour provides an approximate reference for highly exploratory scheduling behaviour prior to policy convergence. During this stage, the reward exhibits substantial fluctuation within the approximate range of 0.45–0.68, reflecting unstable observation selection under exploration-dominated policy behaviour. As training progresses, the reinforcement-learning-based scheduler gradually converges toward more stable timeliness-aware scheduling policies with consistently improved average reward values. Nevertheless, this exploratory phase should not be interpreted as a formally controlled uniform-random baseline, and a dedicated random scheduling comparison remains an interesting direction for future investigation. The random scheduling behaviour corresponds to highly unstable reward fluctuations within the approximate range of 0.45–0.68, with an average reward of approximately 0.60. In comparison, the learned reinforcement-learning-based scheduler converges to a substantially more stable policy with an average reward of approximately 0.70, corresponding to an improvement of around 16.7%. The small performance gap between training and testing indicates good generalisation

under varying communication conditions. Compared with the fixed-threshold baseline analysed in Section 5.6.1, the learned scheduler adaptively adjusts the effective freshness constraint according to the instantaneous AoI profile, thereby achieving consistently high reconstruction quality without manual tuning of MAT parameters. Despite noticeable reward fluctuation during the early exploration stage, the overall reward trend gradually increases as training progresses, indicating stable policy improvement over time. The average reward increases from approximately 0.60 during the initial training stage to around 0.70 after convergence, corresponding to an improvement of approximately 16.7%. The relatively high short-term variance is primarily caused by stochastic communication dynamics and exploration behaviour during policy learning, while the later training stage exhibits improved stability and consistently higher reward values. These results suggest that the proposed RL-based scheduling strategy is able to progressively learn more effective timeliness-aware observation selection policies under the considered communication setting.

## 5.7 Conclusion

In this chapter, we established a unified framework for analysing the tradeoff between timeliness and fidelity in real-time 3D scene representation under communication constraints. By explicitly incorporating Age of Information (AoI) into the perception pipeline, we demonstrated that reconstruction quality is fundamentally influenced by temporal misalignment across multi-view observations.

Through threshold-based analysis, we revealed a non-monotonic relationship between observation freshness and novel view synthesis performance. Specifically, neither maximising freshness nor maximising view availability alone yields optimal reconstruction quality. Instead, an intermediate balance between temporal consistency and geometric coverage is required. Furthermore, we showed that the optimal freshness threshold is not a fixed constant, but increases monotonically with the expected transmission delay, highlighting the dependency of perception performance on communication statistics. Building upon this insight, we formulated the scheduling problem as a reinforcement learning task and proposed a single-step PPO-based scheduler. The learned policy successfully adapts to stochastic delay conditions and consistently achieves high reconstruction quality without manual tuning of threshold parameters. Experimental results across multiple neural scene representation backbones validate the effectiveness and generality of the proposed timeliness-aware scheduling framework.

Overall, this work demonstrates that timeliness should be treated as a first-class system variable in edge-assisted 3D scene representation. The proposed framework provides a principled foundation for integrating communication dynamics and neural scene reconstruction in future real-time robotic and edge-based perception systems.

# Chapter 6

# Task-Oriented Communications for 3D Scene Representation for Multi-robot Telepresense

## 6.1 Motivation

Chapter 5 established the fundamental tradeoff between timeliness and fidelity in real-time 3D scene representation systems. Under stochastic communication delays, simply using the freshest available data does not necessarily lead to the best reconstruction quality, and waiting for additional observations may improve fidelity at the cost of increased information staleness. This analysis highlights the intrinsic tension between communication latency and perception accuracy in networked 3D scene representation systems. However, the formulation in Chapter 5 adopts a simplified system model, where scheduling decisions are primarily driven by the AoI of incoming data streams. In practical robotic perception scenarios, not all observations are equally important. Different parts of the scene may have different semantic relevance and update requirements. In such settings, communication and perception decisions should not be based solely on timeliness metrics. Instead, they must be aligned with the underlying task objectives, prioritizing information that contributes most to the final decision or reconstruction outcome. This motivates a shift from purely timeliness-driven scheduling to task-oriented communication strategies, where both temporal freshness and semantic relevance are considered. Moreover, real-world wireless environments often exhibit bursty or time-varying channel conditions. Fixed threshold or single-step waiting strategies become suboptimal under such settings, as the system must adaptively adjust its waiting horizon based on both communication states and task requirements.

Fig. 6.1 illustrates a representative edge-assisted industrial perception scenario considered in this chapter, where heterogeneous robotic platforms transmit sensing observations to edge infrastructure for timeliness-aware 3D scene representation. The figure is intended primarily as

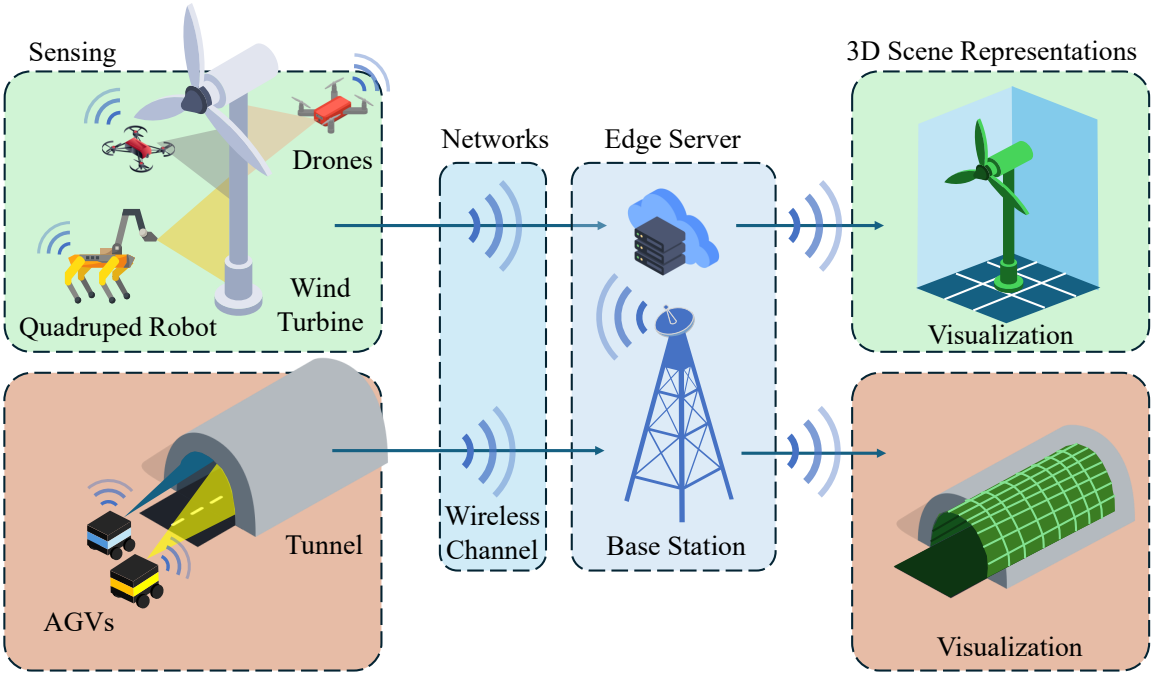



Figure 6.1: System model of edge-assisted 3D scene reconstruction, where heterogeneous sensors (e.g., drones, quadruped robots, and AGVs) capture data from industrial environments such as wind turbines and tunnels. The sensed data are transmitted via wireless channels to the edge server through a base station, and then processed for real-time 3D scene visualization.

a conceptual application framework demonstrating the potential deployment context of the proposed system, rather than a fully implemented end-to-end experimental platform used directly in the reported evaluations. It should be noted that Fig. 6.1 illustrates multiple representative industrial application scenarios for the proposed edge-assisted 3D scene representation framework, rather than a simultaneously deployed multi-workload system. In the current formulation, the communication-aware reconstruction framework assumes a single sensing and reconstruction workload at a time under a shared edge-assisted pipeline. Consequently, the wind-turbine and tunnel inspection examples are intended primarily to demonstrate the potential applicability of the framework across heterogeneous robotic perception scenarios rather than concurrent system operation.

To address these challenges, this chapter proposes a task-oriented communication framework for real-time 3D scene representation. The scheduling problem is formulated as a contextual decision-making process, where the system state incorporates both timeliness indicators and task-related features. An RL-based policy is then used to learn adaptive waiting strategies that balance information freshness and reconstruction fidelity under dynamic communication conditions. The main contributions of this chapter are threefold.

**1) Timeliness-aware communication with semantic context.** We propose a cross-system scheduling formulation that augments AoI-driven decision-making with semantic context extracted from incoming image streams. Rather than relying solely on freshness indicators, the

scheduler conditions its waiting/selection decisions on lightweight pretrained visual embeddings, enabling prioritisation of visually salient and dynamically changing observations in a practical, modular manner.

**2) Contextual-bandit RL strategy for adaptive waiting.** To solve the resulting scheduling problem, we formulate it as a contextual decision-making process and develop a reinforcement learning approach based on a contextual-bandit variant of PPO. Unlike the formulation in Chapter 5, where the state is defined solely by AoI variables, the proposed policy incorporates both timeliness indicators and semantic features extracted from the data streams. This richer state representation enables the agent to dynamically adjust the waiting horizon according to both communication conditions and task priorities. As a result, the learned policy achieves improved reconstruction fidelity while maintaining low information staleness across diverse operating conditions.

**3) Cross-system design bridging communication and 3D perception.** This chapter demonstrates a task-oriented design paradigm that explicitly links communication scheduling decisions to downstream 3D perception performance. Instead of treating communication and perception as separate modules, the proposed framework optimises them jointly with respect to task-level objectives. Extensive experiments across multiple datasets and scene representation models, including neural radiance fields and 3D Gaussian Splatting, show that the proposed method consistently outperforms conventional threshold-based or fixed waiting strategies. These results highlight the importance of semantic-aware scheduling and provide practical insights for designing integrated communication–perception systems in real-time robotic applications.

This formulation extends the tradeoff analysis in Chapter 5 from a purely AoI-driven perspective to a task-aware, cross-system communication–perception design, enabling more efficient and robust real-time 3D scene modelling.

## 6.2 System Model and Scheduling Structure

### 6.2.1 Overview

Building upon the timeliness-aware perception framework developed in Chapter 5.2.3, this chapter investigates how concrete scheduling mechanisms influence real-time 3D scene representation under communication constraints. While the previous chapter formulated a general optimisation objective that couples reconstruction fidelity and information freshness, the present chapter focuses on the operational layer: how scheduling decisions reshape the temporal structure of the data used for 3D scene representations.

While Chapter 5 considers relatively stationary stochastic communication dynamics in order to isolate the fundamental timeliness–fidelity tradeoff, practical robotic sensing systems often exhibit more heterogeneous and bursty traffic patterns. In telepresence and multi-sensor robotic

perception scenarios, sensor updates may become highly event-driven due to robot motion, scene changes, or task-triggered sensing demands. Similar asynchronous and partial-arrival sensing behaviour has been studied in wireless sensor network literature, where temporally heterogeneous update arrivals were shown to significantly influence the achievable timeliness–fidelity tradeoff [37]. Motivated by these observations, this chapter extends the communication setting toward more realistic burst-aware sensing dynamics.

We consider the same multi-robot perception architecture introduced previously. Multiple robots equipped with calibrated cameras continuously observe a scene and transmit visual measurements to an edge server for reconstruction. As discussed in Chapter 5.2.3, stochastic wireless delays introduce temporal heterogeneity in the received multi-view dataset. At any given time slot, the edge server typically holds a mixture of fresh and stale observations originating from different generation instants. This temporal inconsistency cannot be corrected by the perception backbone alone; instead, it must be addressed at the scheduling layer.

In contrast to the abstract policy formulation in Chapter 5, where the scheduling decision was represented generically through a policy $\pi$, this chapter explicitly characterises how different scheduling mechanisms determine the subset of images and camera poses used for reconstruction at each time slot. In other words, we make explicit how the binary decision vector $\omega(t)$ shapes the supervision set of the neural 3D representation.

We study two representative paradigms:

- **$\omega$-threshold policy**: reconstruction proceeds at every time slot, but only frames whose AoI lies within a predefined freshness window are admitted;
- **$\omega$-wait policy**: reconstruction is deferred until sufficiently recent frames from all cameras are available, thereby enforcing stronger temporal alignment.

These two mechanisms embody distinct philosophies of timeliness-aware reconstruction. The threshold policy prioritises responsiveness by operating on whatever sufficiently fresh data are currently available, whereas the wait policy prioritises temporal coherence at the cost of potential reconstruction latency. Their structural differences lead to fundamentally different temporal statistics of the supervision set, which in turn influence both geometric fidelity and dynamic consistency of the reconstructed scene.

### 6.2.2 Temporal Observation Set and Pose Selection

We adopt the time-slotted model introduced in Chapter 5.2.3. Let $\tilde{\tau}_n(t)$ denote the most recently received image from the $n$-th camera at time slot $t$, generated at time $u_n(t)$.

