RoboBIM: A Novel Low-Cost Autonomous Interior BIM 3D Reconstruction Mobile Robotic System

Submitted 2023/7, in partial fulfillment of
the conditions for the award of the degree
Master By Research.

Fuhua JIA
20415354

Supervised by Dr. Adam Rushworth

Department of Mechanical, Materials and Manufacturing Engineering,
Faculty of Science and Engineering,
University of Nottingham, Ningbo China

Abstract

Building Information Modeling (BIM) has greatly aided lean management in construction, providing cost savings, efficiency improvements, collaboration, and sustainability throughout the building lifecycle. Several countries and industries promote the development and practical application of BIM. However, the current scan-based reconstruction process of BIM requires costly specialized equipment and extensive manual data collection and processing, hindering rapid reconstruction and continuous updating. As a solution, Autonomous Mobile Robots (AMR) are expected to be the most promising platform for autonomous 3D BIM reconstruction.

This thesis achieves a brief review of robot-based BIM reconstruction methods through a data flow-based classification approach, identifying three main challenges to robots: limited Field of View (FOV) of sensors, lack of scan quality assessment, and rough autonomous movement control. To address these challenges, a novel rotating Light Detection and Ranging (LiDAR) gimbal design is presented, along with scan distortion removal and scan quality evaluation algorithms. A three-step autonomous navigation method that integrates scan quality and scan parameters is proposed, enabling efficient, reconstruction-oriented autonomous navigation. A robot is designed and built to validate the proposed algorithms, and real-world testing demonstrates their effectiveness.

Keywords

BIM reconstruction, Mobile Robotic, Optimal Scanning Parameters, LiDAR Gimbal, Autonomous Navigation.

Contents

List of Figures

List of Tables

Abbreaviations

AGV Autonomous Guided Vehicle 26

AMR Autonomous Mobile Robots 1

APF Artificial Potential Field 38

BFS Breadth-first Search 38

BIM Building Information Modeling 1

BLDC Brushless Direct Current 52

CAD Computer-Aided Design 13

CPP Coverage Path Planning 35

DFS Depth-first Search 38

FMCW Frequency Modulated Continuous Wave 27

FOV Field of View 1

GNSS Global Navigation Satellite System 30

HT Hough Transform 41

ICP Iterative Closest Point 31

IMU Inertial Measurement Unit 24

KHT Kernel-based Hough Transform 42

L-M Levenberg-Marquardt 33

LiDaR Light Detection and Ranging 1

LOAM LiDAR Odometry and Mapping 33

LOD Level of Detail 18

MEMS Microelectromechanical System 26

MM-W Millimeter Wave 26

NDT Normal Distribution Transform 31

OPA Optical Phased Array 26

PCA Principal Component Analysis 41

PHT Probabilistic Hough Transform 42

RANSAC Random Sample Consistency 41

RGB-D Red-Green-Blue-Depth 24

RL Reinforcement Learning 36

ROS Robot Operation Sytem 33

SL Structured Light 27

SLAM Simultaneous Localization and Mapping 23

STC Spanning Tree Coverage 38

TOF Time of Flight 25

UNNC University of Nottingham Ningbo China 7

Chapter 1

Introduction

1.1 Background

Under the challenge of building a resource-efficient, environment-friendly and sustainable society, the global building industry is seeking more intelligent and efficient ways of designing and building to meet global demand and help create more intelligent and more resilient spaces. Throughout the long life cycle of a building, site construction, maintenance, upkeep, and renovation must be based on a thorough knowledge and assessment of the building's structure and current condition to achieve sound management and resource conservation and recycling. With the development of information science, the digitalisation of buildings has become mainstream, and BIM has come into being.

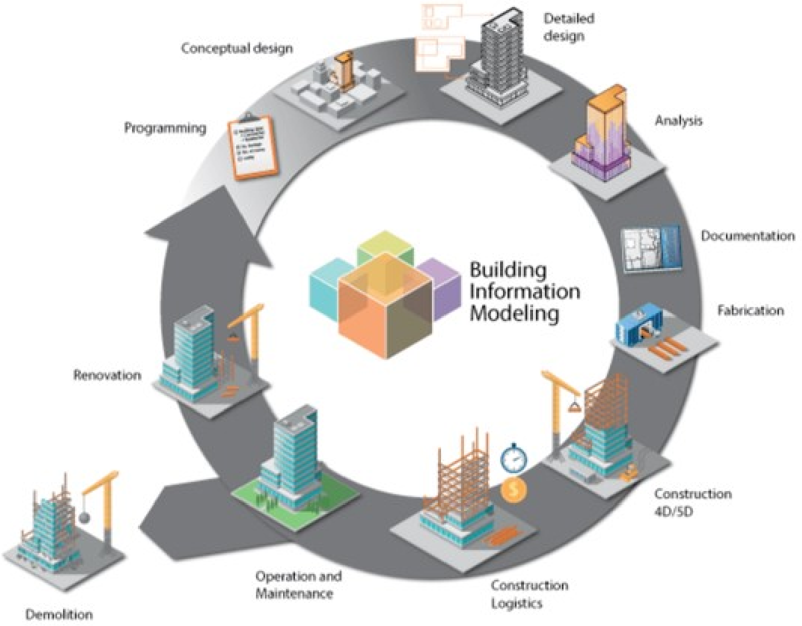


Figure 1.1: The overview of BIM. BIM facilitates a collaborative project process which integrates all stakeholders at the beginning of the design and allows connection through the project life cycle [1].

Early BIM is often closely associated with digital drawings that originated in the 1950s. Along with the birth of Computer-Aided Design (CAD) technology, architectural design and construction digitalisation were first seen. In the initial stage, computers mainly assisted engineers in 2-dimensions(2D) drafting to save time and reduce errors. Over time, CAD gradually became more widely used, and the software evolved so that CAD could communicate product design information directly from the drawing to the computer-driven manufacturing tool. In other words, products could be manufactured directly from the computer without further processing. With the further development of CAD software, engineers were no longer satisfied with 2D drawings, and 3D modelling capabilities were gradually developed. In the 1980s, 3-dimensions(3D) modelling function-assisted architectural design appeared in the industry, showing the powerful ability of information science. In the early 1990s, software integrating graphical analysis and simulation was developed to provide information about how buildings would behave under different conditions, including building orientation, geometry, material properties, and building systems. Shortly after this, parametric modelling software emerged, meaning that if some model element changed, other relevant elements would automatically change. The digital simulation of the construction process model with construction time as a variable was also proposed in this era.

At the beginning of this century, information science developed rapidly, and the above software functions were brought together and further enriched. 3D models of buildings, monitoring of construction progress and quality, and other functions were integrated into a virtual platform that allowed interactive access to all data models. This stage is often referred to as 3D BIM.

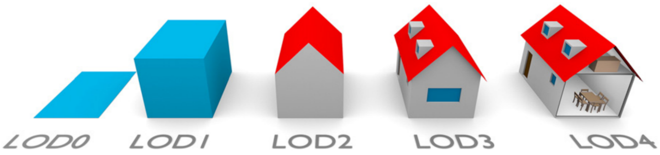


Figure 1.2: To better describe the geometric features of BIM, Level of Detail (LoD) was proposed by CityGML [2] to describe the geometric feature richness of BIM. the geometric feature model described by LoD is usually considered to be 2D and 3D BIM [3].

The addition of BIM over time, which gives BIM a temporal attribute, is called 4-dimensional(4D) BIM. Similarly, adding the cost of construction, maintenance and many other items to BIM is called 5-dimensional(5D)BIM. N-dimensional(N-D) BIM adds more dimensions to BIM, such as energy, carbon footprint analysis [6, 7], environmental effect and other perspectives, which has gradually shown great vitality in the past few years. It has gained wide attention both from academia and industry.

One such example is the UK Government's mandate for BIM Level 2 in

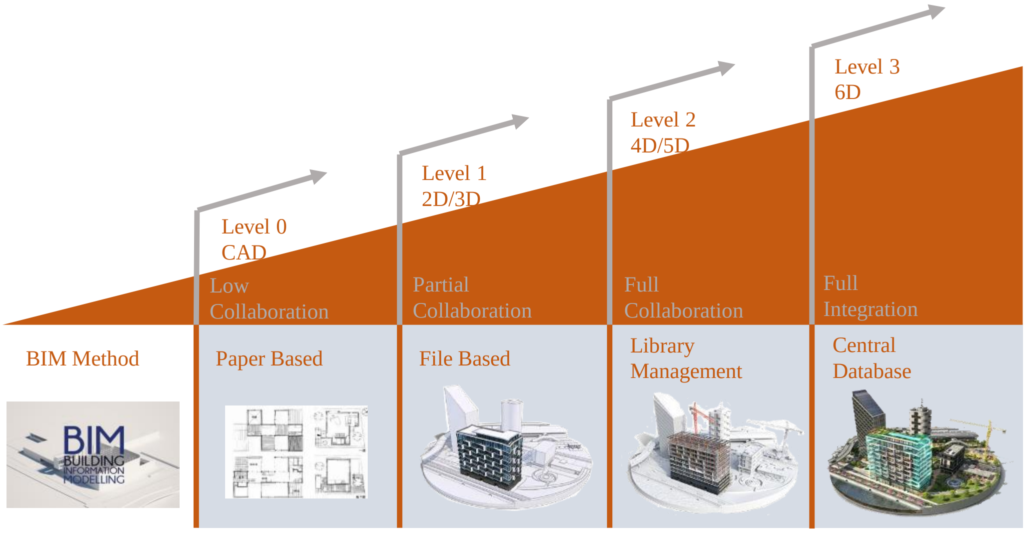


Figure 1.3: The expansion of the dimensions of BIM is essentially an upgrade of the implementation methods. From drawings to construction and maintenance, there are different methods of implementing BIM in each stage. It is worth noting that although the research of BIM has reached more to the dimension, in Implement, BIM is still in the 3D/4D stage.

April 2016 [8]. Similarly, the Ministry of Housing and Urban-Rural Development of the People's Republic of China released new policies and requirements for the BIM industry in 2022 [9]. Reviewing the development history and current application of BIM, it is foreseeable that BIM will develop into a standard platform and tool in the coming years.

In contrast to the potential for development offered by BIM, implementing this technology takes time and effort. In many cases, BIM is simply a model that can be accessed virtually and is more akin to a virtual demonstration. From a commercial perspective, implementing BIM entails a significant financial outlay in terms of project migration costs and the expense of training personnel to use BIM software. Technically, builders must have faith in a hypothetical model. At the same time, technical managers tend to prefer that BIM coexists alongside traditional design and construction methods to verify BIM's credibility.

The application of BIM for new buildings is often ideal and generated as-planned. However, the reality of the building is often different from the as-planned BIM, which is further complicated when considering existing buildings, as the BIM of many is often composed of architectural sketches and construction drawings that need 3D models and detailed documentation. In many cases, heritage sites' architectural sketches and construction drawings are completely missing. These discrepancies between BIM and actual buildings have led to the need for automatic generation and updating of BIM for existing buildings, which has both academic significance and essential engineering applications, as well as the potential to be extended to other usage scenarios of general significance, such as rescue and firefighting operations.

Table 1.1: Common Laser Total Station Scanners and Parameters

Model	Range	Data Noise	Density	Price
Trimble X12 [10] 	365 m	2mm @ 10m	2,180,000 Points per second	$280,000
Leica RTC360 [11] 	130 m	3mm @ 10m	2,000,000 Points per second	$78,000
Topcon GLS-2000 [12] 	500 m	3mm @ 50m	120,000 Points per second	$100,000

Generating BIM from scan data is a state-of-the-art method, with the scan data typically being derived from expensive laser-total station scanners. The typical field BIM generation method involves fixing the laser-total station scanners in the building and measuring the installation position of the total station scanner accurately by operators, followed by manual conversion of the scanning data to BIM. Another popular method is using hand-hold laser scanner to replace the laser-total station, but it suffered precision issues. Common models and their specifications are shown in Fig.1.1. The scanning data is typically stored in the form of a point cloud, named after its cloud-like point data storage structure. This process is labour-intensive and error-prone, thus making it challenging to guarantee BIM precision, efficiency and regular updating. This procedure is usually referred to as building 3D reconstruction.

Given their excellent task-carrying capabilities, mobile robots have demonstrated their potential in several aspects. Automating the BIM generation process through robotics could bring significant progress to the BIM industry, further exploring the functional scenarios of mobile robots and expanding their scope of applications. Compared to manual methods, mobile robot-based approaches to BIM generation are quite different, which presents several challenges:

- The sensors of mobile robots are designed for navigation and obstacle avoidance. They have a narrow FOV and low information density, making it challenging to meet the dense data requirements needed for 3D reconstruction. At the same time, the locomotion of mobile robots will bring motion distortion to the scanning data, which will lose accuracy if the distortion is not dealt with properly.

- Robotic sensors are not able to evaluate scan quality. The uniformity of the point cloud, the range of the distribution and the scanning time per frame can have a significant impact on the quality of the reconstruction. These parameters are closely coupled to the robot's motion.
- Mobile robots need to be able to scan buildings autonomously, which means that the robot should avoid omissions, reduce repeated paths, and scan and explore buildings autonomously. This is a challenge for the robot's navigation and planning algorithm.

1.2 Aims and Objectives

The aim of the project is to develop a mobile robot based solution for BIM 3D reconstruction of building interior scenes. The robot should be able to scan and collect data autonomously, thus reducing the need for expensive specialized equipment and extensive manual data collection and processing, and should address the challenges of robot-based BIM reconstruction methods. In the face of the above challenges, the following objectives should be achieved by this project:

- Determine, through a literature review method, the feasibility of applying robotic-level sensors to BIM reconstruction to meet Level of Detail (LOD) 300 requirements.
- Design a novel structure that extends the FOV of robotic-level sensors to cover roofs, walls, and floors within the robot's operating area.
- Design algorithms to eliminate sensor motion distortion caused by the added structure and calibrate the sensor with the new structure.
- Design algorithms to evaluate scan quality and establish relationships between scan quality and scan parameters.
- Fulfill two different navigation needs with navigation algorithms: exploration navigation and scanning task navigation.
- Design and build the robot, including both software and hardware components. Deploy the above structures and methods on the robot to verify their feasibility.

1.3 Contributions and Overview

This dissertation is organised around several chapters presenting the context for the RoboBIM(Robot BIM) project at UNNC and reporting the findings of this master by research(MRes) as reflected in the most significant output it produced.

Chapter 1 introduces the concept, history, level and significance of BIM and elaborates on the issues of implementing BIM in practice. The contradiction between the vital role of BIM and the current immature BIM implementation method brings this topic. This MRes work shows an effective way to realise robot-based BIM automation. The proposed subtopic and methods will be explained in the following parts.

Chapter 2 summarises the procedure and detail of mobile robot-based BIM, explains how the data of the BIM scanning method is produced from robotic-level sensors hardware and sensing algorithms, and how the robot realised localisation and mapping by itself. It also summarises how the data are further processed for BIM and BIM generation methods. In addition, a brief overview of existing robotic-based BIM methods is included in this chapter.

The current robotic-level sensors and sensing algorithms have many drawbacks. The Field of View(FOV) of light detection and ranging(LiDAR) is usually limited due to its mechanical structure, and the scanning density cannot be evaluated by sensing algorithms. In order to improve the performance of robotic-level sensors and the density of point cloud, Chapter 3 presents a novel structure of sensor gimbal and a new sensing methodology based on a locomotion compensation method. This research revealed a novel view of LiDAR-based scanning and the evaluation of scanning density. This research provides information that can guide LiDAR scanning parameters selection and the design of the BIM robot and contributes a new method for locomotion compensation.

The significance of the robot-based BIM method is automation. After reviewing all related works and papers, as far as the author's knowledge, no such navigation method can take good care of scanning quality based on the scanning data. Motivated by the scanning method of the chapter 3, a novel three-step navigation approach is proposed in the chapter 4. By which the robot can realise a fast, well-organised and 3D reconstruction quality-orientation automatic navigation. This method is a vital part of the RoboBIM project and is summarised as a deliverable part of the RoboBIM project overview paper.

With the overview of existing methods and work of BIM robots and knowledge, a new all-in-one BIM mobile robot solution is proposed in Chapter 5, where the hardware and software systems of the RoboBIM robot are illustrated here. The workflow, implementation detail and experiment result are further explained in this chapter. The RoboBIM robot is the integration of the algorithms and devices of the aforementioned chapters. This project shows great progress on the robot-based and robot-assist BIM reconstruction.

Chapter 6 summarises the research in this MRes, synthesises the main findings, and provides a discussion of future research directions and applications of this work.

Chapter 2

Background

2.1 Robotic-based BIM Automatic Reconstruction Workflow

As explicated in the introduction chapter 1, the production of BIM still requires human involvement, including the crucial steps of data gathering, processing, and management. Recent advancements in mobile robot technology have shown promise in enabling automation of the BIM generation process by leveraging their autonomous positioning, navigation, mapping, and obstacle avoidance capabilities [13, 14]. The robotic-based BIM generation process can be divided into five distinct ***stages*** based on the flow of data and the goals of the algorithms used, namely, ***Data Acquisition, Simultaneous Localization and Mapping (SLAM), Path-planning and Navigation, Data Processing, and BIM Reconstruction***, with the former four stages being facilitated by mobile robots. By automating these stages, mobile robots have the potential to significantly reduce the need for human intervention, resulting in increased efficiency and accuracy in BIM generation.

It is noteworthy that these five stages are partially independent, yet they may also have an effect on each other. For example, the path generated by the navigation stage may lead to low quality of the scanning data, affecting the BIM reconstruction results.

- Data acquisition refers to the procedure of collecting scanning data and the robot itself information from sensors installed on the robot, such as LiDaR, radar, Red-Green-Blue-Depth (RGB-D) camera, Inertial Measurement Unit (IMU), and built-in rotation encoders.
- The SLAM stage aims to position (localization) the robot and creates a rough map to mark the obstacles and reachable zone.
- The Path-planning and Navigation part is vital to the robot-based

automatic BIM reconstruction method. It can guide the robot in working in the known, semi-known or unknown environment.

- Compared to the rough data filter and cluster method of the SLAM stage, the data processing is the preparation for BIM reconstruction where coloured image, background segmentation information, temperature distribution and other time-variable elements will be added to the final BIM.

- BIM reconstruction is the most mature part. Many methods and algorithms have been developed to transfer and convert the input data from the abovementioned stages into BIM.

