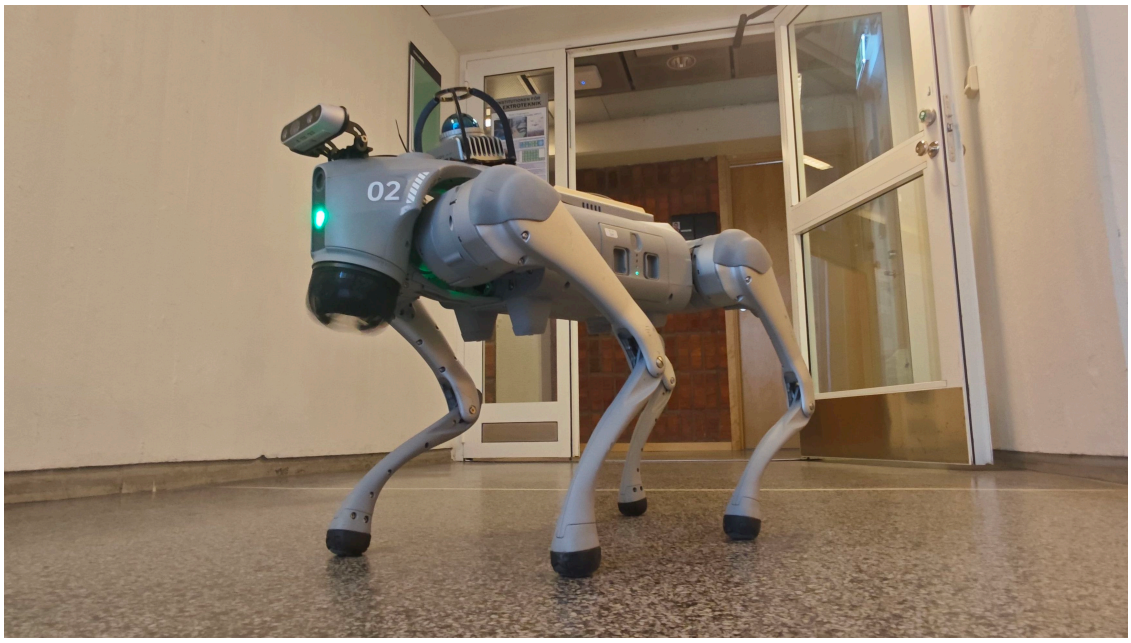
CHALMERS
UNIVERSITY OF TECHNOLOGY



Digital Twin Generation via Robotic SLAM for Wireless Ray-Tracing Applications

Master's thesis in Electrical Engineering

JIARUI HE
YUZHENG LONG

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2026

www.chalmers.se

Master's thesis 2026

Digital Twin Generation via Robotic SLAM for Wireless Ray-Tracing Applications

JIARUI HE
YUZHENG LONG

Department of Electrical Engineering
Information and Communication Technology
Chalmers University of Technology
Gothenburg, Sweden 2026

Digital Twin Generation via Robotic SLAM for Wireless Ray-Tracing Applications
JIARUI HE
YUZHENG LONG

Supervisor: Yu Ge, RAITech AB
Examiner: Henk Wymeersch, Department of Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering

Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The Unitree Go2 robot collects point cloud data in the experimental environment.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Digital Twin Generation via Robotic SLAM for Wireless Ray-Tracing Applications

JIARUI HE, YUZHENG LONG
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Wireless digital twins are useful for analyzing radio propagation in real environments. They can be used to study coverage, propagation paths, and channel characteristics before a wireless system is deployed. However, creating an accurate three-dimensional scene manually is time-consuming, especially in indoor environments with walls, doors, glass surfaces, corridors, and other structures that influence signal propagation. SLAM can provide a point cloud map of such environments, but this output cannot be directly used for wireless ray tracing. A point cloud only contains discrete points and usually does not include continuous surfaces, clean mesh geometry, or radio material information. This thesis proposes a workflow that converts robotic SLAM data into a wireless digital twin that can be used in Sionna RT.

In the proposed workflow, a mobile robot equipped with LiDAR, camera, and IMU sensors is used to collect indoor environmental data. The recorded data are processed by SLAM to reconstruct the main geometry of the environment as a point cloud map. The point cloud is then processed through filtering, surface reconstruction, mesh cleaning, simplification, and material assignment. After these steps, the model is exported as a Sionna RT-compatible scene. In Sionna RT, the transmitter and receiver positions, antenna settings, carrier frequency, and ray-tracing parameters are configured. The simulation is then used to compute propagation paths and channel results, including radio map, PDP, CFR, AoA, and AoD.

The results show that the generated digital twin can be used for ray-tracing-based wireless simulation. The simulation outputs also show that mapping quality, mesh completeness, surface roughness, coordinate consistency, and material assignment can influence the channel results. These factors affect path delay, received power, angular information, and frequency response. Overall, this thesis shows that robotic SLAM and Sionna RT can be combined to generate indoor wireless digital twins, and it provides a practical workflow for converting real indoor mapping data into a ray-tracing scene.

Keywords: Digital Twin; SLAM; LiDAR-inertial mapping; point cloud reconstruction; Sionna RT; ray tracing; wireless channel simulation.

Acknowledgements

Firstly, we would like to express our sincere gratitude to Professor Henk Wymeersch for providing us with this valuable master's thesis topic and for making the necessary research resources available for the experiments. His guidance helped us identify a meaningful research direction and laid the foundation for this work.

We are also deeply grateful to our supervisor, Yu Ge, for his continuous and patient guidance throughout the thesis process. His technical suggestions and detailed feedback greatly supported our research and helped us improve the quality of this thesis. Without his careful supervision, this work would have been much more difficult to complete successfully. We would also like to thank Yuhao Zhang for his support during the project, which contributed greatly to the smooth progress of this work.

We will always cherish the time we spent at Chalmers and in Gothenburg. It was a truly memorable experience, and even decades from now, we believe we will still look back on those days with a smile.

Finally, we would like to thank ourselves for the effort, persistence, and commitment we devoted to this project. Completing this work within more than one hundred days was not easy, and it required continuous learning, problem solving, and cooperation. During this process, music also gave us motivation and emotional support in many difficult moments.

JIARUI HE & YUZHENG LONG, Gothenburg, MAY 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AoA	Angle of Arrival
AoD	Angle of Departure
CFR	Channel Frequency Response
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
LoS	Line-of-Sight
NLoS	Non-Line-of-Sight
PDP	Power Delay Profile
RMS	Root Mean Square
SLAM	Simultaneous Localization and Mapping
SNR	Signal-to-Noise Ratio

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i	Index for a LiDAR point or sampled point
l	Index for a propagation path
t	Index for time step or LiDAR scan
k	Index for subcarrier

Sets

$\mathcal{P}$	Set of valid propagation paths
$\mathcal{M}$	Environment map or reconstructed map
$\mathcal{G}$	Receiver grid used for radio map generation
V_j	Set of points inside the j-th voxel
$\mathcal{N}_{K_{\text{nn}}}(\mathbf{p}_i)$	Set of K_{nn} nearest neighbors of point $\mathbf{p}_i$
$\mathcal{N}_i$	Local neighborhood of point $\mathbf{p}_i$ used for normal estimation

Parameters

c	Speed of light
f_c	Carrier frequency
λ	Wavelength
d	Propagation path length
Δd	Propagation path length error
$\Delta\tau$	Propagation delay error
K	Number of subcarriers
L	Number of valid propagation paths
N	Number of LiDAR points or registration residuals
P_l	Power of the l-th propagation path
$\mathbf{n}_i$	Normal vector of the local map surface corresponding to the i-th point
$\mathbf{q}_i$	Corresponding point on the local map surface
K_{nn}	Number of nearest neighbors used for neighbor-distance estimation
μ_d	Mean value of the average neighbor distances
σ_d	Standard deviation of the average neighbor distances
α	Standard deviation multiplier used for statistical outlier removal

Variables

$\mathbf{x}_t$	Robot state at time step t
$\mathbf{u}_t$	Motion input or inertial measurement at time step t
$\mathbf{z}_t$	Sensor observation at time step t
$\mathbf{m}$	Environment map estimated by SLAM
$\mathbf{w}_t$	Process noise in the SLAM state model
$\mathbf{v}_t$	Observation noise in the SLAM measurement model
$\mathbf{p}_{i,t}^{l}$	The i-th LiDAR point in the LiDAR coordinate frame at scan t

$\mathbf{p}_{i,t}^{w}$	The i-th LiDAR point transformed into the world coordinate frame
$\mathbf{R}_t$	Rotation matrix from the LiDAR frame to the world frame at scan t
$\mathbf{t}_t$	Translation vector from the LiDAR frame to the world frame at scan t
r_i	Point-to-plane registration residual of the i-th point
a_l	Complex gain of the l-th propagation path
τ_l	Propagation delay of the l-th path
$\bar{\tau}$	Power-weighted mean delay
τ_{rms}	RMS delay spread
$h(\tau)$	Channel impulse response in the delay domain
$H(f)$	Channel frequency response
$H(f_k)$	Channel frequency response at the k-th subcarrier
$\overline{P}_h$	Average channel power over the selected subcarriers
$\mathbf{p}_i$	Generic 3D point in the point cloud
$\tilde{\mathbf{p}}_j$	Representative downsampled point of the j-th voxel
$\bar{d}_i$	Mean neighbor distance of point $\mathbf{p}_i$
$\mathbf{C}_i$	Local covariance matrix around point $\mathbf{p}_i$
$\bar{\mathbf{p}}_i$	Centroid of the local neighborhood around point $\mathbf{p}_i$
χ	Implicit function used in Poisson surface reconstruction
$\mathbf{V}$	Vector field defined by the oriented point normals

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

A wireless digital twin is a digital environment model built for wireless communication scenarios. It not only represents the geometric structure of a real space, but also provides the scene information required by wireless propagation models, such as material information and transmitter and receiver locations, and wireless propagation models to analyze signal behavior in a specific environment. Unlike a common three-dimensional model, a wireless digital twin is not mainly used for visual presentation. Its main purpose is to support practical tasks such as wireless network planning, coverage analysis, channel modeling, weak-coverage identification, and deployment evaluation [2, 3, 4]. In practical environments such as office buildings, factories, campuses, stations, and indoor public spaces, this type of model can help engineers evaluate wireless network performance before real deployment. For example, it can be used to predict received power, propagation paths, multipath effects, and coverage blind spots at different locations. This can reduce repeated on-site testing and manual adjustment, while also providing a more realistic simulation basis for future 6G, smart factories, indoor positioning, and integrated sensing and communication applications.

However, wireless propagation in real environments is usually complex because it is affected by building structures and materials, blockage relationships, and spatial layout. Traditional simulation environments often rely on idealized or manually created scene models, which makes it difficult to quickly represent the real structure of a specific site. Therefore, an efficient method is needed to build wireless simulation environments from real scenes, so that the simulation model can stay close to the actual space rather than only relying on simplified assumptions.

Robotic SLAM provides a feasible way to address this problem. Compared with traditional fixed scanning devices or handheld scanning methods, a mobile robot

can carry sensors such as Light Detection and Ranging (LiDAR), cameras, and Inertial Measurement Units (IMUs) into the target area to collect spatial data. A quadruped robot also provides good stability and controllability, which makes it suitable for data collection in corridors, buildings, narrow spaces, or environments where long manual operation is inconvenient. After a series of processing steps, the data collected by the robot can be converted into a digital twin scene suitable for wireless ray tracing.

Sionna RT can then use this scene to compute channel-related results such as propagation paths, received power, power delay profile (PDP), channel frequency response (CFR), angle of arrival (AoA), and angle of departure (AoD)[5, 6]. Therefore, this thesis focuses on building a complete workflow from robotic data collection in a real environment, through data processing and model reconstruction, to Sionna RT-based wireless simulation. The purpose is to develop a practical workflow for generating a digital twin for wireless propagation analysis.

1.2 Related Work

Previous work related to this thesis can be grouped into three areas: wireless digital twins, robotic mapping, and ray-tracing-based wireless simulation. In wireless communications, digital twins have been studied as site-specific models for network planning, channel modeling, coverage analysis, and network optimization in future wireless systems [2, 4]. Some recent works have further shown how digital twins can be used as ray-tracing environments for 6G-related studies. For example, the Boston Twin project presents a city-scale digital twin for ray-tracing research, while other studies have proposed high-precision digital twin platforms based on ray-tracing simulation [7, 8]. These works show that digital twins can provide a more realistic geometric basis for wireless propagation analysis than purely statistical models or idealized simulation scenes. However, these studies mainly focus on prepared or manually constructed digital twin scenes, while the automatic conversion from robotic SLAM outputs to a ray-tracing-ready indoor model remains less explored.

Robotic SLAM has also been widely studied as a key technique for mobile robot perception and three-dimensional mapping. Existing surveys and studies have discussed the use of LiDAR, cameras, IMUs, and odometry for estimating robot motion and constructing environment maps [9, 10, 11]. LiDAR-based and LiDAR-inertial SLAM methods are especially useful when stable geometric reconstruction is required, because they can capture walls, floors, corridors, and other large structures

in indoor environments. FAST-LIO2, for example, combines LiDAR measurements with IMU data to provide efficient LiDAR-inertial odometry and mapping [12].

For wireless channel simulation, ray tracing has been used to compute propagation paths, reflection components, received power, delay, angular information, and channel responses from three-dimensional scenes. Sionna RT is one recent tool for this purpose. It supports radio propagation modeling based on scene geometry, radio materials, antenna settings, and transmitter and receiver locations [5, 6]. Such tools make it possible to convert a prepared three-dimensional scene into channel-level outputs for wireless analysis. Recent studies have further extended the concept of wireless digital twins toward digital twin networks and digital twin channels, where the digital replica is used not only for visualization but also for channel prediction, network planning, and what-if analysis[2, 13].

Overall, these works provide the technical basis for this thesis. However, a practical workflow that connects robotic mapping outputs with a Sionna RT-ready wireless simulation scene still requires further development.

1.3 Research Gap and Problem Statement

Although wireless digital twins, robotic SLAM, and wireless ray tracing have been widely studied as separate topics, the complete workflow from robotic real-environment data collection to Sionna RT-based wireless simulation is still not straightforward. Existing SLAM systems usually output trajectories and point clouds. These results can be used for mapping and visualization, but they cannot be directly used as simulation scenes for wireless ray tracing. A point cloud does not contain continuous surfaces or radio material information, and the initially reconstructed mesh may still contain noise, holes, redundant faces, or excessive geometric complexity. These issues can affect both the efficiency of the later simulation and the computation of propagation paths.

Therefore, the problem addressed in this thesis is how to convert real-environment data collected by a robotic platform and reconstructed through SLAM into a clean, simplified, material-labeled wireless digital twin model that can be correctly loaded and used by Sionna RT. The focus is not only to generate a three-dimensional model, but also to build a reliable data conversion workflow, so that the spatial structure of a real scene can be further used for wireless propagation analysis.

1.4 Objectives and Research Questions

The main objective of this thesis is to build a complete workflow from robotic data collection, SLAM reconstruction, and digital twin generation to Sionna RT-based wireless simulation. The workflow converts real indoor environment data into a scene model suitable for ray tracing, and then extracts wireless propagation and channel-related results.

The specific objectives are to collect multi-sensor data from a real indoor environment, reconstruct a point cloud map using SLAM, convert the point cloud into a mesh-based digital twin, clean and simplify the mesh, assign material labels, and import the processed scene into Sionna RT for wireless ray-tracing simulation.

Based on these objectives, this thesis addresses the following research questions:

- Can the robotic SLAM reconstruction accurately represent the main structure of the real indoor environment?
- How should the point cloud generated by SLAM be processed and converted into a simulation scene usable by Sionna RT?
- What wireless channel results can be obtained from ray tracing based on the reconstructed model?
- How can mapping and mesh quality affect the simulated channel characteristics?

1.5 Contributions and Scope of Claims

The main contribution of this thesis is an end-to-end engineering workflow that connects robotic LiDAR-inertial SLAM reconstruction with Sionna RT-based wireless ray-tracing simulation. This thesis does not propose a new SLAM algorithm or a new ray-tracing solver. Instead, it focuses on the practical integration of existing robotic mapping, point cloud processing, mesh generation, scene editing, material assignment, and wireless simulation tools into one complete pipeline.

The specific contributions of this thesis are summarized as follows:

- A ROS 2-based robotic data collection workflow is implemented using a Unitree Go2 platform with LiDAR, IMU, RGB-D camera, and odometry-related topics. The collected data provide the input for the later SLAM reconstruction stage.
- ICP, FAST-LIO2, and RTAB-Map are compared for the purpose of wireless digital twin generation. Based on structural completeness, visible drift, local

consistency, and suitability for later mesh generation, FAST-LIO2 is selected as the main reconstruction method.

- A conversion pipeline from SLAM point cloud to Sionna RT-compatible digital twin is developed. The pipeline includes point cloud preprocessing, mesh reconstruction, mesh simplification, Blender-based scene editing, material assignment, and export to a Mitsuba/Sionna-compatible scene format.
- The generated mesh is evaluated using both visual inspection and geometric dimension validation. Several main dimensions of the generated mesh are compared with manual measurements from the real environment to check whether the main spatial scale is preserved.
- A Sionna RT-based simulation demonstration is carried out using the generated digital twin. The simulation extracts and analyzes channel-related results such as propagation paths, path gain, PDP, CFR, AoA, and AoD, and compares representative LoS and NLoS cases.

The scope of the claims in this thesis is also limited. The generated digital twin is used to demonstrate the feasibility of the proposed robotic-to-ray-tracing workflow, but it is not claimed to be a fully accurate copy of the real environment. The material assignment is based on simplified surface categories and available Sionna RT material models, rather than measured electromagnetic material parameters. In addition, the ray-tracing results are not calibrated against real wireless channel measurements. Therefore, the results should be interpreted as a feasibility demonstration and an engineering evaluation of the proposed pipeline, rather than as fully measurement-calibrated channel predictions.

1.6 Thesis Outline

The rest of this thesis is organized as follows.

Chapter 2 introduces the background and theory related to wireless digital twins, robotic SLAM, point cloud and mesh representation, wireless ray tracing, and Sionna RT. Chapter 3 presents the overall system workflow and explains how the main modules are connected. Chapter 4 describes the robotic data collection setup, including the platform, sensors, ROS 2 recording system, and data quality issues.

Chapter 5 explains the SLAM reconstruction process and the generation of the point cloud map. Chapter 6 describes how the SLAM output is converted into a digital twin scene through point cloud preprocessing, surface reconstruction, mesh editing, material assignment, and scene export. Chapter 7 presents the Sionna RT

simulation workflow and the main simulation parameters. Chapter 8 reports and evaluates the robotic mapping, digital twin generation, and ray-tracing simulation results. Finally, Chapter 9 summarizes the thesis, discusses the main limitations, and suggests possible future work.

2

Background and Theory

This chapter introduces the main theoretical background required for this thesis. It explains how wireless digital twins, robotic SLAM, point cloud and mesh representation, surface reconstruction, wireless ray tracing, and Sionna RT are connected in the proposed pipeline. The purpose of this chapter is not to present detailed implementation steps or software parameters, but to provide the theoretical basis needed to understand the following chapters on system design, robotic data collection, SLAM reconstruction, digital twin generation, ray-tracing simulation, and result evaluation.

2.1 Digital Twins for Wireless Communications

2.1.1 Concept of Wireless Digital Twin

A digital twin is generally understood as a virtual representation of a physical object, system, or environment[14]. It is not only a static three-dimensional model, but can also combine measurement data, simulation models, and analysis methods to study the behavior of a real system under different conditions. In engineering applications, the value of a digital twin is that it connects the physical environment with a virtual analysis environment, allowing system design, testing, and optimization to be carried out in a controlled simulation setting [15].

In wireless communications, the geometry and material properties of the environment directly affect radio propagation [2, 3, 4]. Walls, floors, glass surfaces, metal objects, door openings, corridor length, and the positions of transmitters and receivers can all change propagation paths, reflection conditions, blockage relationships, and received power. Therefore, a wireless digital twin should not be understood only as a three-dimensional model for visualization. It also needs to contain information that supports radio propagation analysis, such as geometric structure, surface materials, coordinate scale, and scene configuration for wireless simulation.

2.1.2 Role in Wireless Channel Analysis

Compared with traditional empirical models, a wireless digital twin can provide site-specific propagation analysis. Empirical path loss models usually describe the average propagation trend using distance and statistical parameters. In contrast, digital-twin-based ray tracing can use the actual environmental structure to analyze line-of-sight paths, reflection paths, delay, received power, angle of arrival, angle of departure, and channel frequency response between specific transmitter and receiver locations.

For this reason, wireless digital twins are useful for indoor network planning, coverage evaluation, channel characterization, and future 6G or integrated sensing and communication (ISAC) studies. In these applications, environmental geometry and material information are no longer only background information. They become important inputs that directly influence wireless channel results.

Recent studies have also demonstrated the use of digital twins as ray-tracing environments for future wireless networks. For example, the Boston Twin project presents a digital twin for ray-tracing-based 6G network studies, while other work has proposed high-precision digital twin platforms based on ray-tracing simulation [7, 8]. These studies show that digital twins can provide useful site-specific environments for wireless channel analysis. However, such platforms still rely on carefully prepared geometric models, which motivates the robotic SLAM-based digital twin generation pipeline developed in this thesis.

2.1.3 Connection to This Thesis

In this thesis, the digital twin connects robotic mapping with wireless simulation. The robotic platform collects data from the real environment using onboard sensors. The SLAM system generates a geometric map of the environment. The map is then processed through point cloud processing, surface reconstruction, mesh cleaning, and material assignment to form a three-dimensional scene suitable for wireless ray tracing. Finally, the generated scene is imported into Sionna RT for propagation path and channel computation.