The set of the latest available observations at the edge server is defined in eq. 5.6.

Each image $\tilde{\tau}_n(t)$ is associated with a camera pose that specifies the spatial configuration of the sensing platform at the time of image acquisition. We denote the pose of camera $n$ at time $t$

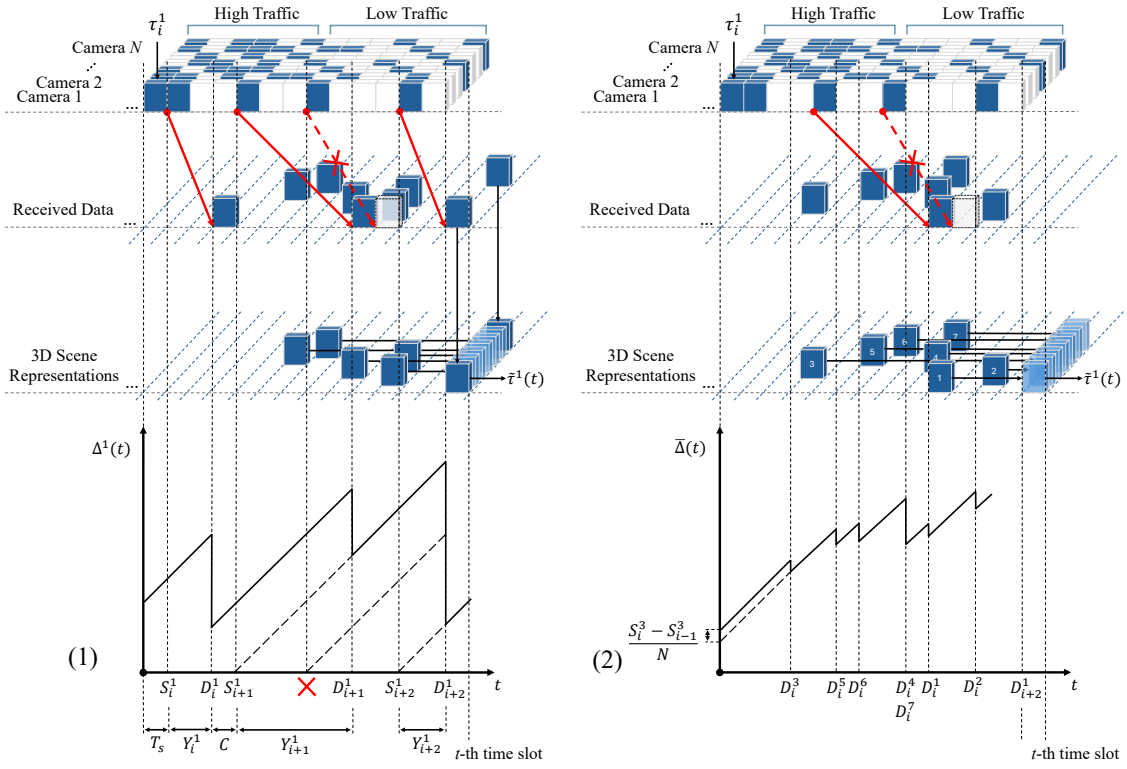



Figure 6.2: Time-sequence diagrams for real-time dynamic 3D scene representation from multi-sensor image streaming. The figure illustrates two baseline scheduling policies: (1) the $\omega$-threshold policy, where the scheduler includes the most recent images from each camera in the training set only if their current AoI is below a global threshold $\omega_t$, and (2) the $\omega$-wait policy, where the scheduler postpones rendering for $\omega_t$ slots, incorporating only the updates that arrive during this waiting horizon.

by

$$\mathbf{p}^n(t) = \left[x^n(t), y^n(t), z^n(t), \theta^n(t), \phi^n(t)\right] \in \mathbb{R}^{1\times 5}, \tag{6.1}$$

where $(x^n(t), y^n(t), z^n(t))$ denotes the 3D position of the camera in the world coordinate frame, and $(\theta^n(t), \phi^n(t))$ parameterise its viewing orientation. More generally, the pose can be equivalently represented as a rigid-body transformation $(\mathbf{R}^n(t), \mathbf{t}^n(t)) \in \mathrm{SE}(3)$, where $\mathbf{R}^n(t) \in \mathrm{SO}(3)$ is the rotation matrix and $\mathbf{t}^n(t) \in \mathbb{R}^3$ is the translation vector. The pose determines the mapping from image pixels to 3D rays in the world coordinate system. Specifically, for each pixel in $\tilde{\tau}_n(t)$, the corresponding camera ray is uniquely defined by the intrinsic matrix and the extrinsic parameters encoded in $\mathbf{p}^n(t)$.

The camera poses establish the multi-view correspondence required for triangulation, volumetric integration, or neural rendering, and directly determine how individual observations contribute to the fused 3D scene representation.

Let the global pose collection be denoted as

$$\mathbf{P}(t) = \left[\mathbf{p}^1(t), \mathbf{p}^2(t), \ldots, \mathbf{p}^N(t)\right]. \tag{6.2}$$

Not all images stored in the receiving buffer are necessarily used for reconstruction at time slot $t$. The edge server maintains, for each camera $n$, the most recently received update $\tilde{\tau}_n(t)$ together with its associated pose $\mathbf{p}^n(t)$. The scheduler determines which of these buffered updates participate in the reconstruction process.

Let

$$\omega(t) = [w_1(t), \ldots, w_N(t)] \tag{6.3}$$

denote the scheduling decision at time slot $t$, where $w_n(t) \in \{0,1\}$ indicates whether the most recent image from camera $n$ is admitted into the reconstruction pipeline.

Under this decision, the effective supervision set used for 3D scene representation at time $t$ becomes

$$\mathscr{D}_s(t) = \{\tilde{\tau}_n(t) \mid w_n(t) = 1\}, \tag{6.4}$$

and the corresponding pose set is

$$\mathbf{P}_s(t) = \{\mathbf{p}^n(t) \mid w_n(t) = 1\}. \tag{6.5}$$

The number of selected views at time slot $t$ is therefore

$$K_t = \sum_{n=1}^{N} w_n(t). \tag{6.6}$$

In this formulation, the scheduling decision does not alter the perception backbone itself, but explicitly reshapes the multi-view supervision structure. Depending on $\omega(t)$, the 3D scene

representation algorithm may operate on a full $N$-view batch, a reduced subset of cameras, or, in extreme cases, a single-view update.

The 3D scene representation operator is instantiated as

$$\hat{\mathbf{S}}(t) = \mathscr{F}_\theta\big(\mathscr{D}_s(t), \mathbf{P}_s(t)\big), \tag{6.7}$$

where $\mathscr{F}_\theta$ denotes the perception backbone introduced previously. In practice, the representation parameters $\theta$ are updated incrementally over time. Therefore, $\mathscr{F}_\theta(\cdot)$ should be interpreted as the current instance of the 3D scene representation model in the $t$-th time slot.

### 6.2.3 Scheduler Agent

We consider a multi-camera wireless perception system where $N$ cameras asynchronously transmit image updates to an edge server for real-time 3D scene representation.

At each decision epoch $t$, the scheduler agent determines which information will be used for reconstruction. Instead of treating scheduling as a purely communication-layer decision, we model it as an information selection problem tightly coupled with scene representation quality.

Let

$$\mathbb{I}(t) \subseteq \widetilde{\mathscr{T}}(t) \tag{6.8}$$

denote the selected image set used for reconstruction at time $t$, where $\widetilde{\mathscr{T}}(t)$ denotes the set of buffered images (ignoring timestamps), i.e., $\widetilde{\mathscr{T}}(t) = \{\tilde{\tau}_n(t)\}_{n=1}^N$.

For any image used in reconstruction, we define an information unit

$$I_n(t) \in \mathbb{I}(t), \quad n = 1, \ldots, N, \tag{6.9}$$

corresponding to the latest buffered observation from camera $n$ admitted at time $t$. Each information unit is associated with the attributes

$$I_n(t) \triangleq \big(\Delta^n(t), \mathbf{p}^n(t), \mathbf{z}^n(t)\big), \tag{6.10}$$

where $\Delta^n(t)$ is its AoI, $\mathbf{p}^n(t)$ denotes the camera pose, and $\mathbf{z}^n(t)$ represents the extracted semantic embedding.

The scheduler agent implements a decision function

$$\mathbb{I}(t) = \mathscr{F}_s(\mathbf{s}_t), \tag{6.11}$$

where $\mathbf{s}_t$ is the system state composed of freshness, pose distribution, and semantic features.

### $\omega$-Threshold Policy

Under the $\omega$-threshold policy, the scheduler selects images based solely on instantaneous freshness. At time slot $t$, a global threshold $\omega_t$ is determined.

An image from camera $n$ is admitted into the reconstruction set if

$$\Delta^n(t) \leq \omega_t. \tag{6.12}$$

According to Eq. 6.3, the scheduling decision variable is defined as

$$w_n(t) = \begin{cases} 1, & \Delta^n(t) \leq \omega_t, \\ 0, & \text{otherwise.} \end{cases} \tag{6.13}$$

Thus, the reconstruction dataset at time $t$ is

$$\mathbb{I}_1(t) = \{I_n(t) \mid w_n(t) = 1,\ 1 \leq n \leq N\}. \tag{6.14}$$

This policy enforces a per-slot freshness constraint while maintaining continuous rendering. When wireless delay increases, some cameras may be excluded due to excessive AoI, reducing multi-view diversity at that time slot.

### $\omega$-Wait Policy

In contrast, the $\omega$-wait policy adopts a horizon-based strategy. At time slot $t$, the scheduler determines a waiting horizon $\omega_t$ and postpones reconstruction until time

$$t' = t + \omega_t. \tag{6.15}$$

During this waiting interval, newly arriving image updates are accumulated in the receiving buffer.

Let $I_n(\tau)$ denote an image update from camera $n$ arriving at time $\tau$. The decision variable under this policy is defined as

$$w_n(\tau) = \begin{cases} 1, & t < \tau \leq t', \\ 0, & \text{otherwise.} \end{cases} \tag{6.16}$$

The reconstruction dataset triggered at time $t'$ is therefore

$$\mathbb{I}_2(t') = \{I_n(\tau) \mid w_n(\tau) = 1,\ 1 \leq n \leq N\}. \tag{6.17}$$

Unlike the threshold policy, reconstruction does not occur at every slot. Instead, temporally

clustered updates collected within the waiting horizon are jointly used for scene representation.

### 6.2.4 Timeliness Embedding Approach

While the $\omega$-threshold and $\omega$-wait policies operate purely along the temporal dimension, practical 3D scene representation quality depends on multiple coupled factors, including geometric coverage, viewpoint diversity, and semantic consistency. To capture these effects, we extend the scheduler from a freshness-only decision rule to an information-aware embedding framework.

At decision epoch $t$, the system state is represented as

$$\mathbf{s}_t = \left[\Delta^1(t), \ldots, \Delta^N(t), \mathbf{p}^1(t), \ldots, \mathbf{p}^N(t), \mathbf{z}^1(t), \ldots, \mathbf{z}^N(t)\right], \tag{6.18}$$

where $\Delta^n(t)$ denotes the AoI of the $n$-th camera, $\mathbf{p}^n(t)$ denotes the camera pose, $\mathbf{z}^n(t)$ represents semantic or structural embeddings extracted from the latest received image.

Let $I_n(t)$ denote the latest available image from camera $n$ at time $t$, and let the scheduler decision variable be $w_n(t) \in \{0,1\}$. The selected reconstruction set is

$$\mathbb{I}(t) = \{I_n(t) \mid w_n(t) = 1\}. \tag{6.19}$$

The key difference from purely temporal policies is that the decision function

$$w_n(t) = \mathscr{F}_s(\mathbf{s}_t) \tag{6.20}$$

is now driven by a joint representation of freshness, spatial alignment, and semantic relevance.

For any selected image $I_n(t) \in \mathbb{I}(t)$, its contribution to representation fidelity is measured by

$$M\big(I_n(t), \hat{I}_n(t)\big) = \sum_j w_j M_j\big(I_n(t), \hat{I}_n(t)\big), \tag{6.21}$$

where $M_j$ includes photometric or perceptual metrics such as PSNR, SSIM, and LPIPS, $\hat{I}_n(t)$ is the rendered image from the same camera pose using the current reconstructed scene, $w_j$ are weighting coefficients controlling the emphasis on different fidelity criteria.

The overall scene fidelity at time $t$ is then defined as

$$Q(t) = \frac{1}{|\mathbb{I}(t)|} \sum_{I_n(t) \in \mathbb{I}(t)} M\big(I_n(t), \hat{I}_n(t)\big). \tag{6.22}$$

Unlike the $\omega$-based policies, which impose hard constraints on temporal freshness, this embedding-based approach evaluates the marginal contribution of each image to reconstruction quality.

### 6.2.5 Network Model

Different from the D/G/1/0-style formulation adopted in Chapter 4, the burst-aware communication setting considered here can be interpreted using an M/D/1/1-type abstraction, where stochastic update arrivals and constrained service availability jointly produce temporally heterogeneous observation freshness.