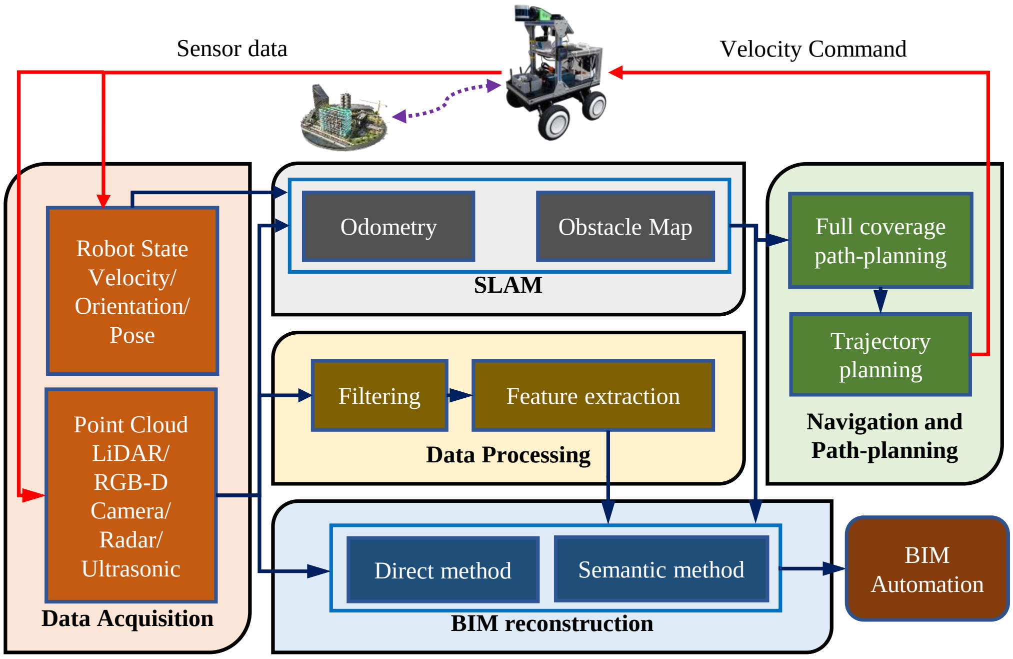


Figure 2.1: The mobile robot-based BIM automatic reconstruction workflow.

2.1.1 Data acquisition

Researchers are concerned about the point cloud and robot state in the data acquisition stage. The point cloud is a discrete set of data points in cartesian coordinates space. The points may represent a 3D shape or object with the coordinates (X, Y, Z). The individual point can not represent complete information about the object being described. In addition, usually, point cloud data does not have an order. Point cloud in any order is equivalent.

LiDaR is the most popular way to get the scanning point cloud [15–18], named after its working principle and function, Light Detection and Ranging. The authors of [19–21] conducted comprehensive reviews of the principles and applications of LiDaR systems. Based on its working principles, LiDaR can be categorized into Time of Flight (TOF) LiDaR and Triangulation LiDaR. ToF LiDaR measures range by comparing the elapsed time between the transmitted and received signal. It dominates the market due to its simple structure, such as laser total stations on the construction site and high-end robotic devices. However, it suffers issues such as interference from sunlight or other LiDaR devices. Triangulation method LiDaR usually has two or more laser transmitting and receiving devices, and the measurement accuracy is obtained by the measurement difference obtained from triangulation. It is usually simple in structure and low in cost and has been widely cited in low-cost scenarios like the sweeping robot and logistics Autonomous Guided Vehicle (AGV) [13, 14].

Based on the laser beam steering mechanism, LiDaR can be further categorized into mechanical LiDaR and solid-state LiDaR. There are motor-driven moving parts where the laser transmitters and receivers are installed in mechanical LiDaR. Solid-state LiDaR comes in multiple forms, including Microelectromechanical System (MEMS) LiDaR, FLASH LiDaR, and Optical Phased Array (OPA) LiDaR [22]. It refers to a steering system without bulky mechanical moving parts, resulting in a relatively small FOV (typically 20–50 degrees horizontally).

However, LiDaR is an optic-based sensor that suffers from some common issues. Tibebu Haileleo et al. [23] found that using LiDaR sensors in an environment with transparent entities, like windows and undecorative glass walls, causes the sensor to report inaccurate range data, leading to a potential collision triggered by errors. Zhou Haoyu et al. found that undecorated walls may degrade the LiDaR with cone FOV from acquiring point cloud information [24]. In the BIM application, they may cause drift and tracking lost issues.

Radar stands for radio detection and ranging, another type of rangefinder [25, 26]. It is based on the emission and detection of electromagnetic waves in radio frequencies ranging from 3 MHz to 300 GHz (with wavelengths from 100 m to 1 mm) [22]. Compared with its counterpart LiDaR, radar has superior detection performance under extreme weather conditions since waves within this spectrum have weak interaction with dust, fog, rain, and snow. The Millimeter Wave (MM-W) spectrum ranges from 30 GHz to 300 GHz, which provides broad bandwidth and narrow beams for sensing, thus allowing finer resolution [27,28]. The shortcomings of millimetre-wave radar, such as high cost, small field of view, and low resolution for small objects, restrict its development. In recent years, the concept of Frequency Modulated Continuous Wave (FMCW) radar has been gradually applied in the development of mm-w radar, which can determine the range and velocity of the object simultaneously and has shown good performance [22].

Ultrasonic is the most mature method of the ranging method, it is based on the TOF principle, but the medium is ultrasonic wave rather than light and radio wave [29]. The core part of the ultrasonic sensor is the transducer, which is used to convert some other type of energy into an ultrasonic vibration. The transducer can be further classified by energy source into mechanical, magnetostrictive and mechatronic types [30]. Compared to LiDaR and Radar, its advantages are simple, robust and easy to read. However, the feedback frequency is relatively low, and the ultrasonic wave speed can be influenced by temperature, moisture and air pressure [31,32].

The vision method, which has recently been a hot research topic, can be concluded by hardware and software categories. The depth camera is the hardware method. Structured Light (SL) and Stereo cameras are the main principles of depth cameras. Structured light uses light of a known pattern projected onto the scene, and then the way the pattern deforms is used in constructing the depth map [33,34]. SL method is vulnerable to sunlight since interference may occur with the projected light pattern, but it does not require an external light source. Stereo cameras try to mimic human vision by using two cameras facing the scene with some distance between them [35,36]. The images from these cameras are acquired and then used to perform visual feature (relevant visual information) extraction and to match to obtain a disparity map between the cameras' views.

The generation of scanning point cloud data by software method is usually related to the stage of SLAM. Generally speaking, the software-based method has no limitation on the sensor. Monocular cameras [37,38], binocular cameras, depth cameras [39,40], event cameras [41,42], infrared cameras [43,44], fisheye cameras [45,46], and even mobile phone and tablet cameras can be used as input devices [47].

Besides the point cloud, the robot's orientation, velocity, and acceleration are vital to the data acquisition and SLAM stage. Generally, IMU is the first choice. The principle and structure of IMU have been widely studied by many researchers [48,49]. Moden robot-level IMU comprises a gyroscope, an accelerometer, and a magnetometer (optional). The gyroscope measures the three-axis angular rate and estimates the relative orientation of the robot to the world frame. The accelerometer measures the acceleration and then projects it to the world frame with the subtracted gravity vector. The velocity and position can be obtained via integration and double integration, respectively [50,51].

Different gyroscopes and accelerometers have been constructed based on working principles. Most modern robot-level IMUs are MEMS types. Since these devices output the acceleration and the angular rate, the robot's velocity and position are obtained by integrating the measured data. Therefore, any drift or bias in measuring acceleration and angular rate will cause an accumulation of errors in the estimation of velocity and position [22]. The common sensors of the data acquisition stage and their properties are shown in table 2.1.

Comparing the table 2.1 and 1.1, it can be easily found that robot-level sensors' range density and precision are weaker than laser total station scanners. Nevertheless, the gap is not too huge to cover, and the precision of robot-level sensors satisfies the BIM reconstruction LOD 300 level requirement. It is foreseeable that robot-level sensors could achieve BIM reconstruction if a proper scanning and data enhancement approach is adopted.

2.1.2 SLAM

SLAM technology is the most promising direction of modern robotic research, whose applications are widely used in robotic, autonomous driving and daily life. Many comprehensive reviews of SLAM technology from the view of computer vision, LiDaR, ultrasonic and mm-w Radar can be found in [52–58].

Odometry is a method for estimating the change of the robot's position over time in the SLAM stage. Usually, the coordinate system where the odometry information is located will have a fixed positional relationship with the global/earth coordinate system, as shown in figure 2.2.

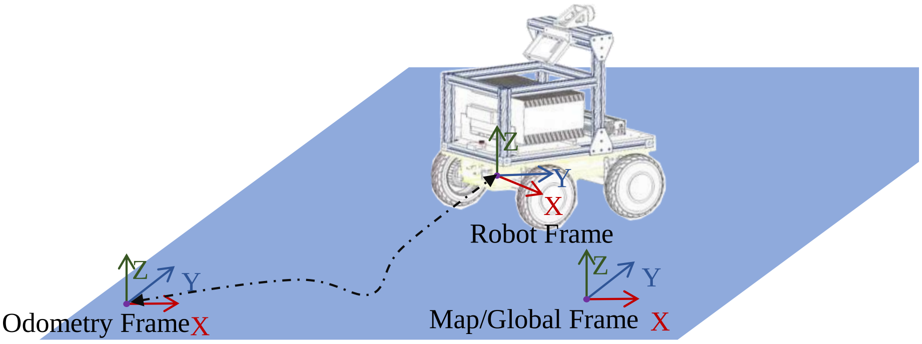


Figure 2.2: The odometry plays a huge role in SLAM. The odometry coordinate is fixed on the earth/global coordinate system while the robot coordinates are varying.

The most direct method is Global Navigation Satellite System (GNSS). In the BIM reconstruction area, the GNSS method is often used on the drone-based scanning system [59, 60]. Another mature and well-developed way to get odometry is robot built-in wheel encoders [61–63], which are

Table 2.1: Common sensors of data acquisition step and their types, principle and parameters

Type	Sensors	Model	Principle	Range	Precision
LiDAR		RpLiDAR S2	Triangulation	40m	2cm
		Livox Mid 70	TOF	260m	5cm
		Sick LMS531	TOF	80m	2cm
		Velodyne VLP32C	TOF	200m	2cm
RGB-D		Realsense D455	Stero Vision	4m	2%
		Kinect V1	SL	2m	1%
		Zed 2	Stero Vision	20m	5%
Radar		Continental ARS408	77Ghz	250m	1.7m
		TI AWR2243	81GHz	4m	2%
IMU		CH108	MEMS	8g and 1000deg/s	0.5%
		Xsens MTI-630	MEMS	1g and 2000deg/s	0.2%

counting the wheel's rotation while spinning. However, the building inside scenario is GNSS-denied, and the slippery wheels will jeopardize the state estimation of the encoder odometry. Due to these issues, scanning data-based odometry methods are becoming increasingly popular due to their robustness to different environments. The scanning data (point cloud and robot information) from the last stage will be further processed into odometry and map in this stage.

The core problem of the scanning-based odometry method can be described as follows: with several consecutive frames of given scanning data from the robot, find a rotation R and translation t to describe the locomotion of the robot. Mathematically, what needs to be built is a loss function *E(R,t)*on R and t obtained from several frames of data. Through optimization, an acceptable solution is obtained.

Many famous matching and optimization algorithms are proposed to solve this classic problem. After the odometry has already been obtained, mapping is no longer complicated. By superimposing all the scan data on the odometry in a certain way, a map relative to the global coordinate system will be obtained. In the case of mapping large scenes where there may be no overlapping areas between scanned data, the concept of closure-loop detection is proposed to optimize the accuracy. Specifically, a localization algorithm can correct the global map when it can identify an area the robot has reached twice or more before and correct the local and global map. Clustering, semantics and many other methods are used for closure-loop detection.

Due to the different characteristics of the sensors, the mileage calculation method can be further divided according to the types of sensors. At the same time, they can generally include matching-based, feature-based, grid-based, surface element-based and semantic-based methods.

In the field of LiDaR sensors, Iterative Closest Point (ICP) and Normal Distribution Transform (NDT) are traditional matching-based methods widely adopted for mobile robots' SLAM problems. A comprehensive review of the ICP algorithms and application evaluation of NDT can be found in [64, 65].

The general idea of the ICP algorithm is the most basic and direct way, which is to find a relationship between several scanning data. The workflow is shown in algorithm1.

Where $X = \{x_1, x_2, \cdots, x_{N_x}\}$ and $Y = \left\{y_1, y_2, \cdots, y_{N_y}\right\}$ are any frame of point cloud with number of N_x or N_y of points in the frame with consecutive timestamp. T_{max} is the maximum mathcing timeout and E_t is the threshold value of error for stopping solving.

A complete ICP algorithm should include functional blocks such as a data filter, initial transformation, associate solver, outlier filter, and error

Algorithm 1:

```
Input: $PointCloud : X_{t0} = \{x_1, x_2, \cdots, x_{N_x}\}, Y = \{y_1, y_2, \cdots, y_{N_y}\}, N_x = N_y, T_{max}, E_t$
Output: $\boldsymbol{R,t}$
while $RunningTime \leq T_{max}$ or $E(R,t) \geq E_t$ do
    $E(R,t) = \frac{1}{N_y} \sum_{i=1}^{N_y} Difference \{(N_x, N_y)_{X_{frame}}\}$;
    $\min(E(R,t))$ via Optimization ;
end
```

minimization [22]. The advantages of the ICP method are easy to implement and high precision with less noise at initialization, but ICP is easily trapped in local minima in noise and dynamic environment [66]. There are a lot of improved works for the ICP and formed the ICP algorithm family. For the matching step, besides basic point coordinate, surface normal [67], curvature [68]), descriptor matching (laser intensity [69]), and mixed method are proposed. The finding process is often accelerated by data structures such as k-D trees [70] to find the correspondences with the shortest distance and/or similar properties. The error function is where most ICP family members differ. The basic ICP method takes the Euclidean distance between selected point sets as the index; Park, Soon-Yong et al. proposed the point-to-plane error index method [71]; the generalized ICP [72], which introduces a probabilistic representation of the points and plane-to-plane error-index to ICP family.

Compared to ICP, NDT is more robust and protective of the raw point cloud because it does not downsample the scanning point cloud. NDT can solve the both 2D registration [73] and 3D registration [74]. The NDT algorithm is a probabilistic model-based implementation of point cloud alignment, where the input point cloud is partitioned into several grids. Then the Gaussian probability distribution of the pair point clouds in the grid is computed, and the distribution is matched to obtain the odometry solution. The NDT algorithm avoids the nearest neighbour search and whole point cloud storage and improves the execution efficiency.

LiDAR Odometry and Mapping (LOAM) is a typical method based on feature matching, which has occupied the first place in the public dataset KITTI since Zhang Ji et al. first proposed it [75]. LOAM computationally selects and classifies feature points with smoothness into three four types: sharp, less_sharp, flat and less_flat, and then uses the Levenberg-Marquardt (L-M) method to find the transition between successive scans by minimizing the point-to-edge distance for selected points and the point-to-plane distance for plane points. Inspired by LOAM, several methods have been proposed, including LeGO-LOAM [76], which first segments the original point cloud using the range image and then extracts features by a similar process to LOAM with a two-step L-M optimization. Table 2.2 shows other LOAM family methods.

Table 2.2: The characteristics and comparisons of some well-known algorithms

Category	Method	Loop-Closure Detection	Downsampling
Scan-matching	ICP [72]	No	Yes
	NDT [74]	No	No
	GMM [82]	No	Yes
	IMLS [83]	No	No
	DLO [84]	No	No
	MULL [85]	Yes	No
Feature-based	LOAM [75]	No	No
	F-LOAM [86]	No	Yes
	A-LOAM [87]	No	No
	Lego-LOAM [76]	No	No
	SA-LOAM [88]	Yes	No
Grid-Matching	Cartographer [77]	Yes	Yes
	Hector SLAM [79]	No	Yes
	CoreSLAM [89]	No	Yes
	KartoSLAM [80]	No	Yes
Surfel-Based	Suma [81]	Yes	No
Segmentation-Based	Suma++ [90]	Yes	No
	Segmap [91]	Yes	No

Cartographer algorithm [77], integrated into Robot Operation Sytem (ROS), is an algorithm grid-based method proposed by Google for mobile robot localization and mapping, which is small in code, fast in solving, and widely used in service robots such as floor sweepers. The grid voxel filter downsamples the collected point clouds. Although some details are lost, the obstacle features can be preserved. The known odometry is differentiated to obtain the guess of angular and linear velocities. The new odometry is then obtained by combining the IMU sensor and odometry guesses after transferring them into the downsampled point cloud data for an optimal solution. Another feature of the Cartographer algorithm is the closer-loop detection based on all the local maps that have been generated and the current scan, which significantly improves its robustness.

In addition to these mentioned algorithms, there are many other well-known algorithms in the LiDaR field, such as Gmapping [78], Hector [79], Karto [80] and Suma [81]. They all use characteristic matching methods and play a great role in various applications. The characteristics and comparisons of some well-known algorithms are shown in the table 2.2.

Radar odometry and mapping algorithms have yet to be well studied compared to LiDaR. An overview of radar applications in robotics can be found in [92, 93]. In analogy to LiDaR methods, radar-related algorithms can be classified into two main categories, direct and indirect methods [94, 95]; the indirect methods involve feature extraction and association, whereas direct methods forego these procedures. The advantage of the radar is the capability of radial velocity measurement, [96, 97] taking full use of these properties, where a RANSAC algorithm is invoked for non-stationary outlier removal.

Radar measurements are noisy and, thus, may worsen the performance of scan-matching algorithms used for LiDaR, such as ICP and NDT [22]. However, some matching and mapping methods for LiDaR have also shown good performance after modification, such as G-ICP [98], NDT with clustering [99, 100]and Surfel-Based method with a new feature-extraction algorithm [95].

SLAM technology, which relies solely on ultrasonic sensors, is typically used in underwater robots and is rarely used in ground robots.

Due to the rising of computer vision technology, CV-based SLAM technology has been a hot topic for a long time. The visual SLAM technique is comprehensively summarized annually and found at [56, 57]. According to the technology base, the technology of visual SLAM can be divided into four types: Filter, Key Frame, Direct Tracking and Spatial Occupancy.

In summary, sensors and sensing technologies for robot sensing have been developed very maturely and are of great value in practical applications. This chapter on data acquisition and SLAM technology only reviews sensor fundamentals, configurations and algorithms and does not provide complete coverage of sensor materials, analysis principles and other issues. A comprehensive review of the above two stages is published as part of this project and can be found in the publication list.

2.1.3 Path-planning and Navigation

Autonomous motion is one of the significant advantages of mobile robots in BIM reconstruction applications. In addition to SLAM, path planning and navigation are other important technical point for autonomous robot movement. In the field of BIM reconstruction, robot path planning and navigation refer that the robot should plan a continuous path in a semi-known building environment autonomously (rough obstacle map and odometry only), enabling the scanning sensors to collect all important scanning information of the building (e.g. corners, windows, corridors) without missing them. The process is to find several essential path points in the environment, connect them to form a continuous path, send them to the trajectory planner for smoothing, and finally send to the robot velocity, acceleration and trajectory tracking commands.

Coverage Path Planning (CPP) determines a path that passes over all points of an area or volume of interest while avoiding obstacles [101]. Many robotic applications require autonomous robots with CPP capabilities, such as windows cleaning [102], automated harvester [103], robotic demining [104] and robotic painting [105]. Cao et al. defined the requirements for full coverage trajectory planning of robots in a flat 2D environment [106], which are also applicable to other coverage scenarios, with the following requirements.

- The robot must avoid all obstacles.
- The robot must move through all points in the target area, covering it completely.
- Simple motion trajectories (e.g., straight lines or circles) should be used (to simplify control)
- The robot fills the area without overlapping paths as much as possible
- The "best" path is required under the available conditions.