Therefore, the digital twin in this thesis is not only the final three-dimensional model. It is also a simulation environment that supports wireless propagation analysis. Its quality can affect propagation paths, received power, PDP, RMS, CFR, and other channel-related results.

2.2 Robotic SLAM and Mapping

2.2.1 Basic Concept of SLAM

SLAM is one of the core techniques in mobile robot perception. Its basic goal is to estimate the pose of a moving robot while constructing a map of an unknown or partially unknown environment. A systematic overview of the development and basic concepts of SLAM can be found in the work by Cadena et al. [9].
For this thesis, SLAM is not only used to obtain the robot trajectory. More importantly, it provides the three-dimensional geometric foundation for later digital twin generation. A SLAM system aligns sensor observations collected at different times into a common coordinate frame, and produces outputs such as a robot trajectory and an environment map. SLAM can be formulated as a joint state estimation and mapping problem. The robot state is predicted from the previous state and motion input, while sensor observations are used to update the state and the map:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t, \qquad \mathbf{z}_t = h(\mathbf{x}_t, \mathbf{m}) + \mathbf{v}_t. \tag{2.1}$$

Here, $\mathbf{x}_t$ denotes the robot state at time t, $\mathbf{u}_t$ denotes the motion input or inertial measurement, $\mathbf{z}_t$ denotes the sensor observation, and $\mathbf{m}$ denotes the environment map. The terms $\mathbf{w}_t$ and $\mathbf{v}_t$ represent process noise and observation noise, respectively. This formulation shows why both reliable motion estimation and accurate sensor measurements are important for generating a consistent map.

2.2.2 Sensor Modalities for Robotic Mapping

SLAM systems can use different sensors, such as LiDAR, camera, IMU, and odometry. Different sensors provide different types of information. LiDAR can directly measure the three-dimensional geometry of surrounding surfaces. Cameras provide texture, color, and possible semantic information. IMUs provide high-rate angular velocity and acceleration measurements. Odometry can also be used as auxiliary motion information.
For robotic mapping aimed at wireless digital twin generation, reliable geometry is especially important. This is because later ray tracing depends on structures such as walls, floors, door frames, and glass partitions to calculate propagation paths. The quality of the geometric map therefore has a direct influence on the quality of the final wireless simulation scene.

2.2.3 LiDAR-based SLAM

LiDAR-based SLAM has the advantage of directly capturing environmental geometry and is less affected by lighting conditions[16]. In indoor corridors, laboratories, and teaching buildings, LiDAR can usually capture walls, floors, door frames, corners, and large static objects in a stable way. These structures are important for later mesh reconstruction and wireless ray tracing.

However, LiDAR-only SLAM also has limitations. In long corridors, repetitive structures, or environments with insufficient geometric features, point cloud registration may suffer from degeneracy. Degeneracy means that the environment does not provide enough geometric constraints in certain directions, making it difficult for the system to accurately estimate motion along those directions. For example, in a long straight corridor, side walls and the floor may provide some lateral and vertical constraints, but there may be limited geometric variation along the forward direction. This can lead to pose drift or scale error.

For normal map visualization, such errors may only appear as bending or stretching of the map. For a wireless digital twin, however, they can further affect path length, propagation delay, AoA, AoD, and received power. Therefore, mapping accuracy is directly related to the reliability of later channel simulation.

2.2.4 Visual and RGB-D SLAM

Visual SLAM and RGB-D SLAM can provide rich image information. RGB images can be used for feature matching and place recognition in visual SLAM systems[17]. They may also provide useful appearance cues for later semantic inspection or material assignment. For example, image information may help distinguish walls, glass, doors, metal objects, and furniture, which can be useful for later material assignment.

However, visual SLAM is sensitive to many practical factors. Illumination changes, motion blur, low-texture surfaces, reflective areas, and rolling-shutter distortion can affect image feature extraction and matching. RGB-D cameras also have limited depth range, and depth measurements can be missing or noisy on glass, metal, black surfaces, or strongly reflective regions. Therefore, if the main goal is to construct stable geometry for wireless ray tracing, relying only on visual or RGB-D SLAM may not be sufficiently robust.

2.2.5 LiDAR-inertial SLAM

LiDAR-inertial SLAM combines LiDAR point clouds with IMU measurements. LiDAR provides three-dimensional geometric observations, while the IMU provides high-rate angular velocity and linear acceleration information[9, 12]. The IMU can provide short-term motion estimation between consecutive LiDAR scans, which helps reduce motion distortion and provides a more stable initial pose for LiDAR registration.

This type of method is highly relevant to this thesis because it combines the geometric measurement capability of LiDAR with the high-rate motion information provided by the IMU[12]. It is suitable for indoor robotic mapping tasks, especially in corridors or environments with repetitive structures. The specific SLAM method selection is discussed later in the SLAM reconstruction chapter, where FAST-LIO2 is selected as the main reconstruction method.

2.2.6 SLAM Output for Digital Twin Generation

It should be emphasized that the SLAM output is not equal to the final digital twin. SLAM usually outputs a trajectory and a point cloud map. A point cloud is only a set of discrete spatial points, and it does not directly contain continuous surfaces, clear object boundaries, or radio material information.

Therefore, the role of SLAM in this thesis is to provide the geometric basis, not to directly generate the final Sionna RT-compatible scene. The following chapters further describe point cloud preprocessing, mesh reconstruction, mesh cleaning, and material assignment.

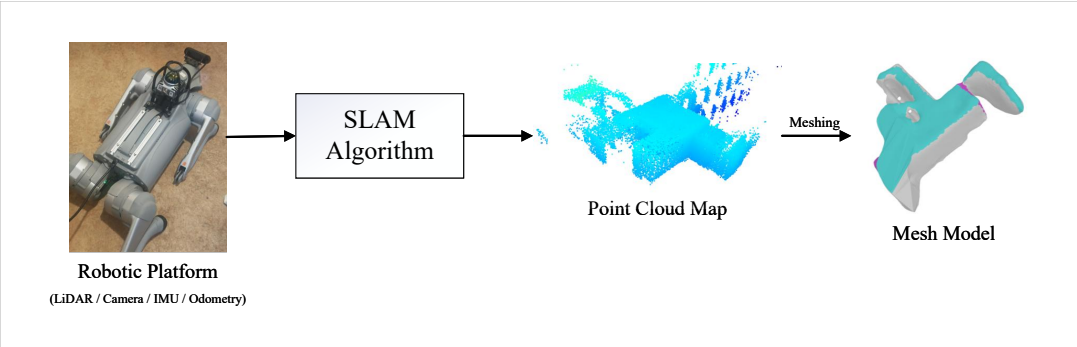


Figure 2.1: Basic input-output structure of the robotic SLAM pipeline.

Fig. 2.1 summarizes the high-level role of SLAM in the proposed workflow. Sensor data from the robotic platform are processed by the SLAM algorithm to generate a

point cloud map, which is then converted into a mesh model for later digital twin generation and ray-tracing simulation.

2.3 Point Cloud, Mesh, and Surface Reconstruction

2.3.1 Point Cloud Representation

A point cloud is a geometric representation composed of discrete points in three-dimensional space. Each point usually contains spatial coordinates, and may also contain additional information such as intensity, color, timestamp, or normal direction[10, 11]. In robotic mapping, point clouds are commonly generated by LiDAR or depth cameras. They can directly represent observed environmental structures, such as walls, floors, door frames, and large indoor objects.

The advantage of a point cloud is that it is close to the original sensor measurement. It is relatively simple to store and process, and it can preserve sampled geometric information from the real environment. However, a point cloud does not explicitly describe continuous surfaces. There are no faces between points, and surface boundaries are not directly defined. For wireless ray tracing, propagation paths interact with continuous surfaces such as walls, floors, and glass, rather than isolated points. Therefore, a raw point cloud is not an ideal ray-tracing scene.

2.3.2 Mesh Representation

A mesh is a surface-based representation, usually composed of vertices, edges, and faces. Compared with a point cloud, a mesh explicitly represents surface geometry. This makes it more suitable for visualization, material assignment, physical simulation, and ray tracing. In a wireless digital twin, the role of a mesh is to convert discrete point cloud data into a continuous geometric scene that can be used by a ray-tracing engine. Main structures such as walls, floors, ceilings, and large static objects should form complete, clear, and properly scaled surfaces. With such surfaces, tools such as Sionna RT can calculate reflection, blockage, penetration, and multipath propagation.

2.3.3 Surface Reconstruction

The conversion from point cloud to mesh usually requires surface reconstruction. The goal of surface reconstruction is to estimate a continuous surface from discrete points. Common methods include local triangulation methods and implicit surface reconstruction methods.

Poisson surface reconstruction is a typical implicit reconstruction method. It uses oriented point clouds with normal vectors to estimate a continuous surface, and then extracts a mesh from the implicit representation [18, 19]. Poisson-based methods are suitable for generating continuous surfaces from oriented point clouds, but they can also smooth local geometric details and may close openings if the input point cloud contains missing regions.

2.3.4 Quality Requirements for Ray Tracing

The quality of surface reconstruction strongly depends on the input point cloud. Noise points, floating points, incorrect normals, uneven point density, and large missing areas can all lead to mesh artifacts. For example, noise points may generate unrealistic small faces, missing regions may produce holes, and over-smoothing may remove important structures.

For ray-tracing simulation, mesh quality should be evaluated from a radio-propagation perspective rather than only from a visual perspective. A visually detailed model is not necessarily a better wireless simulation model. The mesh should preserve large propagation-relevant surfaces such as walls, floors, ceilings, doors, and glass regions, while avoiding unnecessary small fragments and noisy geometry. Correct scale and coordinate consistency are also essential, because path delay, AoA, AoD, and received power are all computed from the spatial relationship between the scene, transmitter, and receiver.

Therefore, when generating a ray-tracing-ready digital twin, the goal is not to create the most complex or visually detailed mesh. Instead, the goal is to generate a model with clear geometry, complete propagation-relevant surfaces, manageable computational complexity, and surfaces suitable for material assignment. This principle also affects the later choices of point cloud preprocessing, mesh cleaning, and simplification.

2.4 Ray Tracing for Wireless Channels

2.4.1 Basic Principle of Ray Tracing

Ray tracing is a wireless propagation modeling method based on geometrical optics. It models the wireless signal as rays emitted from the transmitter and traces their propagation in a three-dimensional environment. Unlike empirical path loss models, ray tracing explicitly considers environmental geometry, material properties, TX/RX positions, and interactions between rays and surfaces. Therefore, it is suitable for site-specific channel simulation based on digital twins.

2.4.2 Propagation Mechanisms

In indoor environments, wireless propagation usually includes not only the direct path, but also reflection, scattering, diffraction, and penetration caused by walls, floors, glass surfaces, door frames, and other structures. A line-of-sight (LoS) path usually has the shortest propagation distance and lower delay. Reflection paths increase propagation distance and may introduce multipath effects. Penetration and scattering are related to material properties and surface roughness.
However, in a specific ray-tracing simulation, only the propagation mechanisms enabled in the solver configuration are considered. Therefore, mechanisms that may exist in theory do not necessarily appear in the simulation results. In this thesis, the later Sionna RT setup focuses mainly on LoS paths and reflection-related multipath components, while other mechanisms are treated according to the selected simulation settings. The electromagnetic properties of materials affect reflection loss, penetration loss, and phase change. Therefore, material assignment is an important part of wireless digital twin generation. The same geometry can lead to different path gain, received power, and CFR if different material settings are used.

2.4.3 Path-level Channel Parameters

The basic output of ray tracing is usually a set of path-level parameters. For each valid propagation path, ray tracing can provide path length, path delay, path gain, interaction points, angle of arrival, and angle of departure. Among them, path delay is directly related to the propagation path length:

$$\tau = \frac{d}{c}, \tag{2.2}$$

where d is the path length and c is the speed of light. Therefore, if the geometry scale or structure position of the digital twin contains errors, the ray-tracing delay will also be affected. This is one reason why mapping quality is important in this thesis: geometric errors affect not only visual appearance, but also wireless channel computation.

2.4.4 Channel Metrics from Multipath Results

From multipath results, higher-level channel metrics can be further derived. For a multipath channel with L valid propagation paths, the $(l)-th$ path can be described by its complex path gain a_l, propagation delay τ_l, angle of arrival, and angle of departure. The complex gain a_l contains both amplitude attenuation and phase information, while the delay τ_l is mainly determined by the propagation path length. The angular information describes the spatial direction of the path at the receiver and transmitter sides.

In a simplified single-antenna case, the channel impulse response can be written as

$$h(\tau) = \sum_{l=1}^{L} a_l \delta(\tau - \tau_l). \tag{2.3}$$

In this expression, $h(\tau)$ denotes the channel impulse response in the delay domain, L is the number of valid propagation paths, a_l is the complex gain of the $(l) - th$ path, τ_l is the propagation delay of the $(l) - th$ path, and $\delta(\cdot)$ is the Dirac delta function. This expression shows that the received channel response can be regarded as the superposition of multiple path components with different gains and delays.

The corresponding channel frequency response can be obtained from the multipath components as

$$H(f) = \sum_{l=1}^{L} a_l e^{-j2\pi f \tau_l}. \tag{2.4}$$

Here, $H(f)$ denotes the channel frequency response at frequency f, and j is the imaginary unit. This scalar form shows that the CFR is determined by the complex gains and delays of all valid propagation paths. If mapping or meshing changes the path set, path length, or reflection conditions, the PDP, RMS delay spread, and CFR may also change.

For antenna-array or MIMO channel modeling, the spatial directions of the propagation paths also need to be included. Inspired by multipath channel modeling in 5G and mmWave systems, each path can be related to the array responses at the

transmitter and receiver sides. A frequency-domain MIMO channel matrix can be written as

$$\mathbf{H}(f) = \sum_{l=1}^{L} a_l e^{-j2\pi f \tau_l} \mathbf{a}_{\mathrm{rx}}(\phi_l^{\mathrm{rx}}, \theta_l^{\mathrm{rx}}) \mathbf{a}_{\mathrm{tx}}^H(\phi_l^{\mathrm{tx}}, \theta_l^{\mathrm{tx}}). \tag{2.5}$$

In this expression, $\mathbf{H}(f)$ denotes the MIMO channel matrix at frequency f. The vector $\mathbf{a}_{\mathrm{rx}}$ is the receiver array response vector, and $\mathbf{a}_{\mathrm{tx}}$ is the transmitter array response vector. They are determined by the AoA and AoD of the $(l)-th$ propagation path, respectively. The superscript $(\cdot)^H$ denotes Hermitian transpose. The angles ϕ_l^{rx} and θ_l^{rx} describe the AoA direction at the receiver, while ϕ_l^{tx} and θ_l^{tx} describe the AoD direction at the transmitter. This model shows that a multipath channel is not only determined by path gain and delay, but also by the spatial directions of the propagation paths. Ge et al. also use delay, angle, and channel gain as important multipath channel parameters for clustering, assignment, and SLAM processing in a 5G SLAM framework [20].

In the current Sionna RT setup of this thesis, both the transmitter and receiver use a 1×1 isotropic antenna. Therefore, the transmitter and receiver array response vectors reduce to scalar responses, and the MIMO channel model in (2.5) reduces to the scalar CFR form in (2.4). However, Sionna RT still provides AoA and AoD for each propagation path. These angular results are useful for analyzing the spatial direction of multipath propagation and provide a basis for future extension to antenna-array or MIMO configurations.

Based on the path powers and delays, the PDP describes how received power is distributed over propagation delay. If $P_l = |a_l|^2$ denotes the power of the $(l) - th$ path, the power-weighted mean delay is

$$\bar{\tau} = \frac{\sum_{l=1}^{L} P_l \tau_l}{\sum_{l=1}^{L} P_l}. \tag{2.6}$$

$\bar{\tau}$ represents the average arrival delay of the multipath components weighted by their path powers. The RMS delay spread is then given by

$$\tau_{\mathrm{rms}} = \sqrt{\frac{\sum_{l=1}^{L} P_l (\tau_l - \bar{\tau})^2}{\sum_{l=1}^{L} P_l}}. \tag{2.7}$$

Here, τ_{rms} measures the time dispersion of the multipath channel. A larger τ_{rms} means that the received multipath components are more spread out in time, while a smaller value means that the received energy is more concentrated around the mean

delay. These quantities provide the theoretical basis for the later analysis of PDP, RMS delay spread, and CFR [21].

2.5 Sionna RT for Wireless Simulation

2.5.1 Role of Sionna RT

Sionna RT is the ray-tracing module of NVIDIA Sionna for radio propagation modeling. According to the Sionna RT paper and official documentation, it is a hardware-accelerated and differentiable ray tracer built on top of Mitsuba 3 and Dr.Jit [5, 1]. In wireless communication research, Sionna RT can calculate propagation paths and channel responses based on three-dimensional scenes, radio materials, antenna configuration, and transmitter and receiver positions.

For antenna-array-based channel modeling, the directional information of each propagation path can be related to the array response. For a general antenna array with M elements, the steering vector for a path arriving or departing from direction (ϕ, θ) can be written as

$$\mathbf{a}(\phi,\theta) = \left[e^{-j\frac{2\pi}{\lambda}\mathbf{r}_1^T\mathbf{u}(\phi,\theta)}, e^{-j\frac{2\pi}{\lambda}\mathbf{r}_2^T\mathbf{u}(\phi,\theta)}, \ldots, e^{-j\frac{2\pi}{\lambda}\mathbf{r}_M^T\mathbf{u}(\phi,\theta)}\right]^T. \tag{2.8}$$

Here, λ is the wavelength, $\mathbf{r}_m$ is the position vector of the m-th antenna element relative to the array reference point, and $\mathbf{u}(\phi,\theta)$ is the unit direction vector defined by the azimuth angle ϕ and elevation or zenith angle θ. In this thesis, the current Sionna RT setup uses a 1×1 isotropic antenna at both the transmitter and receiver, so $M = 1$ and the steering vector reduces to a scalar response. However, the AoA and AoD extracted from Sionna RT still describe the spatial directions of the propagation paths and can be used for later extension to multi-antenna or MIMO configurations.

In this thesis, Sionna RT is the simulation tool that converts the mesh-based digital twin into wireless channel results. It is not used to create the geometry, but to evaluate radio propagation based on the geometry and materials prepared in the previous stages.

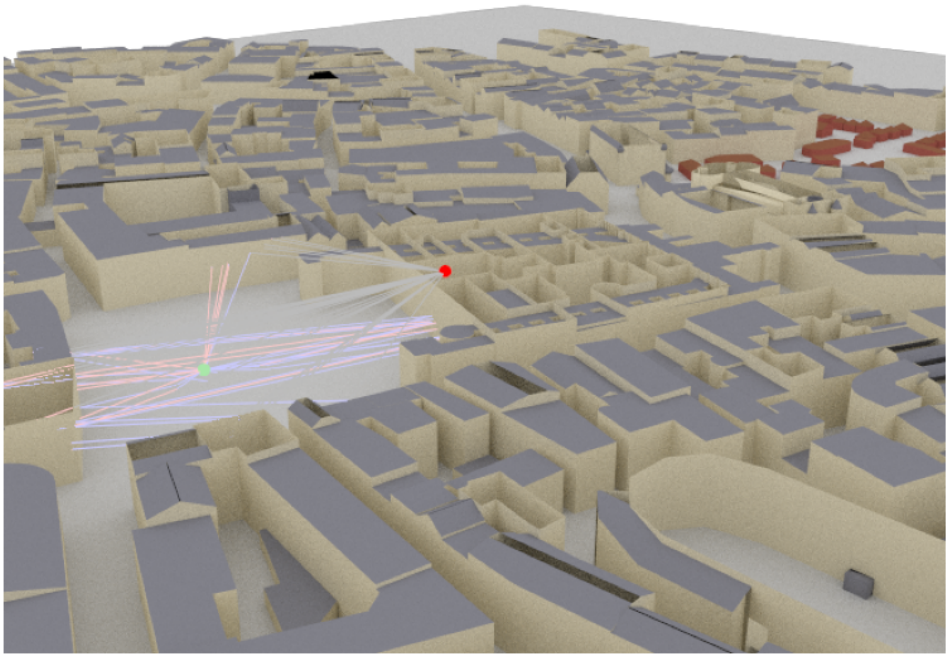


Figure 2.2: Sionna RT scene with radio propagation paths. [1]. Reproduced from NVIDIA.

Fig. 2.2 shows an example of a Sionna RT scene with computed propagation paths. This type of visualization helps check whether the scene geometry, TX/RX positions, and ray trajectories are physically reasonable.

2.5.2 Main Inputs and Outputs

In this thesis, the input to Sionna RT is not the raw SLAM point cloud. Instead, it is a three-dimensional scene after point cloud processing, surface reconstruction, mesh cleaning, and material assignment. The scene should have reasonable scale, correct coordinate orientation, clear surface geometry, and recognizable radio material settings.

The main inputs of Sionna RT include scene geometry, radio materials, carrier frequency, antenna arrays, transmitter configuration, receiver configuration, and ray-tracing parameters. Its outputs can include propagation paths, path gain, path delay, AoA, AoD, CIR, CFR, received power, and radio maps.

In this chapter, radio maps are only introduced as a type of spatial simulation output. An additional official radio map example is not included here, since the later result chapter presents radio maps and simulation figures generated from the

digital twin developed in this thesis.