We model the burstiness of packet arrivals in image transmission using a Markov modulated Poisson process (MMPP) [73], where network traffic transitions between $M$ states, $s_1, s_2, ..., s_M$, governed by a transition matrix. Although the overall system operates in discrete time slots, the underlying network state evolution is modeled as a continuous-time Markov process sampled at slot boundaries.

In state $s_m$, packets arrive following a Poisson process with rate $\lambda_g^m$, and transmission delays follow an exponential distribution with rate $\lambda_d^m$:

$$X_m^n \sim \text{Poisson}(\lambda_g^m), \quad Y_m^n \sim \text{Exp}(\lambda_d^m). \tag{6.23}$$

A simplified model is the switched Poisson process (SPP) [74] with a Gilbert-Elliot (GE) channel [75], where the network alternates between low-traffic ($G$) and high-traffic ($B$) states. The packet generation rate $\lambda_g$ and transmission delay $\lambda_d$ depend on the current channel state:

$$\begin{aligned} &X_i^n \sim \text{Poisson}(\lambda_g), \quad Y_i^n \sim \text{Exp}(\lambda_d), \\ &\lambda_g = \lambda_g^L, \quad \lambda_d = \lambda_d^L, \quad \text{if } s = G, \\ &\lambda_g = \lambda_g^H, \quad \lambda_d = \lambda_d^H, \quad \text{if } s = B. \end{aligned} \tag{6.24}$$

Although the scheduling process evolves in discrete decision slots indexed by $t$, the channel state itself is modeled as a continuous-time Markov chain (CTMC) and sampled at slot boundaries.

Let $P(u) = e^{Q_s u}$ denote the transition probability matrix of the CTMC after $u$ units of continuous time, where $Q_s$ is the generator matrix defined by

$$Q_s = \begin{bmatrix} -\mu_1 & \mu_{12} & \dots & \mu_{1M} \\ \mu_{21} & -\mu_2 & \dots & \mu_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{M1} & \mu_{M2} & \dots & -\mu_M \end{bmatrix}, \tag{6.25}$$

where $\mu_{ij}$ denotes the transition rate from state $s_i$ to $s_j$ and $\mu_i = \sum_{j=1, j\neq i}^{M} \mu_{ij}$.

For example, when $M = 2$, the generator reduces to

$$Q_s = \begin{bmatrix} -\mu_1 & \mu_1 \\ \mu_2 & -\mu_2 \end{bmatrix}. \tag{6.26}$$

The corresponding transition probability matrix is

$$P(u) = \begin{bmatrix} \frac{\mu_2}{\mu_1+\mu_2} + \frac{\mu_1}{\mu_1+\mu_2} e^{-(\mu_1+\mu_2)u} & \frac{\mu_1}{\mu_1+\mu_2}\left(1 - e^{-(\mu_1+\mu_2)u}\right) \\ \frac{\mu_2}{\mu_1+\mu_2}\left(1 - e^{-(\mu_1+\mu_2)u}\right) & \frac{\mu_1}{\mu_1+\mu_2} + \frac{\mu_2}{\mu_1+\mu_2} e^{-(\mu_1+\mu_2)u} \end{bmatrix}, \tag{6.27}$$

where the entry $p_{i,j}(u)$ represents the probability of being in state $s_j$ after $u$ units of continuous time when starting from state $s_i$.

### 6.2.6 3D Scene Representations

For 3D scene representations, we employ a representation function $\mathscr{F}_\theta$ parameterized by $\theta$, which maps camera poses $\mathbf{P}_s(t)$ to the underlying scene representation. Depending on the method, $\mathscr{F}_\theta$ can be instantiated as an MLP-based NeRF or a more explicit 3D Gaussian-based model.

Specifically, in the $t$-th time slot, $\mathscr{F}_\theta$ outputs the volume density $\sigma$ and view-dependent RGB color $\mathbf{c}$:

$$\{\mathbf{c}, \sigma\} = \mathscr{F}_\theta(\mathbf{x}, \mathbf{d}), \tag{6.28}$$

where $\mathbf{x}$ denotes a spatial query point and $\mathbf{d}$ the viewing direction.

To visualize the 3D scene representation, a rendered image $\hat{\mathbf{I}}(t)$ is obtained via the volume rendering function $\mathscr{F}_r(\cdot)$:

$$\hat{\mathbf{I}}(t) = \mathscr{F}_r\big(\mathscr{F}_\theta, \mathbf{p}_v\big), \tag{6.29}$$

where $\mathbf{p}_v$ denotes the desired camera pose.

Let $\mathbb{I}(t)$ denote the image set selected by the scheduler at time $t$. For each selected image $I_n(t) \in \mathbb{I}(t)$ with pose $\mathbf{p}^n(t)$, a predicted image $\hat{I}_n(t)$ is synthesized from the same pose using Eq. (6.29).

The representation function is trained by minimizing the reconstruction loss:

$$\min_\theta \mathscr{L}(t) = \sum_{I_n(t) \in \mathbb{I}(t)} M\big(I_n(t), \hat{I}_n(t)\big), \tag{6.30}$$

where $M(\cdot)$ denotes the image similarity metric defined previously (e.g., PSNR, SSIM, LPIPS).

Thus, the selected image set $\mathbb{I}(t)$ directly determines the supervision signal for updating $\theta$, linking the scheduler decision with the evolution of the 3D scene representation.

## 6.3 Problem Formulation

Unlike Chapter 5, where timeliness affected fidelity implicitly through artefacts in multi-view fusion, here timeliness is incorporated as an explicit penalty in the reward design. To explicitly

incorporate information timeliness into the real-time 3D scene representation pipeline, we formulate the scheduling of visual observations as a reinforcement learning problem. At each time slot, the edge server must decide whether to immediately perform scene representation using the currently available images, or to wait for a short horizon to collect additional updates that may improve multi-view completeness and temporal alignment. This decision is non-trivial due to stochastic wireless delays and bursty arrivals, which jointly determine the instantaneous AoI profile and the availability of synchronised views. Motivated by this stochasticity and the absence of a simple closed-form relationship between waiting decisions and reconstruction fidelity, we employ a learning-based scheduler.

We adopt PPO as the baseline algorithm due to its simplicity and stable performance [78]. Moreover, since the randomness in our setting is dominated by one-step exogenous effects (packet arrivals and delays) and the scheduler mainly affects the reconstruction outcome within the current decision epoch, we further employ a single-step (contextual-bandit) variant of it [77].

### State

In the $t$-th time slot, let $\tilde{\tau}^n(t)$ denote the most recent image from the $n$-th camera available at the edge server. A semantic feature extractor $\mathscr{F}_e(\cdot)$ encodes $\tilde{\tau}^n(t)$ into a compact embedding

$$\mathbf{z}_t^n = \mathscr{F}_e(\tilde{\tau}^n(t)), \quad \mathbf{z}_t^n \in \mathbb{R}^{1\times d}, \tag{6.31}$$

where $d$ is the feature dimension. The semantic feature extractor $\mathscr{F}_e(\cdot)$ is used only to provide a compact, task-related context for scheduling. In our implementation, $\mathscr{F}_e(\cdot)$ is instantiated as an off-the-shelf YOLOv11 backbone with pretrained parameters, and kept frozen throughout all experiments. That is, the scheduler does not jointly optimise the perception backbone, the feature extractor, and the scheduling policy. Instead, $\mathscr{F}_e(\cdot)$ acts as a deterministic mapping from the currently buffered image $\tilde{\tau}^n(t)$ to an embedding $\mathbf{z}_t^n$, which is then concatenated into the scheduler state. This approach preserves the modularity of the cross-system pipeline: the 3D scene representation module $\mathscr{F}_\theta$ can be replaced without re-training the semantics encoder.

The scheduler state aggregates timeliness, semantics, and geometry across all $N$ cameras. Define

$$Z_t = \begin{bmatrix} \mathbf{z}_t^1 \\ \vdots \\ \mathbf{z}_t^N \end{bmatrix} \in \mathbb{R}^{N\times d}, \quad \Delta(t) = \begin{bmatrix} \Delta^1(t) \\ \vdots \\ \Delta^N(t) \end{bmatrix} \in \mathbb{R}^{N\times 1}, \quad \mathbf{P}(t) = \begin{bmatrix} \mathbf{p}^1(t) \\ \vdots \\ \mathbf{p}^N(t) \end{bmatrix} \in \mathbb{R}^{N\times 5}, \tag{6.32}$$

where $\Delta^n(t)$ is the AoI of camera $n$ and $\mathbf{p}^n(t)$ is its pose vector. The resulting state is constructed by concatenation:

$$\mathbf{s}_t = \begin{bmatrix} Z_t & \Delta(t) & \mathbf{P}(t) \end{bmatrix} \in \mathbb{R}^{N\times(d+6)}. \tag{6.33}$$

This design allows the agent to reason jointly about (i) freshness, (ii) semantic relevance, and (iii) viewpoint coverage, without directly operating on raw high-dimensional images.

We emphasise that, under this implementation, semantic information influences scheduling only through the state, enabling the policy to condition its waiting decision on the presence of visually salient or dynamically changing content. Since $\mathbf{z}_t^n$ is computed locally from $\tilde{\tau}^n(t)$ at the edge, it introduces no additional communication overhead beyond the transmitted image stream. In contrast to approaches that require learning an explicit importance function $g(\cdot)$ or a task-specific value estimator, we rely on pretrained visual features as a practical and reproducible proxy for semantic context.

### Action

In the $t$-th time slot, the scheduling agent selects a waiting horizon for the subsequent 3D scene representation, defined as

$$\mathbf{a}_t = \omega_t \in \{0, 1, \ldots, \omega_{\max}\}. \tag{6.34}$$

Here, $\omega_t = 0$ indicates immediate rendering using the currently available images, whereas $\omega_t > 0$ indicates that the system postpones reconstruction for $\omega_t$ slots to incorporate potentially fresher and more complete multi-view updates. This action aligns with the $\omega$-wait policy described in the system model, where the training set at the next reconstruction epoch is formed by collecting updates arriving within the waiting horizon.

### Reward

The timeliness–fidelity tradeoff is quantified through a weighted objective. Let $\mathbb{I}(t)$ denote the reconstruction set formed at decision epoch $t$ under the selected waiting horizon $\omega_t$ (and the corresponding scheduling mechanism described in Sec. 6.3). Define the induced selection indicator $w_n(t) \in \{0,1\}$, where $w_n(t) = 1$ if the most recent update from camera $n$ is admitted into $\mathbb{I}(t)$, and $w_n(t) = 0$ otherwise.

We evaluate timeliness on the effective supervision set rather than on all buffered streams. Specifically, the average Age of Information (aAoI) on the reconstruction set is

$$\bar{\Delta}(t) = \frac{1}{K_t} \sum_{n=1}^{N} w_n(t) \Delta^n(t), \quad K_t = \sum_{n=1}^{N} w_n(t). \tag{6.35}$$

Here, the feasible scheduling set is constrained such that $K_t \geq 1$, ensuring that at least one valid observation is available for reconstruction and preventing division-by-zero cases. This definition aligns the freshness penalty with the updates that actually contribute gradients to the representation model. The average formulation of AoI is adopted in this work because the objective is to capture the overall temporal freshness of the distributed sensing system rather than only the

worst-case sensor behaviour. Using aggregated AoI encourages balanced observation freshness across multiple viewpoints and avoids policies that over-optimise a single sensor stream while neglecting the remaining observations. In contrast, max-AoI formulations primarily emphasise the worst stale observation and may lead to overly conservative scheduling behaviour, while min-AoI objectives are generally less informative for multi-sensor perception quality assessment.

Let the fidelity qualifier be defined as a weighted combination of three image similarity metrics:

$$Q(t) = w_p \times \mathrm{PSNR}(\mathbf{I}(t), \hat{\mathbf{I}}(t)) + w_s \times \mathrm{SSIM}(\mathbf{I}(t), \hat{\mathbf{I}}(t)) + w_l \times \mathrm{LPIPS}(\mathbf{I}(t), \hat{\mathbf{I}}(t)). \tag{6.36}$$

We then define the weighted tradeoff function

$$F_w(t, \omega_t) = w_t \cdot \bar{\Delta}(t) + Q(t), \tag{6.37}$$

The instantaneous reward is set as the negative of the tradeoff function:

$$r(\mathbf{s}_t, \mathbf{a}_t) = -F_w(t, \omega_t), \tag{6.38}$$

so that maximizing the reward is equivalent to minimizing the tradeoff objective. Depending on the application scenario, the weights can be adjusted to prioritise either timeliness (smaller aAoI) or fidelity.