During actual task execution, the robot must balance these above requirements to achieve the task. Coverage algorithms can be classified into heuristic, complete algorithms based on the guarantee of complete coverage, or online and offline methods based on whether they incorporate dynamic sensor data information, a classification originally proposed by Choset et al. [107]. Complete literature reviews of the CPP problem can be found in [101, 107–110]. It is worth mentioning that in recent years, with the development of machine learning and artificial intelligence, especially Reinforcement Learning (RL). New solutions to the CPP problem have been proposed in large numbers. A further classification of the CPP algorithm based on the unknown region segmentation method can be found in the table 2.3.

Initially, an effective way to solve the CPP problem was randomization. Early sweeping robots such as Karcher's RC3000, Electrolux's Trilobite, and iRobot's Roomba [111] represented this approach, moving randomly across the floor so that if the motion time were long enough, the floor would be completely cleaned. The advantages of this approach are low sensor dependency and low computational resource requirements, but the runtime is entirely unpredictable, and the cost is enormous.

The CPP approach based on cell decomposition usually involves decomposing the free space (i.e., the space without obstacles) into simple, non-overlapping regions called meta-cells [131]. All meta-cells fill exactly the available space. These regions have no obstacles, are "easy" to cover, and can be swept by the robot using simple movements. Of these, Breadth-first Search (BFS) and Depth-first Search (DFS) are the most basic algorithms. DFS is a traversal algorithm that starts at the root and continues searching in a specific direction to the end of a branch; breadth-first search BFS is also a traversal algorithm that starts at the root and explores all nodes at the current depth before moving to the next depth level. The algorithm starts at the root node (in the case of graphs, choose any node as the root) and explores each branch as far as possible before backtracking. Additional memory (usually stacks) is needed to keep track of the nodes discovered so far along the specified branches, which helps in the backtracking of the

Table 2.3: Coverage Path Planning Algorithms

Category	Heruistic	Method
Direct Method	N/A	Zigzag
	Random Walk	Brownian motion [86] Levy Fight [112]
	Contact Sensor	Rectilinear decomposition [113]
Cellular Decomposition	Breadth	Breadth-first search
	Depth	Depth-first search
	Motion Cost	A-Star D-Star
	Chaotic Function	Chaotic Coverage [114]
	Tree Search	Spanning Tree Coverage [115]
	Imaginary Field	Artificial Potential Field [116]
	Bio-Heruistic	Genetic-Algorithm [117] Particle Swarm [118] Ant Colony [119]
	Ecology-based	Invasive Weed Optimization [120]
Graph Decomposition	Tree Search	Rapidly Exploration Random Tree [121] Motion Planning [122] View Planning [123] Next-best-view [124]
Learning-based	Value-based RL	Q-Learning [125] Deep Q-Network [126]
	Policy-based RL	Proximal Policy Optimization [127]
	Actor-based RL	DDPG [128] Asynchronous Advantage Actor-critic [129]
	Model-based RL	Markov Decision Process [130]

graph, and both stacks will usually be referred to as the open set and the closed set. The BFS and DFS algorithms are the ideas underlying the CPP method and are widely used in tree searches. Even in other CPP methods, they are usually used as inspiration.

Chaotic CPP is a deterministic technique that consists of chaotic systems that generate coverage trajectories based on chaotic motions. Since the motion is predetermined, faster coverage in the workspace is guaranteed. However, the algorithm relies on the kinematic motion of the robot subject to kinematic constraints, which requires an exhaustive modelling analysis.

The CPP algorithm based on Spanning Tree Coverage (STC) subdivides the workspace into a finite sequence of disjoint cells. Then it uses a tree traversal algorithm such as DFS or BFS to find the best path to cover each unoccupied cell. It has the advantages of being fast and efficient but low processing power for huge obstacles and a high path overlap rate. Artificial Potential Field (APF) algorithms typically generate a fictitious repulsive and attractive force around the surrounding obstacles and the target, respectively, to ensure that the robot reaches the target while maintaining the distance between the robot and the obstacles [132]. GA is a meta-heuristic population-based stochastic algorithm inspired by the natural laws of biogenetics [133] and the survival of the fittest and reproduction for solving search problems [133].GA can generate near-optimal solutions to achieve fast solutions to path planning problems through parallel processing. The genetic algorithm has good global search capability in the area coverage but needs better stability and long computation time due to the large search space complexity [134].

Graph decomposition-based methods usually mean that the decomposition of unknown regions does not rely on grids or cells but is directly based on coordinates and graphs. Among them, RRT is the typical representative. It uses incremental techniques in tree structures to construct a graph to search and explore in the space of free or obstacle configurations [121]. The algorithm is designed to search the high-dimensional space and handle motion dynamics planning efficiently. The advantages of such algorithms come from their non-metric decomposition prior operations that allow such algorithms to reach non-regular points; however, they also bring obvious drawbacks, such as the difficulty for the robot to pass through unstructured environments.

RL is a type of machine learning in which an agent learns to achieve a desired goal by processing sequential decisions [135]. RL is neither supervised nor unsupervised learning but learns from experience through trial-and-error rules.RL is widely used in robotics applications [136], and in the context of CPP problems. In short, the adaptability of RL methods using suitable robotic platforms in dynamically changing environments remains a major challenge in robotics.

2.1.4 Data processing

The workflow diagram presented in Figure 2.1 illustrates that the input for the data processing step primarily comprises scanning data and robot information from the first step discussed in Chapter 2.1.1, as well as odometry and rough positioning map information obtained in the subsequent step described in Chapter 2.1.2. Although BIM reconstruction can be accomplished without data processing if the input data is free from imperfections, such data often contains significant noise, irrelevant information, and unclassified data points. To address this issue, robot information, including the IMU, is typically processed in the SLAM stage utilizing a Kalman filter to minimize noise and generate optimal estimates. Consequently, this stage will focus on the data processing for point cloud data.

The processing of point clouds starts with filtering, smoothing and noise reduction of the point cloud data. After reducing noise interference, it will be easier to realize the classification and feature extraction of the point cloud. By combining the classification and features, the semantic segmentation information of the point cloud can be obtained, and the point cloud can be well used for BIM 3D reconstruction.

Point cloud filtering has been extensively studied, [137–139] reviewed the point cloud filtering methods from the view of signal processing, performance and data structure, respectively. MATLAB has several toolbox built-in methods for de-noising, down-sampling, estimating normal direction and nearest neighbour finding for point cloud, making it one of the easiest-to-use point cloud processing tools [140].

Common numerical filter methods can be divided into several categories based on statistics, neighbourhood, projection and signal processing.

Bayesian filters [141], iterative least squares filters [142], and kernel function clustering filters [143] are all statistical-based filters. Neighbourhood-based filters use a similarity measure between a point and its neighbourhood to filter conditions, and the similarity can be determined by point, normal or plane methods [144]. Bilateral filters are representative of this type of method [145]. Projection-based methods handle point clouds by adjusting the position of each point in the point cloud with different projection strategies [146]; Laplace transforms, as an important fundamental tool in signal processing, can also obtain better results with point cloud filters based on it [147]. In addition, partial differential equations, voxel grids [148] and other methods have also played a great role in point cloud filtering.

It should be noted that since point clouds are disordered and structured, it is necessary to design point cloud filters to preserve these characteristics. Otherwise, they will corrupt the point cloud data.

After filtering, the point cloud's classification operation and feature extraction is the most important step. Feature points, such as ground, wall, and poles, can be extracted to increase the data dimension and enrich the details. From the technical basis classification, it can be divided into three main categories: numerical, model-matching and learning-based algorithms.

Principal Component Analysis (PCA) and clustering methods are the most commonly used numerical algorithms. PCA methods are based on calculating the normal to the point cloud to extract the point cloud plane and are widely used in the open-source library PCL. Clustering-based methods are not based on a specific mathematical theory but group points with similar geometric features, spectral features, or spatial distribution into the same uniform pattern. k-means [149], mean shift [150, 151] and fuzzy clustering [152] methods are widely used.

The core idea of model matching is to match the point clouds to different original geometries. The most widely used model-fitting methods are based on two classical algorithms: the Hough Transform (HT) and Random Sample Consistency (RANSAC). HT is a classical feature detection technique in digital image processing. It originally appeared in [153] for line detection in 2D images. One of the main drawbacks of HT is the lack of boundaries in the parameter space, which can lead to high memory consumption and long computation times [154]. Therefore, several studies have been conducted, and such algorithms include Probabilistic Hough Transform (PHT) [155], adaptive PHT [156], progressive PHT [157], and Kernel-based Hough Transform (KHT) [158]. In addition to planes, other basic geometric elements can also be segmented by HT. For example, reference [159] comprehensively describes the HT-based method for sphere identification. A comprehensive review of the RANSAC method can be found in [160, 161]. RANSAC methods are widely used for planar segmentation, such as building facades [162] and indoor scenes [163]. RANSAC is a non-deterministic algorithm. Therefore the model detected by the RANSAC-based algorithm may not exist. Many RANSAC-based algorithms have emerged in the last decades to improve their efficiency, accuracy, and robustness.

Recent research has focused on learning-based point cloud classification and feature extraction, which has been aided by open-source point cloud databases. This area of research has gained popularity due to its potential applications, and several detailed reviews can be found in [164–166]. Learning-based point cloud classification and feature extraction can be categorized into unsupervised, semi-supervised, and supervised learning. PointNet is a widely recognized deep learning framework that operates directly on points without using convolution operators. However, it is important to note that annotating point cloud datasets is a significantly more time-consuming and laborious task than annotating ordinary image data.

2.1.5 BIM reconstruction

Given a point cloud of a facility, BIM modelling involves three tasks: modelling geometry, object classes and material properties, and establishing relationships between components. These tasks are not necessarily performed sequentially; depending on the workflow, they may be staggered [167].

The goal of geometric modelling tasks is to create a simplified representation of building components by fitting geometric primitives to point cloud data. Geometric primitives can be individual surfaces or volumetric shapes. Simple geometric elements may not model surfaces such as moulding or decorative sculpture. More complex structures can be modelled non-parametrically, for example, using triangular meshes such as STL, or modelled from a database of known object models [168]. Since BIM usually uses solid shape definitions, surface-based representations need to be converted to solid models.

The modelled components are labelled with object categories. Standard BIM categories include walls, roofs, floor slabs, beams, and columns [169]. In addition, custom object categories can be created based on the needs of each project. Objects can be further expanded with additional metadata, such as material properties or links to custom component specifications.

Topological relationships between components and spaces are important in BIM and must be established. Connection relationships indicate which objects are connected to each other and where they are connected. For example, adjacent walls and the floor slab at the bottom will be connected at their boundaries. In addition, containment relationships are used to code the location of components embedded in each other, such as windows and doors embedded in a wall.

In current practice, the creation of BIM is primarily a manual process performed by service providers who are contracted to scan and model the facility [170]. A project can take several months to complete, depending on the complexity of the facility and the modelling requirements [171]. No single software tool can complete all aspects of the process, and information may be lost due to limitations in data exchange standards or errors in the implementation of standards in software tools [172].

There are two main approaches to geometric modelling. The first approach is to fit geometric elements directly to 3D data. Geometric modelling software typically includes tools for fitting geometric primitives to data, as well as specialized tools for pattern modelling. These tools are semi-automated and require a large amount of user input. The second geometric modelling approach uses cross-sectional and surface extrusion. First, horizontal and vertical cross-sections are extracted from the data. Then, vertical cross sections are extracted to determine the height of walls and any windows and doors relative to the floor and ceiling. Finally, the walls are modelled by vertically extruding the horizontal cross-sections according to the constraints of the vertical cross-sections. This method is less computationally intensive than surface-fitting methods. However, it can lead to errors when components do not follow their idealized geometry (for example, if the walls are not truly vertical).

For a medium-sized building [173], manual merge modelling can take up to several months, which is often a bottleneck for completing completed BIM creation projects. Ideally, a system could be developed that takes a point cloud of the facility as input and produces a fully annotated as-built BIM of the facility as output. However, building such a system is a challenging problem for several reasons. Facilities can be complex environments, often with many unrelated objects, such as furniture and wall hangings, that obscure the view of the components to be modelled. These unrelated objects (called clutter) often do not need to be included in the BIM, and obscured surfaces can result in incomplete BIM representations unless assumptions are made about them (e.g., walls extend until they touch the floor). Even without clutter and occlusion, the geometry of a facility can be so complex that the resulting model is essentially a CAD model. The level of modelling accuracy and detail required for a particular use case remains an open question, but GSA provides accuracy tolerances ranging from ± 51 mm to ± 3 mm, with artefact sizes ranging from 152 mm to 13 mm [170]. Commonly used commercial and free softwares are Autodesk Netfabb [174], Revit [169], and mashlab [175].

2.2 Related work

The utilization of robots in BIM reconstruction is undoubtedly an innovative concept being explored in both the commercial and academic sectors. Some conferences and journal papers have shown BIM-related robotic research in the last two decades. Robot-based solutions for BIM reconstruction are not a simple overlay of various related BIM and robotic algorithms but a coupled combination of robot configuration, dynamic model, motion parameters, sensor characteristics and reconstruction methods. This section will briefly overview multiple perspectives on various relevant research studies and industrial work concerning robots and BIM reconstruction.

Boston Dynamics is a famous robotics company from the United States known for its walking robots. Based on the SPot quadruped robot dog, Boston Dynamics has introduced a robot 3D reconstruction and digital twin solution. Based on the built-in RGB-D camera and external multi-line rotating LiDaR of the Spot robot, the robot's localization and rough mapping are realized. The robot is equipped with a laser total station scanner, which scans and 3D reconstructs the robot by the positioning information given by the robot, as shown in the figure. The overall cost of this solution is high, and the stability of the walking robot is yet to be investigated. In addition, the massive amount of data from the laser total station is difficult to handle on the robot's built-in computing platform.

The QuicaBot robot from Singapore's Nanyang Technological University and Transforma company is a wheeled robot-based building scanning and reconstruction robot. It can move autonomously to scan a room, with a structured light camera and LiDaR, to detect building defects such as cracks and uneven surfaces in about half the time it takes to inspect them manually. The robot can upload the scanned 3D data to the cloud and notify operators of the project, which aims to perform quality finishing inspections according to the Construction Authority of Singapore (BCA) Quality Assessment System CONQUAS-9 standard. The advantage is the reliable mobility and high efficiency granted by the wheeled robot chassis. However, due to the shortcomings of its sensor solution, the maximum scanning height is only 3 meters, and the positioning capability is weak.

In academia, BIM reconstruction has attracted concern since the beginning of this century. The concept of learning 3D models of indoor environments with mobile robots was proposed by Yufeng LIU et al. in 2001 [176]. The output result is VRML format, a universal BIM format. Then, in 2004, Nuchter et al. proposed a wheeled robot for scanning data-based reconstruction [177]. Ding Lieyuan et al. proposed a BIM-based robotic assembly model that contains all the required information for planning was proposed [178]. Sungjin Kim et al. proposed a prototype built upon a robot operating system (ROS), focusing on reconstruction robot task plans for indoor wall painting based on BIM [179].

A summary of the BIM reconstruction robots containing state-of-art work from industry and academia can be found in table 2.4.

The brief review of automatic robot-based BIM reconstruction technologies reveals that, despite the challenges of robot use in BIM reconstruction, the ability of robots to automate real-time BIM updates will bring about a huge industrial change. Existing solutions in the industrial world focus on using robots as carriers with expensive laser total station scanners for BIM reconstruction, which brings not only a considerable increase in cost but also the laser total station is not designed for robots, and issues such as data storage, time synchronization and motion distortion, can bring unexpected errors and mistakes. Over time, the robotics of BIM reconstruction based entirely on robotic sensors has gradually been recognized by the academic community. In this thesis, an automated BIM reconstruction robot scheme called RoboBIM is proposed at UNNC to solve the two main challenges of low performance of robot-level sensors and automatic robot navigation in BIM reconstruction circumstances. A well-developed autonomous robot platform is designed on which the relevant algorithms are verified. The results demonstrate the novelty and reliability of the RoboBIM scheme.

Table 2.4: The brief summary of several relevant state-of-art works from industry and academia.

Project	Description	Localization	Scanning	Function	Automation Level
	Boston Dynamics Spot all-in-one method released in 2022 [180].	Built-in RGB-D camera and external LiDaR	Leica laser total station	Transporting scanners and positioning the system.	Motion: Semi-auto & Auto, Scanning: Auto, BIM: Manual.
	Leica BLK2FLY all-in-one drone scanning method [181].	GPS	Leica laser total station and mono-camera	Generating coloured point cloud outside the building by GPS and LiDaR scanner.	Motion: Semi-auto & Auto, Scanning: Auto, BIM: Manual.

Table 2.4. The brief summary of several relevant state-of-art works from industry and academia.

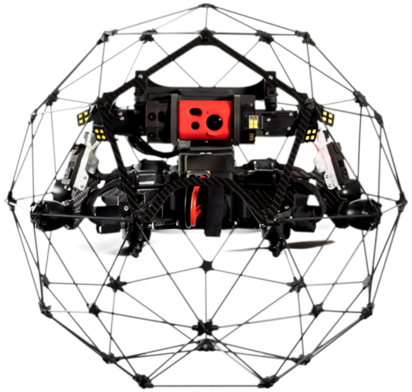
Project	Description	Localization	Scanning	Function	Automation Level
	Flyability Elios 2 indoor drone scanning method [182].	Indoor SLAM, remote control	LiDAR, thermal camera, and high-resolution industrial camera.	Intuitive indoor inspection including high-quality images under different light conditions and coloured point clouds.	Motion: Semi-auto & Manual, Scanning: Auto, BIM: Manual.
	Transforma Robotics QuicaBot indoor scanning robot [183].	Indoor LiDaR SLAM and remote control	LiDAR and structural light camera.	Detecting wall evenness, measuring floor slope and wall alignment and scanning and mapping rooms autonomously or semi-autonomously.	Motion: Semi-auto & Manual, Scanning: Semi-auto, BIM: Manual.

Table 2.4. The brief summary of several relevant state-of-art works from industry and academia.