2.5.3 Relation to the Proposed Pipeline

The results produced by Sionna RT depend on the quality of all previous stages. If the SLAM map contains drift or missing structures, the propagation paths in ray tracing may change. If the mesh surface is broken or too rough, unrealistic reflections may be produced. If material assignment is inaccurate, path gain and received power may deviate from the expected behavior.
Therefore, Sionna RT is not only a simulation tool in this thesis, but also a way to evaluate whether the generated digital twin is suitable for wireless propagation analysis. Detailed parameter settings, PathSolver and RadioMapSolver usage, and result extraction are described later in the ray-tracing simulation chapter.

2.6 Summary

This chapter introduced the main theoretical background required for this thesis and explained the relationship between wireless digital twins, robotic SLAM, point clouds, meshes, surface reconstruction, ray tracing, and Sionna RT. A wireless digital twin is not only a three-dimensional model for visualization, but also a simulation-ready representation of the physical environment. Its geometry, material properties, coordinate scale, and TX/RX configuration directly affect LoS and NLoS propagation, multipath reflections, received power, and channel responses.
The chapter then discussed the role of robotic SLAM in digital twin generation. SLAM can use LiDAR, camera, IMU, or multi-sensor fusion to capture the spatial structure of a real environment and produce point cloud maps for further processing. However, raw point clouds are usually not suitable for direct use in Sionna RT, since ray tracing requires a mesh-based scene with continuous surfaces, reasonable scale, limited holes, and assigned radio materials. Therefore, the point cloud must be processed through filtering, registration, downsampling, surface reconstruction, mesh cleaning, and material assignment before it can be converted into a digital twin for wireless simulation.
Finally, the chapter explained the basic principle of ray tracing and its role in wireless channel modeling. By computing propagation paths, path gain, delay, AoA, AoD, PDP, RMS delay spread, and CFR, ray tracing provides more site-specific channel analysis than empirical propagation models. In this thesis, Sionna RT performs

this simulation step by using the processed digital twin, radio materials, antenna configuration, and TX/RX positions to generate propagation paths and channel outputs. Therefore, this chapter establishes the theoretical basis of the complete workflow from robotic mapping to mesh-based digital twin generation and finally to Sionna RT wireless simulation, supporting the later chapters on system design, data collection, SLAM reconstruction, digital twin generation, and result evaluation.

3

System Overview

After introducing the theoretical background in the previous chapter, this chapter presents the system-level workflow used in this thesis. The purpose of this chapter is not to explain the internal details of each algorithm, software tool, or parameter setting. Instead, it describes how data are transferred between the main modules, how each intermediate output is used by the next stage, and how the complete system connects real-world robotic mapping with Sionna RT-based wireless channel simulation.

The following chapters describe each stage of the workflow in detail, including robotic data collection, SLAM reconstruction, digital twin generation, Sionna RT simulation, and final result evaluation.

3.1 Overall Workflow

The proposed system follows an end-to-end workflow from robotic data collection to wireless RT simulation. Unlike a fully manual modeling approach, this thesis first uses a mobile robotic platform to collect sensor data from the target environment. The recorded data are then processed through SLAM, digital twin generation, and Sionna RT simulation. In this way, real-world spatial measurements are gradually converted into a simulation-ready wireless digital twin.

As shown in Fig. 3.1, the workflow consists of four main stages: robotic data collection, SLAM reconstruction, digital twin generation, and Sionna RT simulation. The digital twin generation stage is further divided into three internal steps: point cloud preprocessing, mesh reconstruction, and scene editing with material assignment. This structure keeps the overall workflow clear while still showing the main operations required to transform a SLAM output into an RT-ready scene.

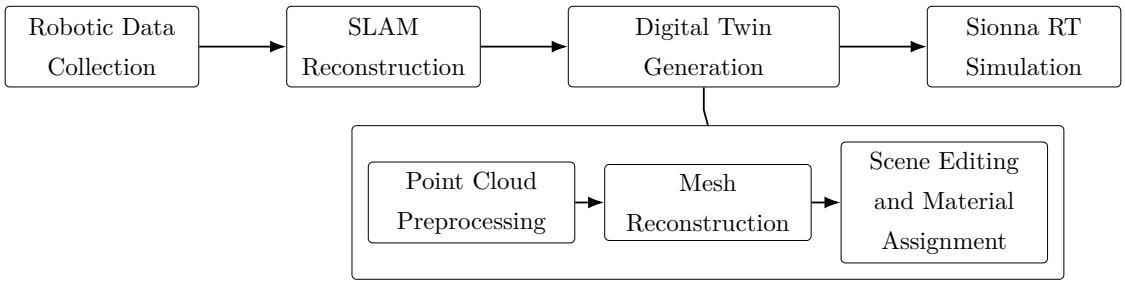


Figure 3.1: Overall workflow of the proposed system. The workflow contains four main stages, while the digital twin generation stage is further divided into point cloud preprocessing, mesh reconstruction, and scene editing with material assignment.

In the robotic data collection stage, the robot platform moves through the target environment and records LiDAR, IMU, RGB-D camera, odometry, and TF data. These data provide the raw geometric, motion, and coordinate-frame information required for later reconstruction.
In the SLAM reconstruction stage, the recorded rosbag data are used to estimate the robot trajectory and align LiDAR scans into a common map frame. The main output of this stage is a global point cloud map. This point cloud captures the main geometry of the environment, but it cannot be directly used for wireless RT because it does not provide continuous surfaces or material information.
In the digital twin generation stage, the SLAM point cloud is converted into an RT-ready scene. This stage first preprocesses the point cloud through filtering, cropping, downsampling, and denoising. The processed point cloud is then converted into a mesh through surface reconstruction. Finally, the mesh is edited and assigned with materials so that the scene becomes suitable for Sionna RT. This stage is the key bridge between robotic reconstruction and wireless simulation.
In the Sionna RT simulation stage, the prepared scene is imported into Sionna RT. TX/RX positions, antenna settings, carrier frequency, and RT parameters are then defined. The resulting simulation outputs include propagation paths, path delays, path gains, received power, PDP, RMS delay spread, CFR, and other channel-related quantities.

3.2 System Architecture

The system architecture can be understood as a sequence of data transformations. Each module does not only perform an isolated operation, but also converts the

output of the previous stage into a form that can be used by the next stage. Table 3.1 summarizes the main input, output, and purpose of each stage.

Table 3.1: Main stages, inputs, outputs, and purposes of the proposed workflow.

Stage	**Main Input**	**Main Output**	**Purpose**
Robotic data collection	Physical environment and robot motion	Rosbag data with LiDAR, IMU, RGB-D, odometry, and TF	Records raw spatial and motion data from the target environment.
SLAM reconstruction	Recorded rosbag data	Point cloud map and estimated trajectory	Aligns sensor measurements into a global map frame.
digital twin generation	SLAM point cloud map	Cleaned and material-labeled RT-ready scene	Converts discrete mapping output into a continuous and simplified scene for RT.
Sionna RT simulation	RT-ready scene and TX/RX settings	Propagation paths and channel outputs	Computes wireless propagation and channel characteristics.

From a system-level perspective, the first two stages mainly focus on obtaining reliable spatial data from the real environment. The robotic platform records the raw data, while SLAM converts these measurements into a global geometric map. The second part of the workflow focuses on preparing and using this map for wireless simulation. The digital twin generation stage transforms the SLAM point cloud into a mesh-based and material-labeled scene, and Sionna RT then computes the corresponding propagation and channel results.

A key point in this architecture is that the SLAM point cloud is only an intermediate result. Although it contains useful geometric information, it is not yet a complete digital twin for wireless simulation. It must be further processed to produce continuous surfaces, remove irrelevant structures, reduce noise, simplify unnecessary details, and assign materials. Only after these steps can the scene be used as a meaningful input to Sionna RT.

Therefore, the proposed architecture can be summarized as a sensing-to-simulation

data conversion chain. Robotic data collection captures the physical environment, SLAM provides the global geometric map, digital twin generation prepares the RT-ready scene, and Sionna RT produces the wireless channel results. This structure also defines the organization of the following methodology chapters.

4

Robotic Data Collection

This chapter describes the robotic data collection stage of the proposed workflow. It introduces the robotic platform, sensor setup, ROS 2 recording system, recorded topics, and the actual data collection procedure. The goal of this stage is to obtain the raw sensor data required for later SLAM reconstruction and digital twin generation. The recorded data mainly include LiDAR point clouds, IMU measurements, RGB-D camera data, robot odometry, and TF information.

4.1 Robotic Platform

The robotic platform used in this work is the Unitree Go2 quadruped robot. It carries the sensing devices, moves through the target environment, and collects spatial measurements for later reconstruction and simulation.

The use of a mobile robot provides several practical advantages. First, it allows the sensing system to observe the environment from multiple viewpoints. This is important because the reconstruction of a three-dimensional scene depends on measurements collected from different positions. Second, the robot can carry multiple sensors at the same time, which makes it possible to collect geometric, visual, and motion-related data in one recording process. Third, compared with a fully manual measurement process, a robotic platform provides a more repeatable and structured way to scan real environments.

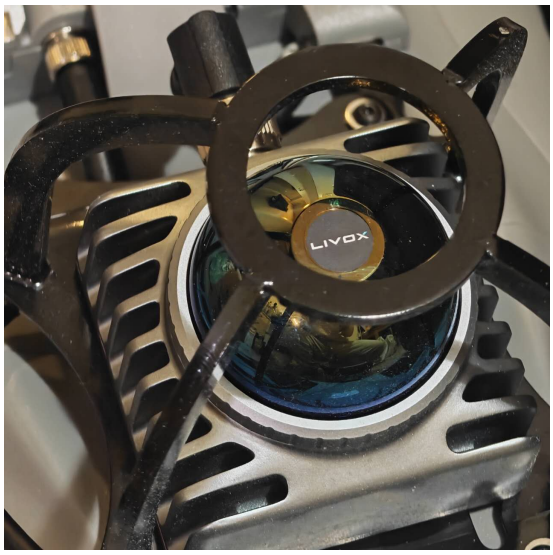
The quadruped design is also useful for indoor data collection. Unlike a fixed tripod or a handheld scanner, the robot can move through corridors, rooms, and other spaces while keeping a relatively stable sensor setup. This helps reduce the amount of manual handling during data collection. It is also useful when repeated scanning is needed or when manual scanning is inconvenient.

The robot can provide state information such as odometry and motion status. These data are useful for coordinating sensor measurements and checking the consistency of

the recorded data. Together with the external sensors mounted on the platform, the robot provides the physical basis for collecting real-world spatial data in a controlled and repeatable way.

4.2 Sensor Setup

The robotic platform was equipped with an external Intel RealSense D435i RGB-D camera and a Livox MID-360 LiDAR sensor. The RealSense D435i camera was mounted on the upper part of the robot, while the Livox MID-360 was fixed to the robot using a rigid bracket. This setup provides geometric, visual, depth, and inertial measurements for different reconstruction and mapping methods. The two main external sensors used in the data collection system are shown in Fig. 4.1.



(a) Livox MID-360 LiDAR sensor.

(b) RealSense D435i RGB-D camera.

Figure 4.1: Main external sensors mounted on the robotic platform.

The Livox MID-360 provides direct three-dimensional measurements of the surrounding surfaces. Its point cloud data are suitable for LiDAR-based mapping, and the associated IMU measurements can support LiDAR-inertial SLAM. This is important for reconstructing the main geometric structure of the environment, especially when reliable depth measurements are needed over a wider field of view. The RealSense D435i provides both visual and depth information. The RGB images record the appearance of the scene, while the depth images provide distance measurements from the camera viewpoint. These data can be used by RGB-D SLAM

methods such as RTAB-Map, where visual features and depth measurements are combined to estimate camera motion and reconstruct the environment. They also provide a useful data source for possible scene inspection, material analysis, and comparison with LiDAR-based reconstruction.
Both sensors were rigidly mounted on the robot platform. Therefore, their relative poses were treated as fixed during each recording. The corresponding frame relationships were recorded or published through the ROS 2 TF system, which allows later checking of the coordinate relationships between the sensor frames and the robot frame. Although multiple sensor streams were recorded, the main reconstruction workflow relied primarily on the Livox MID-360 LiDAR and IMU data. The RGB-D camera and robot state topics were mainly recorded for auxiliary checking, alternative reconstruction tests, and data completeness validation.

Table 4.1: Main sensor streams used during robotic data collection.

Sensor / Stream	**Main Data**	**Role in This Work**
Livox MID-360 LiDAR	3D point cloud	Main geometric input for LiDAR-based and LiDAR-inertial SLAM.
Livox MID-360 IMU	Angular velocity and linear acceleration	Inertial input for LiDAR-inertial SLAM.
Intel RealSense D435i RGB-D camera	RGB image, depth image, and camera information	Used for RGB-D reconstruction tests, visual inspection, and comparison with LiDAR-based reconstruction.
Robot odometry	Robot motion estimate	Used as auxiliary motion information and for consistency checking.
TF / TF static	Coordinate-frame relationships	Used to verify sensor and robot frame consistency.

4.3 ROS 2 System and Recorded Data

4.3.1 ROS 2 System

ROS 2 was used as the communication and recording framework for the robotic data collection system. In ROS 2, sensors and processing programs run as separate nodes and exchange data through topics. Each topic carries a specific type of message, such as point clouds, images, IMU measurements, odometry, or coordinate

transformations.
This structure allows the LiDAR, RGB-D camera, robot odometry, and IMU data to be recorded during the same experiment. The Livox driver publishes LiDAR point clouds and IMU measurements. The RealSense driver publishes RGB images, depth images, camera calibration parameters, and camera IMU data. The robot platform publishes odometry, pose, and inertial measurements.
Coordinate frame information was handled through the ROS 2 TF system. The TF messages describe the spatial relationships between frames, such as the odometry frame, robot base frame, LiDAR frame, and camera frame. In this work, an auxiliary ROS 2 node was used to convert the robot odometry topic into a TF transform from `odom` to `base_link`. The node subscribes to `/utlidar/robot_odom`, copies the position and orientation from the odometry message, and broadcasts them as a time-stamped TF transform. This keeps the robot motion information available in the TF tree during recording and later processing.
The recorded data were stored in rosbag format. A rosbag file stores the selected topics with their timestamps, which allows the same dataset to be replayed and processed offline. This thesis uses an offline processing workflow, so the data collection stage and the later SLAM reconstruction stage are separated. This makes it possible to reuse the same recorded data with different SLAM methods and parameter settings.

4.3.2 Recorded Topics

The main recorded topics include LiDAR point clouds, LiDAR IMU data, RGB-D camera streams, camera calibration information, robot odometry, robot pose, robot IMU data, and TF transformations. These topics provide the raw data for LiDAR-inertial mapping, RGB-D mapping, point cloud processing, and scene reconstruction.

Listing 4.1: Core ROS 2 bag recording command used for the sensor dataset.

```
ros2 bag record -o thesis_new \
/livox/lidar /livox/imu \
/utlidar/robot_odom /utlidar/robot_pose /utlidar/imu \
/camera/color/image_raw /camera/color/camera_info \
/camera/depth/image_rect_raw /camera/depth/camera_info \
/camera/imu \
/tf /tf_static
```

Although many individual topics were recorded, they can be grouped according to their main function, as shown in Table 4.2. This grouping avoids repeating the same information from the recording command while still showing how the recorded data support later processing.

Table 4.2: Recorded ROS 2 topics grouped by function.

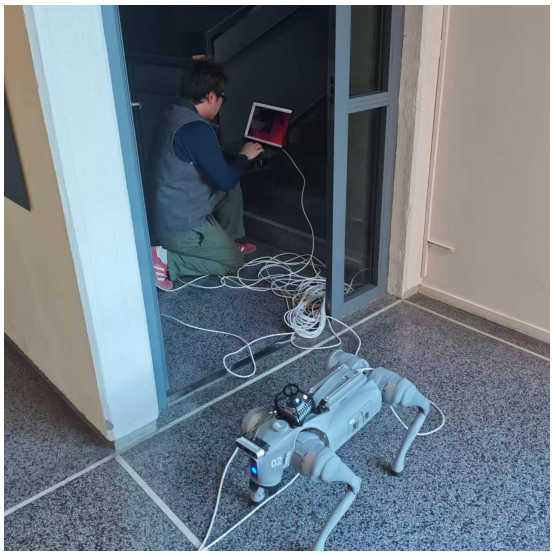
Category	**Topics**	**Purpose**
LiDAR and inertial data	`/livox/lidar`, `/livox/imu`	Main input for LiDAR-based and LiDAR-inertial SLAM.
Robot state	`/utlidar/robot_odom`, `/utlidar/robot_pose`, `/utlidar/imu`	Auxiliary motion information and consistency checking.
RGB-D camera	`/camera/color/image_raw`, `/camera/color/camera_info`, `/camera/depth/image_rect_raw`, `/camera/depth/camera_info`, `/camera/imu`	Used for RGB-D reconstruction tests and visual inspection.
TF information	`/tf`, `/tf_static`	Records coordinate-frame relationships between sensors and robot frames.

The LiDAR and IMU topics were used for LiDAR-inertial reconstruction. The RGB-D camera topics provided image, depth, calibration, and camera IMU data for RGB-D mapping methods. The odometry and TF topics provided robot motion and coordinate frame information, which were needed when comparing data from different sensors or checking the spatial consistency of the recorded dataset.

4.4 Data Collection Procedure

The data collection was carried out in an indoor corridor environment inside a teaching building. The scene contains corridor structures, walls, doors, glass surfaces, and other common indoor objects. This type of environment is suitable for evaluating the proposed pipeline because it contains large planar structures for reconstruction and also includes objects and materials that can affect wireless propagation.

The experimental operation setup is shown in Fig. 4.2. The robot was accessed and controlled from a computer during the recording, and the operator stayed outside the main scanning area when possible to reduce human interference in the LiDAR point cloud.



(a) Computer-based operation and monitoring during recording.

(b) Remote operation from behind the glass door.

Figure 4.2: Operation setup during robotic data collection.

Before data collection, the robot system, Livox MID-360 driver, RealSense D435i driver, and the required ROS 2 nodes were started. The sensor topics were then checked to confirm that they were being published correctly, and the TF tree was inspected to ensure that the main frames needed for later processing were available. After the system status was verified, `rosbag record` was used to record the complete experiment.

During data collection, the robot moved through the target area and covered the main walls, corridors, door frames, and open spaces as much as possible. The collection trajectory was planned to avoid unnecessary repeated coverage while still keeping sufficient overlap between consecutive observations for stable SLAM registration. For corridors and other areas with repetitive structures, the robot motion was kept smooth. Fast turns or sudden acceleration were avoided to reduce the risk of motion distortion and registration failure.

Before each recording, the system time was checked again to reduce timestamp inconsistency between different sensor streams. However, the robot odometry may follow the internal microcontroller unit(MCU) time reference, which can introduce an offset relative to other ROS 2 topics. Therefore, this issue needed to be considered during later preprocessing and SLAM evaluation, rather than being assumed to be completely solved during recording.

After data collection, each rosbag was checked before being used for reconstruc-

tion. The checks included whether the required topics were present, whether the message counts were reasonable, whether the LiDAR point cloud could be visualized correctly, whether the camera topics were non-empty, and whether `/tf` and `/tf_static` contained the necessary frame relationships. Rosbags with missing key topics or clearly abnormal message counts were not used as the main reconstruction input.

Several motion patterns were tested during data collection. In one recording, the robot moved along a loop path in the corridor and returned close to the starting area. In another recording, the robot followed a more linear path along the corridor. The loop trajectory provided observations of the environment from different directions and could help evaluate map consistency, while the linear trajectory provided a simpler motion pattern for comparison.

The final dataset used for the main reconstruction was selected after checking topic availability, message counts, LiDAR visualization, camera streams, and TF information. Its basic recording information is summarized in Table 4.3.

Table 4.3: Summary of the selected rosbag used for the final reconstruction.

Item	Value
Rosbag file	`thesis_new30_0.db3`
Storage format	SQLite3
Bag size	371.3 MiB
Recording duration	45.26 s
Total messages	36,157
Main LiDAR input	`/livox/lidar`: 452 messages
Main inertial input	`/livox/imu`: 9,034 messages
Robot motion data	`/utlidar/robot_odom`: 6,755 messages; `/utlidar/robot_pose`: 848 messages; `/utlidar/imu`: 11,228 messages
Camera data	`/camera/color/image_raw`: 270 messages; `/camera/depth/image_rect_raw`: 271 messages
TF data	`/tf`: 6,756 messages; `/tf_static`: 1 message

4.5 Data Quality and Synchronization Issues

Several practical issues were observed during data collection. These issues mainly came from coordinate frame handling, timestamp synchronization, hardware limitations, sensor bandwidth, robot motion, and environmental interference.

The first issue was related to TF availability. The robot provided odometry through the topic `/utlidar/robot_odom`, but this information was not directly available as a TF transform in the required form. To make the odometry usable in the TF tree, an auxiliary ROS 2 node was written to convert the odometry message into a time-stamped transform from `odom` to `base_link`. The node copies the position and orientation from the odometry message and broadcasts them through the TF system. This step was needed to keep the robot motion information consistent with the other recorded frame transformations.

Timestamp synchronization was another important issue. The LiDAR, camera, IMU, odometry, and TF streams must be temporally consistent for later SLAM and reconstruction. During startup, the sensor timestamps were checked through the prepared launch script. However, the robot odometry was based on the internal MCU timing logic rather than exactly the same time reference used by the other sensor streams. Because of this, the possible time offset was considered during later preprocessing and SLAM evaluation.

The hardware and bandwidth limitations of the robot also affected the recording setup. Recording high-rate LiDAR data, RGB images, depth images, IMU data, odometry, and TF data at the same time created a high load on the system. In practice, the recording setup was adjusted to reduce unnecessary data streams and avoid message loss. The RealSense point cloud topic was not included in the main recording because the Livox MID-360 already provided the main point cloud data. This helped reduce recording load and avoided cases where large data streams were dropped during rosbag recording.