**Policy and Objective**

The policy $\pi_\phi$ maps the state $\mathbf{s}_t$ to a categorical distribution over the discrete waiting actions:

$$\rho_t \triangleq \begin{pmatrix} \Pr\{\omega_t = 0 \mid \mathbf{s}_t\} \\ \Pr\{\omega_t = 1 \mid \mathbf{s}_t\} \\ \vdots \\ \Pr\{\omega_t = \omega_{\max} \mid \mathbf{s}_t\} \end{pmatrix} \in \mathbb{R}^{(\omega_{\max}+1)\times 1}. \tag{6.39}$$

The policy is represented by a neural network $\pi_\phi(\mathbf{s}_t)$ with parameters $\phi$. Following $\pi_\phi$, the discounted long-term return is

$$R^{\pi_\phi} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right], \tag{6.40}$$

where $\gamma \in (0,1)$ is the discount factor. The learning objective is to find an optimal policy that maximizes the expected return:

$$\pi^* = \arg\max_{\pi_\phi} R^{\pi_\phi}. \tag{6.41}$$

**Contextual-Bandit PPO**

To optimize $\pi_\phi$, we employ a contextual-bandit (single-step) variant of PPO. At each time slot, the agent observes $\mathbf{s}_t$, samples $\omega_t$ from $\pi_\phi(\cdot \mid \mathbf{s}_t)$, triggers reconstruction according to the selected waiting horizon, and receives the instantaneous reward $r(\mathbf{s}_t, \omega_t)$ computed from Eq. (6.37). The policy parameters are then updated using the clipped PPO objective:

$$\mathscr{L}(\mathbf{s}_t, \omega_t, \phi) = \min\Bigg( \frac{\pi_\phi(\omega_t \mid \mathbf{s}_t)}{\pi_{\phi_t}(\omega_t \mid \mathbf{s}_t)} A^{\pi_{\phi_t}}(\mathbf{s}_t, \omega_t), \tag{6.42}$$

$$\mathrm{clip}\left( \frac{\pi_\phi(\omega_t \mid \mathbf{s}_t)}{\pi_{\phi_t}(\omega_t \mid \mathbf{s}_t)}, 1-\varepsilon, 1+\varepsilon \right) A^{\pi_{\phi_t}}(\mathbf{s}_t, \omega_t) \Bigg), \tag{6.43}$$

where $A^{\pi_{\phi_t}}(\mathbf{s}_t, \omega_t)$ is the advantage estimate [79].

Although the overall problem is cast as reinforcement learning, the state transition in our setting is primarily driven by exogenous communication processes (bursty arrivals and random delays) rather than by the waiting decision itself. Consequently, the action $\omega_t$ has limited influence on future AoI evolution beyond the current decision epoch, while it directly impacts the immediate reconstruction outcome through the timeliness–fidelity objective. This weak action-to-state coupling motivates a single-step approximation, where policy optimization is performed using one-step rewards while still learning a robust stochastic decision rule under stationary network conditions.

## 6.4 Experiment Setup

### 6.4.1 Evaluation of 3D Scene Representations

To comprehensively evaluate the performance of 3D scene representations, we consider both controlled benchmarks and realistic communication scenarios. The DyNeRF dataset provides high-quality inward-facing multi-view sequences for assessing representation accuracy and timeliness, while the Eyeful Tower dataset introduces large-scale outward-facing scenes with greater environmental complexity. In addition, we examine the influence of network burstiness using a GE channel model to capture the impact of packet-level dynamics on real-time scene representation.

In our configuration, the packet generation rate and transmission delay follow the state-dependent parameters $\lambda_g^H = 1/30$, $\lambda_g^L = 1/120$, $\lambda_d^H = 1/60$, and $\lambda_d^L = 1/30$. The underlying Markov chain is a two-state process ($M = 2$), where the transition parameter is set to $\mu_1 = \mu_2 = 1/30$. This configuration captures the contrast between high-state burstiness with longer delays and low-state sparsity with shorter delays.

**Algorithm 2** Contextual-Bandit PPO for Timeliness-Aware Scheduling

**Input:** Channel parameters $\lambda$, initial policy parameters $\phi_0$, representation parameters $\theta$, maximum waiting horizon $\omega_{\max}$, training steps $T_t$.
**for** $t = 1, 2, \ldots, T_t$ **do**
  Observe latest images $\tilde{\tau}^n(t)$ and compute semantic features $\mathbf{z}_t^n = \mathscr{F}_e(\tilde{\tau}^n(t))$.
  Construct state $\mathbf{s}_t = [Z_t \ \Delta(t) \ \mathbf{P}(t)]$.
  Sample waiting action $\omega_t \sim \pi_{\phi_t}(\cdot \mid \mathbf{s}_t)$.
  **if** $\omega_t > 0$ **then**
    Wait for $\omega_t$ slots and collect arriving updates.
  **end if**
  Form reconstruction set $\mathbb{I}(t)$ according to the selected waiting horizon.
  Update representation parameters $\theta$ by minimizing $\mathscr{L}(t) = \sum_{I_n(t)\in\mathbb{I}(t)} M(I_n(t), \hat{I}_n(t))$.
  Render novel view $\hat{\mathbf{I}}(t)$ from pose $\mathbf{p}_v$.
  Compute fidelity metric $Q(t)$ and aAoI $\bar{\Delta}(t)$.
  Compute reward $r_t = -F_w(t, \omega_t)$.
  Estimate advantage $A_t = Q^{\pi_{\phi_t}}(\mathbf{s}_t, \omega_t) - V^{\pi_{\phi_t}}(\mathbf{s}_t)$.
  Update policy parameters by maximizing clipped PPO objective:

$$\phi_{t+1} = \arg\max_{\phi} \mathscr{L}_{\text{PPO}}(\mathbf{s}_t, \omega_t, \phi_t, \phi).$$

**end for**
**Output:** Optimized scheduling policy $\pi_\phi^*$.

### Evaluation on the DyNeRF and the ZJU-MoCap Dataset

The DyNeRF dataset [34] is widely used for 3D scene representation and novel view synthesis. It consists of 10-second, 30-FPS multi-view videos captured from 19 cameras positioned at different angles, providing high-quality data for training and evaluation. Fig. 6.4 compares the reconstruction quality achieved under the $\omega$-wait and $\omega$-threshold policies across different timeliness thresholds $\omega_t$ on the DyNeRF dataset. Qualitative comparisons show that the $\omega$-wait policy generally preserves clearer structural consistency and fewer temporal artefacts in the dynamically changing regions highlighted by the bounding boxes. In particular, the reconstructed human body and foreground object regions remain more visually coherent under the $\omega$-wait formulation, while the $\omega$-threshold policy exhibits stronger blurring and ghosting effects as temporally stale observations accumulate. The quantitative LPIPS results further support these observations. As $\omega_t$ increases, the $\omega$-wait policy consistently improves perceptual reconstruction quality, reducing LPIPS from approximately 0.515 to 0.193, corresponding to an improvement of about 62.5%. In contrast, although the $\omega$-threshold policy initially improves from 0.643 to 0.232, the reconstruction quality gradually degrades again at larger $\omega_t$ values, eventually increasing to approximately 0.325. This behaviour suggests that the $\omega$-threshold formulation becomes increasingly sensitive to stale or temporally inconsistent observations under relaxed freshness constraints. Overall, the results indicate that the proposed $\omega$-wait policy

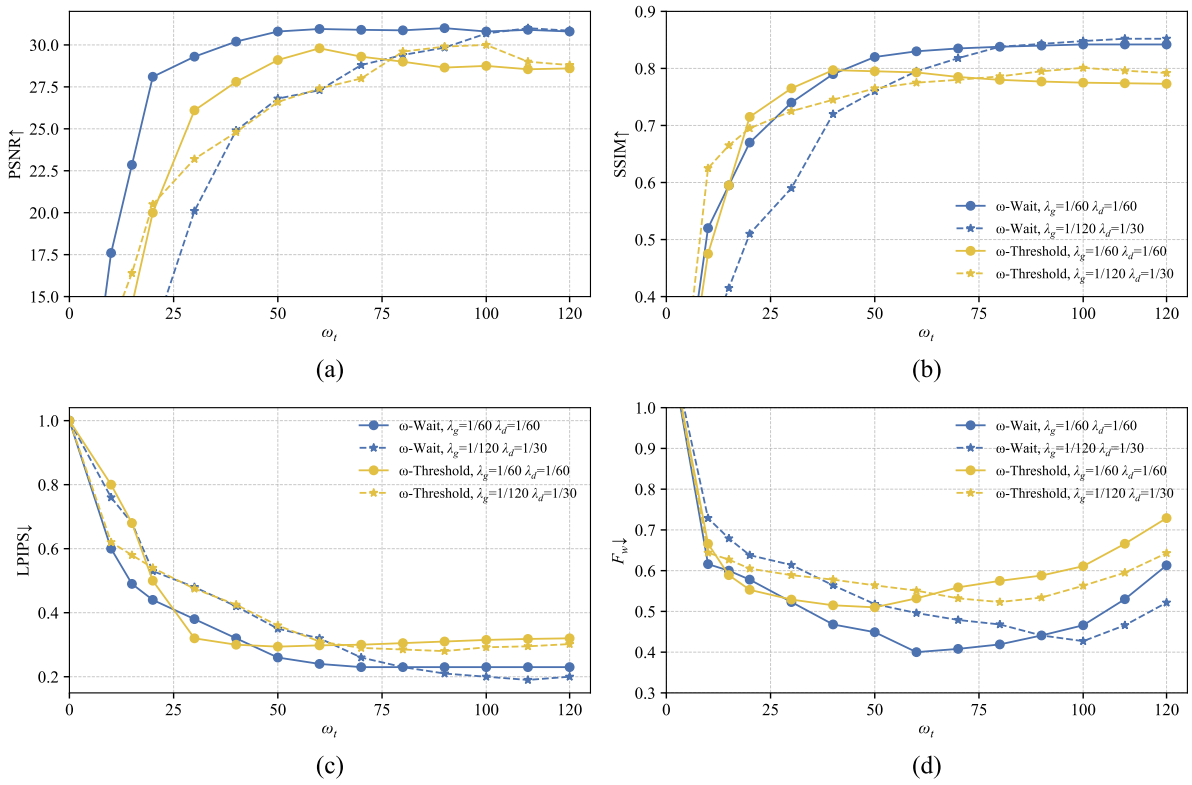



Figure 6.3: Comparison of representation quality and overall performance under different scheduling strategies averaged across datasets with Instant-NGP. The four subfigures report (a) PSNR↑, (b) SSIM↑, (c) LPIPS↓, and (d) $F_w \downarrow$ as functions of the parameter $\omega_t$. Results are shown for two traffic intensities ($\lambda_g$=1/60, $\lambda_d$=1/60 and $\lambda_g$=1/120, $\lambda_d$=1/30) and two policies ($\omega$-wait and $\omega$-threshold).

achieves more stable and perceptually consistent scene representation performance in dynamic multi-robot telepresence scenarios. Both datasets adopt inward-facing camera configurations, which are particularly suited for neural rendering methods that rely on dense multi-view observations.

The ZJU-MoCap dataset [81, 82] contains human motion sequences recorded in a multi-view studio setup with synchronized cameras. Fig. 6.5 illustrates the influence of the number of transmitted observations on both reconstruction quality and temporal freshness in the multi-robot telepresence scenario. As the number of images increases, the perceptual reconstruction quality consistently improves, with LPIPS decreasing from approximately 0.323 to 0.101, corresponding to an improvement of approximately 68.7%. Qualitative comparisons further confirm that increasing the number of observations significantly improves structural completeness and texture consistency, particularly in the highlighted regions containing fine-grained clothing patterns and human body boundaries. However, this improvement in reconstruction fidelity is accompanied by a substantial increase in aAoI, which rises from approximately 16 to 82 as more observations are incorporated into the reconstruction process. This trend demonstrates the fundamental timeliness–fidelity tradeoff considered in this thesis: incorporating more viewpoints