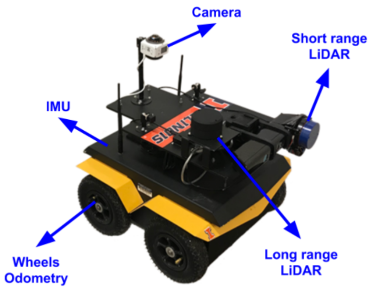
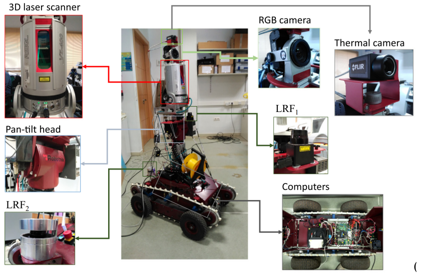
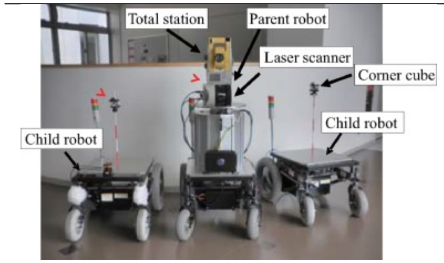
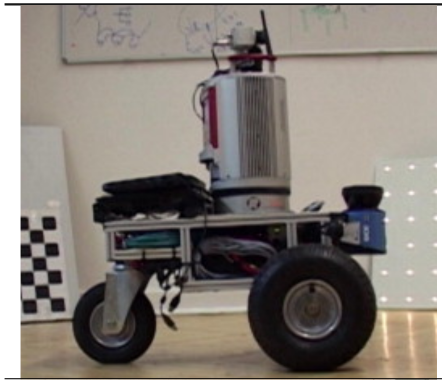
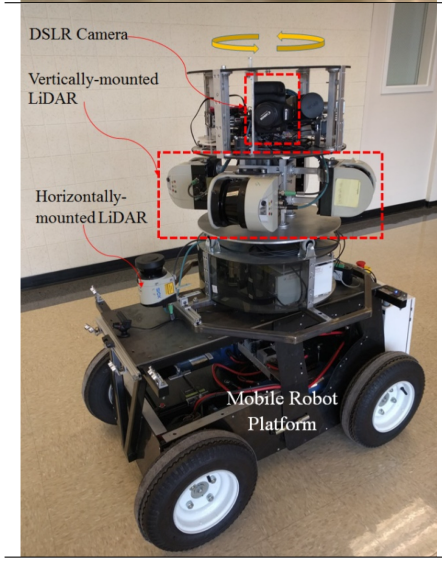
Project	Description	Localization	Scanning	Function	Automation Level
 Camera Short range LiDAR IMU Wheels Odometry Long range LiDAR	A ground robot-based scanning system proposed by Amir Ibrahim et al., using two 2d LIDARs mounted orthogonally to achieve BIM reconstruction in 2019. [184].	LIDAR SLAM	LIDAR	BIM-driven data collection planning, automatic navigation and dynamic obstacle avoidance.	Motion: Auto, Scanning: Auto, BIM: Semi-Auto & Manual.
 3D laser scanner RGB camera Thermal camera LRF₁ Pan-tilt head Computers LRF₂	Antonio Adan et al. proposed a multi-sensor and laser scanner scheme for a mobile robot to collect point cloud and BIM generation in 2020. [185].	LIDAR SLAM	Laser total station scanner	Automatic generation of detailed semantic models of indoor buildings with autonomous robots.	Motion: Auto, Scanning: Auto, BIM: Semi-Auto & Manual.
 Total station Parent robot Laser scanner Corner cube Child robot Child robot	Ryo Kurazume et al. proposed a multi robot 3d scanning collaboration scheme in 2017 [186].	LIDAR SLAM	Laser total station scanner	Multiple robots collaborate in 3D scanning work and an automatic planning technique for laser measurement.	Motion: Auto, Scanning: Auto, BIM: Only point cloud.

Table 2.4. The brief summary of several relevant state-of-art works from industry and academia.

Project	Description	Localization	Scanning	Function	Automation Level
	Dorit Borrmann et al. proposed a fully autonomous system for 3D and 3D thermal modelling of buildings, proving the demonstrated robot's feasibility for N-D BIM reconstruction. [187].	LiDAR SLAM	Laser total station scanner & thermal camera	Automatic data acquisition route planning;	Motion: Auto, Scanning: Auto, BIM: Semi-Auto & Manual.
 DSLR Camera Vertically-mounted LiDAR Horizontally-mounted LiDAR Mobile Robot Platform	Kim, Pileun et al. proposed a SLAM-driven automatic point cloud alignment framework based on a two-stage scanning method depending on the movement status of the robot. [188].	LiDAR SLAM	LiDAR array and DSLR camera	Automatic scanning state choosing, static scanning for better localization position while dynamic scanning during robot motion.	Motion: Auto, Scanning: Auto, BIM: Auto, Semi-auto and Manual.

Chapter 3

Structure and Control of LiDAR Rotation Gimbal

Compared with LiDAR total station scanners, robot-level sensors usually have apparent shortcomings in terms of FOV, the density of scanning points, and the uniformity of scanning, as shown in table 1.1 and table 2.1. Taking Leica RTC360 and Velodyne VLP32c as examples for comparison, VLP32c's number of scanning points per second is only one-third of that of RTC360, and the FOV is only one-twelfth of that of RTC360. Using LiDAR Scanning the building without any augmentation for BIM reconstruction can cause many issues, such as broken surfaces and missing scans, as shown in the figure 3.1.

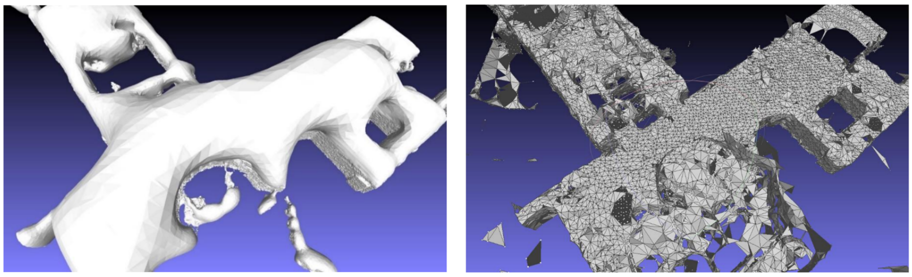


Figure 3.1: Point clouds without data augmentation are directly used for BIM reconstruction. Broken surfaces, distortions and noise can be clearly noticed. The data comes from the robot-level LiDAR Velodyne VLP32C. Fig. 3.1.A is the Ball-Pivoting surface method. Fig. 3.1.B is the Poisson reconstruction method.

In order to improve the performance of robot-level LiDAR in the field of BIM reconstruction, a LiDAR gimbal was proposed to improve the FOV and data density of LiDAR. Its mechanical structure, control and data processing methods are also presented together. Since its scanning parameters can be adjustable, an optimal scanning parameter identification method based on it is proposed and performs well.

This idea comes from the camera gimbal mounted on the drone, which has multiple degrees of freedom and allows the camera to rotate freely around the x, y, and z axes. With such a gimbal, detection without dead angles can be realized. However, LiDAR has mechanical structures inside that cause gyroscopic effects and vibrations compared to drone cameras. It requires the LiDAR gimbal more strong in mechanics and robust in control. The data of the LiDAR usually has the characteristic of a time pattern: the scanning data is sent from a specific angle and gradually according to the scanning time, instead of transmitting one frame of an image at a time like a camera, which will take a while to obtain an entire scanning data. While LiDAR moves, the sending data will be based on time-variable coordinate frames, distorting the data. In short, it is challenging to solve the design of the mechanical structure. The stability of the control system and the distortion of the data caused by the gimbal movement must be considered. This design is one of the deliverables of this project, which has obtained Chinese patent authorization, and a related paper is ongoing.

3.1 Design of gimbal

3.1.1 Mechanical design

Considering the structure, control, and motion distortion compensation of the gimbal, the main mechanical structure of the gimbal is designed as a cantilever structure. The LiDAR's base part (L-Arm) rotates around the Y-axis with a rotation angle of ±90 degrees. It is driven by a Brushless Direct Current (BLDC) motor, and the angle and angular velocity are closed-loop controlled, respectively. The specific structure is shown in the figure3.2.

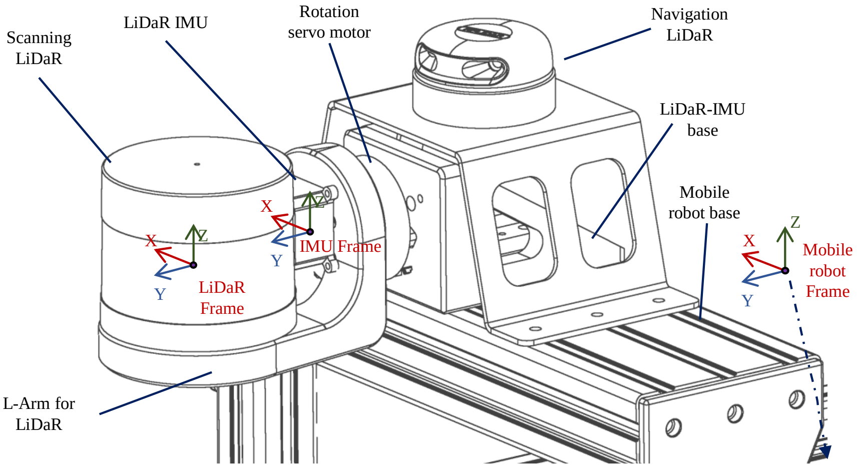


Figure 3.2: Schematic diagram of the structure of *LiDAR Rotation Gimbal.* The LiDAR Base is the main structure used to be fixed on the robot chassis. The motor drives the LiDAR to rotate about the Y-axis, controlled by an STM32MCU. The L-Arm is the base of the LiDAR, which is used to connect the rotating motor, LiDAR and IMU.

In this example, a Velodyne VLP-32c is used as a scanning LiDAR sensor. The Y axes of the IMU and LiDAR are pointing approximately collinearly. The other parts, including L-Arm and base, are made of ABS plastic with 3D printing. The scanning FOV has been improved significantly with the gimbal workspace, as shown in the figure3.3.

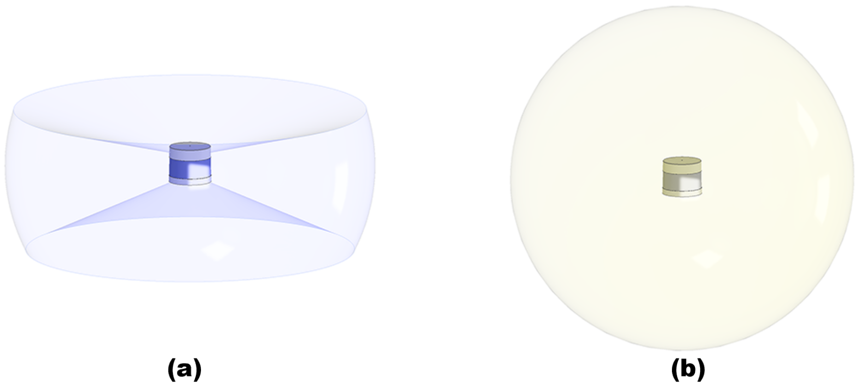


Figure 3.3: FOV of Velodyne VLP32C original (a) and gimbal extended (b). There is no dead angle in the scanning area anymore, so the robot-level sensor has the FOV and control capabilities of a laser total station scanner.

3.1.2 Control system design

The gimbal is a vital part of RoboBIM. The robot's controller should be able to control its movement directly, and the scanning results can be optimized by controlling the scanning parameters. The controller of the robot is implemented based on ROS. More details are in Chapter 5. This section mainly describes the control system of the gimbal. The gimbal controller is developed based on STM32 and receives messages from ROS through the USB-VCPinterface. The specific commands are serialized via JSON and contain the gimbal's angle and angular velocity information, which correspond to the angle control mode and angular velocity control mode of the gimbal, respectively. The control system's block diagram is shown in fig 3.4.

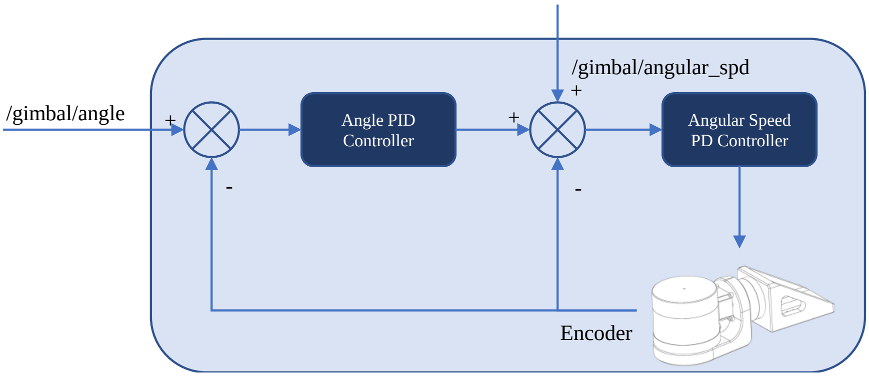


Figure 3.4: The control system of the head consists of a dual-loop PID, with an outer-loop position-loop PID control and an inner-loop velocity-loop PD control. The feedback information comes from the encoder of the BLDC motor.

3.2 Motion distortion compensation

LiDAR does not take single snapshots of the environment, which is the principle of global-shutter cameras. On the contrary, LiDAR collects a succession of 3D points generally grouped in scans [189]. The interval between each data frame causes a time difference in the scanning data. During this time difference, the motion of the LiDAR itself will bring distortion in the scanning data.

In SLAM research, the impact caused by the aberration of LiDAR is not negligible. There are three proposed solutions. The first solution is to increase the data rate of LiDAR. When the data frame rate of the LiDAR is high enough, the aberrations caused by the motion can be ignored. The second solution is to assume that the robot is moving at a uniform speed and then calculate the coordinate transformation of each scan data by linear interpolation. The third option is to combine the IMU with other external sensors for calibration.

Unlike traditional LiDAR robotics applications, in the gimbal, the LiDAR data distortion is not only from the robot's motion but also from the swing of the gimbal around the Y-axis. Figure 3.5 shows six consecutive frames of scan at 1s intervals, and the distortion due to the robot's motion and the swing of the gimbal can be detected.

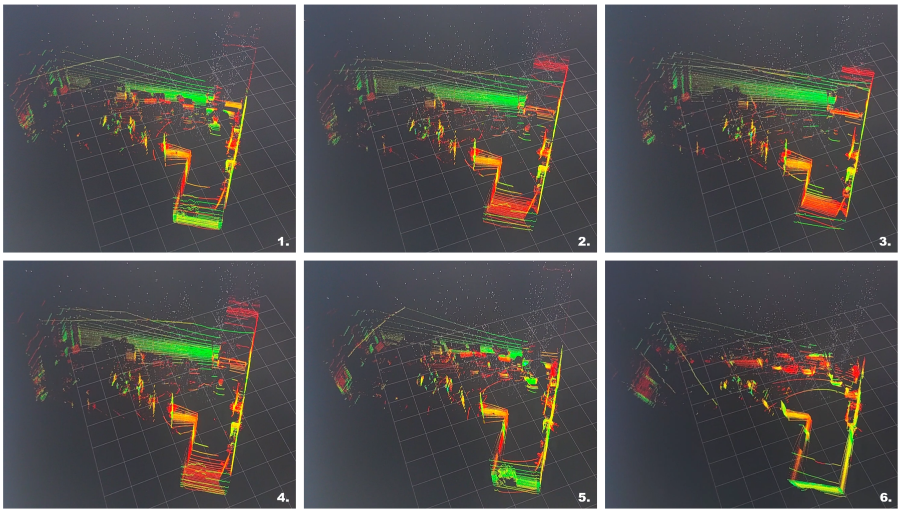
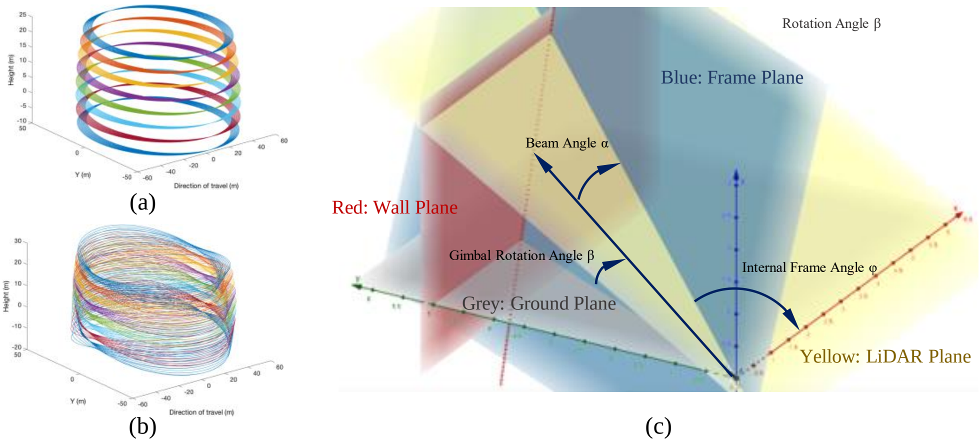


Figure 3.5: Six consecutive scans at 1s interval, the robot is doing uniform linear motion at 1m/s, the gimbal swing rate is 30° per second

Most of the LIDAR point cloud data transfer is based on the frame as the smallest unit. Frame holds the range data in an angular range. For the Velodyne VLP-32C LiDAR used in this project, the following fundamental values can be obtained according to its user manual. The interval between each two adjacent data frames is $46.08\mu s$. The angular difference between two adjacent frames is related to the rotation speed, assuming a rotation speed of $600RPM$, the $resolution = 600RPM * 1/60 * 360 * 46.08 * 10^{-6} =$

$0.165888(deg/frame)$. Correspondingly, if the speed is set 50% lower to $300RPM$, the resolution will be doubled to $0.082944(deg/frame)$. Within a data frame, the 32 lasers are not transmitted simultaneously but sequentially, with a time interval of $1.152\mu s$ and a delay of $9.216\mu s$ after 32 transmissions.

Due to the small time interval between laser frames, it can be approximately assumed that the coordinate changes between consecutive frames are small. Therefore, interpolation can be used to supplement the time interval of coordinate transformation updates. As shown in fig 3.6, the LiDAR's scanning pattern and beam are modelled.



(a)

(b)

(c)

Figure 3.6: (a) is the normal scanning LiDAR result pattern, (b) is the rotating LiDAR result pattern, (c) is the scan model of a single beam, where the robot structure is simplified to focus on the beam shot angle model of the LiDAR itself.

For each laser beam, the following conclusions exist. The vertical angle and internal rotation angle α and ϕ of the LIDAR are known, the gimbal rotation angle β is collected by an encoder, and the real linear and angular velocities of the vehicle noted as $v = [v_x, v_y, 0]^{\mathrm{T}}$ and ω are obtained by an IMU and an odometer model. Thus, for a laser beam measurement result $X(t)_{LiDAR} = [x_t, y_t, z_t]^{\mathrm{T}}$, it is expressed on the robot coordinate system as:

$$\boldsymbol{X}(t)_{Body} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} X(t)_{LiDAR} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} \tag{3.1}$$

Further, converted to the world coordinate system/odometer coordinate system, it is expressed as:

$$\boldsymbol{X}(t)_{World/Odom} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 & v_x \\ \sin\beta & \cos\beta & 0 & v_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X(t)_{Body} \\ 1 \end{bmatrix} \tag{3.2}$$

Finally, the calculation is performed on all the laser beams of a single rotation scan of the LiDAR, resulting in the undistorted raw point cloud $P_{World/Odom}$ of a single scan.

$$P_{World/Odom} = \sum_{\alpha} \sum_{\phi} X(t)_{World/Odom} \tag{3.3}$$

3.3 Scanning parameter evaluation

The rotation speed of the internal structure of the LiDAR and the robot's gimbal and movement speed can significantly affect the scanning results. In order to evaluate the quality of the scan, Ripley's K function is used, which is named after the statistician W. O. Ripley, a statistical method used to analyze and measure the spatial distribution pattern of a set of points in a two-dimensional plane. The K function calculates the expected number of points within a certain distance from each point in the data set and then averages the results across all points. This function can identify the data's clustering or dispersion patterns. The results of the K function analysis can be plotted on a graph, where the y-axis represents the K value and the x-axis represents the distance between points. The graph can be used to determine whether the points are randomly distributed or if the spatial pattern has clustering or regularity.

$$\widehat{K}(t)=\hat{\lambda}^{-1} \sum_i \sum_{j \neq i} w\left(l_i, l_j\right)^{-1} \frac{I\left(d_{ij}<t\right)}{N} \tag{3.4}$$

By considering the rotation model and scanning mode of each laser beam of the LiDAR, as well as the robot's movement speed and the gimbal's rotation speed, the impact of these factors on the quality of the scan was analyzed. The results are shown in the figure 3.7.