Robot motion affected the visual data quality. When the robot turned quickly or moved while the camera was recording, image frames could show rolling-shutter distortion. This effect was more visible during turning motions and could reduce the quality of visual features for RGB-D based methods. For this reason, the robot was moved at a low and stable speed, and sudden turns were avoided when possible.

Environmental interference also affected the point cloud. If a person appeared close to the scanning area, the LiDAR point cloud could contain irregular moving shapes that did not belong to the static environment. These points would reduce the quality

of the reconstructed map and could introduce unwanted geometry during later mesh generation. To reduce this problem, the robot was operated from behind a glass door when possible, so that the scanned corridor remained clear during recording.
These issues were considered when selecting the final datasets for reconstruction. Recordings were checked using topic inspection and `ros2 bag info`, and datasets with missing important streams or abnormal message counts were avoided. The final recording settings were chosen as a balance between data completeness, sensor quality, and the hardware limits of the robotic platform.

4.6 Summary

This chapter described the complete robotic data collection process used in this thesis. The purpose of this stage was to obtain the raw sensor data required for later SLAM reconstruction and digital twin generation. The experimental platform was based on the Unitree Go2 quadruped robot, equipped with a Livox MID-360 LiDAR sensor and an Intel RealSense D435i RGB-D camera. The Livox MID-360 provided the main three-dimensional point cloud and IMU data for LiDAR-based and LiDAR-inertial SLAM, while the RealSense D435i provided RGB images, depth images, and camera information for RGB-D reconstruction tests, visual inspection, and comparison with LiDAR-based reconstruction.
The chapter then explained the ROS 2 system and rosbag recording setup. ROS 2 was used to organize data exchange between different sensor nodes, and the selected topics recorded LiDAR data, IMU measurements, RGB-D camera streams, robot odometry, robot pose, and TF information. To maintain coordinate consistency during later processing, an auxiliary ROS 2 node was used to convert `/utlidar/robot_odom` into a TF transform from `odom` to `base_link`. The final selected dataset was `thesis_new30_0.db3`, which contained 45.26 s of recording, 36,157 messages, and complete LiDAR, IMU, camera, odometry, and TF data.
Finally, this chapter discussed the practical issues observed during data collection, including TF availability, timestamp synchronization, hardware bandwidth, robot motion distortion, and environmental interference. Since recording multiple high-rate sensor streams at the same time increased the system load, unnecessary large data streams, such as the RealSense point cloud topic, were excluded from the main recording. To reduce rolling-shutter distortion and the risk of SLAM registration failure, the robot was moved at a low and stable speed, and sudden turns were avoided when possible. To reduce the influence of moving people on the LiDAR point

cloud, the operator controlled the robot from behind a glass door when possible. Overall, this chapter completed the data collection workflow from robotic platform setup to final rosbag selection, providing a reliable raw-data foundation for the following SLAM reconstruction and digital twin generation stages.

5

SLAM Reconstruction

This chapter describes the SLAM reconstruction process from the recorded rosbag data to the point cloud map. The dataset collected in the previous chapter contains LiDAR, IMU, RGB-D camera, odometry, and TF information. However, the main reconstruction workflow in this chapter is based on the Livox MID-360 LiDAR and IMU data. The purpose of SLAM in this thesis is not to directly generate the final digital twin.

5.1 Choice of SLAM Method

Several mapping methods were considered for the indoor robotic reconstruction task, including ICP-based LiDAR mapping, FAST-LIO2, and RTAB-Map. Since the later goal of this thesis is to generate a geometry model for Sionna RT, the method selection was not based only on whether a trajectory could be estimated. The quality and usability of the generated map for later mesh reconstruction were also important.

Table 5.1 summarizes the main differences between the candidate methods.

FAST-LIO2 was selected as the main LiDAR-inertial reconstruction module in this thesis[22, 12, 23]:

First, it matches the available Livox MID-360 LiDAR and IMU data collected by the robotic platform. Second, it produces a dense point cloud map that can be further processed for mesh reconstruction. Third, its computational cost is acceptable for offline experiments and repeated parameter tests.

In addition, alternative mapping approaches were evaluated during the preliminary experiments. However, the FAST-LIO2 used here is mainly to generate a geometric map for digital twin generation. The aim is not to evaluate the complete SLAM performance of FAST-LIO2 or to prove that it is optimal for all mapping scenarios. In this work, the important question is whether the output point cloud is complete,

Table 5.1: Comparison of candidate SLAM methods considered for the proposed pipeline.

Method	**Main Input**	**Advantages**	**Limitations**
ICP-based LiDAR mapping	LiDAR point clouds	Simple structure; easy to understand and debug; can directly align point clouds.	Sensitive to weak geometric constraints in long corridors and repetitive structures; more affected by motion distortion without IMU support.
FAST-LIO2	LiDAR point clouds and IMU	Matches the Livox MID-360 LiDAR and IMU data; generates a dense point cloud map; suitable for offline reconstruction and later point cloud processing.	Mainly used here for LiDAR-inertial odometry and mapping, rather than full global loop-closure optimization; map quality still depends on trajectory, IMU timing, and scene structure.
RTAB-Map	RGB-D images, camera information, optional odometry / TF	Can use both appearance and depth information; useful for RGB-D reconstruction tests and visual inspection.	Sensitive to image quality, synchronization, RGB-D alignment, and depth data; limited camera FoV may leave ceilings and upper wall regions insufficiently observed, causing holes in the reconstructed mesh.

aligned, and clean enough to support later mesh reconstruction and RT simulation. The ICP-based mapping method accumulates point clouds frame by frame without a tightly coupled state estimation framework. As a result, small registration errors may accumulate over time, leading to noticeable drift and scan-layer artifacts in the reconstructed map, particularly in long corridor environments. These artifacts appear as ring-like or layered structures and reduce the geometric consistency required for subsequent mesh reconstruction.

RTAB-Map was also considered because of its ability to combine visual and depth information. However, the computational cost is significantly higher than that of FAST-LIO2 for the available hardware platform. Furthermore, the limited field of view of the onboard RGB-D camera resulted in incomplete observations of certain wall and ceiling regions, causing missing geometric structures in the reconstructed model. Since the objective of this thesis is to generate a complete and geometrically consistent point cloud for digital twin generation rather than visual scene understanding, FAST-LIO2 provided a more suitable balance between reconstruction quality, computational efficiency, and robustness. For this reason, the main workflow relies on the more stable geometric coverage provided by LiDAR-inertial reconstruction, and FAST-LIO2 is used as the main reconstruction method. These observations are further illustrated and discussed in Chapter 8, where the reconstruction results of ICP, FAST-LIO2, and RTAB-Map are compared using both qualitative observations and mapping quality indicators.

5.2 SLAM Processing Workflow

The FAST-LIO2 processing workflow is based on the rosbag data recorded during the data collection stage. Since data collection and SLAM reconstruction are performed separately, the same rosbag can be replayed multiple times and tested with different parameter settings in offline processing.

During processing, the rosbag is replayed, and FAST-LIO2 subscribes to the Livox LiDAR and IMU topics. The LiDAR provides 3D point measurements of the surrounding surfaces, while the IMU provides high-rate motion information for short-term motion prediction and motion compensation. The system then updates the robot pose and the map through LiDAR scan-to-map registration. As the robot moves, LiDAR scans collected at different times are aligned into a common map frame, and a global point cloud map is finally generated.

Figure 5.1: SLAM processing workflow from recorded rosbag data to the reconstructed point cloud map.

Fig. 5.1 summarizes the main workflow from rosbag replay to point cloud map generation. The point cloud map is then used as the input for point cloud preprocessing and mesh reconstruction in the next chapter.

The reconstructed map is formed by transforming LiDAR points from the local LiDAR frame into a common map or world frame. For the i-th LiDAR point in scan t, this transformation can be written as

$$\mathbf{p}_{i,t}^{w} = \mathbf{R}_t \mathbf{p}_{i,t}^{l} + \mathbf{t}_t. \tag{5.1}$$

Here, $\mathbf{p}_{i,t}^{l}$ is the point measured in the LiDAR coordinate frame, $\mathbf{R}_t$ and $\mathbf{t}_t$ are the estimated rotation and translation of the LiDAR frame with respect to the world frame, and $\mathbf{p}_{i,t}^{w}$ is the transformed point in the world frame. This equation also explains why pose estimation errors can directly lead to misalignment, drift, or deformation in the final point cloud map.

In LiDAR-based scan-to-map registration, the current scan is aligned with the existing map by minimizing the geometric residual between transformed LiDAR points and local map surfaces. A commonly used point-to-plane residual can be written as

$$r_i = \mathbf{n}_i^T \left(\mathbf{R}_t \mathbf{p}_{i,t}^{l} + \mathbf{t}_t - \mathbf{q}_i \right), \qquad \min_{\mathbf{R}_t, \mathbf{t}_t} \sum_{i=1}^{N} r_i^2. \tag{5.2}$$

In this expression, $\mathbf{q}_i$ is a corresponding point on a local map surface, and $\mathbf{n}_i$ is the normal vector of that surface. The residual r_i measures the distance from the transformed LiDAR point to the local map plane. By minimizing the sum of squared residuals, the SLAM system refines the estimated pose and improves the alignment between the current scan and the accumulated map.

Besides the LiDAR and IMU data, TF and robot odometry were also checked during processing. They are not the core inputs of FAST-LIO2, but they are useful for checking frame relationships, trajectory behavior, and the consistency of the recorded dataset. For example, missing frames in the TF tree or a clear timestamp

offset in robot odometry can make it harder to interpret the reconstruction result correctly.

The main output of FAST-LIO2 includes an estimated trajectory and a point cloud map. The trajectory is used to check whether the robot motion is continuous and whether sudden jumps or large drift appear. The point cloud map is the main geometric input for the next chapter. It is usually exported in PCD format and inspected in tools such as RViz, CloudCompare, or Open3D. The point cloud map generated by SLAM is still not a final digital twin. It is only a discrete set of points and does not contain continuous surfaces or material information for Sionna RT. Therefore, the map must still go through filtering, cropping, downsampling, mesh reconstruction, scene cleaning, and material assignment before it can be used as an RT-ready scene.

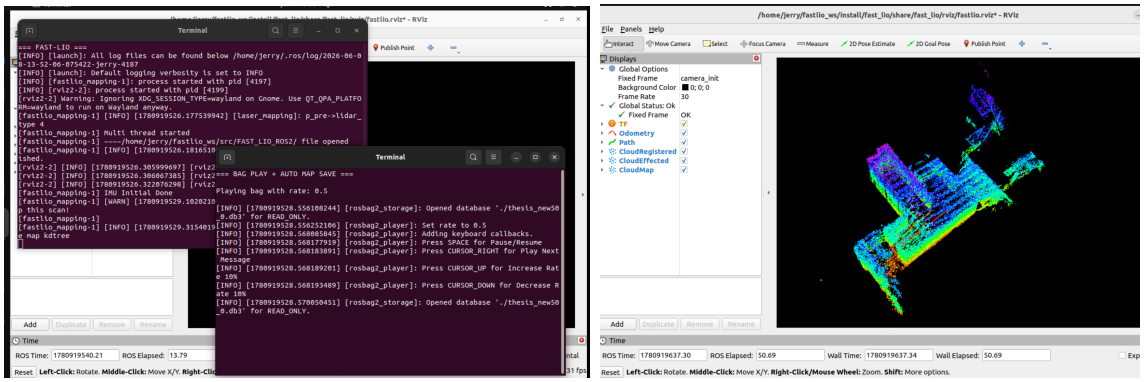
5.3 Point Cloud Map Output

The point cloud map generated by FAST-LIO2 is the main geometric output of the SLAM reconstruction stage. It contains the main structures observed by the robot during scanning, including walls, floor regions, door frames, corridor boundaries, and part of the indoor objects. Compared with a single LiDAR scan, the SLAM map integrates scans collected at different times into a common coordinate frame. Therefore, it provides a more complete representation of the experimental environment and can be used as the geometric input for later digital twin generation.

The FAST-LIO2 processing was integrated into an automated workflow. After the ROS 2 and FAST-LIO2 environments were sourced, the selected rosbag was replayed at a reduced playback rate of 0.5. This slower playback rate was used to reduce the risk of message congestion and to allow the mapping node to process the LiDAR and IMU data more stably. During playback, FAST-LIO2 continuously estimated the robot motion and accumulated the LiDAR scans into a global point cloud map, the point cloud map could also be inspected in RViz during the process. After the bag playback finished, the workflow automatically triggered the map-saving procedure and exported the reconstructed map as a PCD file.

The automated FAST-LIO2 processing and the corresponding RViz visualization are shown in Figure 5.2.

The generated PCD files were typically around 2–5 MB, depending on the selected recording, the scanned area, and the amount of retained map points. After filtering and map saving, approximately 50–70% of the accumulated point information was



(a) Automated FAST-LIO2 processing workflow.

(b) Visualized point cloud map in RVIZ.

Figure 5.2: FAST-LIO2 reconstruction and visualization during the automated SLAM processing workflow. The left image shows the bag replay and mapping process, while the right image shows the accumulated point cloud map used for later mesh reconstruction.

preserved for the following digital twin generation stage. The coordinate values were treated in meters, which allowed the map to be further processed for mesh reconstruction and later geometric validation. In addition to serving as the input for mesh generation, the point cloud map was also used for a sanity check of the SLAM result, including whether there were obvious duplicated structures, severe drift, large missing regions, or strong local artifacts.

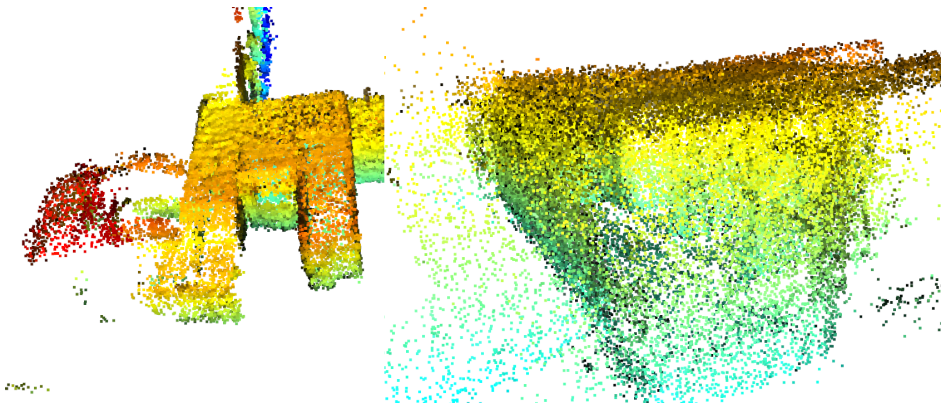


Figure 5.3: Visualization of the reconstructed point cloud map used for SLAM result evaluation.

Fig. 5.3 shows the reconstructed point cloud map used for visual inspection. The overview is used to observe the environmental coverage and possible global drift, while local regions are used to check wall alignment, floor continuity, and obvious

noise.

5.4 SLAM Result Evaluation

Since the SLAM output is used as the geometric input for digital twin generation, the evaluation focuses on map quality rather than only localization accuracy. The reconstructed map was checked in terms of coverage, local alignment, visible drift, point density, noise, and suitability for mesh reconstruction. These criteria are summarized in Table 5.2.

Table 5.2: Criteria used to evaluate the SLAM reconstruction result.

Criterion	**What to Check**	**Relevance to Digital Twin and RT**
Coverage	Whether main walls, corridors, door frames, floor regions, and open spaces are scanned.	Determines whether the later mesh can contain the main propagation-relevant structures.
Local alignment	Whether repeated observations of walls, floors, and door frames overlap correctly.	Affects mesh surface quality and helps avoid duplicated walls or shifted structures.
Visible drift	Whether the map shows bending, stretching, or large global misalignment.	Affects scene scale, path length, AoA, AoD, and path delay in RT simulation.
Point density	Whether the main structures contain enough points for reconstruction.	Influences the stability of surface reconstruction.
Noise and artifacts	Whether there are dynamic objects, floating points, or clear outliers.	Increases mesh cleaning work and may create unrealistic reflecting surfaces.
Meshing suitability	Whether the point cloud can be cropped, downsampled, and reconstructed into a mesh.	Determines whether the SLAM output can be used as input for digital twin generation.

The first check is whether the point cloud map covers the main structures in the target area. If important walls, floor regions, or corridor boundaries are missing, the

map is not suitable for later mesh reconstruction even if the estimated trajectory looks continuous. Coverage is especially important because large surfaces usually dominate the main reflection and blockage behavior in RT simulation.

The second check is local alignment. When the robot scans the same area more than once, repeated observations of walls, floors, and door frames should overlap. If the same wall appears as two shifted layers, the mesh reconstruction stage may produce duplicated or broken surfaces. This can later introduce unrealistic geometry in the digital twin.

Visible drift is also important. In long corridors or repetitive indoor environments, SLAM can slowly drift, which may appear as bending, stretching, or incorrect closing of the map. For normal visualization, a small amount of drift may still be acceptable. For RT simulation, however, geometry errors can directly change path lengths and reflection relationships. Therefore, maps with clear drift are not suitable as the main reconstruction input.

Point density and noise are checked together. Sparse points can make it difficult to recover continuous surfaces, while too many noisy points increase the later processing cost. Dynamic objects and floating points should also be reduced, because they can become unrealistic small surfaces during mesh reconstruction. These surfaces may affect the RT result if they remain in the final scene.

Based on these checks, the final SLAM output was selected according to its usefulness for digital twin generation. The selected map should contain the main indoor structures, show acceptable local alignment, have limited visible drift, and be suitable for later point cloud preprocessing and mesh reconstruction. This choice is based on the needs of the later digital twin and RT stages, rather than on a single SLAM metric.

In addition to the qualitative checks above, dimensional consistency is also used as a simple quantitative indicator of the reconstruction quality. Since the final goal is to use the reconstructed environment for ray-tracing simulation, the large-scale geometry of the map is more important than small surface details. Therefore, several physically measurable dimensions in the real environment, such as the main transverse width, the narrow passage width, and the indoor height, are compared with the corresponding dimensions in the generated mesh. This comparison is reported in Chapter 8 as part of the geometric validation of the digital twin.

5.5 Summary

This chapter described the SLAM reconstruction process from the recorded rosbag data to the reconstructed point cloud map. Although the dataset collected in the previous chapter contained LiDAR, IMU, RGB-D camera, odometry, and TF information, the main reconstruction workflow in this chapter was based on the Livox MID-360 LiDAR and IMU data. By comparing ICP-based LiDAR mapping, FAST-LIO2, and RTAB-Map, FAST-LIO2 was selected as the main SLAM method. This is because FAST-LIO2 matches the LiDAR-IMU data used in this thesis, produces a relatively dense and geometrically stable point cloud map, and has acceptable computational cost for offline processing and repeated experiments. In comparison, ICP-based mapping is more likely to accumulate drift and generate layered scan artifacts in corridor environments with weak geometric constraints, while RTAB-Map is more sensitive to RGB-D image quality, synchronization, and camera field of view, which can lead to missing wall or ceiling regions.

The FAST-LIO2 processing workflow was then explained. The selected rosbag was replayed at a reduced playback rate of 0.5 to reduce the risk of message congestion and to improve the stability of LiDAR and IMU processing. During replay, FAST-LIO2 continuously estimated the robot pose and transformed LiDAR scans collected at different times into a common map frame, resulting in a global point cloud map. Although TF and robot odometry were not the core inputs of FAST-LIO2, they were checked to verify frame relationships, trajectory continuity, and dataset consistency. The final PCD files were typically around 2–5 MB, and approximately 50–70% of the accumulated point cloud information was preserved for the following digital twin generation stage.

Finally, this chapter evaluated the SLAM output from the perspective of digital twin generation and RT simulation. The evaluation did not focus only on localization accuracy, but mainly on whether the point cloud was suitable as the geometric basis for later mesh reconstruction. The checked aspects included coverage, local alignment, visible drift, point density, noise and artifacts, and meshing suitability. A map was considered suitable only when it covered the main walls, floor regions, door frames, and corridor boundaries, and did not contain obvious duplicated structures, severe drift, or a large amount of floating noise. In addition, dimensional consistency was introduced as a simple quantitative check by comparing measurable dimensions in the real environment with the corresponding dimensions in the reconstructed model. Overall, this chapter completed the key step from robotic sensor data to a

usable SLAM point cloud map, providing the basis for point cloud preprocessing, surface reconstruction, and mesh-based digital twin generation in the next chapter.

6

Digital Twin Generation

This chapter describes how the SLAM point cloud map is converted into a scene that can be used for Sionna RT simulation. The process includes point cloud preprocessing, surface reconstruction, Blender-based scene editing, material assignment, and scene export.

6.1 Point Cloud Preprocessing

The SLAM output from the previous chapter could not be used directly for surface reconstruction. The point cloud still contained redundant points, uneven density, isolated outliers, and regions that were not important for the final simulation scene. If such data are used directly for mesh reconstruction, the resulting mesh can contain rough surfaces, floating fragments, unnecessary faces, and holes that make later editing more difficult. Voxel downsampling was first used to reduce the point cloud density. This method divides the space into small three-dimensional voxels and replaces the points inside each voxel with a representative point.

For each occupied voxel V_j, the representative point was computed as the centroid of all points inside that voxel:

$$\tilde{\mathbf{p}}_j = \frac{1}{|V_j|} \sum_{\mathbf{p}_i \in V_j} \mathbf{p}_i, \tag{6.1}$$

where V_j denotes the set of points inside the j-th voxel and $\tilde{\mathbf{p}}_j$ is the downsampled point. It reduces the number of points without strongly changing the main structure of the scene. In this work, the voxel size was chosen to keep large structures such as walls, floors, and corridor boundaries while removing redundant local details.