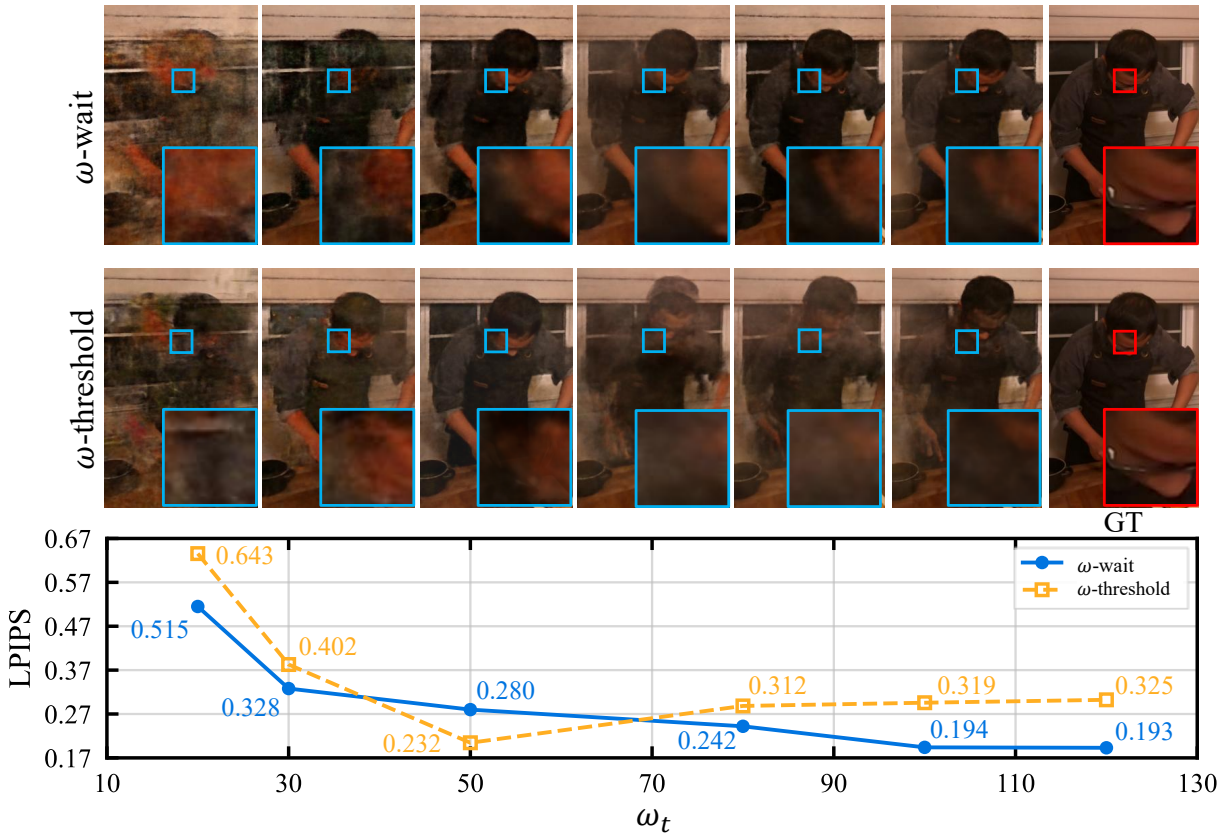



Figure 6.4: Comparison of 3D scene representation quality under the $\omega$-wait and the $\omega$-threshold policy, trained with Instant-NGP.

generally improves multi-view consistency and perceptual quality, but also increases communication latency and observation staleness. Beyond approximately 12 images, the LPIPS improvement gradually saturates while the aAoI continues to increase rapidly, suggesting diminishing reconstruction returns under increasingly stale observation conditions.

For benchmarking, we adopt the following methodology:

- **Training and evaluation:** Eighteen out of the 19 available videos are used to train the 3D scene representations, while the remaining one serves as the ground truth for evaluating novel view synthesis performance, i.e., $N = 18$.
- **Frame capture interval:** Each camera captures frames at a fixed interval of 30 ms (30 FPS).

### Evaluation on the VR-NeRF Eyeful Tower Dataset

The VR-NeRF Eyeful Tower dataset [80] is an outward-facing multi-view dataset specifically designed for evaluating large-scale 3D scene representation and novel view synthesis. Unlike inward-facing datasets such as DyNeRF, the cameras in Eyeful Tower are positioned around

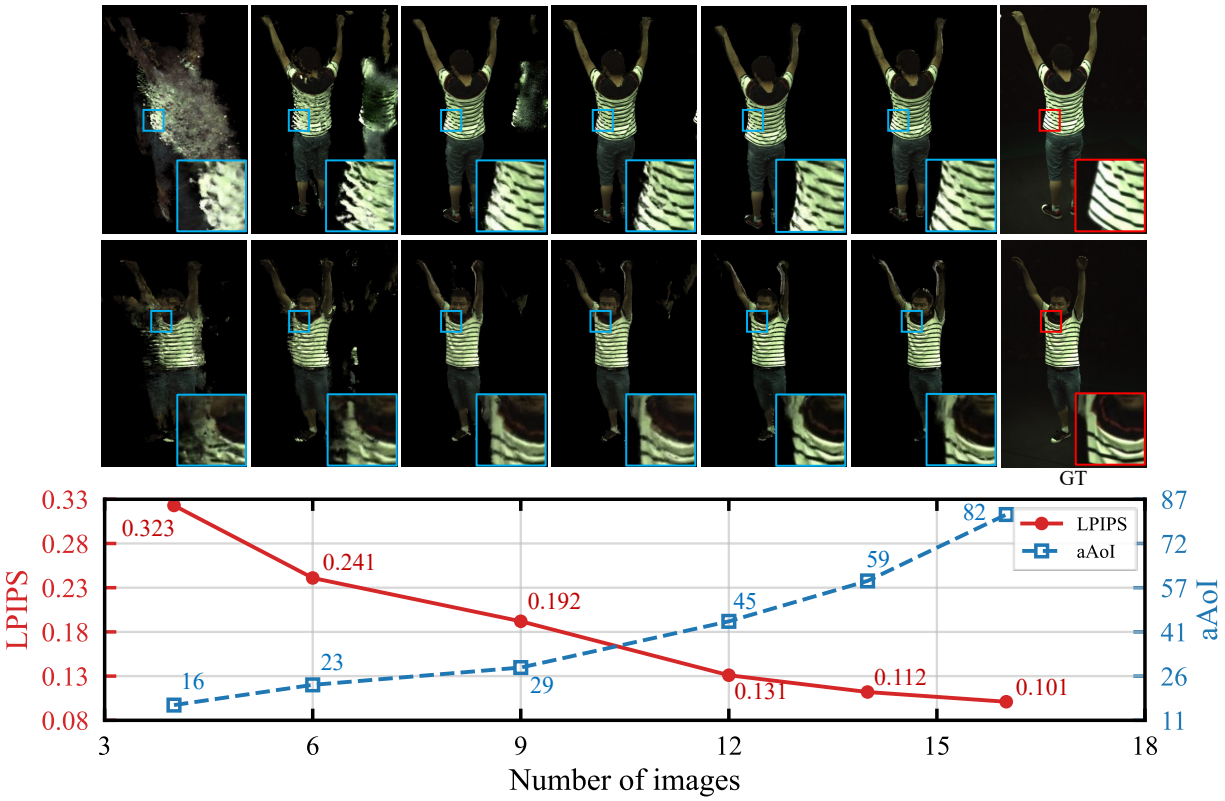



Figure 6.5: Results on the ZJU-MoCap dataset, trained with Nerfacto. The plots report LPIPS↓ and aAoI↑ as the number of training images increases.

outdoor landmarks and point outward, capturing diverse views of the scene under natural illumination. This setting introduces greater challenges in terms of scale variation, occlusion, and background complexity. Fig. 6.6 presents the reconstruction results on the VR-NeRF Eyeful Tower dataset using 3D Gaussian Splatting under different numbers of transmitted observations. Similar to the previous experiment, increasing the number of images consistently improves perceptual reconstruction quality, with LPIPS decreasing from approximately 0.472 to 0.214, corresponding to an improvement of approximately 54.7%. The qualitative comparisons further demonstrate progressively improved geometric completeness and sharper structural boundaries in the highlighted regions as additional observations are incorporated into the reconstruction process. At the same time, the aAoI increases substantially from approximately 37 to 146, indicating that higher reconstruction fidelity is achieved at the cost of increased temporal staleness. Compared with the previous multi-robot telepresence experiment, the aAoI growth in this dataset is significantly more pronounced, suggesting that the larger-scale and more complex scene introduces greater communication and reconstruction overhead during multi-view fusion. These results further validate the central timeliness–fidelity tradeoff studied in this thesis: incorporating more observations generally improves reconstruction quality, but simultaneously increases communication delay and observation staleness within distributed perception systems.

For benchmarking, we adopt the following methodology:

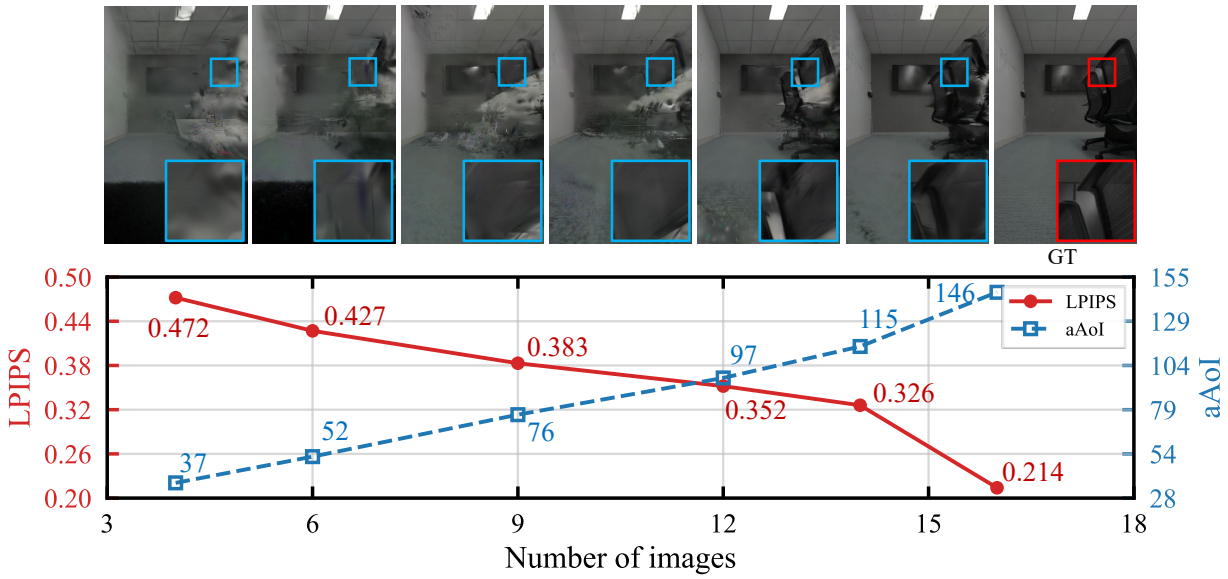



Figure 6.6: Results on the VR-NeRF Eyeful Tower dataset, trained with 3D Gaussian Splatting. The plots report LPIPS↓ and AoI↑ as the number of training images increases.

- **Training and evaluation:** A subset ($N = 18$) of the outward-facing camera views is used for training the 3D scene representations, while the remaining views are held out for evaluating novel view synthesis performance.
- **Frame capture interval:** Each camera records frames at a fixed interval of 33 ms (corresponding to 30 FPS).

### 6.4.2 Evaluation on Packet Burstiness

To further investigate the impact of communication dynamics on 3D scene representations, we simulate a GE channel model, which introduces packet burstiness and network latency into the pipeline. Before the simulation begins, a time series map is generated for each of the 19 sensor nodes, containing both the sending and receiving frame sequences. At each packet interval $T_i$, a frame is sampled and transmitted. After experiencing the total uplink delay $Y_i^n$, the frame is stored in the training buffer of the corresponding node within the scene representation module.

### 6.4.3 3D Scene Representation Methods

To analyze the timeliness–fidelity tradeoff in real-time 3D scene representations, we evaluate our approach using three widely used neural scene representation methods.

**Instant-NGP** [52]

Instant-NGP is a NeRF-based 3D scene representation method. Utilizing a C++ embedded neural network structure, hash coding, and CUDA, Instant-NGP achieves state-of-the-art training

speed among NeRF-style methods.

**3D Gaussian Splatting** [54]

3DGS represents a scene using a set of anisotropic 3D Gaussians instead of an implicit neural field. Each Gaussian is parameterized by its position, covariance, opacity, and color, and the scene rendering is achieved by splatting these Gaussians along camera rays. This explicit representation enables real-time training and rendering, offering a significant speedup compared to traditional NeRF-based methods while preserving high visual fidelity.

**Nerfacto** [57]

Nerfacto integrates camera pose refinement and per-image appearance conditioning to augment representation quality. It applies the hash coding from Instant-NGP to accelerate training. Compared with Instant-NGP's CUDA-based core computing module, Nerfstudio is programmed in Python and thus requires more computation time.

## 6.5 Performance Evaluation

In Fig. 6.3, to demonstrate the timeliness–fidelity tradeoff, we consider both the $\omega$-threshold and the $\omega$-wait methods. In Fig. 6.8, to show the optimal performance achieved by the proposed contextual-bandit PPO algorithm, we evaluate the training and testing results. The parameters of the weighted objective $F_w(t, \omega_t)$ are set to $w_p = -0.02$, $w_s = 0.2$, $w_l = 0.3$, and $w_t = 0.015$.

Before presenting the detailed comparisons, it is important to clarify a fundamental mechanism difference between the $\omega$-threshold and the $\omega$-wait policies. Under the $\omega$-threshold policy, image generation and transmission are continuous. At each time slot, the scheduler filters the most recently received frames based on instantaneous AoI. Packets may already be in transmission before the decision is made, and previously generated frames can still arrive and be considered. The policy therefore operates on asynchronously received updates.

In contrast, under the $\omega$-wait policy, the waiting horizon is initiated at the decision epoch. Only frames generated and transmitted after that time are accumulated for reconstruction. Prior to the waiting decision, no packets are considered in transit for that reconstruction step. Because the generation interval is much smaller than the average delay, during the waiting horizon almost all cameras generate new frames. As a result, the reconstruction step at the end of the waiting period effectively operates on a near-synchronized multi-view batch.