In conclusion, the robot's movement speed, the gimbal's rotation angle, and the distance between the robot and the wall can all be used to generate the optimal scanning parameters. The point cloud data obtained under the optimal scanning parameters is more uniform than that obtained under the fine-tuned scanning parameters. The LiDAR gimbal is feasible for practical use in BIM reconstruction.

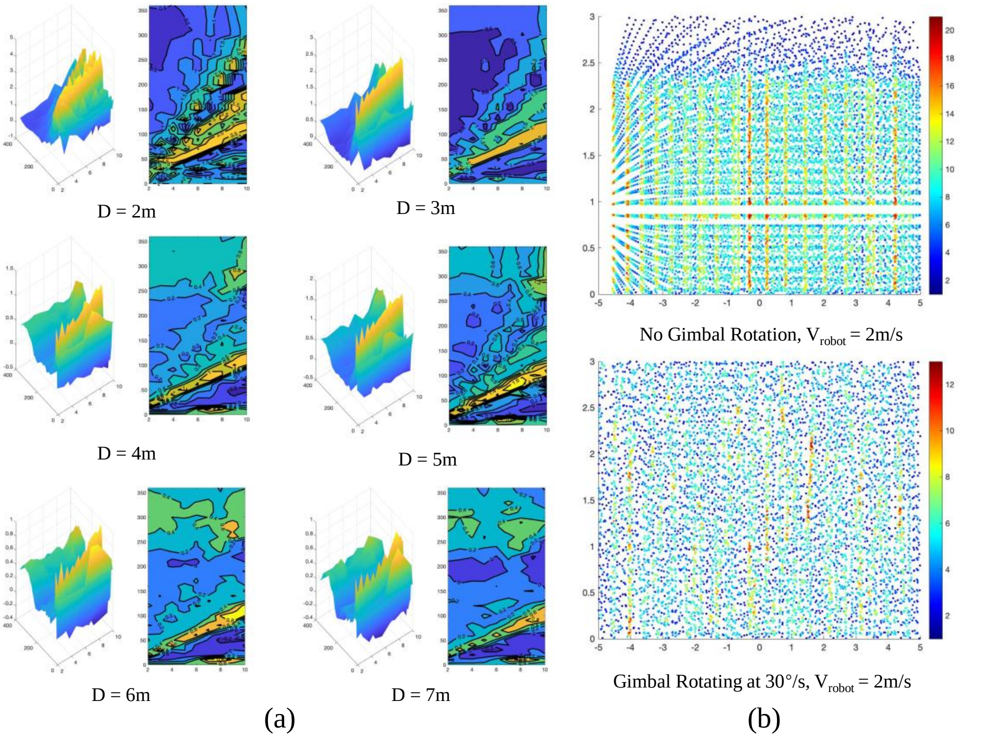


Figure 3.7: Calculation was performed on several scanning parameters of the gimbal. In the (a) group of figures, the horizontal axis represents the movement speed of the robot, and the vertical axis represents the angular velocity of the gimbal. In the (b) group of figures, the impact of having or not having a rotating gimbal on the uniformity of the LIDAR data at a specific distance is intuitively presented.

Chapter 4

BIM Reconstruction Oriented Robot Scanning Approach

Autonomous of a robot usually refers to the process of a robot performing work without commands. It requires the robot can deal with task scheduling and planning autonomously. Specifically for BIM reconstruction-oriented workflows, its autonomy usually means the robot can achieve two main tasks: autonomous motion and scanning. In the task of autonomous motion, there are three sub-tasks to be implemented: navigation point setting, path planning and trajectory generation. Navigation point setting reflects the autonomous nature of the robot, and automatic navigation point generation allows the robot to be free from command and remote control. Path planning refers to generating usable routes directly at the navigation points (avoiding obstacles). Trajectory generation is the conversion of paths into usable velocity and acceleration commands for the robot, allowing the robot to achieve paths in a manner that satisfies kinematic and dynamic constraints. The trajectory tracking capability is one of the critical metrics for evaluating robot motion controllers. Autonomous scanning is a new concept in robotic, laser total station scanners usually have exclusive scanning modes to enable automatic scanning. This chapter describes an automatic scanning method based on the dynamic adjustment of scanning parameters of the gimbal described in chapter 3. A novel framework called the three-step method is proposed for solving the problem.

4.1 Robot's three-step autonomous scanning workflow

The three-step workflow with optimal scanning parameters is one of the innovations of this project, as shown in fig4.1. To the best of the authors' knowledge, this is the first framework for automatic BIM reconstruction work that combines scanning optimal parameters, robot environment exploration and path planning. This framework solves the situation where robots have to trade off between scan quality, scan speed, safety and reliability when used for BIM reconstruction. It provides an effective solution to the problems of potential scan misses, low scan quality and low automation.

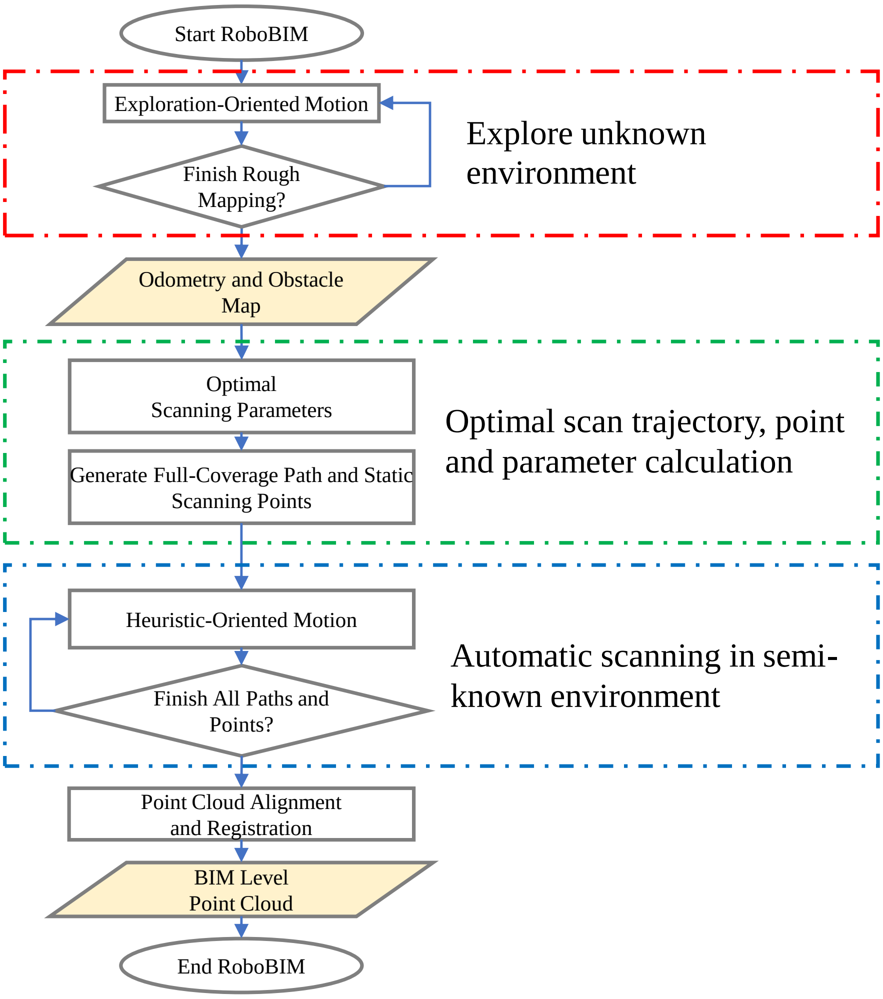

Start RoboBIM
Exploration-Oriented Motion
Finish Rough Mapping?
Explore unknown environment
Odometry and Obstacle Map
Optimal Scanning Parameters
Generate Full-Coverage Path and Static Scanning Points
Optimal scan trajectory, point and parameter calculation
Heuristic-Oriented Motion
Finish All Paths and Points?
Automatic scanning in semi-known environment
Point Cloud Alignment and Registration
BIM Level Point Cloud
End RoboBIM

Figure 4.1: The workflow of *RoboBIM* automatic scanning robot. The three-step scanning method with optimal scanning parameters is one of the novelties of this project, which is highly automated, accurate and easy to implement.

In the first stage, when the robot enters an entirely new environment (newly constructed building, renovated and redecorated existing building), The robot slowly advances through the unknown environment executing an exploration-oriented autonomous movement., and a high-frequency navigation LiDAR and RGB-D camera obtain a rough map through loosely coupled sensor fusion, which contains the boundaries of the working area, the reachable zone and the location of obstacles.

In the second stage, the robot generates scan waypoints based on the existing rough map and calculates the optimal scan parameters based on the scan distance and moving speed, as described in the chapter. Based on the generated waypoints and moving speed constraints, full coverage path and static scan points (scan waypoints where the robot is not moving while scanning) are generated.

In the third stage, the environment is no longer utterly unknown to the robot but semi-unknown. The robot can execute heuristic-oriented autonomous motions, sequentially passing scan waypoints at the optimal speed calculated in the second stage and stopping at static scan points for detailed scanning. Commercial and free software can then use the resulting point cloud and other information to produce 3D and N-D BIM.

In this process, the robot's awareness of the environment gradually increases. In the first stage, exploration-oriented autonomous motion ensures the robot can fully explore the unknown environment. The second stage of the path, waypoints and scanning parameter calculation can ensure that the scanning sensors work under the appropriate parameters as much as possible, which can bridge the performance gap between robot-level sensors and laser total station scanners and improve the automation of BIM reconstruction. The third stage of heuristic-oriented motion is based on partial knowledge of the environment, which can ensure the safety of the robot scanning process and improve the safety compared with the conventional method of building a map while scanning to avoid the robot getting lost, falling, and getting trapped in an unknown environment.

4.2 Path planning and trajectory generation

Both exploratory and heuristic trajectory planning algorithms require the robot to be able to first reach the order target point accurately from the starting point, avoid the collision, and update whether the target point is reachable in time. In this phase, the target point constraint usually also includes the final pose of the robot. When designing exploration algorithms, the first consideration is the robot's motion constraints, such as velocity and position constraints. Specifically, if the robot is omnidirectional and unconstrained in both velocity and position, such as the mecanum wheel and walking robot, then any exploration algorithm can be employed. Suppose the robot can move omnidirectionally in position but not in velocity, such as a differential drive chassis like a sweeping robot. In that case, its most basic exploration method can be zigzag. While the robot has non-holonomic constraints on velocity and non-strict positional constraints, such as an Ackermann chassis, the available exploration algorithms for the robot are significantly limited, such as the head-first problem in parking.

In the RoboBIM project, a chassis with an Ackermann-like configuration, which has the properties of an Ackermann structure, was adopted as the robot carrier. Its structural and kinematic relationships are shown in fig.4.2. The body's central reference point is located at the rear axle of the chassis. The track noted as L measures the distance between the front and rear axles, while the wheelbase noted as D measures the distance between the left and right wheels. The steering angle, labelled as Θ, is the angle between the front axle and the body's Y-axis direction. Lastly, the diameter of the wheels is noted as d.

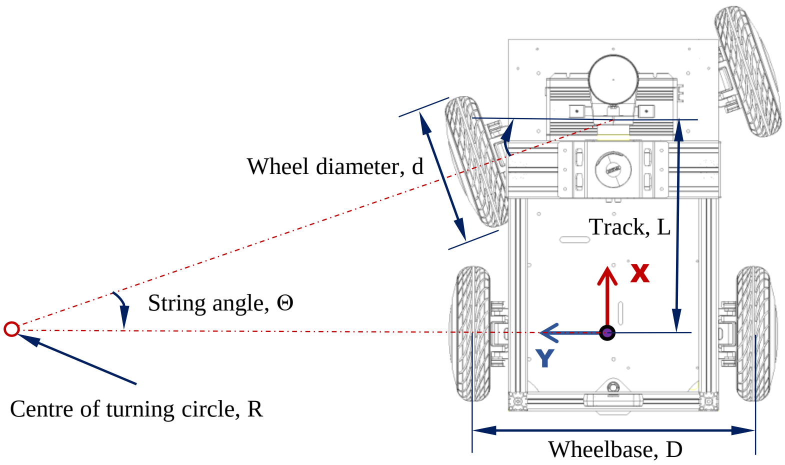


Figure 4.2: Structural and kinematic relationships of Robobim vehicle chassis.

Unlike the standard Ackermann chassis, it has an integrated front axle design, thus simplifying the Ackermann quadrilateral. The vehicle's forward and reverse kinematics can be obtained, where V_L and V_R are the linear speed of the left and right wheels at the rear end, ω and v are the angular and linear speed of the vehicle.

$$\begin{cases} \omega = \dfrac{V_R - V_L}{D} \\ v = \dfrac{V_R + V_L}{D} \\ \tan\Theta = L\dfrac{W}{\omega} \end{cases} \quad \begin{cases} V_R = v + \dfrac{\omega D}{2} \\ V_L = v - \dfrac{\omega D}{2} \\ \Theta = \arctan L\dfrac{W}{\omega} \end{cases} \tag{4.1}$$

Based on the kinetic model, the odometry model can be obtained, where X_r, Y_r and φ are the position of the middle point of the robot's rear axle and heading angle on global or odometry frame, v_r is the linear velocity projected on global or odometry frame.

$$\begin{bmatrix} \dot{\mathrm{X}}_{\mathrm{r}} \\ \dot{\mathrm{Y}}_{\mathrm{r}} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \frac{\tan\Theta}{\mathrm{L}} \end{bmatrix} \mathrm{v}_{\mathrm{r}} \tag{4.2}$$

For a robot chassis with kinematic constraints, its global and local trajectory planners need to satisfy the kinematic constraints. In this project, the Kinodynamic-RRT* is chosen as the global path planner, and the lattice planner is chosen as the local trajectory planner.

RRT algorithm, Rapid-exploration Random Tree, is an algorithm that extends the search by random sampling in a wholly known environment. The RRT algorithm is probabilistically complete, and as long as the planning time is long enough, a path does exist to get from the start point to the goal point. It will be found, but the paths planned by the RRT algorithm are usually not optimal, and the paths need to be smoother. The kinodynamic-RRT* algorithm selects the points that satisfy the kinematic constraints at each iteration in the workspace so that the generated paths satisfy the kinematic constraints better, as shown in figure 4.3. (a) and (b). Its specific algorithm flow is shown as the algorithm. 2.

Algorithm 2:

```
Input: Map, $X_{init}$, $X_{init}$
Output: A path P from $X_{init}$ to $X_{init}$
P.init();
for $i = 1$ to $n$ do do
    $X_{rand} \leftarrow KinematicSmaple(M)$;
    $X_{near} \leftarrow Near(X_{rand}, P)$;
    $X_{new} \leftarrow Update(X_{rand}, X_{near}, Step)$;
    if $CollisionFree(X_{new})$ then
        $X_{near} \leftarrow NearC(P, X_{new})$;;
        $X_{min} \leftarrow FindParentNode(X_{near}, X_{new})$;
        $P.addNodeEdge(X_{min}, X_{new})$;
        $P.reWrite()$;
    end
end
```

Lattice Planner is named for its discrete approach to the robot motion workspace, and the visualization is similar to that of lattice, as shown in the figure 4.3. (c). Its advantage is that it satisfies the robot motion constraint with low computational overhead, and the resulting trajectory is straightforward. Hence, it is also widely used for autonomous driving. The disadvantage is that the motion constraint at the trajectory connection is difficult to guarantee, so it usually requires more constraint parameters than the actual degrees of freedom of the robot, resulting in more parameters and cumbersome debugging.

4.3 Exploration and Scanning path-planning

Exploring unknown environments is one of the challenges of autonomous robots and has been widely researched. Various approaches have been used to address this challenge, from the most basic zigzag method to complex and diverse learning-based methods. In this project, due to the robot's kinematic constraints, the zigzag method would obviously lead to repeated

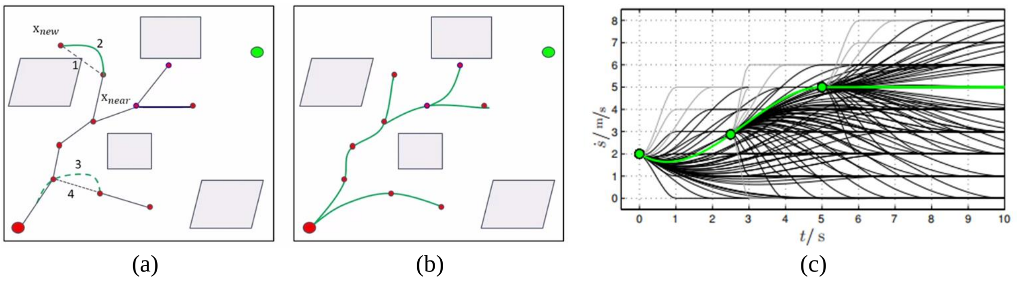


Figure 4.3: The figures of global and local planning algorithms. (a) is the normal RRT algorithm while (b) is the Kinodynamic-RRT*, it can be found that Kinodynamic-RRT* can search new nodes following kinetic constrain rather than straight lines [4]. (c) is the result of the discretization of the state space has a shape similar to that of a lattice, from which the lattice planning algorithm gets its name [5].

reversing. A method based on right-hand priority is proposed for the exploration path planning of this project. Specifically, the robot keeps the right-hand edge along the wall to move forward. This method ensures that the robot moves along the edge of the environment. If the robot encounters a dead end or an unreachable area, it will backtrack and return to the edge of the environment after bypassing the obstacle. The termination condition of the exploration is determined by detecting mixed loop closure information, which means that the robot undergoes both loop closure detection and is in an approximate pose state. The distance between the robot and the right-hand wall is variable. It can be updated promptly after each planning failure, ensuring the robot can always find a feasible closed trajectory. This method is particularly suitable for building reconstruction applications, where the relationship between the walls inside the building is usually straightforward, so the robot can easily find a path and complete the mapping of the interior of the building. However, the right-hand rule may only be applicable to some environments, especially when there are no edges in the environment or multiple exits. The flowchart of the algorithm is shown in the algorithm.3.

Several common cases are discussed and studied separately, including corners, inaccessible and accessible slits, and many other cases, as shown in the figure.

After completing a rough obstacle map, it becomes easy to plan the trajectory for scanning the interior of the building. The scanning path follows the same pattern of right-hand priority, except that the distance between the robot and the right-hand wall is determined by the optimal scanning parameters calculated in Chapter 3. The robot's motion speed is also provided to the trajectory planner as a constraint. In some cases, the optimal scanning parameters cannot be satisfied, and the robot chooses one feasible path randomly to finish the scanning task. The flowchart of the algorithm is shown in the algorithm.4.