After downsampling, statistical outlier removal was applied to remove isolated noisy points. For each point, the method computes the average distance to its neighboring points.

The mean neighbor distance of point $\mathbf{p}_i$ can be written as

$$\bar{d}_i = \frac{1}{K_{\mathrm{nn}}} \sum_{\mathbf{p}_j \in \mathcal{N}_{K_{\mathrm{nn}}}(\mathbf{p}_i)} \|\mathbf{p}_i - \mathbf{p}_j\|, , \tag{6.2}$$

where $\mathcal{N}_{K_{\mathrm{nn}}}(\mathbf{p}_i)$ is the set of K_{nn} nearest neighbors of $\mathbf{p}_i$. A point was treated as an outlier when

$$\bar{d}_i > \mu_d + \alpha\sigma_d, \tag{6.3}$$

where μ_d and σ_d are the mean and standard deviation of all mean neighbor distances, and α is the standard deviation multiplier.

Points whose average distance is much larger than the local distribution are treated as outliers and removed. This filtering step helps remove sparse noise that would otherwise produce unwanted surfaces during mesh reconstruction.

The preprocessing step provides a cleaner input for surface reconstruction. The goal of this step is to obtain a point cloud suitable for surface reconstruction, rather than to keep all local details from the raw SLAM output. A moderate point cloud density is useful because it keeps the main geometry of the environment while reducing unnecessary mesh complexity.

6.2 Surface Reconstruction

After point cloud preprocessing, the cleaned point cloud was converted into a mesh model. This step is needed because Blender editing and RT simulation require surface geometry instead of isolated points. In this work, the surface reconstruction was implemented using Open3D , and Poisson surface reconstruction was used to generate the initial mesh from the processed point cloud [18].

The quality of the preprocessed point cloud directly affects the reconstructed mesh. If many floating points or dynamic objects remain in the point cloud, Poisson reconstruction may create unrealistic small surfaces. If important regions such as walls, ceiling areas, or corridor boundaries are missing, the reconstructed mesh may contain holes or incomplete edges. For this reason, the point cloud was cleaned before mesh generation.

Before mesh generation, normal vectors were estimated for the point cloud. Poisson surface reconstruction uses oriented points to infer a continuous surface, so the direction of the local surface around each point is required. For each point, neighboring points were searched within a local radius, and the normal direction was estimated from the nearby geometry.

For a local neighborhood around point $\mathbf{p}_i$, the covariance matrix can be expressed as

$$\mathbf{C}_i = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{p}_j \in \mathcal{N}_i} (\mathbf{p}_j - \bar{\mathbf{p}}_i)(\mathbf{p}_j - \bar{\mathbf{p}}_i)^T, \tag{6.4}$$

where $\bar{\mathbf{p}}_i$ is the centroid of the local neighborhood. The normal direction is given by the eigenvector corresponding to the smallest eigenvalue of $\mathbf{C}_i$.

The normals were then oriented consistently to reduce direction conflicts during reconstruction.

Poisson surface reconstruction formulates the reconstruction problem as a spatial Poisson problem and recovers a continuous surface from oriented point samples. In simplified form, the method solves for an implicit function χ, whose gradient is expected to be consistent with the vector field $\mathbf{V}$ defined by the oriented point normals:

$$\nabla\chi \approx \mathbf{V}, \tag{6.5}$$

where ∇ denotes the gradient operator and describes the spatial variation of χ. Taking the divergence on both sides gives the Poisson equation:

$$\Delta\chi = \nabla \cdot \mathbf{V}, \tag{6.6}$$

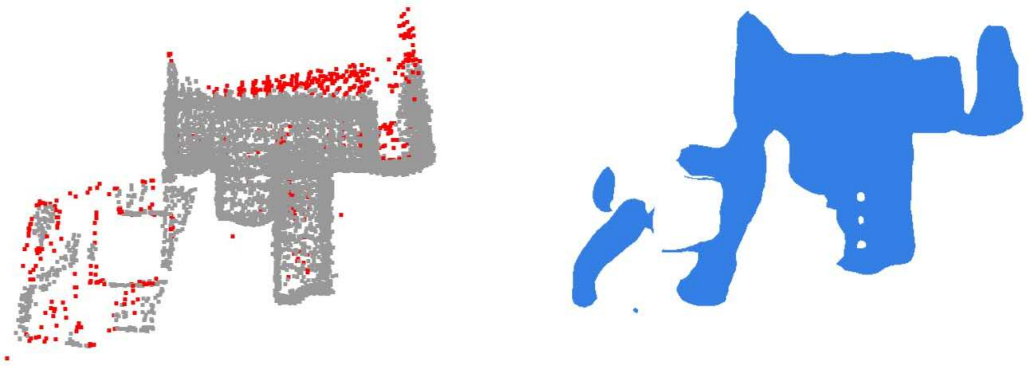
where Δ is the Laplacian operator and $\nabla \cdot \mathbf{V}$ is the divergence of the normal vector field. The reconstructed surface is then extracted as an iso-surface of the implicit function χ.

This method is suitable for the scanned corridor environment because the scene mainly contains large structures such as walls, floors, doors, and corridor boundaries. The reconstruction depth controls the level of geometric detail. A larger depth can preserve more local details, but it also increases the number of triangles and may produce unnecessary small surfaces. In this thesis, the depth value was selected to keep the main structure of the scene while avoiding excessive mesh complexity.

After the initial reconstruction, density-based filtering was applied to remove weakly supported surfaces. Poisson reconstruction may create low-confidence geometry in sparse or noisy regions, especially near the boundary of the reconstructed area. To reduce this effect, vertices with low density values were removed using a percentile-based threshold. This step helped remove unstable surfaces before mesh simplification.

The mesh was then simplified and smoothed. Quadric decimation was used to reduce the number of triangles while preserving the main shape of the environment. A simple smoothing filter was applied to reduce local roughness on reconstructed surfaces.

For the later Sionna RT simulation, a cleaner mesh with large and stable surfaces is more suitable than a highly detailed mesh with many small faces, since excessive geometric detail can increase processing time and make material assignment more difficult.
Fig. 6.1 shows an example of the point cloud after preprocessing and the mesh generated by Poisson surface reconstruction.



(a) Preprocessed point cloud and removed noise points.

(b) Mesh generated by Poisson surface reconstruction.

Figure 6.1: Example of point cloud preprocessing and Poisson-based mesh reconstruction.

The final mesh was exported as a PLY file for Blender editing and later scene conversion. At this stage, the goal was to obtain a usable geometric model with the main surfaces preserved.

6.3 Blender-based Scene Editing and Material Assignment

The reconstructed mesh was further processed in Blender before being exported for wireless RT simulation. Blender 4.0.0 was used as a 3D scene editing tool for mesh inspection, geometry repair, scene simplification, and material preparation. Although the Open3D reconstruction produced a surface model, the mesh could still contain small fragments, holes, rough boundaries, or incorrect face orientations.

A Blender Python script was developed to make this step more repeatable.
First, to make the reconstructed model more suitable for inspection, material labeling, export, and later use, the script automatically corrected the orientation of the model. It computed the covariance matrix of all mesh vertex coordinates and performed eigenvalue decomposition on this matrix. The eigenvectors represent the main directions of the model in three-dimensional space, while the eigenvalues indicate how much the model extends along each corresponding direction.
For the corridor-like indoor scene scanned in this thesis, the model usually extends the most along the corridor direction, the second most along the width direction, and the least along the height direction. Therefore, the script used a heuristic assumption: the direction with the smallest variance can be treated as an approximate vertical direction of the indoor scene. The script then rotated the model so that this estimated vertical direction was aligned with the vertical axis of the world coordinate system. In this way, the model orientation was corrected.
It should be noted that the orientation correction and the following positioning operation in Blender can change the scene origin used later for simulation export, but they do not change the actual metric scale of the model.
The script then used surface normal information to adjust the orientation of the reconstructed model. It removed small disconnected fragments and repaired simple holes in the mesh. These operations reduced the amount of manual cleaning needed before material assignment and export.

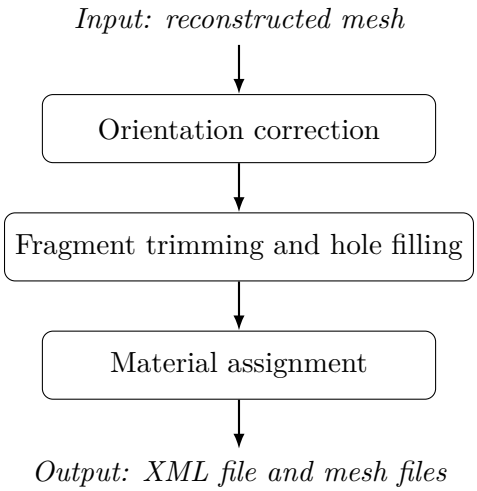


Figure 6.2: Main steps of the Blender-based automation script.

The same script was also used for coarse material labeling. The labeling was based on the orientation and position of mesh faces. Vertical surfaces were treated as wall

regions, horizontal lower surfaces were treated as floor regions, and upper horizontal surfaces were treated as ceiling regions. This rule-based method is simple, but it matches the main structure of the corridor scene, where large planar surfaces dominate the environment.

This material assignment is not a full semantic segmentation method. It is only used to reduce repeated manual work, not to replace manual checking. The script can label the main wall, floor, and ceiling regions, but it cannot reliably distinguish special objects such as glass doors, windows, or small metal structures only from the mesh geometry[24, 25]. These parts were therefore checked manually in Blender when they were relevant to the RT scene. This step is important because glass, metal, and other material regions can affect reflection and received power in Sionna RT.

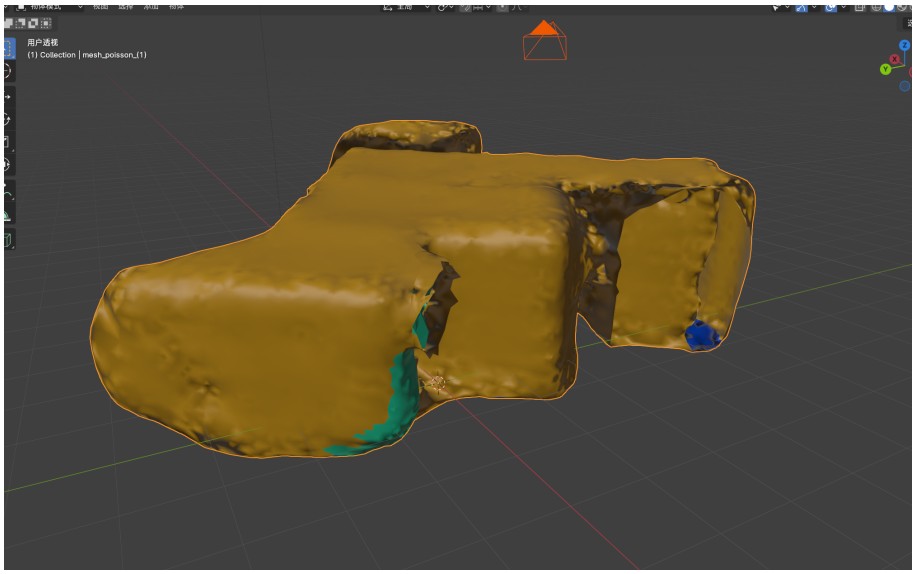


Figure 6.3: Mesh model after Blender-based cleaning and material assignment.

The edited mesh was then prepared for export. Scale, face orientation, and the main material regions were checked before the model was used in the Sionna RT workflow. A simplified scene with large and stable surfaces is more suitable for RT than a noisy mesh with many small faces, since excessive geometric detail can increase processing time and make material assignment more difficult.

6.4 Export to Sionna RT-compatible Format

After Blender-based editing, the scene was exported as a scene package for Sionna RT. The export was performed through the Mitsuba plugin in Blender. This plugin

converts the edited Blender scene into a Mitsuba XML scene description and exports the referenced mesh geometry into external mesh files. In this work, the exported package included an XML file, an OBJ file, and a `meshes/` folder containing multiple PLY files.

The XML file is the main scene description used in the Sionna RT workflow. It stores the scene structure, mesh references, object names, and material assignments. Sionna RT uses the Mitsuba scene format for defining geometry and materials in the RT scene [5]. However, the XML file does not contain all geometric data by itself. It references external mesh files through file paths. Therefore, the exported XML file and the `meshes/` folder must keep the same relative path relationship. If the XML file is moved without the corresponding mesh files, or if the folder structure is changed, Sionna RT may fail to load the scene geometry correctly.

Before export, the material names also need to be checked. They should either be directly recognizable by Sionna RT or correctly mapped to radio materials in the XML file. This step is needed because material labels affect how Sionna RT handles reflection, scattering, and other propagation interactions.

The `meshes/` folder contains the separated geometry exported from Blender. In this workflow, different geometry parts were exported as multiple PLY files, including parts associated with different material labels. These labels can be preserved in the exported scene and later mapped to radio-relevant materials in Sionna RT.

The exported OBJ file was kept as an auxiliary geometry representation. It can be used to inspect the exported model in common 3D tools and check whether the overall scene geometry is correct. The main file used by Sionna RT is still the XML scene description together with the referenced mesh files.

Table 6.1: Exported files for the Sionna RT scene.

Exported item	**Content**	**Role**
XML file	Scene description, mesh references, and material assignment	Main scene file loaded in the Sionna RT workflow through the Mitsuba scene format.
OBJ file	Auxiliary geometry file	Used for geometry inspection and compatibility with common 3D tools.
`meshes/` folder	Multiple PLY mesh files	Stores separated scene geometry parts referenced by the XML file.

6.5 Summary

This chapter described the complete process of generating a Sionna RT-compatible digital twin scene from the SLAM point cloud. First, the raw SLAM point cloud had to be preprocessed, since using the uncleaned point cloud directly for surface reconstruction could lead to rough surfaces, floating fragments, unnecessary faces, and holes. Therefore, voxel downsampling was applied to reduce the point cloud density, and statistical outlier removal was used to remove isolated noisy points. These steps reduced redundant details while preserving the main structures of the environment, such as walls, floors, and corridor boundaries.

The preprocessed point cloud was then converted into a mesh model. Since Blender editing and Sionna RT simulation require continuous surface geometry instead of isolated points, Open3D and Poisson surface reconstruction were used to generate the initial mesh. This method uses the estimated point cloud normals to infer a continuous surface, which is suitable for the corridor environment in this thesis because the scene mainly consists of large-scale structures such as walls, floors, doors, and corridor boundaries. To improve the usability of the mesh, density-based filtering, mesh simplification, and smoothing were further applied. These operations produced a cleaner and more stable geometric model for later material assignment and ray tracing.

Finally, this chapter explained the Blender-based scene editing, material assignment, and export process. A Blender Python script was used to correct the model orientation, remove small fragments, fill simple holes, and assign coarse material labels to walls, floors, and ceilings based on surface orientation and position. Since this rule-based method is not a full semantic segmentation approach, special objects such as glass, metal parts, and other detailed structures still required manual checking. After editing, the scene was exported through the Mitsuba plugin as an XML scene description for Sionna RT, together with the referenced external mesh files and material information. Overall, this chapter completed the key conversion from SLAM reconstruction to a simulation-ready digital twin, providing the geometric and material basis for the subsequent Sionna RT simulation.

7

Ray-Tracing Simulation

This chapter describes the ray-tracing simulation process based on Sionna RT. Building upon the previously completed robotic data collection, SLAM reconstruction, and digital twin generation, this chapter focuses on the complete simulation workflow—from importing the digital twin scene into Sionna RT to extracting the final wireless propagation outputs.

Distinct from the theoretical overview in Chapter 2, this implementation-oriented chapter details the specific simulation workflow and parameter configurations used in this thesis. The pipeline encompasses scene loading, radio and material configuration, TX/RX setup, solver computations (PathSolver and RadioMapSolver), and result extraction. Through these steps, the mesh-based digital twin is effectively converted into key wireless channel metrics, including path delay, path gain, received power, PDP, CFR, AoA/AoD, and radio maps.

7.1 Simulation Workflow

The ray-tracing simulation workflow starts from the Sionna RT-compatible scene exported in the previous chapter. The scene is loaded from `Blender2.xml` together with the corresponding `meshes/` folder, while keeping the original relative path structure from the export step.

After the scene is loaded, the carrier frequency, radio materials, antenna arrays, TX/RX positions, and solver parameters are configured. The PathSolver is then used to compute path-level results between the selected TX/RX pair, while the RadioMapSolver is used to compute the radio map over the selected receiving area. The overall workflow is shown in Fig. 7.1.

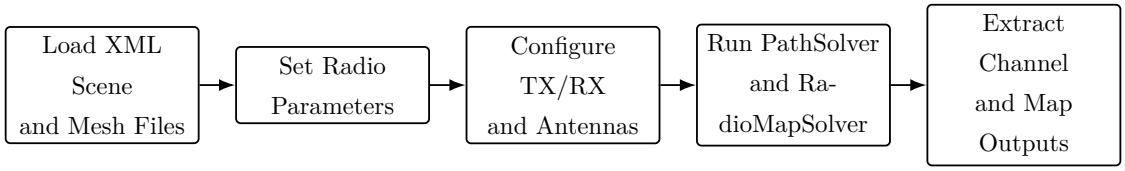


Figure 7.1: Sionna RT simulation workflow from scene loading to result extraction.

Sionna RT is used in this chapter only as the RT simulation engine, not as a geometry creation tool. The scene geometry and material information come from the digital twin scene generated in the previous chapter. Before extracting numerical results, a visual check is performed to confirm that the XML file and the `meshes/` folder are loaded correctly, that the main material regions are displayed as expected, and that the TX/RX are placed at the intended positions.

7.2 Parameter Settings

This section records the main parameters used in the Sionna RT simulation. The aim is not to analyze the influence of each parameter, but to make the simulation setup reproducible. The propagation path, PDP, CFR, AoA, AoD, and radio map results are analyzed in the next chapter.

In this thesis, the scene is loaded from the exported XML file and the corresponding `meshes/` folder. The carrier frequency is set to 3.5 GHz. Both TX and RX use a 1×1 isotropic antenna with vertical polarization. The PathSolver is used to compute path-level results between the selected TX/RX pair, while the RadioMapSolver is used to generate a radio map over the receiving area.

Table 7.1 summarizes the main simulation parameters and inputs.

Table 7.1: Main Sionna RT simulation parameters.

Configuration Item	**Setting**
Scene file(Main input)	`Blender2.xml` and corresponding `meshes/` folder
Carrier frequency	3.5 GHz
Wavelength	0.0857 m
Antenna spacing	$\lambda/2 \approx 0.0429$ m
TX position	$(0.2, -0.2, 3.8)$
RX position	$(0.3, -0.4, -1.8)$
TX/RX antenna	1×1 isotropic antenna with vertical polarization
PathSolver setting	Maximum depth = 10; LoS, specular reflection, and diffuse reflection enabled; refraction disabled
RadioMapSolver setting	Maximum depth = 50; cell size = $(0.1, 0.1)$; samples per TX = 10^6
Main extracted outputs	Propagation paths, radio map, PDP, CFR, AoA, and AoD

7.2.1 Radio and Material Configuration

The carrier frequency is set to 3.5 GHz. The corresponding wavelength is computed as

$$\lambda = \frac{c}{f_c} \tag{7.1}$$

where c is the speed of light and f_c is the carrier frequency. This gives a wavelength of approximately 0.0857 m. The antenna spacing is set to $\lambda/2$, which is approximately 0.0429 m.

In the current simulation, both TX and RX use a 1×1 antenna array. Therefore, the antenna spacing does not affect the current result in practice. The parameter is still kept in the array definition to keep the Sionna RT antenna configuration complete and to allow possible future extension to multi-antenna setups.

The radio material information comes from the imported XML scene. The material labels assigned in the previous chapter are used by Sionna RT to determine how different surfaces affect reflection, scattering, and received power. This chapter only records the material source and basic radio settings, while the result interpretation

is given in the next chapter.

7.2.2 TX/RX Configuration

The TX and RX are added to the Sionna RT scene as radio devices. The TX is placed at $(0.2, -0.2, 3.8)$, and the RX is placed at $(0.3, -0.4, -1.8)$. These coordinates follow the exported Sionna RT scene coordinate system processed by Blender, not the original robot coordinate system. The RX height was adjusted according to the loaded scene geometry, so it is located at the correct receiving position in the visual check.

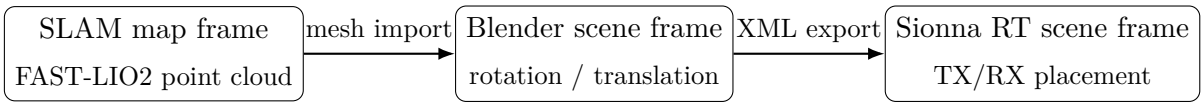


Figure 7.2: Coordinate-frame flow from SLAM reconstruction to Sionna RT simulation.

Both TX and RX use a 1×1 isotropic antenna with vertical polarization. Since the current setup contains only a single antenna element, the array configuration mainly defines the antenna type and polarization rather than forming an actual beam pattern.