Therefore, the difference between the two policies is not merely a matter of delaying reconstruction versus filtering by freshness. It originates from two fundamentally different temporal acquisition mechanisms:

- $\omega$-threshold: asynchronous filtering of already in-flight updates;

- $\omega$-wait: synchronized accumulation of newly generated updates.

This distinction is essential for interpreting the experimental results below.

### 6.5.1 Timeliness-Fidelity Tradeoff with $\omega$-Threshold Policy

Figure 6.3 illustrates the effect of the scheduling parameter $\omega_t \in (0, 120)$ in the $\omega$-threshold policy on the performance of 3D scene representations, measured by PSNR, SSIM, LPIPS, and $F_w$.

The results reveal an inherent tradeoff between timeliness and fidelity. As $\omega_t$ increases, more packets satisfy the freshness constraint, improving the effective delivery rate. However, a larger threshold also increases the probability that outdated frames are incorporated into the reconstruction process. When $\omega_t$ becomes excessively large, stale information contaminates the supervision set, leading to degraded representation quality.

In the high-traffic state ($\lambda_g = \lambda_d = 1/60$), the initial reconstruction quality is relatively low due to high latency, yet it improves rapidly as $\omega_t$ increases and more packets are admitted. The tradeoff point is reached earlier at $\omega_t = 50$. The corresponding $F_w$ curve exhibits a sharp decrease at small $\omega_t$, reflecting improved timeliness, but subsequently rises as stale-frame contamination dominates.

In the low-traffic state ($\lambda_g = 1/120, \lambda_d = 1/30$), representation quality increases more slowly due to reduced packet arrival rates. Nevertheless, the tradeoff point shifts to a higher threshold at $\omega_t = 72$, since packet drops are less frequent and the probability of outdated frame contamination is reduced.

Overall, these results confirm that under the $\omega$-threshold policy, the choice of $\omega_t$ directly induces a timeliness–fidelity tradeoff in image metrics alone. This behaviour stems from the asynchronous nature of filtering already in-flight updates.

### 6.5.2 Timeliness-Fidelity Tradeoff with $\omega$-Wait Policy

Figure 6.3 also illustrates the effect of $\omega_t$ under the $\omega$-wait policy. In contrast to the $\omega$-threshold strategy, the $\omega$-wait curves do not exhibit a pronounced internal tradeoff point in the image similarity metrics.

This behaviour is consistent with the mechanism described earlier. Because the $\omega$-wait policy accumulates only newly generated frames within the waiting horizon, it largely avoids contamination from stale data. As a result, PSNR and SSIM increase steadily while LPIPS decreases monotonically until convergence. Representation quality remains consistently higher across a wide range of $\omega_t$ values.

In the high-traffic setting ($\lambda_g = \lambda_d = 1/60$), the curves start from a lower baseline due to latency but rise quickly and converge around $\omega_t = 62$. In the low-traffic setting ($\lambda_g = 1/120, \lambda_d = 1/30$), the curves increase more slowly yet eventually achieve a higher steady-state quality at

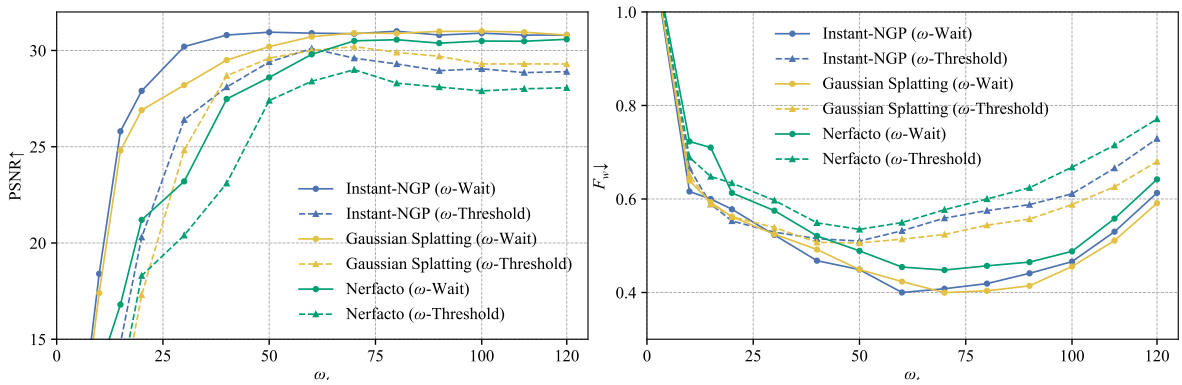



Figure 6.7: Comparison of representation quality and overall performance under different scheduling strategies and 3D scene representation methods. The two subfigures report PSNR↑ and the $F_w$↓ as functions of the parameter $\omega$. Results are shown for two policies ($\omega$-wait and $\omega$-threshold).

$\omega_t = 105$. However, when the AoI-aware penalty $F_w$ is considered, a clear tradeoff re-emerges. Increasing $\omega_t$ improves reliability by raising packet delivery probability, but simultaneously reduces freshness, resulting in higher penalties in $F_w$. Therefore, although the $\omega$-wait policy mitigates stale-data degradation in image metrics, it still reveals an inherent timeliness–reliability tradeoff once freshness is explicitly evaluated. Moreover, as illustrated in Fig. 6.7, this timeliness–fidelity tradeoff consistently emerges across different 3D scene representation methods, demonstrating that the phenomenon is structural rather than specific to a single representation model.

### 6.5.3 3D Scene Representations with Contextual-Bandit PPO

Figure 6.8 presents the training dynamics of the proposed RL scheduler under the SPP channel model. Due to the increased state-space complexity introduced by semantic information and burst-aware communication dynamics, the scheduler agent required approximately one hour of offline training on an NVIDIA RTX 4090 GPU. Both scheduling paradigms exhibit steady performance improvement as training progresses, confirming that the contextual-bandit PPO framework is capable of capturing the timeliness–fidelity tradeoff through interaction with the communication environment. Notably, the $\omega$-wait policy consistently outperforms the $\omega$-threshold policy in terms of instantaneous reward after convergence. Specifically, the learned scheduler achieves $r(\mathbf{s}_t, \mathbf{a}_t) = -0.46$, PSNR = 30.05, SSIM = 0.793, and LPIPS = 0.248 under the $\omega$-wait mechanism. Although a fully random scheduling baseline was not explicitly implemented, the early-stage exploratory behaviour provides an approximate reference for highly randomised scheduling prior to policy convergence. The reward fluctuations during this stage are approximately within the ranges of $-0.60$ to $-0.44$ for the $\omega$-wait policy and $-0.62$ to $-0.49$ for the $\omega$-threshold policy. After convergence, the reinforcement-learning-based scheduler stabilises

around average rewards of approximately $-0.45$ and $-0.48$, respectively, corresponding to improvements of approximately 8.2% and 17.2% over the corresponding exploratory scheduling behaviour. Nevertheless, this exploratory phase should not be interpreted as a formally controlled uniform-random baseline, and a dedicated random scheduling comparison remains an interesting direction for future investigation. Both the $\omega$-wait and $\omega$-threshold policies demonstrate stable reinforcement learning convergence behaviour over the training process. The $\omega$-wait policy improves its average instantaneous reward from approximately $-0.57$ to $-0.45$, corresponding to an improvement of about 21%. Similarly, the $\omega$-threshold policy improves from approximately $-0.67$ to $-0.48$, corresponding to an improvement of about 28%. After convergence, the proposed $\omega$-wait formulation consistently achieves approximately 6% higher reward than the $\omega$-threshold policy while also exhibiting lower long-term reward variance. These results indicate that the proposed scheduling strategy achieves more stable and effective optimisation behaviour under the considered timeliness-aware communication setting.

This performance gap is consistent with the mechanism analysis presented earlier. Because the frame generation interval is significantly smaller than the average delay, the $\omega$-wait strategy effectively constructs near-synchronized multi-view batches, thereby reducing stale-data contamination and improving reconstruction stability. The PPO agent exploits this structural advantage by learning an appropriate waiting horizon that balances reliability and freshness under the given traffic state. In contrast, under the $\omega$-threshold policy, the agent must operate on asynchronously received and potentially in-flight updates. The policy, therefore, faces a more delicate tradeoff between admitting additional views and preventing stale-frame contamination. Although PPO successfully learns an effective threshold $\omega_t$, the achievable reward is inherently constrained by the asynchronous filtering mechanism.

Importantly, these results demonstrate that contextual-bandit PPO does not merely tune a scalar scheduling parameter. Instead, it adapts the temporal acquisition strategy to the stochastic communication dynamics, learning a state-dependent decision rule that aligns reconstruction fidelity with observation freshness. This enables real-time 3D scene representation to operate closer to the optimal timeliness–fidelity operating point under both scheduling paradigms.

## 6.6 Conclusion

This chapter investigated the fundamental timeliness–fidelity tradeoff in real-time 3D scene representation under stochastic wireless communication. Unlike conventional neural rendering pipelines that assume temporally consistent multi-view observations, we explicitly modelled the impact of transmission delay and packet dynamics on reconstruction quality, and demonstrated that timeliness constitutes a first-order system constraint rather than a secondary implementation detail.

We first analysed two baseline scheduling mechanisms, namely the $\omega$-threshold and $\omega$-wait

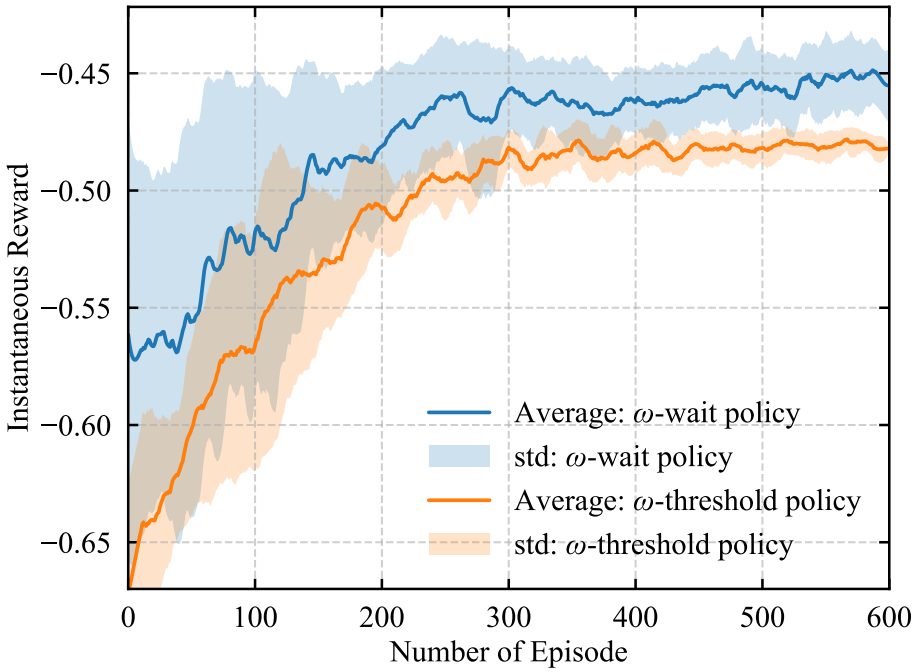



Figure 6.8: Training performance comparison of the two scheduling policies. The curves show the average instantaneous reward as a function of the training episode, while the shaded areas represent the standard deviation. Results are reported for the $\omega$-wait and $\omega$-threshold policies.

policies, and showed that both induce inherent tradeoffs between observation freshness and multi-view reliability. Through theoretical reasoning and empirical evaluation, we revealed that the difference between these policies arises from their distinct temporal acquisition mechanisms: the $\omega$-threshold policy filters asynchronously received updates, whereas the $\omega$-wait policy constructs quasi-synchronous batches by explicitly delaying reconstruction. This structural difference leads to fundamentally different operating characteristics under bursty communication dynamics.

To overcome the rigidity of fixed scheduling rules, we formulated the problem as a contextual-bandit reinforcement learning task and proposed a single-step PPO-based scheduler. The learned policy adapts the waiting or selection strategy according to the instantaneous AoI profile, enabling dynamic balancing between freshness and reconstruction fidelity. Extensive experiments across inward-facing and outward-facing datasets, as well as under switched Poisson channel models, confirmed that the proposed approach consistently achieves improved fidelity while maintaining controlled AoI, demonstrating robustness across scene types and representation methods.

Overall, this study establishes that timeliness-aware scheduling is a critical component of networked 3D perception systems. By bridging communication dynamics and neural scene representation, the proposed framework provides a principled foundation for integrating learning-based scheduling into real-time robotic perception pipelines. Future directions include extending

the formulation to multi-hop communication architectures, task-driven perception objectives, and closed-loop embodied systems where perception, communication, and control are tightly coupled.