Algorithm 3:

```
Input: Pose, Scan, ClosureLoop
Output: $X_{goal}$
Position.init();
Tolerance.reset();
while $ClosureLoop = False$ do
    Pose.update();
    Scan.update();
    $X_{next} \leftarrow Lookup(Scan, Pose, Tolerance)$;
    if $CollisionFree(X_{next})$ then
        Tolerance.reset();
        $X_{goal} \leftarrow X_{new}$;
    end
    else
        Tolerance.update();
    end
end
```

Algorithm 4:

```
Input: $Map, D_{gimbal}, Pose, Vel_{gimbal}, X_{goal}, ClosureLoop$
Output: $X_{scan}$
Position.init();
while $ClosureLoop = False$ do
    Pose.update();
    $D_{gimbal}$.update();
    $X_{next} \leftarrow Update(D_{gimbal}, X_{goal})$;
    if $CollisionFree(X_{next})$ then
        $X_{scan} \leftarrow Lookup(Map, X_{goal}, Pose, D_{gimbal})$;
        $GlobalPlanner \leftarrow (X_{scan}, Vel_{gimbal})$;
    end
    else
        $X_{scan} \leftarrow X_{goal}$;
        $Vel_{gimbal}.reset()$;
    end
    $X_{goal}.next()$;
end
```

4.4 Experimental results

The actual test and experiment of the above algorithm were carried out in IMAET 2F at UNNC. It should be noted that during the test period, the scanning parameters calculated by the LiDAR gimbal were not the optimal parameters and only merely met the coverage scanning requirements.

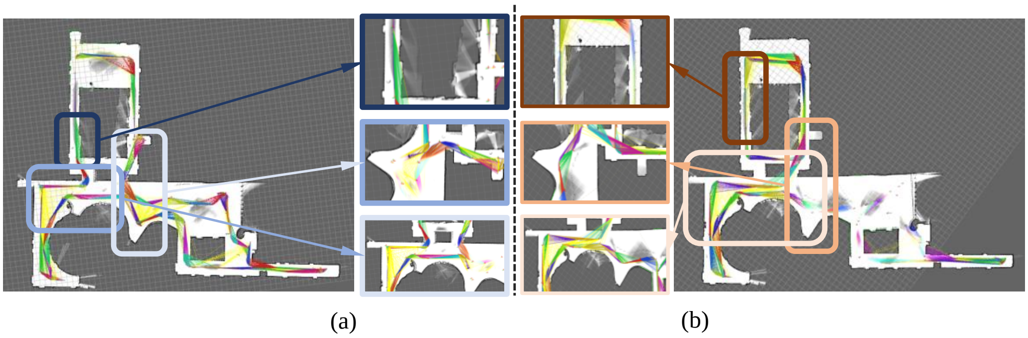


Figure 4.4: The exploration and scanning approaches test. (a) is the exploration planning method, while (b) is the scanning one. It can be clearly observed that the scanning path planning algorithm ignores many dead ends and narrow passages that cannot be entered or optimally scanned. On the other hand, the exploration path planning algorithm scans all corners without omission, leading to the presence of turning points in the robot's path.

Chapter 5

All-in-one BIM Reconstruction Mobile Robot-based Solution

RoboBIM project is named after Robots and Building Information Modelling; specifically, it is based on AMR technology to realize BIM generation automation. Through a comprehensive review of related works in the industry and academic research and an overview of several key technology points in the chapters mentioned above, an all-in-one BIM generation solution is proposed at UNNC. The *RoboBIM* project consists of an automated scanning robot and a remote database. The automated scanning robot scans the building, acquires dense point clouds and other information, and loads the information to the remote database; the remote database converts the collected information into BIM using commercial or free software and adds the information needed for N-D BIM. In addition, a *RoboBIM* simulator based on the physical simulation engine gazebo was developed for researchers to study relevant robotics algorithms and verify experimental results. The structure of the *RoboBIM* project is shown in figure 5.1.

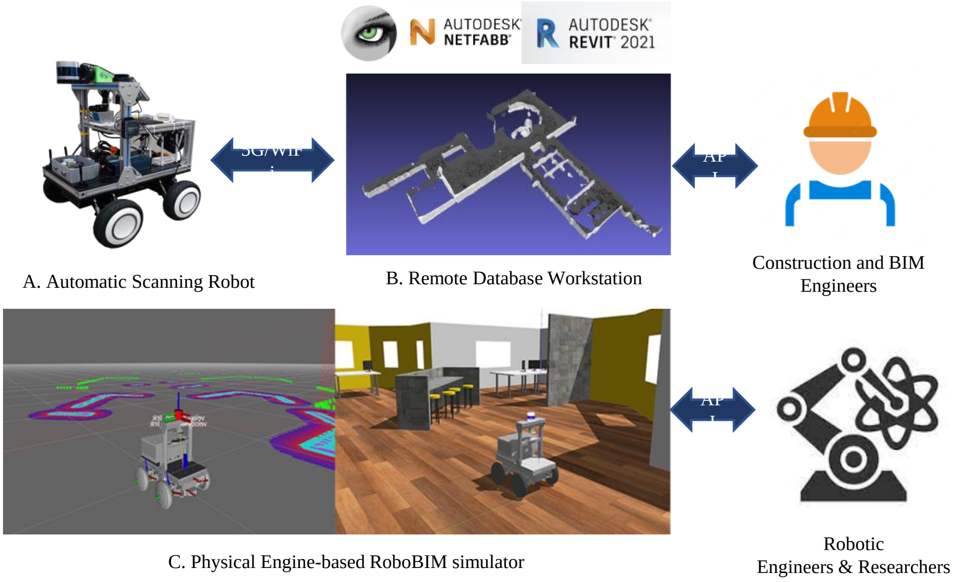

AUTODESK NETFABB
AUTODESK REVIT 2021
5G/WIF
i
AP
I
A. Automatic Scanning Robot
B. Remote Database Workstation
Construction and BIM Engineers
AP
I
C. Physical Engine-based RoboBIM simulator
Robotic Engineers & Researchers

Figure 5.1: The overview of *RoboBIM* project, 5.1.A is the automatic scanning robot, 5.1.B is the remote database and workstation with a user interface to construction and BIM engineers, and commercial and free software are installed to deal with the point cloud. The reconstruction result of IMAET 2F at UNNC is also shown in 5.1.B. 5.1.C is the physical engine-based *RoboBIM* simulator to help robotic researchers and engineers develop related algorithms.

5.1 Hardware Design of *RoboBIM* Robot

5.1.1 Mechanical Design

Considering various factors such as movement ability, stability, load capacity, and off-road capability, the robot carrier of this project chooses Segway rmp401 chassis, which adopts a similar structure to Ackermann's layout and BLDC motors from Bosch directly drive four wheels. It has barrier-crossing solid performance and an open interface, which is convenient for secondary development.

A multi-layered payload bay was designed for the *RoboBIM* project, with its overall frame built from aluminium profiles and divided into three layers. The bottom layer mainly installs hardware facilities for external communication and internal power supply; the middle layer installs on-board computing devices, high-precision on-board IMU sensors and mechanical narrow FoV high-density LiDAR; the top layer installs low-precision LiDAR for navigation, sensor gimbal and display screen for human-robot interaction. Figure 5.2 shows the overall layout.

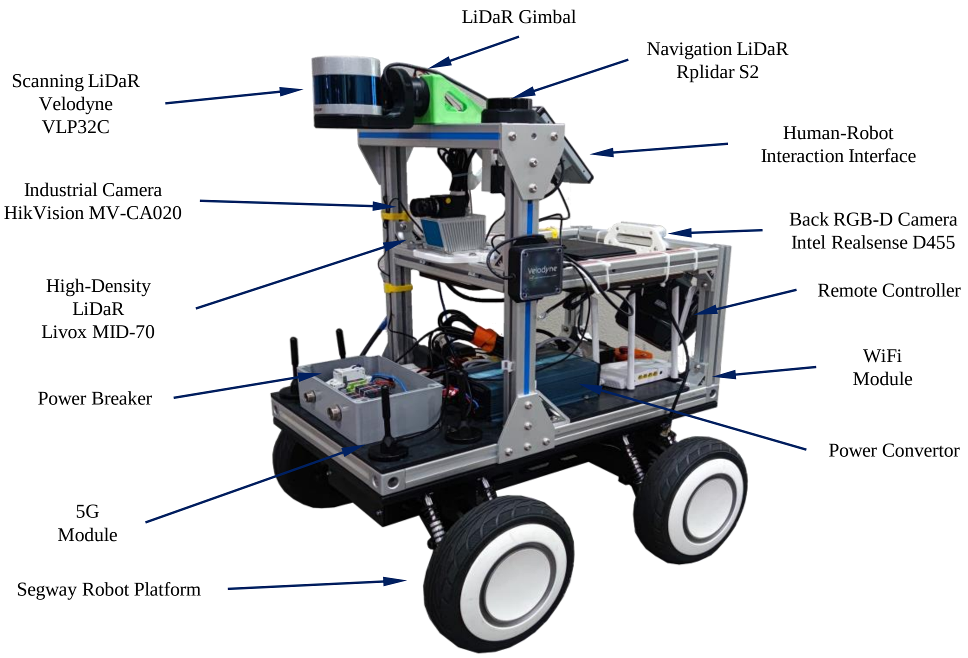


Figure 5.2: The overview of the automatic scanning robot of ***RoboBIM*** project, the payload bay was designed based on the chassis of the Segway RMP401 robot, with a three-layer layout that houses multiple sensors, computing units, power distribution units, and communication units.

5.1.2 Sensors and hardware

After considering the task requirements and sensor performance, the sensors and hardware attributes are shown in the table 5.1.

Table 5.1: Sensor and hardware for *RoboBIM* automatic scanning robot.

Type	Application	Note
Intel NUC11 I7	Computation Unit	CPU I7-1165G7 GPU Nvidia RTX2060
Go bigger Display Touch	Human Robot Interaction	1920x1080 with Touch
Velodyne VLP32C	Scanning LiDAR	
RpLiDAR S2	Navigation LiDAR	
HikVision MV-CA020	Point Cloud Color Information	
Luvox Mid-70	High density Point Cloud	
Intel Realsense D455	Back RGB-D Camera	
Xsens MTI-630	Robot Chassis IMU	
Quectel QM-500	5G Communication Module	

5.1.3 Circuit and Communication Design

In order to integrate as many functions as possible and enhance motion, the robot should be designed to be cable-free driven and can rely on its electrical power to perform tasks. Because the body is equipped with multiple sensors, computing units and communication devices, the structure is complex, and the power distribution requirements are different, so a complex power distribution system is required. The chassis's battery power passes through circuit breakers and fuses, then enters the inverter, which is converted to 220V voltage. Since the chassis's voltage is 48V, which comes directly from the power battery, using the inverter can stabilize the downstream voltage, simplify the system structure, ensure the response of the leakage protector and improve safety. 220V voltage is further passed through the AC leakage protector and subsequently through multiple AC-DC converters, i.e., converted to 5V, 12V, 19V, 24V and 48V, respectively, for the controller, status indicator, scanning radar, computing units and communication devices, the overview of power system is shown in fig 5.3.

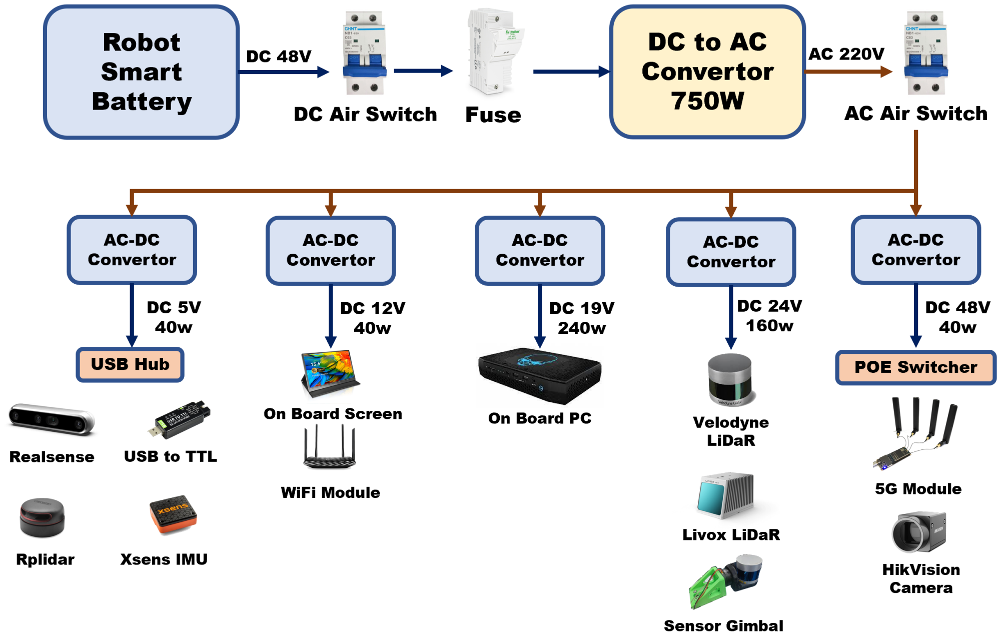


Figure 5.3: The power circuit of the automatic scanning robot of *Robo-BIM* project with inverter-based DC-to-AC power supply and distribution schemes are chosen. Compared to the DC-to-DC scheme, this scheme provides step-down failure protection and reduces the number of distribution components, ensuring the sensitivity of leakage protection.

The communication part mainly considers internal communication and external communication. Internal communication is mainly the two-way communication between the computing unit and the robot chassis, each sensor and the gimbal; external communication is the high-speed communication between the computing unit and the remote database and workstation. The internal communication is via Gigabit Ethernet with LiDaR and industrial cameras and via USB with IMU, LiDaR, gimbal and robot chassis. External communication is divided into two modes, 5G and WiFi. For on-site robot debugging and development, WiFi is mainly used, and for remote databases, remote transmission is performed via 5G. It is worth noting that the communication part is implemented using a web video server based on FTP technology to improve bandwidth utilization. The main process is to package the data and forward it with the help of the video API interface. The remote server unpacks the data according to a predefined protocol. The overview of communication of the RoboBIM robot is shown in fig 5.4.

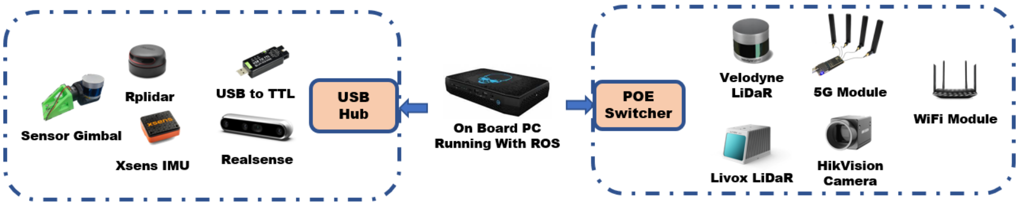


Figure 5.4: The communication system of the automatic scanning robot of *RoboBIM* project, the industrial camera and the 5G module are powered by Power on Ethernet (POE) technology.

5.2 Multi-sensor co-calibration

Robots are fitted with a variety of sensors whose coordinate translation relationships to each other are important. For single sensor, the calibration is realized in different way, the internal reference calibration of the camera is performed using a checkerboard calibration board, while the internal reference calibration of the LiDaR follows the instruction manual.

Multi-sensor calibration works by determining the relative position and attitude relationships between multiple sensors so that they can sense and measure the same scene in a consistent manner. This process is typically accomplished using a known scene and a known reference point. Before calibration, the time synchronization between the network devices should be achieved. On *RoboBIM* robot, there are two lines of internal communication as shown in 5.4, one ethernet line and one USB line. For ethernet port sensors such as Velodyne, Livox and industrial camera, pps For USB port devices including chassis, RpLiDAR and IMU, the Linux system of the computing unit is installed with a real-time patch [191] to achieve relative time synchronization between the network port and these devices.

The joint calibration of the industrial camera and the Livox radar was performed using the LiDAR_camera_calib toolkit [190], and the Livox and the Velodyne LiDAR were indirectly aligned using a measurement approach.

Indirect alignment between the back Realsense RGB-D camera and the navigation LiDaR is also performed by measurement. This alignment method does not jeopardize the accuracy and stability of the system due to the loosely coupled sensor fusion of the localization method.

The calibration of gimbal and navigation LiDaR to the robot base is realized in another way. The mounting errors between the robot chassis and the LiDaR gimbal and navigation LiDaR, respectively, can be represented by two sets of coordinate transformation relations that incorporate displacements and rotations. Three independent sets of odometry information are computed via the navigation LiDaR mounted on the robot chassis, the built-in encoders inside the robot drive wheels and the LiDaR gimbal data, respectively. Then the coordinate transformation relationship between the three sets of odometers is obtained by the L-M method to realize the joint calibration. The detailed sensors coordinate conversion relationships are shown in the appendix 1.

5.3 Framework of *RoboBIM* software system

The software framework for RoboBIM's automated scanning machine was developed based on ROS. The open-source Robot Operating System is widely used in modern robotics development, first proposed by Stanford University [192]. The ROS is not an operating system in the strict sense, such as Windows or Linux, but a software framework that enables and standardizes the communication between software packages and robot hardware components. ROS follows the computational graph principle, i.e., each executable program is a node, and nodes communicate with each other in three ways: *Topic* communication, which requires no answer, *Service* communication, which requires an answer, and *Action* communication, which requires action as feedback. ROS package refers to a container of several nodes with functions organized via Cmake [193]. ROS message refers to the protocol in the communication. Note that there is a base node called ROS Master, which is responsible for maintaining the ROS computation graph, log and parameter server.

The overview of *RoboBIM* software framework is shown in fig 5.5.

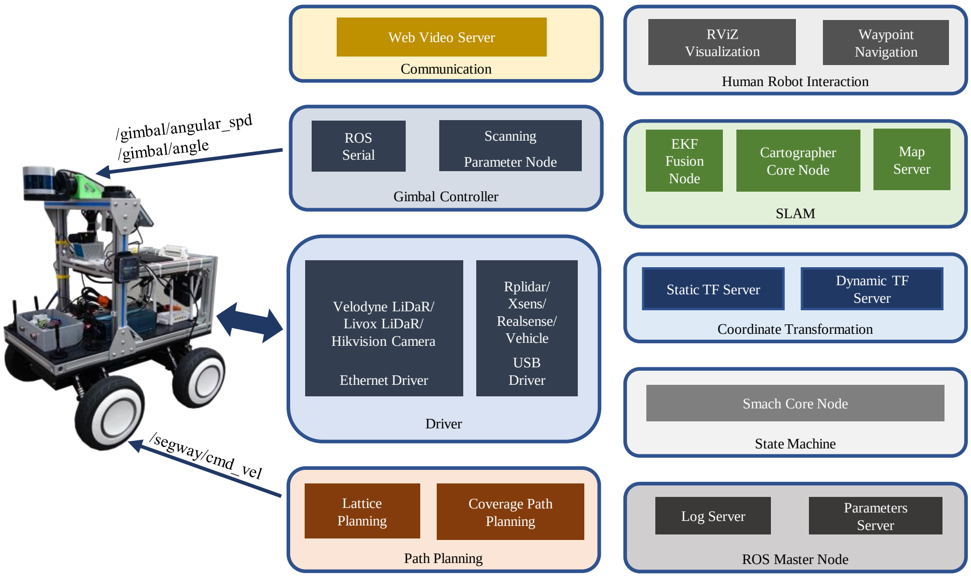


Figure 5.5: The overview of *RoboBIM* project software framework. Every block stands for a group of related ROS nodes. The framework is based on ROS and consists of nine parts, with a universal robot control interface and an interface for human-robot interaction.