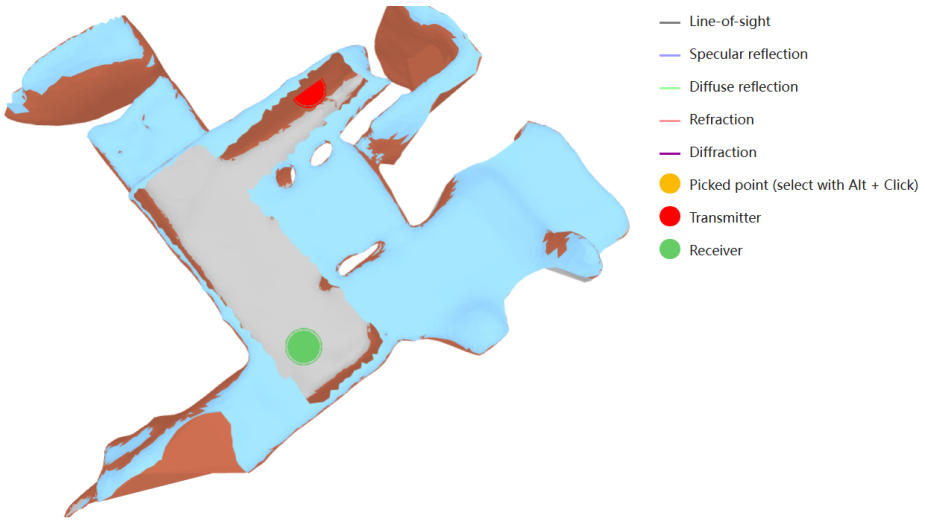


Figure 7.3: Transmitter and receiver locations in the generated digital twin scene.

Fig. 7.3 shows the TX/RX locations in the digital twin scene and confirms that the relative placement between the TX/RX and the scene geometry is reasonable.

7.2.3 Ray-Tracing Parameters

Two Sionna RT solvers are used in this thesis: PathSolver and RadioMapSolver[6]. The PathSolver is used to compute path-level results between the selected TX/RX pair. The RadioMapSolver is used to compute the spatial radio map over the selected receiving area.

For the PathSolver, the maximum depth is set to 10. LoS, specular reflection, and diffuse reflection are enabled, while refraction is disabled. This setting is used to obtain the main direct path and reflection-related multipath components. Since this thesis mainly focuses on propagation changes caused by indoor geometry, refraction is not treated as the main mechanism in the path-level analysis.

For the RadioMapSolver, the maximum depth is set to 50, the cell size is set to $(0.1, 0.1)$, and the number of samples per TX is set to 10^6. Since the tested scene is relatively simple and small, a higher maximum depth is still acceptable in terms of computation. It can also avoid truncating possible propagation paths too early during radio map computation. For larger or more complex scenes, this value may need to be reduced to control computation time and memory usage.

These parameters define the simulation setup used for the results in the next chapter. The path-level outputs, PDP, CFR, AoA, AoD, and radio map are interpreted in the results and evaluation chapter.

7.3 Channel Outputs and Result Extraction

The outputs from Sionna RT can be divided into path-level outputs, radio-map outputs, channel-level outputs, and visualization outputs.

Path-level outputs include propagation paths, path gain, path delay, AoA, AoD, and interaction points. These results are used to analyze how the signal travels from the TX to the RX and which surfaces affect reflection or blockage. Although the propagation path result figure is not shown in this chapter, the path-level data are saved and used for later result analysis.

Radio-map outputs describe the spatial distribution of a selected radio metric, such as path gain or received power, over the receiving area. They can be used to observe the relationship between digital twin geometry and wireless coverage. They also provide a basis for comparing the influence of different mesh or mapping quality levels.

Channel-level outputs can be further computed from the paths. These include CFR,

PDP, RMS delay spread, and average channel power. If the CFR is extracted over K subcarriers, the average channel power can be computed as

$$P_h = \frac{1}{K} \sum_{k=1}^{K} |H(f_k)|^2, \tag{7.2}$$

where $H(f_k)$ is the CFR at the k-th subcarrier. This metric can be used to compare the overall channel strength at different RX positions or under different digital twin models.

Table 7.2 summarizes the main outputs extracted from the Sionna RT simulation.

Table 7.2: Main outputs extracted from the Sionna RT simulation.

Output Type	**Examples**	**Role in This Work**
Path-level outputs	Propagation paths, path gain, delay, AoA, AoD, interaction points	Used to analyze the influence of geometry on individual propagation paths.
Radio-map outputs	Spatial radio map, path gain map, received-power-related map	Used to analyze spatial radio behavior in the digital twin scene.
Channel-level outputs	CFR, average channel power, PDP, RMS delay spread	Used to evaluate wireless channel characteristics derived from the paths.
Visualization outputs	Scene preview, TX/RX placement, radio map visualization	Used to check whether the simulation setup and outputs are physically reasonable.

All simulation outputs need to remain traceable to the corresponding TX, RX, and scene configuration. For each RX position or radio map cell, the related path data and channel metrics should be stored together with the simulation settings. This makes it possible to compare how different digital twin models, mesh processing settings, or mapping quality levels affect the wireless simulation results.

7.4 Summary

This chapter described the implementation of the ray-tracing simulation based on Sionna RT. Different from the previous chapter on digital twin generation, the focus of this chapter was no longer on creating the geometric model, but on explaining how

the processed mesh-based digital twin was imported into Sionna RT and configured as a wireless simulation environment. The complete workflow included loading the XML scene and the corresponding mesh files, setting the carrier frequency and radio materials, configuring the TX/RX positions and antenna parameters, running the PathSolver and RadioMapSolver, and extracting the final wireless propagation outputs.

The main simulation settings used in this thesis were also recorded in this chapter. The scene was loaded from the Blender-exported `Blender2.xml` file and the corresponding `meshes/` folder. The carrier frequency was set to 3.5 GHz, and both TX and RX used a 1×1 vertically polarized isotropic antenna. The TX/RX coordinates followed the Sionna RT scene coordinate system after Blender export, rather than the original robot coordinate system. Therefore, before running the numerical simulation, a visual check was required to confirm that the scene, material regions, and TX/RX positions were loaded and placed correctly. The PathSolver was used to compute path-level propagation results between the selected TX/RX pair, while the RadioMapSolver was used to generate the radio map over the receiving area.

Finally, this chapter explained the main types of outputs extracted from Sionna RT. The path-level outputs included propagation paths, path gain, path delay, AoA, AoD, and interaction points, which can be used to analyze the propagation behavior and reflection mechanisms inside the digital twin. The radio-map outputs described the spatial distribution of path gain or received-power-related metrics over the receiving area, making it possible to observe how the geometry affects wireless coverage. The channel-level outputs included CFR, PDP, RMS delay spread, and average channel power, which can be further used for channel characterization. Overall, this chapter completed the conversion from a simulation-ready digital twin to wireless channel outputs, providing the simulation basis for the radio map, path behavior, and channel metric analysis in the following results and evaluation chapter.

8

Results and Evaluation

8.1 Robotic Mapping Results

8.1.1 Raw Sensor Data

The raw data collected by the robotic dog mainly include LiDAR point clouds, RGB-D images, IMU data, robot odometry, and TF coordinate transformation information. Among them, the LiDAR point cloud is the most direct data source for later geometric reconstruction, because it reflects the direct observation of surrounding environmental surfaces during sensor movement.

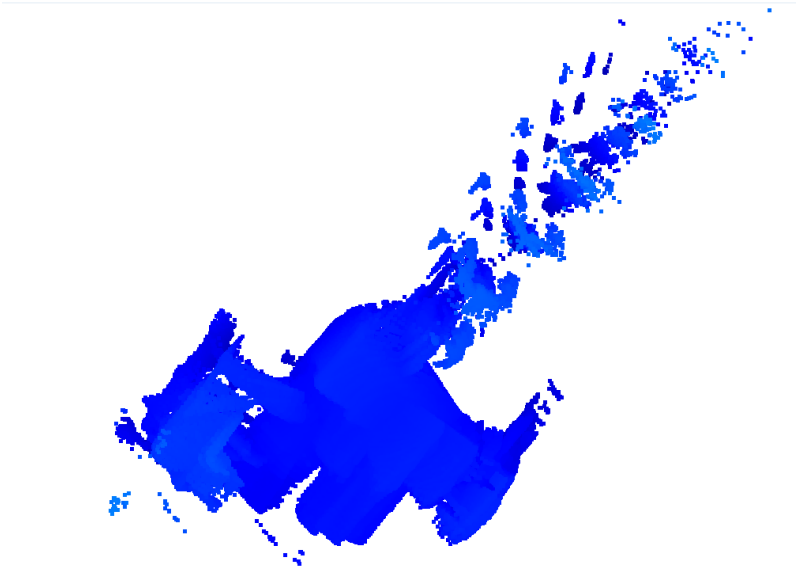


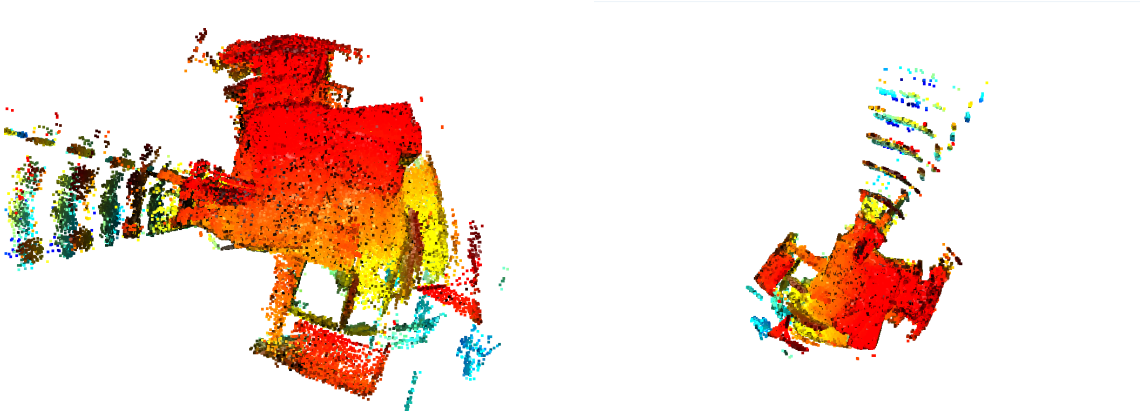
Figure 8.1: Raw sensor data result.

From the raw point cloud, the spatial distribution of corridors, walls, outdoor glass window areas, and indoor structures can be observed. However, this type of data is still only a collection of continuous sensor observations, rather than a usable environmental map. The raw point cloud lacks unified global registration, and it

also contains noise, duplicated points, and other abnormal points. Therefore, the raw point cloud is mainly used to check the operating quality of the sensors.

8.1.2 LiDAR-only Odometry Mapping (ICP) Result

This section presents the mapping result based on LiDAR-only odometry. This type of method mainly relies on geometric registration between consecutive LiDAR point clouds to estimate robot motion and align point clouds collected at different times into a unified coordinate system.



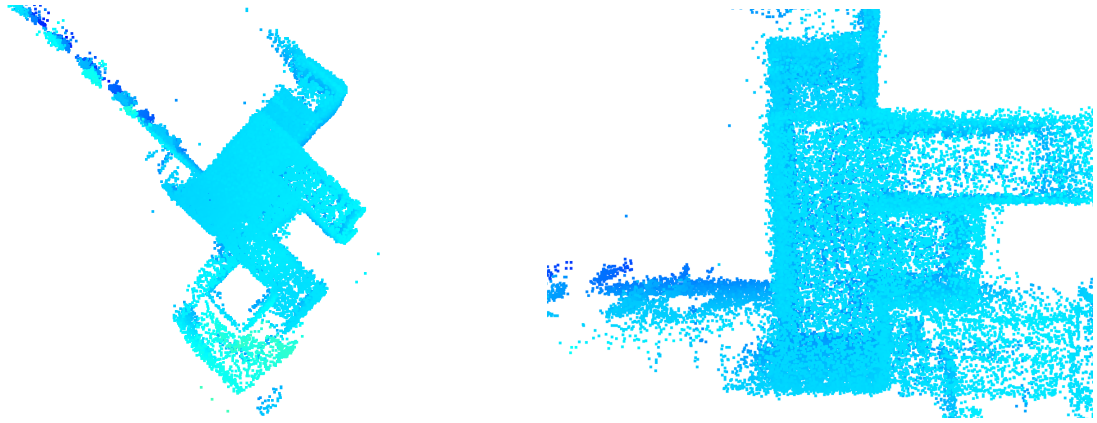
(a) Drifting of the ICP SLAM result. (b) Overview of the ICP SLAM result.

Figure 8.2: Mapping result generated by LiDAR-only odometry based on ICP registration.

From the result, it can be observed that this method can reconstruct the general spatial outline of the experimental environment. The central corridor area, the upper and lower stairway areas, and some outdoor structures scanned through the windows can be identified in the point cloud.
However, since this type of method mainly relies on geometric matching between point clouds, its robustness is relatively limited, and local errors can easily occur during registration. As the robot continues to move, these local errors may gradually accumulate, causing unstable wall boundaries, local drift and misalignment of structures, which is especially obvious in the outdoor structures on the left side, or overlap between different scanned regions.
These problems increase the difficulty of later mesh reconstruction. If this result is directly used for surface reconstruction, the local misalignment and noise may generate incorrect surfaces, thick walls, overlapping surfaces, or irregular triangle meshes.

8.1.3 LiDAR-inertial Odometry Mapping(FAST-LIO2) Result

This section presents the mapping result based on LiDAR-inertial odometry. This method combines LiDAR point clouds and IMU measurements for pose estimation and map construction.



(a) Overview of the FAST-LIO2 SLAM result.

(b) Detail view of the FAST-LIO2 SLAM result.

Figure 8.3: Mapping result generated by LiDAR-inertial odometry.

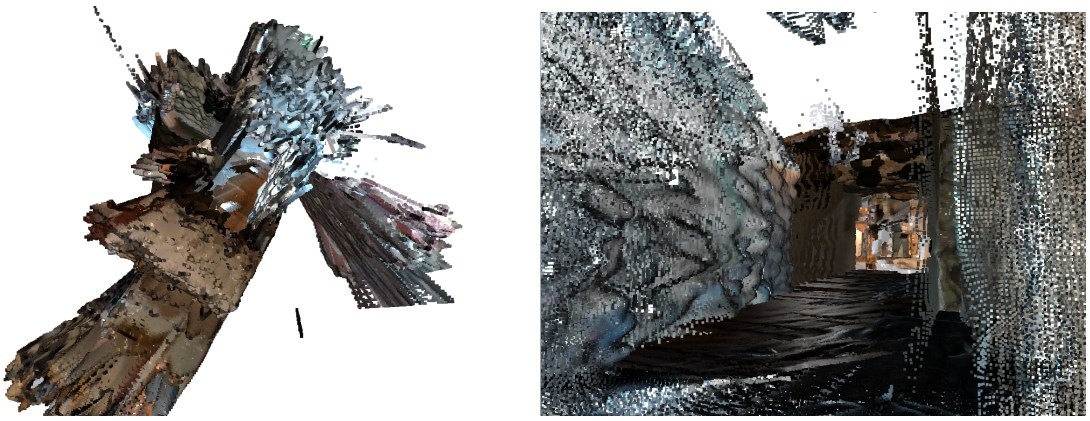
From the overall result, LiDAR-inertial odometry can reconstruct the main spatial structures of the experimental environment relatively completely. The corridor outline, wall boundaries, floor area, local door openings, and outdoor structures can all be clearly observed in the point cloud. Compared with LiDAR-only odometry, this result is more stable in terms of geometric continuity, and no obvious large-scale folding, tearing, or severe misalignment can be observed.

In addition, the main planar structures obtained by this method, such as walls and floors, are relatively flat, and the boundaries are also clearer. This is very important for later mesh reconstruction and Sionna RT ray-tracing simulation. Irregular objects or planes can greatly increase the number of faces, thereby significantly increasing the computational load of ray tracing.

However, although LiDAR can scan window edges, stairways, or other external structures outside glass doors through glass areas, these structures behind the glass are not part of the main simulation area that needs to be preserved for the purpose of this thesis. Therefore, in the later digital twin generation stage, these irrelevant regions need to be removed through filtering, denoising, and other processing methods to avoid affecting mesh reconstruction and ray-tracing simulation.

8.1.4 RGB-D and Visual-feature-based SLAM(RTAB-Map) Result

This section presents the SLAM result based on RGB-D data and visual features. This method uses RGB images, depth images, and visual features for pose estimation and environmental reconstruction. Compared with LiDAR-based methods, the main advantage of this method is that it can preserve the color, texture, and scene appearance information of the recorded corridor environment.



(a) Overview of the RGB-D and visual-feature-based SLAM result.

(b) Local view of the RGB-D and visual-feature-based SLAM result.

Figure 8.4: Mapping result generated by RGB-D and visual-feature-based SLAM.

From the result, it can be seen that RGB-D and visual-feature-based SLAM has certain advantages in visual representation. Inside the model, a relatively complete floor area, as well as the real colors and appearance of the corridor, walls, and door openings, can still be observed. This shows that RGB-D data can indeed provide visual information that LiDAR point clouds do not have, and image data can also assist scene understanding and recorded-data inspection.
However, the stability of this method is clearly limited in this experiment. First, the hardware bandwidth of the robotic dog platform is limited. In order to record LiDAR, IMU, RGB-D camera, odometry, TF, and other multi-sensor data at the same time, the resolution and frame rate of the RGB-D camera had to be significantly reduced. This directly affects visual feature extraction, depth image quality, and the matching stability between consecutive frames. Second, the sensor mounting position of the robotic dog is relatively low, so the RGB-D camera mainly observes the forward and lower areas, making it difficult to capture the ceiling completely. Therefore, holes can easily appear in the ceiling region of the reconstruction result.

In addition, the robotic dog produces continuous pose changes during walking and turning, and the RGB images may suffer from rolling-shutter distortion, which is similar to a "jello effect" in the image. This reduces the accuracy of visual feature matching and further affects camera pose estimation and depth point cloud fusion.

Therefore, although this result can preserve the color and appearance information of the corridor inside the model, the overall geometric structure is still not stable enough. In the figure, obvious fluctuations, holes, and discontinuous regions can be observed on the walls and spatial boundaries.

8.1.5 Comparison Result

Overall, the comparison shows that FAST-LIO2, as a LiDAR-inertial odometry method, is the most suitable choice for further processing. This is mainly because the other methods are more strongly affected by the hardware limitations of the robotic dog, sensor bandwidth, camera viewpoint, rolling-shutter distortion, and other practical factors. In addition, the later ray-tracing simulation requires relatively flat and stable geometric surfaces, which makes the LiDAR-inertial odometry result more appropriate for mesh reconstruction and digital twin generation.

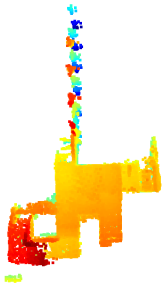
8.2 Digital Twin Generation Results

8.2.1 Point Cloud Processing and Mesh Reconstruction Results

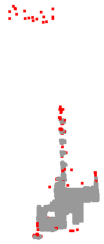
The main implementation parameters used for point cloud preprocessing and surface reconstruction are summarized in Table 8.1.

Table 8.1: Main parameters used for point cloud preprocessing and surface reconstruction.

Step	**Parameter Value**	**Purpose**
Voxel downsampling	Voxel size: 0.10 m	Reduces point cloud density while keeping the main scene structure.
Statistical outlier removal	Neighbors: 20; standard deviation ratio: 2.0	Removes isolated points that are far from their local neighborhoods.
Normal estimation	Search radius: 0.3 m; maximum neighbors: 30	Estimates local surface directions for Poisson reconstruction.
Normal orientation	Consistent tangent plane: 100 neighbors	Improves normal direction consistency before surface reconstruction.
Poisson reconstruction	Depth: 7	Generates an initial mesh from the oriented point cloud.
Density filtering	Lowest 5% density removed	Removes weakly supported surfaces in sparse or noisy regions.
Mesh simplification	Target triangles: 50,000	Reduces mesh complexity before Blender editing and RT simulation.
Mesh smoothing	Simple smoothing: 5 iterations	Reduces local roughness in the reconstructed mesh.



(a) Point cloud after downsampling and denoising.



(b) Comparison of preserved points and removed outliers.

(c) Point cloud after extracting the main region.

(d) Poisson mesh generated from the processed point cloud.

Figure 8.5: Point cloud processing and mesh reconstruction results.

Fig. 8.5 shows the main processing results from the LiDAR-inertial odometry point cloud to mesh reconstruction. This process includes point cloud downsampling, denoising, and Poisson surface reconstruction. Since the original SLAM point cloud still contains noise and scattered points, point cloud preprocessing is required before mesh reconstruction.

Fig. 8.5(a) shows the point cloud result after downsampling and denoising. Compared with the original point cloud, this result reduces redundant points and part of the isolated noise, while still preserving the main structure of the experimental environment.

Fig. 8.5(b) shows the comparison before and after processing. The gray region represents the preserved main structure, while the red region represents the points identified as noise or outliers.

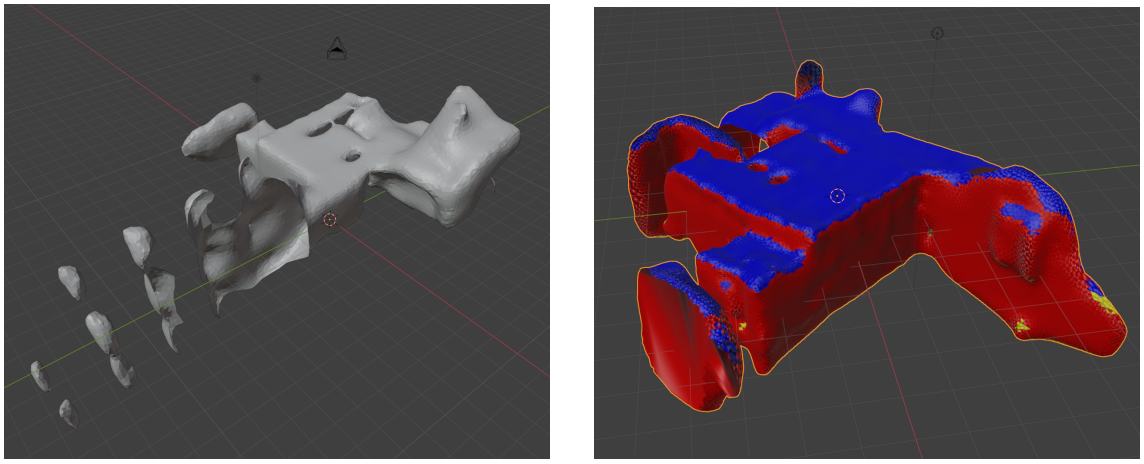
Fig. 8.5(c) shows the point cloud result after extracting the main region.