# Chapter 7

# General Conclusion and Future Directions

## 7.1 Thesis Overview

This thesis investigated task-oriented, timeliness-aware 3D scene representation for robotic systems operating under sensing and communication constraints. Rather than treating perception and communication as loosely coupled components, the thesis adopted a system-level perspective in which real-time 3D scene representation is shaped by physical acquisition, network dynamics, and downstream task objectives.

The central premise is that real-time 3D scene representation for robotics cannot be designed in isolation from how observations are generated, transmitted, and consumed. In practical inspection and interaction scenarios, reconstruction fidelity, update freshness, and resource usage (e.g., bandwidth and compute) must be jointly considered, since each directly affects what the robot (or operator) can reliably perceive and act upon.

Across Chapters 4–6, the thesis developed this perspective progressively: Chapter 4 established an embodied real-time 3D scene representation platform; Chapter 5 formalised the timeliness–fidelity coupling under stochastic delays; and Chapter 6 extended the formulation toward task-oriented communication and adaptive scheduling under bursty network dynamics.

## 7.2 Summary of Contributions

The main contributions of this thesis can be summarised along three connected directions.

### 1. Embodied Real-Time 3D Scene Representation Platform

Chapter 4 presented an embodied robotic platform for real-time 3D scene representation developed within the UKAEA RAICo project. The system integrates teleoperation-driven data acquisition with deterministic camera pose recovery derived from robot kinematics and hand–eye calibration, enabling reliable synchronization between captured images and camera poses

during robotic operation.

On top of this acquisition pipeline, real-time neural 3D scene representation is performed using Instant-NGP, allowing rapid updates of the scene representation as new observations are collected. To improve robustness in cluttered inspection environments, object-centric preprocessing, including detection and segmentation, is introduced prior to neural optimisation. For deployment-level outputs required in inspection tasks, the neural radiance field can further be converted into surface-continuous meshes through PGSR-based post-processing.

Furthermore, the platform serves as a physical testbed for active viewpoint planning, supporting both human-in-the-loop exploration and algorithmic next-best-view strategies under realistic robotic sensing constraints.

## 2. Timeliness–Fidelity Tradeoff Modelling

Chapter 5 introduced a modelling framework that links Age of Information (AoI) dynamics with the quality of learned 3D scene representations. By integrating discrete-time AoI evolution with reconstruction performance metrics, the chapter characterised how communication delay and update staleness alter the supervision set used for multi-view learning, and thereby affect 3D scene representation fidelity.

Building on this formulation, a learning-based scheduling approach was developed to optimise sensing and transmission decisions under timeliness constraints. Under the evaluated bursty traffic settings, the learned scheduling policies achieved consistently better timeliness–fidelity tradeoffs than fixed periodic baselines and simple heuristic rules, indicating that the operating point of real-time neural 3D scene representation is strongly shaped by communication dynamics.

This contribution reframed real-time 3D scene representation as a communication-constrained optimisation problem, where temporal properties of the dataset (not only its geometric coverage) become a first-order design variable.

## 3. Task-Oriented Communications for 3D Scene Representation

Chapter 6 extended the framework from timeliness-only scheduling toward task-oriented communication strategies. Instead of treating all updates as equally valuable and minimising AoI globally, the proposed formulation incorporates task-related context into the decision process, allowing the scheduler to prioritise updates that are more likely to contribute to downstream 3D scene representation quality under limited resources.

A joint state–action–reward design was introduced, enabling contextual scheduling policies that integrate (i) timeliness indicators, (ii) bursty network conditions, and (iii) task/semantic context extracted from the incoming visual stream. This aligns communication decisions with task objectives, reducing unnecessary transmissions while preserving 3D scene representation

fidelity where it matters most for inspection.

Together, Chapters 5 and 6 support the broader conclusion that communication policies for robotic perception should be aware of both temporal dynamics and task objectives, rather than optimising a single network-layer metric in isolation.

## 7.3 Limitations

Despite the demonstrated effectiveness of the proposed framework, several limitations remain.

First, experimental validation was conducted primarily in laboratory and semi-structured inspection environments. Real nuclear decommissioning sites may introduce additional sensing challenges, including radiation-related sensor degradation, extreme illumination variation, and constrained communication infrastructure.

Second, the communication models assume that burstiness characteristics are either known or can be estimated reliably. In highly non-stationary or adversarial conditions, the scheduling policy may require explicit online adaptation or uncertainty-aware decision mechanisms.

Third, while reinforcement learning improves empirical performance under the tested settings, interpretability remains limited and strong theoretical optimality guarantees are not provided.

Finally, the current platform is evaluated primarily in single-robot scenarios. Extending to multi-robot cooperative 3D scene representation introduces coordination, synchronisation, and inter-agent communication constraints that are not fully addressed in this thesis.

## 7.4 Future Research Directions

Building upon this thesis, several research directions emerge.

#### 1. Fully Autonomous View Planning

While the current framework supports teleoperation and provides a basis for active planning, future work can integrate task-aware scheduling with continuous trajectory optimisation to enable fully autonomous inspection under communication constraints.

#### 2. Multi-Robot Distributed 3D Scene Representations

Extending timeliness-aware modelling to distributed multi-robot systems would enable scalable inspection in large industrial environments. Key challenges include joint AoI modelling across agents, coordinated scheduling, and distributed fusion of partially asynchronous observations.

**3. Robustness in Extreme Environments**

Adapting neural 3D scene representation and scheduling policies to radiation, underwater, or low-visibility conditions remains an important practical extension, particularly when sensing quality and network reliability degrade simultaneously.

**4. Cross-Layer Co-Design**

Future work may explore tighter integration between physical-layer communication controls (e.g., rate adaptation and retransmission strategies) and high-level 3D scene representation objectives, enabling end-to-end optimisation from wireless transmission to task-level perception performance.

**5. Outstanding Challenges for Real-World Deployment**

Despite the promising results presented in this thesis, several challenges remain before timeliness-aware 3D scene representation systems can be fully deployed in robotic environments. Current neural reconstruction pipelines remain computationally expensive for continuously updating high-fidelity representations under streaming conditions. Dynamic scenes, asynchronous sensor updates, and communication instability may further introduce temporal inconsistencies and reconstruction artefacts. In addition, future systems will require tighter integration between sensing, communication, edge computation, and task-oriented semantic reasoning in order to support robust long-term robotic operation.

**6. Ablation Study for State Info in Chapter 6**

Although the proposed scheduler jointly considers observation freshness, semantic information, and camera-related state variables, this thesis does not perform a comprehensive ablation analysis of each individual state component. Such analysis could provide additional insight into the relative contribution of different information sources to scheduling performance and policy behaviour, and therefore represents an important direction for future work. In particular, understanding how different semantic, temporal, and geometric state representations influence scheduling decisions may further improve the interpretability and generalisability of timeliness-aware perception policies.

## 7.5 Concluding Remarks

A central insight of this thesis is that, in networked robotic perception, the supervision set used for 3D scene representation is not a static dataset but a stochastic object shaped by sensing dynamics, communication delays, and scheduling decisions. Consequently, reconstruction quality

is not determined solely by model capacity or optimisation strategy, but also by how, when, and which observations are acquired and transmitted.

This thesis advanced the understanding of how real-time 3D scene representation should be designed for robotic systems operating under realistic sensing and communication constraints. By combining an embodied reconstruction platform with timeliness-aware modelling and task-oriented scheduling, the thesis showed that reconstruction fidelity and resource efficiency need not be treated as opposing goals; instead, they can be jointly optimised when communication and perception are co-designed around the task.

The resulting framework provides a foundation for future research in communication-aware robotic perception and supports progress toward deployable real-time 3D modelling for safety-critical inspection environments.

# Bibliography


[1] Shannon, C. E. (1948). *A Mathematical Theory of Communication*. *Bell System Technical Journal*, 27(3), pp. 379–423.

[2] Meng, Z., Chen, K., Diao, Y., She, C., Zhao, G., Imran, M. A., and Vucetic, B. (2023). *Task-oriented cross-system design for timely and accurate modeling in the metaverse*. *IEEE Journal on Selected Areas in Communications*, 42(3), pp. 752–766.

[3] She, C., Sun, C., Gu, Z., Li, Y., Yang, C., Poor, H. V., and Vucetic, B. (2021). *A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning*. *Proceedings of the IEEE*, 109(3), pp. 204–246.

[4] Diao, Y., Zhang, Y., She, C., Zhao, P. G., and Li, E. L. (2025). *Aligning task-and reconstruction-oriented communications for edge intelligence*. *IEEE Journal on Selected Areas in Communications*.

[5] Kizilkaya, B., She, C., Zhao, G., and Imran, M. A. (2023). *Task-oriented prediction and communication co-design for haptic communications*. *IEEE Transactions on Vehicular Technology*, 72(7), pp. 8987–9001.

[6] Qin, Z., Gao, F., Lin, B., Tao, X., Liu, G., and Pan, C. (2023). *A generalized semantic communication system: From sources to channels*. *IEEE Wireless Communications*, 30(3), pp. 18–26.

[7] Xie, H., Qin, Z., Li, G. Y., and Juang, B. H. (2021). *Deep learning enabled semantic communication systems*. *IEEE Transactions on Signal Processing*, 69, pp. 2663–2675.

[8] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). *MonoSLAM: Real-time single camera SLAM*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), pp. 1052–1067.

[9] Klein, G., and Murray, D. (2007). *Parallel tracking and mapping for small AR workspaces*. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, pp. 225–234.

[10] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). *ORB-SLAM: A versatile and accurate monocular SLAM system*. *IEEE Transactions on Robotics*, 31(5), pp. 1147–1163.

[11] Whelan, T., Leutenegger, S., Salas-Moreno, R. F., Glocker, B., and Davison, A. J. (2015). *ElasticFusion: Dense SLAM without a pose graph*. In *Proceedings of Robotics: Science and Systems (RSS 2015)*, Vol. 11, No. 3.

[12] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. (2015). *Keyframe-based visual–inertial odometry using nonlinear optimization*. *The International Journal of Robotics Research*, 34(3), pp. 314–334.

[13] Qin, T., Li, P., and Shen, S. (2018). *VINS-Mono: A robust and versatile monocular visual-inertial state estimator*. *IEEE Transactions on Robotics*, 34(4), pp. 1004–1020.

[14] Hartley, R., and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

[15] Kaul, S., Yates, R., and Gruteser, M. (2012). *Real-time status: How often should one update?* In *Proceedings of IEEE INFOCOM 2012*, pp. 2731–2735. doi:10.1109/INFCOM.2012.6195689.

[16] Sun, Y., Uysal-Biyikoglu, E., Yates, R. D., Koksal, C. E., and Shroff, N. B. (2017). *Update or wait: How to keep your data fresh*. *IEEE Transactions on Information Theory*, 63(11), pp. 7492–7508. doi:10.1109/TIT.2017.2735804.

[17] Yates, R. D., Sun, Y., Brown, D. R., Kaul, S. K., Modiano, E., and Ulukus, S. (2021). *Age of information: An introduction and survey*. *IEEE Journal on Selected Areas in Communications*, 39(5), pp. 1183–1210. doi:10.1109/JSAC.2021.3065072.

[18] Yates, R. D., and Kaul, S. K. (2019). *The age of information: Real-time status updating by multiple sources*. *IEEE Transactions on Information Theory*, 65(3), pp. 1807–1827. doi:10.1109/TIT.2018.2871079.

[19] Zhou, B., and Saad, W. (2019). *Joint status sampling and updating for minimizing age of information in the Internet of Things*. *IEEE Transactions on Communications*, 67(11), pp. 7468–7482.

[20] Kam, C., Kompella, S., Nguyen, G. D., Wieselthier, J. E., and Ephremides, A. (2016). *Age of information with a packet deadline*. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2016)*, pp. 2564–2568. doi:10.1109/ISIT.2016.7541762.

[21] Agheli, P., Pappas, N., Popovski, P., et al. (2024). *Effective communication: When to pull updates?* In *Proceedings of the IEEE International Conference on Communications (ICC 2024)*, pp. 183–188.

[22] Mildenhall, B., Srinivasan, P. P., Tancik, M., et al. (2021). *NeRF: Representing scenes as neural radiance fields for view synthesis*. *Communications of the ACM*, 65(1), pp. 99–106.

[23] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., et al. (2002). *Wireless sensor networks: A survey*. *Computer Networks*, 38(4), pp. 393–422.

[24] Rosinol, A., Leonard, J. J., and Carlone, L. (2023). *NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023)*, pp. 3437–3444.

[25] Yu, Z., Peng, S., Niemeyer, M., et al. (2022). *MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction*. *Advances in Neural Information Processing Systems*, 35, pp. 25018–25032.

[26] Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., and Srinivasan, P. P. (2021). *Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021)*, pp. 5855–5864.

[27] Sucar, E., Liu, S., Ortiz, J., and Davison, A. J. (2021). *iMAP: Implicit mapping and positioning in real-time*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021)*, pp. 6229–6238.

[28] Rosinol, A., Abate, M., Chang, Y., and Carlone, L. (2020). *Kimera: An open-source library for real-time metric-semantic localization and mapping*. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2020)*, pp. 1689–1696.