There are nine parts to this framework. The ROS master node contains the log and parameter server, which maintains the software framework logs and dynamically updates the parameters. The parameter server interaction interface is shown in the Rviz visualization section of the human-robot interaction part.

All sensor drivers and control drivers for the robot chassis are contained in the driver package. Sensor data are transmitted through the structure shown in figure 5.4.

The coordinate information of the sensor data is unified and managed by the Coordinate Transformation node (TF). The coordinate transformation between the sensors fixed on the robot and the chassis is handled by the static TF service. The dynamic TF service handles the one between the gimbal and the robot chassis and between the robot and the global coordinate system.

The SLAM part first uses the Pose_EKF open-source toolkit for data fusion of visual odometry and laser odometry to obtain high-precision odometry information. Then it obtains a rough map by Cartographer open-source slam package. The map server is responsible for maintaining the global map with a low-frequency update (0.5hz) and the local map information with a high-frequency update (2hz).

The state machine package part is implemented based on the SMACH state machine, a robust and scalable hierarchical state machine library based on python that supports multi-conditional transfer. The state list and state transfer relationship are shown in the table.

Two sub-packages are included in the path-planning part. The CPP package generates scanning waypoints and publishes them to the trajectory generation package by topic, described in the previous chapter. The trajectory generation package is implemented based on lattice planning, a method for finding available trajectories in control space based on Sobel's operator and kinematic constraints. Its output trajectory is a smooth and safe local without collisions satisfying the vehicle's kinematic and velocity constraints. The specific implementation is based on the sbpl_lattice_planning package with customized parameter [194].

The gimbal controller package contains the node for control parameters and functions of the gimbal, the node for serialization and deserialization of ROS serial data and the node for relative coordinate system release of the gimbal. Details are described in the previous chapter.

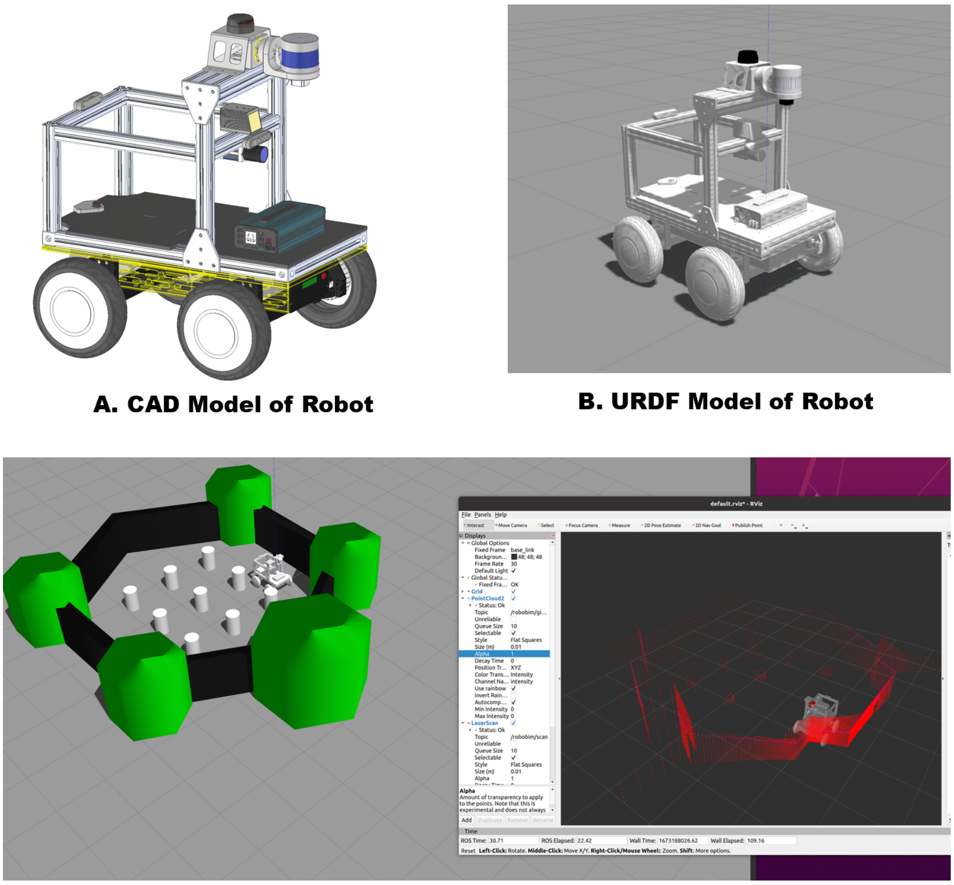
The Human-robot interaction package is the sum of the RViZ node for data visualization, the parameter modification node, the pointing navigation function node, and the safety control node. This part mainly provides the operator with a visual data monitoring and command interface.

The communication package is used to achieve high bandwidth utilization communication, as mentioned above.

5.4 Physical engine-based *RoboBIM* simulator

Using computer-based simulations to develop new robot designs and algorithms is good practice before building and executing code on physical robotic systems. To rapidly develop and verify relevant robotics algorithms and reduce time and testing costs, a physics-based simulator, Robobim Simulator, was developed. With real-time factors, physics engine support and simulation accuracy as the primary reference criteria, Gazebo was chosen for the development of the RoboBIM simulator after a detailed comparison of four common robot simulators, CoppeliaSim (formerly known as V-REP), Gazebo, MORSE and Webots [195, 196].

After assigning mass to the CAD model of the robot, export it as a Universal Robot Description Format (URDF) file, and program the driver node for the robot simulation file, then add simulation programs for various sensors to it. By replacing the part of the RoboBIM program framework that interacts with the real robot with ROS_Gazebo_bridge, the simulator can be used as the main development tool to verify the algorithm, as shown in fig 5.6.



```
Global Options
  Fixed Frame       base_link
  Backgroun...      48; 48; 48
  Frame Rate        30
  Default Light     ✓
Global Statu...
  Fixed Fra...      OK
Grid                ✓
PointCloud2         ✓
  Status: Ok
  Topic             /robobim/gi...
  Unreliable        □
  Queue Size        10
  Selectable        ✓
  Style             Flat Squares
  Size (m)          0.01
  Alpha             1
  Decay Time        0
  Position Tr...    XYZ
  Color Trans...    Intensity
  Channel Na...     intensity
  Use rainbow       ✓
  Invert Rain...    □
  Autocomp...       ✓
  Min Intensity     0
  Max Intensity     0
LaserScan           ✓
  Status: Ok
  Topic             /robobim/scan
  Unreliable        □
  Queue Size        10
  Selectable        ✓
  Style             Flat Squares
  Size (m)          0.01
  Alpha             1

Alpha
Amount of transparency to apply
to the points. Note that this is
experimental and does not always

ROS Time: 30.71   ROS Elapsed: 22.42   Wall Time: 1673188026.62   Wall Elapsed: 109.16
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
```

A. CAD Model of Robot

B. URDF Model of Robot

C. Sensor and Data Visualization in Simulator

Figure 5.6: The overview of *RoboBIM* project simulator. 5.6.A is a CAD model of a robot, which can be converted into a URDF file, as shown in fig. 5.6.B with adding inertia matrix. fig. 5.6.C shows the sensor and data visualization in the simulator.

5.5 Experiment result

Tests for the RoboBIM project were conducted in the IAMET building at UNNC, testing the robot's autonomous motion capabilities, scanning capabilities, workflow and the feasibility of the LiDAR gimbal, with the results shown in the figure 5.7, figure 5.8 figure 5.9 and table 5.2.

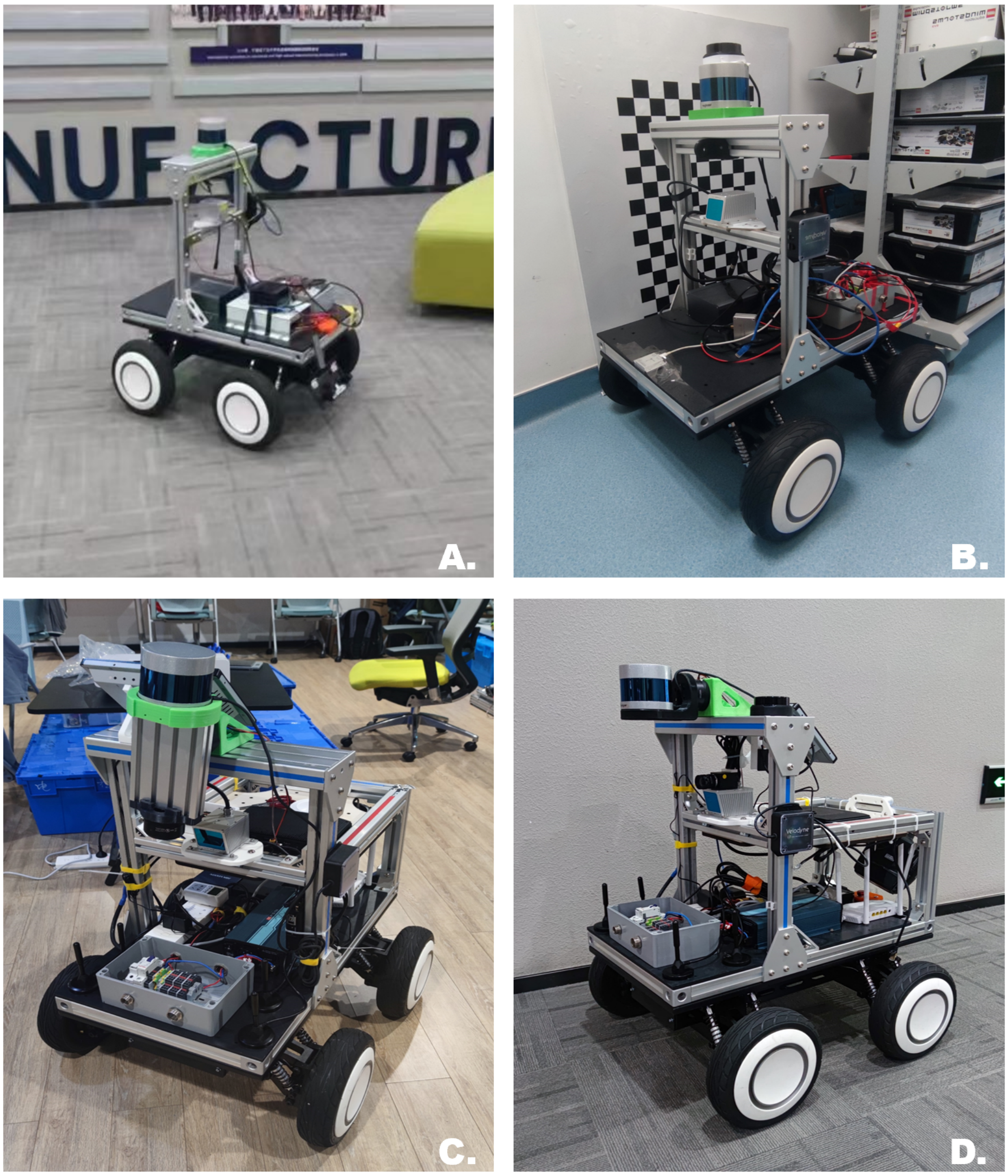


Figure 5.7: A showed test of the SLAM function of the robot in November 2021. B shows a test of the robot with the SLAM function separated from the scanning function in February 2022. C shows the first test of the robot with the LiDAR gimbal installed in June 2022. D showed a joint test of the robot with the eccentric distance-free gimbal in September 2022.

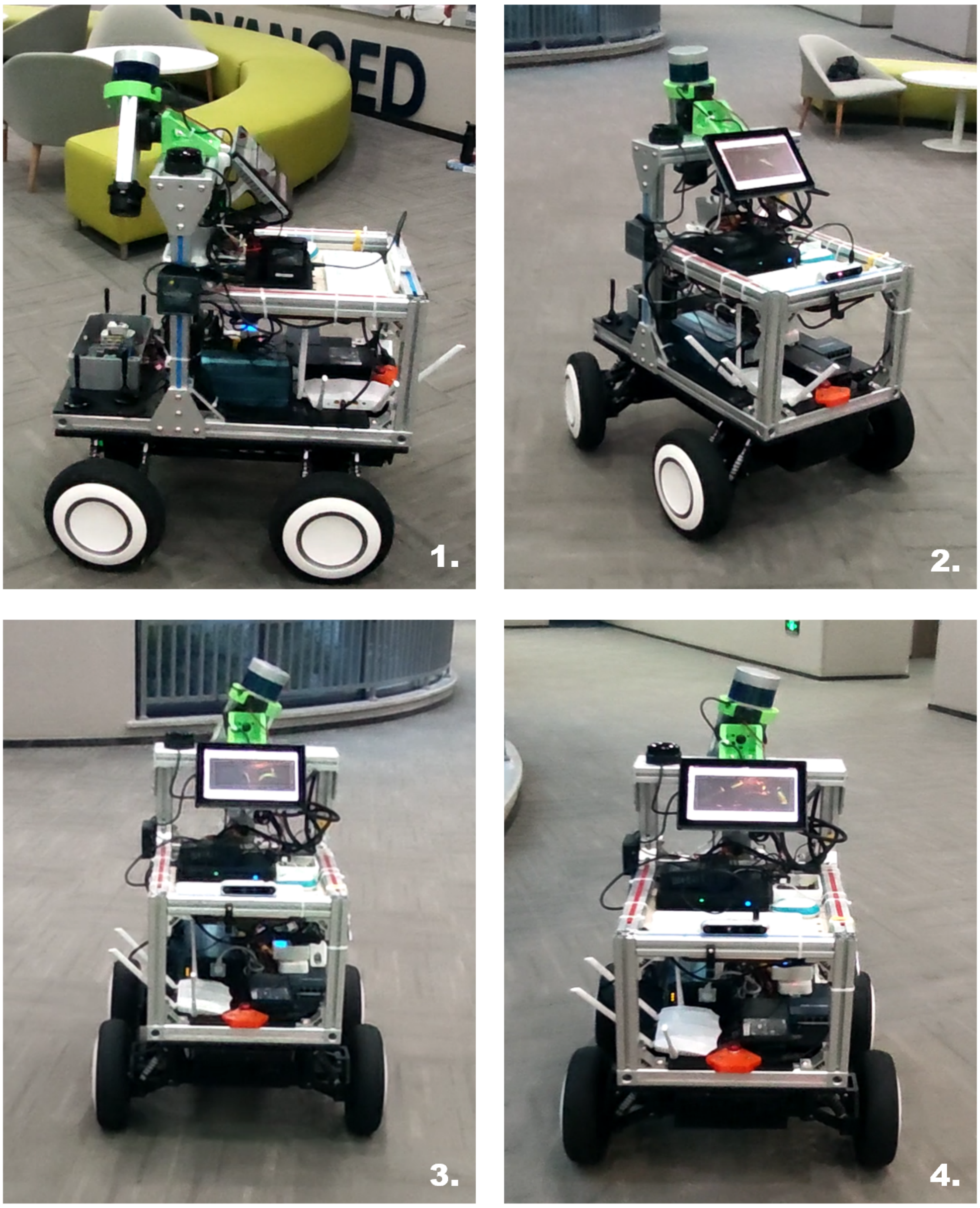


Figure 5.8: *RoboBIM* automatic scanning robot undergoes fully functional testing at IAMET 2f at UNNC. Fig.5.8.1-4 demonstrate the robot's continuous autonomous motion.

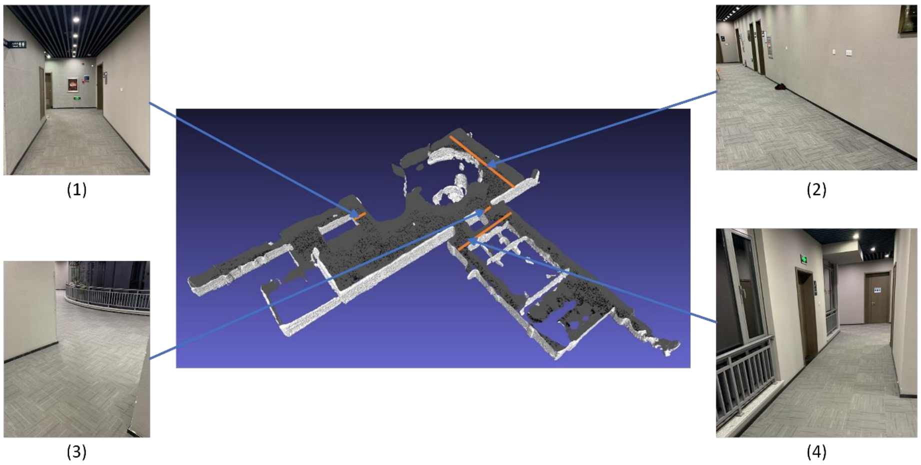


Figure 5.9: Reconstruction results of IAMET 2F indoor from *RoboBIM* robot, 1-4 are four selected measure points.

Table 5.2: The error estimation was performed for four selected points in the reconstruction results obtained by the *RoboBIM* robot at IAMET 2F, UNNC.

Selected Point	Truth Data (cm)	Reconstruction Data (cm)	Error (cm)	Percentage(%)
Figure.5.9.(1)	243.5	242.5	-0.95	0.392
Figure.5.9.(2)	1951.5	1947.8	-3.7	0.190
Figure.5.9.(3)	229.4	228.7	-0.7	0.305
Figure.5.9.(4)	1270.3	1268.4	-1.9	0.149

Chapter 6

Conclusion

In the face of many challenges in building intelligence, longevity, and low new construction rates, BIM has played an irreplaceable role. This thesis presents a BIM reconstruction method based on mobile robot technology and comprehensively reviews the method required for BIM reconstruction using robot-level sensors. The main conclusions are as follows.

- The LiDAR rotating gimbal proposed in Chapter 3 effectively extends the FOV of the LiDAR so that the robot-level LiDAR meets the needs for use in scanning BIM reconstruction, i.e., the LOD300 accuracy requirement. This makes it possible to realize BIM reconstruction at low cost.
- The rotational speed and the measured distance of the rotating gimbal can be composed into a set of parameters to optimize the scanning uniformity of the LiDAR in indoor building scanning operations and to improve the smoothness of the scanning results, which in turn affects the reconstruction results. The calculation of this set of parameters is presented in Chapter 3.
- The three-step navigation algorithm proposed in Chapter 4 can effectively improve the automatic performance of the BIM reconstruction robot, which is tested and validated in some scenarios. The results prove that the navigation strategy of separating exploration and building effectively improves the scanning efficiency and reduces the possibility of scanning failure, and the scanning process with optimized scanning parameters is more reasonable.
- Testbed and hardware built on a mobile robot, software configuration in Chapter 5. It fully demonstrates the feasibility of the mobile robot for use as an indoor BIM reconstruction and proves the reliability and superiority of the above algorithms.