Fig. 8.5(d) shows the Poisson mesh generated from the processed point cloud. The mesh provides a continuous surface representation, converting the environmental structure from a set of discrete points into a geometric model that can be used for

editing and simulation. This mesh still contains redundant faces, holes, or irregular surfaces in some local regions, so further processing in Blender is still required.

8.2.2 Blender Scene Editing Results

After the initial mesh reconstruction was completed, the model was imported into Blender for further processing. The main goal of Blender scene editing was to clean irrelevant structures in the reconstructed model, adjust the model scale and orientation, reduce the model complexity through smoothing and face reduction, perform material labeling, and prepare the model for ray-tracing simulation.



(a) Mesh model before Blender processing.

(b) Mesh model after Blender cleaning and material labeling.

Figure 8.6: Comparison of the mesh model before and after Blender-based scene editing.

Fig. 8.6 shows the comparison of the model before and after Blender processing. Although the mesh before processing already has continuous surfaces, it still contains some fragments and holes. If these regions are directly preserved, they will add unnecessary complexity to the Sionna RT simulation.
After Blender cleaning and material labeling, the main structure of the model becomes clearer. The main regions such as the corridor, walls, and floor are preserved, while some irrelevant fragments and obvious holes are removed or repaired. The processed model is closer to the geometric input required for the later ray-tracing simulation. In other words, it does not need to preserve all visual details, but should preserve the large-scale structures that have major influence on wireless propagation. At the same time, to improve efficiency and automation, an automatic script was

written in Blender to label the ceiling, walls, and floor as three different materials according to face normals and related geometric information. The wall region was labeled as `MAT_WALL` with the actual material `itu_brick`, the floor region was labeled as `MAT_FLOOR` with the actual material `itu_concrete`, and the ceiling region was labeled as `MAT_CEILING` with the actual material `itu_glass`. The corresponding viewport colors were red, yellow, and blue, respectively. This realizes an automated processing method from data processing to material labeling, which is useful for the later simulation.

It should be noted that the actual ceiling material is closer to the wall material or other common indoor building materials. In the current implementation, the ceiling label is mapped to `itu_glass` mainly to keep it as a separate material category in Sionna RT and to show that the pipeline can handle multiple material labels. This setting is a simplified material assumption and does not mean that the real ceiling is made of glass. In addition, refraction is disabled in the experimental setup.

8.2.3 Geometric Dimension Validation

To further evaluate the geometric reliability of the generated mesh, several main dimensions of the final mesh were measured in Blender and compared with manual measurements from the real environment. The purpose of this validation is not to provide a complete high-accuracy 3D ground truth, but to check whether the generated digital twin keeps the main geometric scale of the real environment. For wireless ray tracing, dimensions such as corridor width, narrow passage width, and indoor height are important because they affect propagation path lengths, reflection positions, blockage relations, and propagation delays.

Figure 8.7 shows the dimension measurements performed on the final mesh. The object scale in Blender was kept at 1.0 during the measurement, so the measured values can be directly interpreted in meters. In the real environment, the main transverse width is approximately 5.50 m, the narrow passage width is approximately 1.90 m, and the indoor height is approximately 2.70 m. In the generated mesh, the corresponding measured values are about 5.46–5.54 m, 1.80–1.85 m, and 2.60–2.66 m, respectively. The comparison is summarized in Table 8.2.

The result shows that the main transverse width is very close to the real measurement, with a mean relative error below 1%. This indicates that the overall horizontal scale of the generated mesh is well preserved. The narrow passage width and the indoor height show slightly larger errors, with mean relative errors of approximately

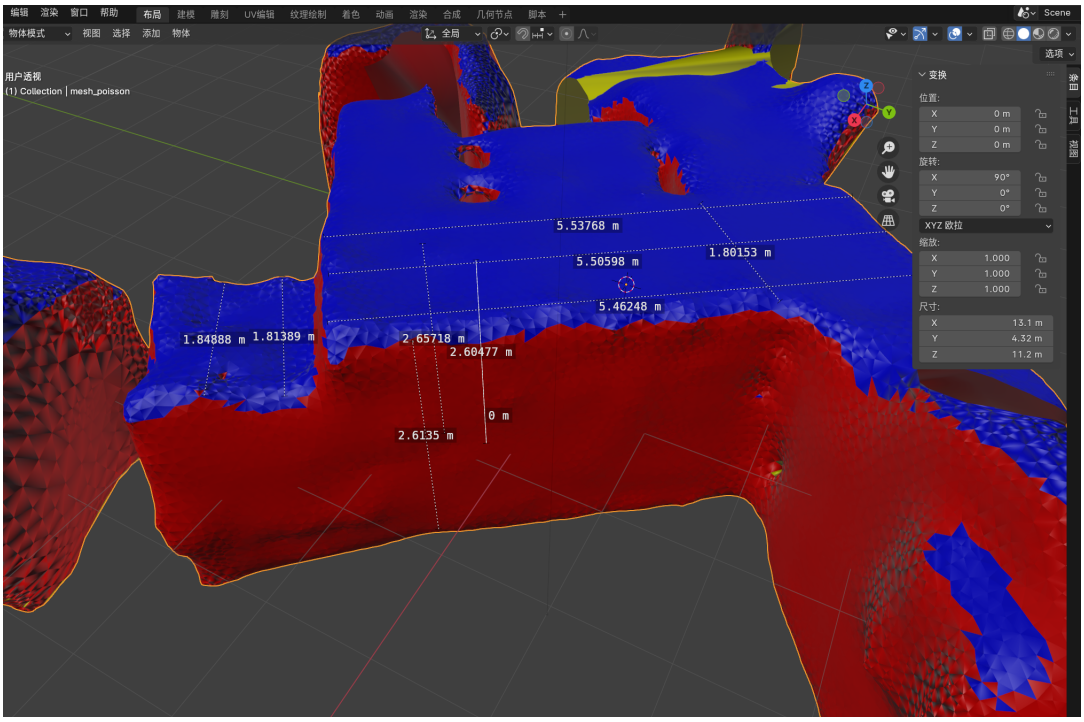


Figure 8.7: Dimension measurements of the generated mesh in Blender.

Table 8.2: Comparison between real-environment dimensions and generated mesh dimensions. The mesh values are taken from the marked measurements in Blender.

Measured item	**Real [m]**	**Mesh [m]**	**Abs. err. [m]**	**Rel. err. [%]**
Main transverse width	5.50	5.46–5.54	0.03	0.49
Narrow passage width	1.90	1.80–1.85	0.08	4.14
Indoor height	2.70	2.60–2.66	0.07	2.77

4.14% and 2.77%, respectively. These errors are still reasonable for the purpose of demonstrating the proposed digital twin generation pipeline, but they also show that local geometric deviations remain in the generated model.
Several factors may contribute to these errors. First, the SLAM point cloud contains local surface roughness and small boundary irregularities. Second, Poisson surface reconstruction smooths the point cloud and may slightly change the position of walls, floors, and ceilings. Third, the Blender editing process removes holes and artifacts, which can also introduce small local shape changes. Therefore, the generated mesh should not be considered a perfect copy of the real environment.
Overall, the geometric dimension validation shows that the generated mesh keeps the main spatial structure of the real environment. This supports the use of the mesh as a reasonable geometric basis for the following Sionna RT simulation. However, the ray-tracing results should still be interpreted as simulation results based on the current geometry and material assumptions, rather than fully measurement-calibrated channel predictions.

8.3 Ray-Tracing Simulation Results

This section presents the RT simulation results obtained from the generated digital twin scene in Sionna RT. Based on the TX, RX, carrier frequency, and RT parameters defined in Chapter 7, Sionna RT computes the valid propagation paths and the corresponding channel results. The extracted results include spatial-domain, delay-domain, frequency-domain, and angular-domain information.
This section contains two types of results. The first type is the radio map, which shows the spatial distribution of path gain over the receiver grid. The second type is the point-level channel result extracted from selected TX/RX positions, including PDP, CFR, AoA, and AoD. To compare the influence of propagation condition, two TX/RX cases are presented: an NLoS case and a LoS case. The NLoS case is mainly affected by reflected paths and local scene geometry, while the LoS case contains a clear direct path between the TX and RX.
First, Fig. 8.8 shows the propagation path visualization in the generated digital twin scene. To make the path structure easier to observe, part of the scene surface is removed, while the main geometric structures are kept for visualizing the spatial relationship between the TX, RX, and propagation paths. The figure shows that wireless propagation in the current indoor scene is not only determined by direct propagation, but is also strongly affected by walls, boundaries, and the geometry of

the reconstructed environment.

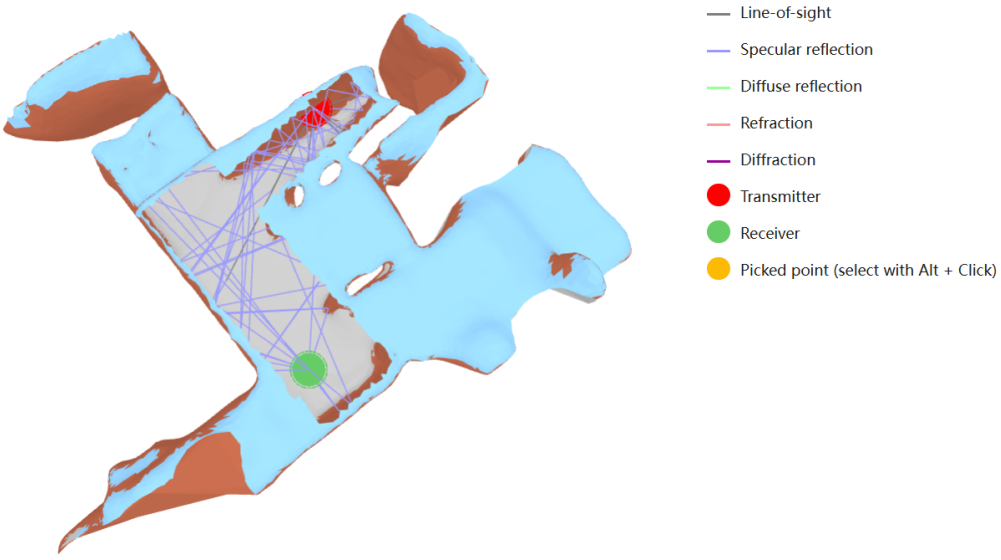
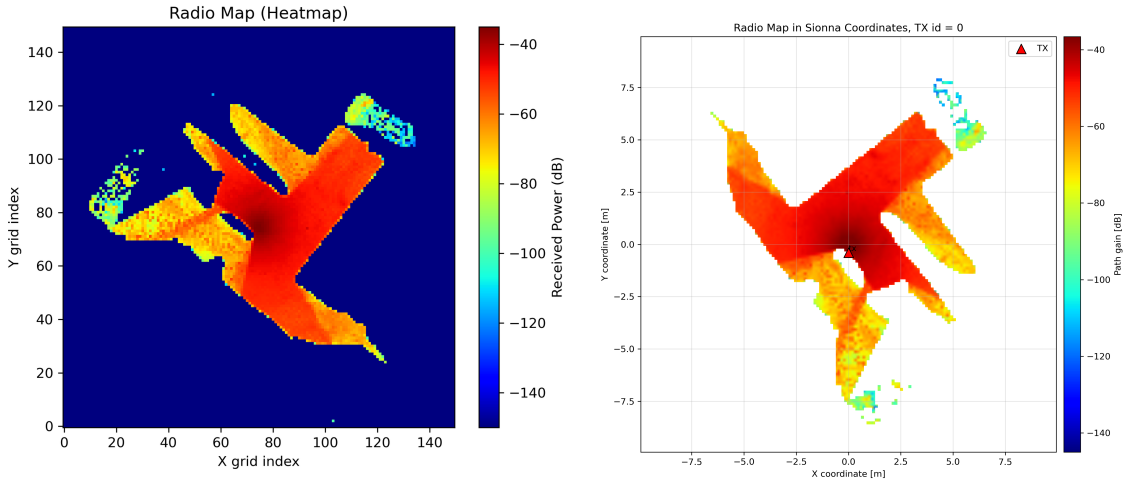


Figure 8.8: Visualization of propagation paths in the generated digital twin scene. Part of the scene surface is removed to make the TX, RX, and ray trajectories easier to observe.

From Fig. 8.8, it can be observed that the current scene supports multipath propagation. Some paths reach the RX after reflections from walls and corridor boundaries, which is consistent with the later PDP, AoA, and AoD results. This also shows that the generated digital twin model can be used by Sionna RT to model and visualize indoor propagation paths.

Fig. 8.9 shows the radio map results generated from the current digital twin scene. Fig. 8.9(a) presents the radio map in grid coordinates, where the horizontal and vertical axes represent the X and Y grid indices. The color scale indicates the path gain at each valid cell. A color closer to red represents a higher path gain, while a color closer to blue represents a lower path gain. The large dark-blue background in Fig. 8.9(a) should not be interpreted as low path gain over the physical environment. It mainly corresponds to invalid matrix entries outside the generated measurement surface.



(a) Grid coordinates.

(b) Sionna coordinates.

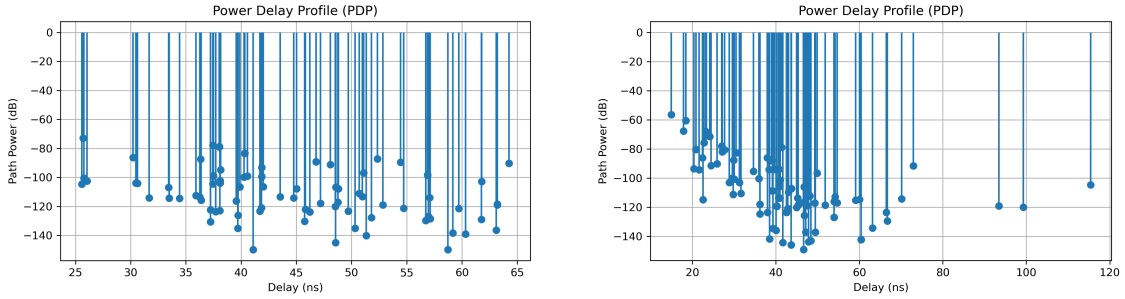
Figure 8.9: Radio map in grid and Sionna coordinate representations.

From Fig. 8.9(a), the valid radio map region has an irregular spatial shape. This is because the measurement surface only covers the effective region where radio map cells are generated. The higher-gain region is mainly concentrated near the central part of the map and expands along the main open propagation regions. This indicates that, under the current TX configuration, positions close to the TX or positions with more direct propagation paths have higher path gain. In contrast, some edge regions, corner regions, and regions farther away from the TX show lower path gain. These low-gain regions may be caused by longer propagation distances, geometric blockage, fewer available reflection paths, or local structural obstruction. To further relate the radio map to the actual digital twin scene, Fig. 8.9(b) shows the same radio map after converting the valid cells into Sionna physical coordinates. In this figure, the X and Y axes are expressed in meters rather than grid indices. The term "physical coordinates" here refers to the metric coordinate system used in the Sionna scene, not geographic coordinates. The red triangle marks the TX position, while the colored points represent valid radio map cells and their corresponding path gain values. After coordinate conversion and mirroring correction, the spatial shape in Fig. 8.9(b) remains consistent with Fig. 8.9(a), but it provides a more direct interpretation of where high- and low-gain regions are located in the scene.

According to the converted radio map data, the current result contains 4301 valid radio map cells. The X-coordinate range is approximately from $-9.67\,\mathrm{m}$ to $6.60\,\mathrm{m}$, and the Y-coordinate range is approximately from $-8.60\,\mathrm{m}$ to $7.93\,\mathrm{m}$. The path gain ranges approximately from $-123.39\,\mathrm{dB}$ to $-35.00\,\mathrm{dB}$, with a median value of

approximately −56.00 dB. The high-gain region around the TX and the gradual reduction in path gain toward more distant or geometrically blocked regions show that the generated digital twin scene can support spatial coverage analysis. Compared with the grid-coordinate representation, the Sionna-coordinate representation is more useful for later analysis of weak-signal areas, geometric blockage, and the relationship between the TX location and surrounding structures.

Fig. 8.10 shows the PDP results for the NLoS and LoS cases. The PDP describes the distribution of path power over propagation delay. In the figures, the markers represent the path power values, while the vertical lines are used only for stem-plot visualization.



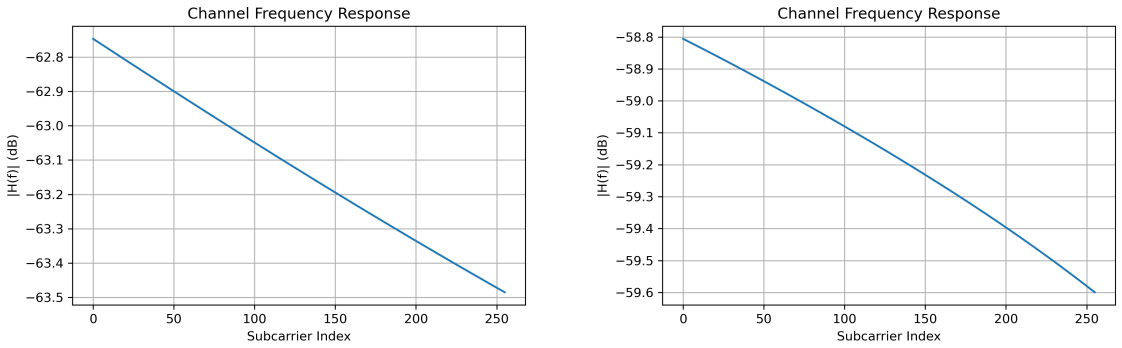
(a) NLoS case. (b) LoS case.

Figure 8.10: PDP comparison between the NLoS and LoS cases.

In the NLoS case, shown in Fig. 8.10(a), the path delays are mainly distributed from approximately 25.57 ns to 64.25 ns. This result shows that the received signal contains several multipath components rather than a single path. The earlier paths usually correspond to shorter propagation routes, while the later paths are related to longer reflected paths.

In the LoS case, shown in Fig. 8.10(b), the delay range extends from approximately 14.95 ns to 115.39 ns. Compared with the NLoS case, the LoS case contains an earlier strong path, which is consistent with the existence of a direct path between the TX and RX. At the same time, later paths are still present. This shows that the LoS condition does not make the channel a single-path channel. Instead, the received signal is formed by the direct path together with reflected and diffuse-reflection-related multipath components from indoor surfaces and boundaries.

Fig. 8.11 shows the CFR magnitude for the two selected TX/RX cases. The horizontal axis is the subcarrier index, and the vertical axis is the channel magnitude in dB.



(a) NLoS case. (b) LoS case.

Figure 8.11: CFR magnitude comparison between the NLoS and LoS cases.

For the NLoS case, the CFR magnitude over the selected 256 subcarriers is approximately between $-63.49\,\text{dB}$ and $-62.75\,\text{dB}$, with an overall variation of about $0.74\,\text{dB}$. For the LoS case, the CFR magnitude is approximately between $-59.60\,\text{dB}$ and $-58.81\,\text{dB}$, with an overall variation of about $0.79\,\text{dB}$. The LoS case therefore has a clearly higher overall channel magnitude than the NLoS case. The average CFR level is increased by approximately $3.95\,\text{dB}$, which indicates that the direct path improves the overall channel strength.

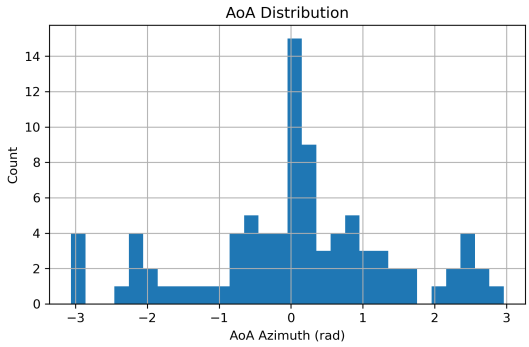
At the same time, both cases show relatively smooth frequency-domain responses over the selected subcarrier range. Neither case shows strong deep fading in the plotted CFR magnitude. This means that the main difference between the two selected cases is not strong frequency selectivity, but the overall channel magnitude. The CFR variation is determined by the complex gains and delays of the propagation paths. A simplified multipath expression of the frequency response can be written as

$$H(f) = \sum_{g=1}^{G} a_g e^{-j2\pi f \tau_g}, \tag{8.1}$$

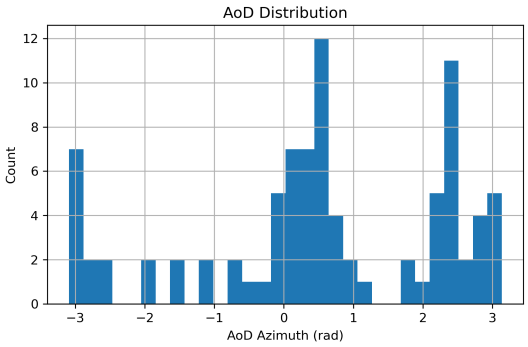
where G is the number of valid paths, a_g is the complex gain of the g-th path, and τ_g is the corresponding propagation delay. If strong propagation paths have clearly different delays, the CFR may show stronger frequency-domain fluctuations. The relatively smooth CFR curves in the current results indicate that the coherent combination of the main paths does not create strong frequency-selective variation over the selected subcarrier range.