[29] Rosinol, A., Leonard, J. J., and Carlone, L. (2023). *NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023)*, pp. 3437–3444.

[30] Zhang, K., Riegler, G., Snavely, N., and Koltun, V. (2020). *NeRF++: Analyzing and improving neural radiance fields*. arXiv preprint arXiv:2010.07492.

[31] Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., et al. (2022). *NICE-SLAM: Neural implicit scalable encoding for SLAM*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, pp. 12786–12796.

[32] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). *You only look once: Unified, real-time object detection*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 779–788.

[33] Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., et al. (2024). *4D Gaussian splatting for real-time dynamic scene rendering*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024)*, pp. 20310–20320.

[34] Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., et al. (2022). *Neural 3D video synthesis from multi-view video*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, pp. 5521–5531.

[35] Weaver, W. (1949). *Recent contributions to the mathematical theory of communication*. In *The Mathematical Theory of Communication*, University of Illinois Press, pp. 1–28.

[36] Chen, Z., Yang, T., Pappas, N., Yang, H. H., Tian, Z., Wang, M., and Quek, T. Q. (2024). *Improving information freshness via multi-sensor parallel status updating*. *IEEE Transactions on Communications*, 73(1), pp. 540–554.

[37] Chen, Z., Xu, M., She, C., Jia, Y., Wang, M., and Li, Y. (2023). *Improving timeliness-fidelity tradeoff in wireless sensor networks: Waiting for all and waiting for partial sensor nodes*. *IEEE Transactions on Communications*, 71(7), pp. 4151–4164.

[38] Yin, B., Wang, Q., Zhang, P., Zhang, J., Wang, K., Wang, Z., et al. (2025). *Spatial mental modeling from limited views*. In *Structural Priors for Vision Workshop at the IEEE/CVF International Conference on Computer Vision (ICCV 2025)*.

[39] Curless, B., and Levoy, M. (1996). *A volumetric method for building complex models from range images*. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1996)*, pp. 303–312.

[40] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). *DeepSDF: Learning continuous signed distance functions for shape representation*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pp. 165–174.

[41] Kajiya, J. T., and Von Herzen, B. P. (1984). *Ray tracing volume densities*. *ACM SIGGRAPH Computer Graphics*, 18(3), pp. 165–174.

[42] Debevec, P., Hawkins, T., Tchou, C., Duiker, H. P., Sarokin, W., and Sagar, M. (2000). *Acquiring the reflectance field of a human face*. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000)*, pp. 145–156.

[43] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). *Occupancy networks: Learning 3D reconstruction in function space*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pp. 4460–4470.

[44] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). *Poisson surface reconstruction*. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, Vol. 7, No. 4.

[45] Lorensen, W. E., and Cline, H. E. (1998). *Marching cubes: A high resolution 3D surface construction algorithm*. In *Seminal Graphics: Pioneering Efforts That Shaped the Field*, pp. 347–353.

[46] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., et al. (2011). *KinectFusion: Real-time dense surface mapping and tracking*. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2011)*, pp. 127–136.

[47] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). *PointNet: Deep learning on point sets for 3D classification and segmentation*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pp. 652–660.

[48] Hartley, R., and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

[49] Peterson, L. E. (2009). *K-nearest neighbor*. *Scholarpedia*, 4(2), 1883.

[50] Osher, S., Fedkiw, R., and Piechor, K. (2004). *Level set methods and dynamic implicit surfaces*. *Applied Mechanics Reviews*, 57(3), pp. B15–B15.

[51] Yariv, L., Gu, J., Kasten, Y., and Lipman, Y. (2021). *Volume rendering of neural implicit surfaces*. *Advances in Neural Information Processing Systems*, 34, pp. 4805–4815.

[52] Müller, T., Evans, A., Schied, C., and Keller, A. (2022). *Instant neural graphics primitives with a multiresolution hash encoding*. *ACM Transactions on Graphics*, 41(4), pp. 1–15.

[53] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). *The unreasonable effectiveness of deep features as a perceptual metric*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, pp. 586–595.

[54] Kerbl, B., Kopanas, G., Leimkühler, T., and Drettakis, G. (2023). *3D Gaussian splatting for real-time radiance field rendering*. *ACM Transactions on Graphics*, 42(4).

[55] Matsuki, H., Murai, R., Kelly, P. H., and Davison, A. J. (2024). *Gaussian splatting SLAM*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024)*, pp. 18039–18048.

[56] Chen, A., Xu, Z., Geiger, A., Yu, J., and Su, H. (2022). *TensoRF: Tensorial radiance fields*. In *Proceedings of the European Conference on Computer Vision (ECCV 2022)*, pp. 333–350.

[57] Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., and Kanazawa, A. (2023). *Nerfstudio: A modular framework for neural radiance field development*. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*.

[58] Kehoe, B., Patil, S., Abbeel, P., and Goldberg, K. (2015). *A survey of research on cloud robotics and automation*. *IEEE Transactions on Automation Science and Engineering*, 12(2), pp. 398–409.

[59] Waibel, M., Beetz, M., Civera, J., d'Andrea, R., Elfring, J., Galvez-Lopez, D., et al. (2011). *RoboEarth*. *IEEE Robotics & Automation Magazine*, 18(2), pp. 69–82.

[60] Chen, L., Tang, W., John, N. W., Wan, T. R., and Zhang, J. J. (2018). *SLAM-based dense surface reconstruction in monocular minimally invasive surgery and its application to augmented reality*. *Computer Methods and Programs in Biomedicine*, 158, pp. 135–146.

[61] Li, J. Q., Zhang, Y. F., Chen, Z. Z., Wang, J., Fang, M., Luo, C. W., and Wang, H. (2020). *A novel edge-enabled SLAM solution using projected depth image information*. *Neural Computing and Applications*, 32(19), pp. 15369–15381.

[62] Cunningham, A., Indelman, V., and Dellaert, F. (2013). *DDF-SAM 2.0: Consistent distributed smoothing and mapping*. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2013)*, pp. 5220–5227.

[63] Schmuck, P., Ziegler, T., Karrer, M., Perraudin, J., and Chli, M. (2021). *Covins: Visual-inertial SLAM for centralized collaboration*. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct 2021)*, pp. 171–176.

[64] Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., and Yang, M. H. (2018). *Flow-grounded spatial-temporal video prediction from still images*. In *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, pp. 600–615.

[65] Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., and Sastry, S. S. (2007). *Foundations of control and estimation over lossy networks*. *Proceedings of the IEEE*, 95(1), pp. 163–187.

[66] Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M. I., and Sastry, S. S. (2004). *Kalman filtering with intermittent observations*. *IEEE Transactions on Automatic Control*, 49(9), pp. 1453–1464.

[67] Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., and Martin-Brualla, R. (2021). *Nerfies: Deformable neural radiance fields*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021)*, pp. 5865–5874.

[68] Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). *A survey of recent results in networked control systems*. *Proceedings of the IEEE*, 95(1), pp. 138–162.

[69] Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., and Chen, M. (2019). *In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning*. *IEEE Network*, 33(5), pp. 156–165.

[70] Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., and Tang, L. (2017). *Neurosurgeon: Collaborative intelligence between the cloud and mobile edge*. *ACM SIGARCH Computer Architecture News*, 45(1), pp. 615–629.

[71] Kadota, I., Sinha, A., Uysal-Biyikoglu, E., Singh, R., and Modiano, E. (2018). *Scheduling policies for minimizing age of information in broadcast wireless networks*. *IEEE/ACM Transactions on Networking*, 26(6), pp. 2637–2650.

[72] Chen, Z., Xu, M., She, C., Jia, Y., Wang, M., and Li, Y. (2023). *Improving timeliness-fidelity tradeoff in wireless sensor networks: Waiting for all and waiting for partial sensor nodes*. *IEEE Transactions on Communications*, 71(7), pp. 4151–4164.

[73] Fischer, W., and Meier-Hellstern, K. (1993). *The Markov-modulated Poisson process (MMPP) cookbook*. *Performance Evaluation*, 18(2), pp. 149–171.

[74] Hou, Z., She, C., Li, Y., Quek, T. Q., and Vucetic, B. (2018). *Burstiness-aware bandwidth reservation for ultra-reliable and low-latency communications in tactile Internet*. *IEEE Journal on Selected Areas in Communications*, 36(11), pp. 2401–2410.

[75] Gilbert, E. N. (1960). *Capacity of a burst-noise channel*. *Bell System Technical Journal*, 39(5), pp. 1253–1265.

[76] Jocher, G., Qiu, J., and Chaurasia, A. (2023). *Ultralytics YOLO (Version 8.0.0) [Computer software]*. Available at: `https://github.com/ultralytics/ultralytics`.

[77] Ghraieb, H., Viquerat, J., Larcher, A., Meliga, P., and Hachem, E. (2021). *Single-step deep reinforcement learning for open-loop control of laminar and turbulent flows*. *Physical Review Fluids*, 6(5), p. 053902.

[78] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). *Proximal policy optimization algorithms*. arXiv preprint arXiv:1707.06347.

[79] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015). *High-dimensional continuous control using generalized advantage estimation*. arXiv preprint arXiv:1506.02438.

[80] Xu, L., Agrawal, V., Laney, W., Garcia, T., Bansal, A., Kim, C., et al. (2023). *VR-NeRF: High-fidelity virtualized walkable spaces*. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–12.

[81] Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., and Zhou, X. (2021). *Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*.

[82] Fang, Q., Shuai, Q., Dong, J., Bao, H., and Zhou, X. (2021). *Reconstructing 3D human pose by watching humans in the mirror*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*.

[83] Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., et al. (2024). *PGSR: Planar-based Gaussian splatting for efficient and high-fidelity surface reconstruction*. *IEEE Transactions on Visualization and Computer Graphics*, 31(9), pp. 6100–6111.

[84] Tsai, R. Y., and Lenz, R. K. (1989). *A new technique for fully autonomous and efficient 3D robotics hand/eye calibration*. *IEEE Transactions on Robotics and Automation*, 5(3), pp. 345–358.

[85] Anderson, R. J., and Spong, M. W. (1992). *Asymptotic stability for force reflecting teleoperators with time delay*. *The International Journal of Robotics Research*, 11(2), pp. 135–149.

[86] Niemeyer, G., and Slotine, J. J. (1998). *Towards force-reflecting teleoperation over the internet*. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1998)*, Vol. 3, pp. 1909–1915.

[87] Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). *A survey of recent results in networked control systems*. *Proceedings of the IEEE*, 95(1), pp. 138–162.

[88] Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. Cambridge, MA: MIT Press.

[89] Drascic, D., and Milgram, P. (1996). *Perceptual issues in augmented reality*. In *Stereoscopic Displays and Virtual Reality Systems III*, Vol. 2653, pp. 123–134.

[90] Tao, F., Zhang, H., Liu, A., and Nee, A. Y. (2018). *Digital twin in industry: State-of-the-art*. *IEEE Transactions on Industrial Informatics*, 15(4), pp. 2405–2415.

[91] Schönberger, J. L., and Frahm, J.-M. (2016). *Structure-from-Motion revisited*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*.

[92] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). *A comparison and evaluation of multi-view stereo reconstruction algorithms*. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, Vol. 1, pp. 519–528.

[93] NVIDIA (2024). *Isaac Sim (Version 5.1.0) [Computer software]*. Available at: `https://github.com/isaac-sim/IsaacSim`.

[94] Jiang, W., Lei, B., and Daniilidis, K. (2024). *FisherRF: Active view selection and mapping with radiance fields using Fisher information*. In *Proceedings of the European Conference on Computer Vision (ECCV 2024)*, pp. 422–440.

[95] Howard, A. (2006). *Multi-robot simultaneous localization and mapping using particle filters*. *The International Journal of Robotics Research*, 25(12), pp. 1243–1256.

[96] Schmuck, P., and Chli, M. (2019). *CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams*. *Journal of Field Robotics*, 36(4), pp. 763–781.

[97] Furukawa, Y., and Ponce, J. (2009). *Accurate, dense, and robust multiview stereopsis*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8), pp. 1362–1376.

[98] Schönberger, J. L., Zheng, E., Frahm, J. M., and Pollefeys, M. (2016). *Pixelwise view selection for unstructured multi-view stereo*. *In European Conference on Computer Vision* (pp. 501–518). Cham: Springer International Publishing.

[99] Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D. (2021). *NeRF in the wild: Neural radiance fields for unconstrained photo collections*. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7210–7219).

[100] Ferrell, W. R. (1966). *Delayed force feedback*. *Human Factors*, 8(5), pp. 449–455.

[101] Mourikis, A. I., and Roumeliotis, S. I. (2007). *A multi-state constraint Kalman filter for vision-aided inertial navigation*. *In Proceedings 2007 IEEE International Conference on Robotics and Automation* (pp. 3565–3572). IEEE.

[102] Bloesch, M., Omari, S., Hutter, M., and Siegwart, R. (2015). *Robust visual inertial odometry using a direct EKF-based approach*. *In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 298–304). IEEE.