Fig. 8.12 shows the azimuth distributions of AoA and AoD for the two cases. AoA describes the direction from which the signal arrives at the RX, while AoD describes

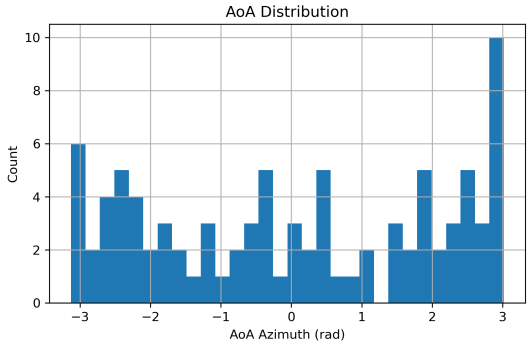
the direction in which the signal departs from the TX.



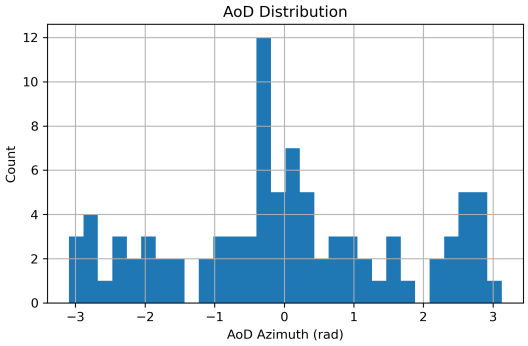
(a) AoA, NLoS case.



(b) AoD, NLoS case.



(c) AoA, LoS case.



(d) AoD, LoS case.

Figure 8.12: AoA and AoD distributions for the NLoS and LoS cases.

Both the NLoS and LoS cases show angular components distributed over multiple azimuth regions. In the NLoS case, the angular distribution is mainly determined by reflected paths and the surrounding scene geometry. The valid paths arrive at the RX and depart from the TX through several directions, which is consistent with the absence of a dominant direct path.

In the LoS case, the direct path provides an important propagation direction, but the AoA and AoD results are still not concentrated in a single angle. Several reflected paths remain visible in the angular domain. This means that even with a clear LoS path, the indoor digital twin scene still creates multipath components through walls, floor, ceiling, and corridor boundaries. The AoA and AoD results further show that the generated scene can support angular-domain analysis of indoor multipath propagation.

The main RT simulation results are summarized in Table 8.3.

Table 8.3: Summary of the RT simulation results for the NLoS and LoS TX/RX cases.

Result	NLoS Case	LoS Case	Main Observation
Valid path entries	89	91	Both cases contain multiple paths.
Delay range	25.57–64.25 ns	14.95–115.39 ns	LoS introduces an earlier direct path.
CFR magnitude range	-63.49 to -62.75 dB	-59.60 to -58.81 dB	LoS gives higher channel magnitude.
CFR variation	0.74 dB	0.79 dB	Both responses are relatively smooth.
Angular distribution	Multiple regions	Multiple regions	Both cases show directional multipath.
Radio map	4301 valid cells; path gain: -123.39 to -35.00 dB		Spatial path-gain distribution over the receiver grid.

As a simple baseline, the free-space path loss was computed for the selected LoS TX/RX configuration. The Euclidean distance between the TX position $(0.2, -0.2, 3.8)$ and the RX position $(0.3, -0.4, -1.8)$ is approximately 5.60 m. The free-space path loss is calculated as

$$PL_{\mathrm{FS}} = 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right), \tag{8.2}$$

where d is the TX–RX distance and λ is the wavelength. With $\lambda \approx 0.0857$ m at 3.5 GHz, the free-space path loss is approximately 58.3 dB, corresponding to a free-space path gain of about -58.3 dB. This value is close to the LoS CFR magnitude range of -59.60 dB to -58.81 dB, while the NLoS case has a lower magnitude range of -63.49 dB to -62.75 dB. This comparison provides a simple sanity-check baseline, but it should not be interpreted as a full validation of the digital twin because the free-space model does not include reconstructed geometry, material assignment, reflection, blockage, or multipath effects.

Overall, the results in this section indicate that the generated digital twin scene can be used for RT simulation in Sionna RT. The radio map provides the spatial path-gain distribution over the receiver grid. The PDP results show that both NLoS and LoS cases contain multiple delay components. The CFR results indicate that the LoS case has a higher overall channel magnitude, while both cases remain relatively smooth over the selected subcarriers. The AoA and AoD distributions

further confirm that the propagation paths are distributed over multiple spatial directions. These results provide the basis for comparing how TX/RX propagation conditions affect the simulated channel characteristics.

8.4 Comparison of Channel Characteristics Under NLoS and LoS Conditions

This section compares the channel characteristics of the NLoS and LoS TX/RX cases. The purpose is not to compare different map-quality levels. Instead, the comparison uses two different propagation conditions to show how the geometry of the digital twin scene affects channel outputs. Since refraction is disabled in the current RT settings, the differences discussed in this section mainly come from LoS availability, reflection, diffuse reflection, blockage, and scene geometry.

First, LoS availability has a direct influence on path delay and channel strength. In the LoS case, the earliest path delay is approximately 14.95 ns, while in the NLoS case the earliest path delay is approximately 25.57 ns. This shows that the direct path creates a shorter propagation path between the TX and RX. The relation between path length error and delay error can be written as

$$\Delta\tau = \frac{\Delta d}{c}, \tag{8.3}$$

where Δd is the path length difference and c is the speed of light. Therefore, changes in TX/RX position, wall position, or coordinate conversion can directly shift the delay-domain result. This is why the geometry and scale of the digital twin scene are important for PDP analysis.

Second, the LoS case has a clearly higher CFR magnitude than the NLoS case. The CFR magnitude is about -63.49 to -62.75 dB in the NLoS case, while it is about -59.60 to -58.81 dB in the LoS case. The average CFR level is increased by approximately 3.95 dB in the LoS case. This indicates that the direct path improves the overall channel strength. In contrast, when the direct path is not available, the received signal depends more on reflected paths and local scene geometry, which leads to a lower overall channel magnitude.

Third, the LoS condition does not remove multipath propagation. Although the LoS case contains an earlier direct component, it still contains later path components with delays up to approximately 115.39 ns. These later paths are caused by reflections and diffuse-reflection-related interactions with indoor surfaces and boundaries.

Therefore, the LoS case should not be interpreted as a single-path channel. It is still a multipath channel formed by the direct path together with reflected components. Fourth, the CFR variation is small in both cases. The NLoS case has a CFR variation of approximately 0.74 dB, while the LoS case has a variation of approximately 0.79 dB. This means that neither case shows strong deep fading over the selected subcarrier range. The main difference between the two cases is the overall channel magnitude, rather than frequency selectivity.

Fifth, the AoA and AoD distributions show the influence of scene geometry on propagation directions. In both cases, the valid paths are distributed over several azimuth regions. For the LoS case, the direct path gives one important propagation direction, but reflected paths still arrive and depart from other directions. For the NLoS case, the angular distribution depends more strongly on reflected paths from walls, corridor boundaries, and other surfaces. Thus, AoA and AoD are determined not only by the TX/RX locations, but also by the positions and orientations of surfaces in the digital twin scene.

Sixth, the radio map shows that scene geometry affects spatial coverage. The high-gain regions mainly appear near the TX or in more open propagation regions, while low-gain regions appear near edges, corners, or blocked regions. This means that the digital twin geometry affects not only one selected TX/RX pair, but also the spatial path-gain distribution over the receiver surface.

Overall, the LoS case has an earlier path delay, a higher CFR magnitude, and a stronger direct component than the NLoS case. However, both cases still show clear multipath behavior. The NLoS case has lower overall channel strength and depends more on reflected paths and local geometry. The comparison shows that the digital twin geometry, coordinate consistency, and surface quality can directly influence path selection, delay distribution, channel magnitude, angular distribution, and radio map interpretation.

8.5 Research Questions Answered

This section summarizes how the research questions defined in Chapter 1 are answered by the results of this thesis. Since the purpose of this thesis is to evaluate an engineering workflow from robotic SLAM reconstruction to Sionna RT-based ray tracing, the answers are discussed together with the available evidence and the remaining limitations.

For RQ1, this thesis investigates whether robotic SLAM reconstruction can accu-

rately represent the main structure of the real indoor environment. The results show that, among raw sensor data, ICP, FAST-LIO2, and RTAB-Map, FAST-LIO2 provides the most suitable reconstruction result for the following digital twin generation stage. Compared with the other methods, the FAST-LIO2 result shows better structural continuity, less visible drift, and better preservation of the main wall, floor, and ceiling structures. The geometric dimension validation further shows that the main transverse width, narrow passage width, and indoor height of the generated mesh are close to the manual measurements from the real environment. The mean relative errors are approximately 0.49%, 4.14%, and 2.77%, respectively. Therefore, the results indicate that robotic LiDAR-inertial SLAM can represent the main large-scale geometry needed for the later ray-tracing simulation. However, this does not mean that the reconstructed model is a fully accurate copy of the real environment, since no high-precision ground-truth trajectory or complete 3D reference scan is used.

For RQ2, this thesis investigates how the point cloud generated by SLAM should be processed and converted into a simulation scene usable by Sionna RT. The results show that a SLAM-generated point cloud cannot be directly used as a Sionna RT simulation scene. It must first be converted into a mesh-based model with continuous surfaces, manageable geometric complexity, and compatible material labels. In this thesis, this is achieved through point cloud downsampling, denoising, normal estimation, Poisson surface reconstruction, mesh simplification, Blender-based scene editing, material assignment, and export to a Mitsuba/Sionna-compatible XML format. After these steps, the generated digital twin can be successfully loaded into Sionna RT and used for wireless ray-tracing simulation.

For RQ3, this thesis investigates what wireless channel results can be obtained from ray tracing based on the reconstructed model. The Sionna RT results show that the generated digital twin can provide several channel-related outputs, including propagation paths, radio maps, path gain, PDP, CFR, AoA, and AoD. These outputs show that the reconstructed model is not only a visual 3D model, but also a usable geometric input for wireless channel simulation. The comparison between LoS and NLoS cases further shows that different propagation conditions lead to different delay-domain, frequency-domain, and angular-domain characteristics. The LoS case has a stronger direct component and an earlier dominant path, while the NLoS case has lower channel strength and depends more on reflected paths and local geometry.

For RQ4, this thesis investigates how mapping and mesh quality can affect the simulated channel characteristics. The results indicate that the quality of the point

cloud and mesh can directly influence propagation path selection, reflection point positions, blockage relations, delay distribution, channel magnitude, AoA/AoD distribution, and radio map interpretation. Geometric dimension errors affect the propagation path length and therefore the propagation delay. Local surface roughness, holes, and irregular boundaries may also change local reflection behavior and blockage relations. Therefore, mapping and mesh quality must be considered when interpreting the ray-tracing results. However, since this thesis does not include real wireless channel calibration, this influence is analyzed through geometric validation and simulation-based results rather than through fully quantified measured-channel errors.

9

Conclusion

9.1 Summary

9.1.1 Key Findings

This thesis addressed the conversion from real-world environmental data to wireless ray-tracing simulation, and implemented a complete workflow from robotic data collection, SLAM reconstruction, digital twin generation, to Sionna RT simulation.

In the data collection stage, a Unitree Go2 robot equipped with a Livox MID-360 LiDAR, an Intel RealSense D435i RGB-D camera, and other sensors was used to collect data from the real environment. ROS 2 rosbag recording was used to store multi-sensor data, including LiDAR, IMU, RGB-D images, odometry, and TF information. The experimental results show that the platform can capture the main spatial information required for later reconstruction in an indoor corridor environment.

In the SLAM reconstruction stage, LiDAR-only odometry, LiDAR-inertial odometry, and RGB-D visual-feature-based SLAM were compared. The comparison shows that the LiDAR-inertial point cloud map generated by FAST-LIO2 provides better structural continuity, flatter wall and floor surfaces, less local drift, and better suitability for later mesh reconstruction. Therefore, the point cloud map generated by FAST-LIO2 was selected as the main geometric input for digital twin generation.

In the digital twin generation stage, the SLAM point cloud was converted into a mesh-based digital twin model with continuous surfaces and manageable geometric complexity for wireless simulation. This was achieved through point cloud downsampling, denoising, mesh reconstruction, and Blender-based optimization.

The geometric dimension validation further shows that the generated mesh keeps the main scale of the real environment. The measured main transverse width, narrow passage width, and indoor height in the mesh are close to the manual measurements

from the real environment. The mean relative errors are approximately 0.49%, 4.14%, and 2.77%, respectively. This indicates that the generated mesh can preserve the main spatial structure required for ray-tracing simulation, although it should not be considered a fully accurate copy of the real environment.

In the ray-tracing simulation stage, the generated digital twin scene was successfully imported into Sionna RT. The simulation produced wireless channel results, including propagation paths, radio maps, PDP, CFR, AoA, and AoD. The results show that the generated scene can support the analysis of indoor propagation paths, spatial path gain distribution, delay-domain behavior, frequency-domain response, and angular distribution. The comparison between LoS and NLoS cases further shows that direct paths, reflected paths, blockage, and scene geometry all affect the final channel characteristics.

Overall, this thesis verifies the feasibility of using robotic SLAM-based digital twin generation for wireless ray-tracing simulation. The experiments also show that the final wireless simulation results are not only determined by the Sionna RT parameter settings, but are also affected by the quality of the previous stages, including mapping quality, mesh completeness, geometric dimension accuracy, surface smoothness, and material assignment.

9.1.2 Limitations

This thesis still has several limitations.

First, the experimental scene is relatively small. The experiments mainly focus on an indoor corridor and a limited indoor area. This environment is suitable for verifying the basic workflow from robotic mapping to wireless ray-tracing simulation, but it cannot fully represent more complex large-scale buildings, open spaces, or mixed indoor-outdoor scenarios.

Second, the robotic data collection was limited by the hardware performance and the recording setup. LiDAR, camera, IMU, odometry, and TF data had to be recorded in the same system. To avoid occupying too much data bandwidth and to keep the recording process stable, the image resolution had to be reduced. In addition, the viewing range of the camera was limited by its mounting angle on the robot. These factors may affect the quality of the recorded data and the final reconstruction result.

Third, the SLAM reconstruction can still be affected by point cloud noise, glass regions, moving objects such as people, and occlusions during scanning. Although

FAST-LIO2 produced relatively stable results in the experiments, its output is still not a perfectly accurate model of the real environment.

Fourth, the current Blender-based model repair, optimization, and material labeling method is mainly based on the normal directions of large surfaces. This simplified method can quickly separate large wall and floor regions, but it cannot reliably identify detailed materials in the scene, such as glass doors, windows, wooden doors, or other small objects.

Finally, the ray-tracing results in this thesis are mainly based on Sionna RT simulation outputs and have not been fully compared with real wireless measurements. For example, received power, PDP, and CFR have not been calibrated using measured channel data. Therefore, this thesis can show that the proposed pipeline can generate and run a wireless ray-tracing scene, but it cannot fully prove that the simulated numerical results are identical to the real wireless channel.

9.2 Engineering, Societal, and Sustainability Implications

From an engineering perspective, the proposed workflow can reduce the dependence on manual modeling for wireless simulation scenes. By collecting real-environment data with a robotic platform and combining SLAM, mesh reconstruction, and Sionna RT, the workflow can generate a wireless digital twin that is closer to the actual spatial structure. This method can support indoor wireless network planning, coverage analysis, weak-signal area identification, and TX/RX placement evaluation. It therefore provides a more practical geometric basis for site-specific wireless simulation.

From a societal and ethical perspective, robotic data collection may include RGB-D images, LiDAR point clouds, and indoor spatial information. Privacy protection should therefore be considered during data collection and data processing. Data should be collected only in authorized environments, and personally identifiable information, such as faces, personal belongings, or sensitive indoor details, should not be stored or published. When the robot operates indoors, pedestrian safety, route planning, and site management should also be considered.

From a sustainability perspective, digital-twin-based simulation can be used to evaluate wireless coverage and channel characteristics before real deployment. This can reduce repeated on-site measurements and repeated adjustment work. More accu-

rate network planning can also help avoid unnecessary device deployment and energy waste. Therefore, the proposed workflow has practical value for both engineering efficiency and sustainable wireless network deployment.

9.3 Future Work

Future work can be carried out in the following directions:
First, the current workflow is still split across different platforms. Data recording on the robotic dog relies on ROS 2 in a Linux environment, while the later data processing and ray-tracing simulation are mainly performed on Windows. Future work can improve the system integration by migrating the required environment and algorithms, so that data collection, processing, reconstruction, and simulation can be completed on a single platform.
Second, the automation level of digital twin generation can be further improved. The current material assignment mainly relies on surface normal directions, which can quickly separate large structures such as walls, floors, and ceilings. However, it is still limited when identifying more detailed objects and materials. Future work can introduce automatic semantic segmentation to recognize specific object types, such as walls, glass, doors, and furniture, and then convert the semantic labels into radio materials usable by Sionna RT.
Third, the proposed workflow should be tested in larger and more complex environments, together with real wireless measurements. The experiments in this thesis mainly focus on an indoor corridor environment. Future experiments can be extended to multi-room areas, large buildings, open indoor spaces, and some outdoor environments. Real measurement equipment can also be introduced to compare the simulation results with measured channel data. This would make it possible to evaluate more clearly how mapping quality, mesh completeness, material settings, and ray-tracing parameters affect the channel results, and to assess the reliability of the pipeline for practical wireless network planning.

Bibliography

[1] NVIDIA, "Introduction to sionna rt," Online documentation, 2026, available: https://nvlabs.github.io/sionna/rt/tutorials/Introduction.html.

[2] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 70–76, 2022.

[3] P. Öhlén, C. Johnston, H. Olofsson, S. Terrill, and F. Chernogorov, "Network digital twins –outlook and opportunities," *Ericsson Technology Review*, vol. 2022, no. 12, pp. 2–11, 2022.

[4] H. Wang, J. Zhang, G. Nie, L. Yu, Z. Yuan, T. Li, J. Wang, and G. Liu, "Digital twin channel for 6G: Concepts, architectures and potential applications," *IEEE Communications Magazine*, vol. 63, no. 3, pp. 24–30, 2025.

[5] J. Hoydis, F. Aït Aoudia, S. Cammerer, M. Nimier-David, N. Binder, G. Marcus, and A. Keller, "Sionna rt: Differentiable ray tracing for radio propagation modeling," in *2023 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2023, pp. 317–321.

[6] F. A. Aoudia, J. Hoydis, M. Nimier-David, S. Cammerer, and A. Keller, "Sionna rt: Technical report," *arXiv preprint arXiv:2504.21719*, 2025.

[7] P. Testolina, M. Polese, P. Johari, and T. Melodia, "Boston twin: the boston digital twin for ray-tracing in 6g networks," in *Proceedings of the 15th ACM Multimedia Systems Conference*, 2024, pp. 441–447.

[8] Y. J. Noh and K. W. Choi, "High-precision digital twin platform based on ray tracing simulation," in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2024, pp. 1464–1465.

[9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[10] Z. Fan, L. Zhang, X. Wang, Y. Shen, and F. Deng, "Lidar, imu, and camera fusion for simultaneous localization and mapping: a systematic review," *Artificial Intelligence Review*, vol. 58, no. 6, pp. 1–59, 2025.

[11] J. Zhu, H. Li, and T. Zhang, "Camera, lidar, and imu based multi-sensor fusion slam: A survey," *Tsinghua Science and Technology*, vol. 29, no. 2, pp. 415–429, 2023.

[12] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[13] H. Wang, J. Zhang, G. Nie, L. Yu, Z. Yuan, T. Li, J. Wang, and G. Liu, "Digital twin channel for 6g: Concepts, architectures and potential applications," *IEEE Communications Magazine*, vol. 63, no. 3, pp. 24–30, 2025.

[14] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," in *White paper*, vol. 1, 2014, pp. 1–7.

[15] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[16] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, 2014.

[17] C. Campos, R. Elvira, J. J. Gómez Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[18] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 2006, pp. 61–70.

[19] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 1–13, 2013.

[20] Y. Ge, F. Wen, H. Kim, M. Zhu, F. Jiang, S. Kim, L. Svensson, and H. Wymeersch, "5G SLAM using the clustering and assignment approach with diffuse multipath," *Sensors*, vol. 20, no. 16, p. 4656, 2020.

[21] ITU-R, "Recommendation itu-r p.1407-1: Multipath propagation and parameterization of its characteristics," International Telecommunication Union, Radiocommunication Sector, 2003, available: https://www.itu.int/rec/R-REC-P.1407.

[22] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[23] M. Labbé and F. Michaud, “Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[24] International Telecommunication Union, “Recommendation itu-r p.2040-4: Effects of building materials and structures on radiowave propagation in the range of 1 mhz to 450 ghz,” International Telecommunication Union, Recommendation ITU-R P.2040-4, 2025. [Online]. Available: https://www.itu.int/rec/R-REC-P.2040

[25] H. Obeidat, A. Ullah, A. AlAbdullah, W. Manan, O. Obeidat, W. Shauieb, Y. Dama, C. Kara-Zaïtri, and R. Abd-Alhameed, “Channel impulse response at 60 ghz and impact of electrical parameters properties on ray tracing validations,” *Electronics*, vol. 10, no. 4, p. 393, 2021.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden

CHALMERS
UNIVERSITY OF TECHNOLOGY