This project provides a low-cost, high-autonomy BIM reconstruction method and robot platform, which can effectively improve the automation of BIM reconstruction, increase the speed and efficiency of building BIM reconstruction, and make BIM reconstruction an important part of building renovation and maintenance.

6.1 Future work

This project has some limitations that can be further improved in future work. These are described below:

- Create a dataset for indoor BIM reconstruction based on the Robo-BIM project, containing a variety of information such as robot status, LIDAR, cameras, etc., for setting up a benchmark.
- Consider extending the range of motion of the robot to the building exterior. The challenge is to handle joint state estimation of GNSS and existing sensors by installing a GNSS system to enable task execution both indoors and outdoors.
- To improve scan quality assessment of the gimbal section, rotational parameters within the radar are added to the scan parameters to obtain better scan optimization parameters and adaptability in narrow spaces.
- For sensor calibration of the gimbal section, an IMU is installed on the radar rotating axis, and the gimbal is changed to continuous rotation, eliminating initial installation errors of the IMU through continuous integration and improving the accuracy of the rotating section of the gimbal.
- The autonomous navigation algorithm of the robot performed well in testing, but further verification is required to determine whether it can achieve the same results in more complex environments.
- This project relies relatively more on distance measurement information from sensors and less on image information. In recent years, algorithms such as Neural Radiance Fields (NERF) have shown excellent performance in the field of 3D reconstruction. The expected research direction of this project could focus on combining machine learning, deep learning algorithms, and distance measurement information.

Publications

Publication

- Yang, M.; Sun, X.; **Jia, F.**; Rushworth, A.; Dong, X.; Zhang, S.; Fang, Z.; Yang, G.; Liu, B. Sensors and Sensor Fusion Methodologies for Indoor Odometry: A Review. **Polymers** 2022, 14, 2019.
  https://doi.org/10.3390/polym14102019
  (JCR Q1, CAS Region III, Impact Factor: 4.967).

Patent

- **JIA FUHUA**; ADAM RUSHWORTH; GEORGIOS KAPOGIANNIS; KEGE XIE; MENGSHEN YANG. A kind of lidar platform and laser radar system, CN202221876135, Application patent, Disclosure Date: 1st, January, 2023.

- **JIA FUHUA**; MENGSHEN YANG; ADAM RUSHWORTH; GEORGIOS KAPOGIANNIS; LECHEN ZHANG. A novel BIM 3d reconstruction robot, 202310194701.2, Invention patent, Under review.

Acknowledgments

This dissertation represents a significant milestone in my long and challenging learning journey, and I am deeply grateful for the support of numerous individuals during this period. I extend my heartfelt appreciation to my supervisor, Dr. Adam Rushworth, for his consistent and valuable guidance throughout my Master's degree. His advice has not only guided my research, but has also deepened my understanding of the scientific problem. I am fortunate to have benefited from his vast knowledge and exceptional qualities of patience, support, empathy, and humor.

I also express my gratitude to my second and third supervisors, Dr. Shunbai and Georges, respectively, for their encouraging discussions that provided me with a multifaceted perspective on this project. Additionally, my collaborators during my Master's degree, Dr. Salman and Dr. Tianxiang Cui, have offered direction and inspiration for further research and exposed me to research methods in other related fields that have profoundly influenced my future development. I am also thankful for the collaborative efforts of my colleagues, including Mengshen, Lechen, Xiaoying, and others, with whom I navigated the research process's hardships and difficulties.

I extend my thanks to all the reviewers for their invaluable comments that have significantly contributed to the quality of this thesis. I am deeply grateful to my parents for their unwavering motivation, patience, and care during my education and life journey. Their support has been instrumental in completing this dissertation. Finally, I would like to thank all the individuals who have supported me and wish them happiness and success in their endeavors.

Bibliography

[1] I. Petri, S. Kubicki, Y. Rezgui, A. Guerriero, and H. Li, "Optimizing energy efficiency in operating built environment assets through building information modeling: A case study," *Energies*, vol. 10, no. 8, 2017. [Online]. Available: https://www.mdpi.com/1996-1073/10/8/1167

[2] "Citygml." [Online]. Available: https://www.ogc.org/standards/citygml

[3] F. Biljecki, H. Ledoux, and J. Stoter, "An improved lod specification for 3d building models," *Computers, Environment and Urban Systems*, vol. 59, pp. 25–37, 2016.

[4] "A novel rrt extend function for efficient and smooth mobile robot motion planning," Jun 2014. [Online]. Available: https://www.youtube.com/watch?v=RB3g_GP0-dU

[5] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation.* IEEE, 2010, pp. 987–993.

[6] G. Cantisani, J. D. C. Panesso, G. Del Serrone, P. Di Mascio, G. Gentile, G. Loprencipe, and L. Moretti, "Re-design of a road node with 7d bim: Geometrical, environmental and microsimulation approaches to implement a benefit-cost analysis between alternatives," *Automation in Construction*, vol. 135, p. 104133, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580522000061

[7] S. Kaewunruen, J. Sresakoolchai, and Z. Zhou, "Sustainability-based lifecycle management for bridge infrastructure using 6d bim," *Sustainability*, vol. 12, no. 6, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/6/2436

[8] I. . S. Department for Business, "Industrial strategy: Uk sector analysis." Research and analysis Report No., London: UK, 2012 [Online]. [Online]. Available: https://www.gov.uk/government/publications/industrial-strategy-uk-sector-analysis

[9] "Webpage of ministry of housing and urban-rural development of the people's republic of china." [Online]. Available: https://www.mohurd.gov.cn/xinwen/gzdt/202104/20210415_249788.html

[10] "3d laser scanning." [Online]. Available: https://geospatial.trimble.com/products-and-solutions/trimble-x12

[11] "Leica rtc360 3d laser scanner." [Online]. Available: https://leica-geosystems.com/products/laser-scanners/scanners/leica-rtc360

[12] "Gls-2000," Apr 2022. [Online]. Available: https://www.topconpositioning.com/ie/scanning/laser-scanners/gls-2000#panel-product-info

[13] A. Yilmaz and H. Temeltas, "Self-adaptive monte carlo method for indoor localization of smart agvs using lidar data," *Robotics and Autonomous Systems*, vol. 122, p. 103285, 2019.

[14] Z. Rozsa and T. Sziranyi, "Obstacle prediction for automated guided vehicles based on point clouds measured by a tilted lidar sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2708–2720, 2018.

[15] S. Park, S. Wang, H. Lim, and U. Kang, "Curved-voxel clustering for accurate segmentation of 3d lidar point clouds with real-time performance," in *2019 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS)*, ser. IEEE International Conference on Intelligent Robots and Systems. IEEE; RSJ, 2019, pp. 6459–6464, iEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, PEOPLES R CHINA, NOV 04-08, 2019.

[16] M. Langerwisch, M. S. Kraemer, K.-D. Kuhnert, and B. Wagner, "Construction of 3d environment models by fusing ground and aerial lidar point cloud data," in *INTELLIGENT AUTONOMOUS SYSTEMS 13*, ser. Advances in Intelligent Systems and Computing, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds., vol. 302. Univ Padova, 2016, pp. 473–485, 13th International Conference on Intelligent Autonomous Systems (IAS), Centro Congressi Padova, Padova, ITALY, JUL 15-18, 2014.

[17] M. Cao, P. Su, H. Chen, S. Tang, and Y. Liu, "3-d dense rangefinder sensor with a low-cost scanning mechanism," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 70, 2021.

[18] H. Yoshisada, Y. Yamada, A. Hiromori, H. Yamaguchi, and T. Higashino, "Indoor map generation from multiple lidar point clouds," in *2018 IEEE INTERNATIONAL CONFERENCE ON SMART COMPUTING (SMARTCOMP 2018)*. IEEE; IEEE Comp Soc, 2018, pp. 73–80, 4th IEEE International Conference on Smart Computing (SMARTCOMP), Sicily, ITALY, JUN 18-20, 2018.

[19] Y. Li and J. Ibanez-Guzman, “Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems,” *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[20] B. Behroozpour, P. A. M. Sandborn, M. C. Wu, and B. E. Boser, “Lidar system architectures and circuits,” *IEEE Communications Magazine*, vol. 55, no. 10, pp. 135–142, 2017.

[21] S. Royo and M. Ballesta-Garcia, “An overview of lidar imaging systems for autonomous vehicles,” *Applied sciences*, vol. 9, no. 19, p. 4093, 2019.

[22] M. Yang, X. Sun, F. Jia, A. Rushworth, X. Dong, S. Zhang, Z. Fang, G. Yang, and B. Liu, “Sensors and sensor fusion methodologies for indoor odometry: A review,” *Polymers*, vol. 14, no. 10, p. 2019, 2022.

[23] H. Tibebu, J. Roche, V. De Silva, and A. Kondoz, “Lidar-based glass detection for improved occupancy grid mapping,” *Sensors (Basel, Switzerland)*, vol. 21, no. 7, p. 2263, 2021.

[24] H. Zhou, Z. Yao, and M. Lu, “Uwb/lidar coordinate matching method with anti-degeneration capability,” *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3344–3352, 2020.

[25] K. Retan, F. Loshaj, and M. Heizmann, “Radar odometry on se(3) with constant velocity motion prior,” *IEEE ROBOTICS AND AUTOMATION LETTERS*, vol. 6, no. 4, pp. 6386–6393, OCT 2021.

[26] F. Li, X. Wang, S. Li, X. Gu, K. Gao, Q. Aina, and Z. Huang, “An improved indoor map construction method based on millimeter-wave radar,” in *2021 7TH INTERNATIONAL CONFERENCE ON AUTOMATION, ROBOTICS AND APPLICATIONS (ICARA 2021)*. IEEE; IEEE Robot & Automat Soc; Beijing Inst Control Robot & Intelligence Technol; Czech Univ Life Sci Prague, 2021, pp. 253–257, 7th International Conference on Automation, Robotics and Applications (ICARA), ELECTR NETWORK, FEB 04-06, 2021.

[27] C. Waldschmidt, J. Hasch, and W. Menzel, “Automotive radar—from first efforts to future systems,” *IEEE Journal of Microwaves*, vol. 1, no. 1, pp. 135–148, 2021.

[28] B. van Berlo, A. Elkelany, T. Ozcelebi, and N. Meratnia, “Millimeter wave sensing: A review of application pipelines and building blocks,” *IEEE Sensors Journal*, vol. 21, no. 9, pp. 10 332–10 368, 2021.

[29] I. Andersone, “Probabilistic mapping with ultrasonic distance sensors,” in *ICTE 2016*, ser. Procedia Computer Science, E. Ginters and J. Kohlhammer, Eds., vol. 104, 2017, pp. 362–368, iCTE Conference, Riga Tech Univ, Riga, LATVIA, 2016.

[30] M. Anjanappa, K. Datta, and T. Song, "Introduction to sensors and actuators," *The mechatronics handbook*, vol. 1, 2002.

[31] X. Liu, S. Cheng, H. Liu, S. Hu, D. Zhang, and H. Ning, "A survey on gas sensing technology," *Sensors*, vol. 12, no. 7, pp. 9635–9665, 2012.

[32] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, "An overview of autonomous vehicles sensors and their vulnerability to weather conditions," *Sensors*, vol. 21, no. 16, p. 5397, 2021.

[33] Z. Song, R. Chung, and X.-T. Zhang, "An accurate and robust strip-edge-based structured light means for shiny surface micromeasurement in 3-d," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 3, pp. 1023–1032, 2012.

[34] J. Geng, "Structured-light 3d surface imaging: a tutorial," *Advances in Optics and Photonics*, vol. 3, no. 2, pp. 128–160, 2011.

[35] G. Pavlidis, A. Koutsoudis, F. Arnaoutoglou, V. Tsioukas, and C. Chamzas, "Methods for 3d digitization of cultural heritage," *Journal of cultural heritage*, vol. 8, no. 1, pp. 93–98, 2007.

[36] "Overview of depth cameras," Feb 2022. [Online]. Available: https://www.aivero.com/overview-of-depth-cameras/

[37] C. Jaramillo, I. Dryanovski, R. G. Valenti, and J. Xiao, "6-dof pose localization in 3d point-cloud dense maps using a monocular camera," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2013, pp. 1747–1752.

[38] W. Zeng, S. Karaoglu, and T. Gevers, "Inferring point clouds from single monocular images by depth intermediation," *arXiv preprint arXiv:1812.01402*, 2018.

[39] D. Bradley, T. Boubekeur, and W. Heidrich, "Accurate multi-view reconstruction using robust binocular stereo and surface meshing," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[40] M. Yuda, Z. Xiangjun, S. Weiming, and L. Shaofeng, "Target accurate positioning based on the point cloud created by stereo vision," in *2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2016, pp. 1–5.

[41] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *European conference on computer vision*. Springer, 2016, pp. 349–364.

[42] X. Lu, X. Mao, H. Liu, X. Meng, and L. Rai, "Event camera point cloud feature analysis and shadow removal for road traffic sensing," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3358–3369, 2021.

[43] L. Hoegner and U. Stilla, “Mobile thermal mapping for matching of infrared images with 3d building models and 3d point clouds,” *Quantitative Infrared thermography journal*, vol. 15, no. 2, pp. 252–270, 2018.

[44] M. Puliti, G. Montaggioli, and A. Sabato, “Automated subsurface defects’ detection using point cloud reconstruction from infrared images,” *Automation in Construction*, vol. 129, p. 103829, 2021.

[45] J. Moreau, S. Ambellouis, and Y. Ruichek, “Equisolid fisheye stereovision calibration and point cloud computation,” in *ISPRS-SSG 2013, conference on Serving Society with Geoinformatics*, 2013, p. 6p.

[46] ——, “3d reconstruction of urban environments based on fisheye stereovision,” in *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 2012, pp. 36–41.

[47] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg, “Global localization from monocular slam on a mobile phone,” *IEEE transactions on visualization and computer graphics*, vol. 20, no. 4, pp. 531–539, 2014.

[48] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on various inertial measurement unit (imu) sensor applications,” *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.

[49] I. A. Faisal, T. W. Purboyo, and A. S. R. Ansori, “A review of accelerometer sensor and gyroscope sensor in imu sensors on motion capture,” *J. Eng. Appl. Sci*, vol. 15, no. 3, pp. 826–829, 2019.

[50] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*. Springer, 2008, vol. 200.

[51] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[52] 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, DEC 2016.

[53] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, “Recent advances in indoor localization: A survey on theoretical approaches and applications,” *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*, vol. 19, no. 2, pp. 1327–1346, 2017.

[54] S. Y. Chen, “Kalman filter for robot vision: A survey,” *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, vol. 59, no. 11, pp. 4409–4420, NOV 2012.

[55] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES*, vol. 2, no. 3, pp. 194–220, SEP 2017.

[56] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, “A survey on vision-based uav navigation,” *GEO-SPATIAL INFORMATION SCIENCE*, vol. 21, no. 1, SI, pp. 21–32, 2018.

[57] E. Garcia-Fidalgo and A. Ortiz, “Vision-based topological mapping and localization methods: A survey,” *ROBOTICS AND AUTONOMOUS SYSTEMS*, vol. 64, pp. 1–20, FEB 2015.

[58] G. Huang, “Visual-inertial navigation: A concise review,” in *2019 INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA)*, ser. IEEE International Conference on Robotics and Automation ICRA, A. Howard, K. Althoefer, F. Arai, F. Arrichiello, B. Caputo, J. Castellanos, K. Hauser, V. Isler, J. Kim, H. Liu, P. Oh, V. Santos, D. Scaramuzza, A. Ude, R. Voyles, K. Yamane, and A. Okamura, Eds. Bosch; DJI; Kinova; Mercedes Benz; Samsung; Argo AI; Clearpath Robot; Element AI; Fetch Robot; Huawei; iRobot; KUKA; Quanser; SICK; Toyota Res Inst; Uber; Waymo; Zhejiang Lab; Amazon; Applanix; Cloudminds; Honda Res Inst; MathWorks; Ouster, 2019, pp. 9572–9582, iEEE International Conference on Robotics and Automation (ICRA), Montreal, CANADA, MAY 20-24, 2019.

[59] Q. Chen, J. Chen, and W. Huang, “Pathfinding method for an indoor drone based on a bim-semantic model,” *Advanced Engineering Informatics*, vol. 53, p. 101686, 2022.

[60] S. Goessens, C. Mueller, and P. Latteur, “Feasibility study for drone-based masonry construction of real-scale structures,” *Automation in Construction*, vol. 94, pp. 458–480, 2018.

[61] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *IEEE Transactions on robotics and automation*, vol. 12, no. 6, pp. 869–880, 1996.

[62] ——, “Gyrodometry: A new method for combining data from gyros and odometry in mobile robots,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1996, pp. 423–428.

[63] K. S. Chong and L. Kleeman, “Accurate odometry and error modelling for a mobile robot,” in *Proceedings of International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 2783–2788.

[64] F. Pomerleau, F. Colas, R. Siegwart *et al.*, “A review of point cloud registration algorithms for mobile robotics,” *Foundations and Trends® in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[65] W. Wen, L.-T. Hsu, and G. Zhang, “Performance analysis of ndt-based graph slam for autonomous vehicle in diverse typical driving scenarios of hong kong,” *Sensors*, vol. 18, no. 11, p. 3928, 2018.

[66] D. L. Lu, “Vision-enhanced lidar odometry and mapping,” Ph.D. dissertation, Carnegie Mellon University Pittsburgh, PA, USA, 2016.

[67] D. Münch, B. Combès, and S. Prima, “A modified icp algorithm for normal-guided surface registration,” in *Medical Imaging 2010: Image Processing*, vol. 7623. SPIE, 2010, pp. 426–433.

[68] J. Serafin and G. Grisetti, “Nicp: Dense normal based point cloud registration,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 742–749.

[69] G. Godin, M. Rioux, and R. Baribeau, “Three-dimensional registration using range and intensity information,” in *Videometrics III*, vol. 2350. SPIE, 1994, pp. 279–290.

[70] M. Greenspan and M. Yurick, “Approximate kd tree search for efficient icp,” in *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.* IEEE, 2003, pp. 442–448.

[71] S.-Y. Park and M. Subbarao, “An accurate and fast point-to-plane registration technique,” *Pattern Recognition Letters*, vol. 24, no. 16, pp. 2967–2976, 2003.

[72] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp.” in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[73] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3. IEEE, 2003, pp. 2743–2748.

[74] M. Magnusson, A. Lilienthal, and T. Duckett, “Scan registration for autonomous mining vehicles using 3d-ndt,” *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.

[75] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time.” in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[76] T. Shan and B. Englot, “Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[77] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar slam,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[78] "Openslam gmapping." [Online]. Available: https://openslam-org.github.io/gmapping.html

[79] "Hector slam." [Online]. Available: http://wiki.ros.org/hector_slam

[80] Ros-Perception, "Ros wrapper and node for openkarto." [Online]. Available: https://github.com/ros-perception/slam_karto

[81] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in *Robotics: Science and Systems*, vol. 2018, 2018, p. 59.

[82] R. W. Wolcott and R. M. Eustice, "Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 292–319, 2017.

[83] J.-E. Deschaud, "Imls-slam: Scan-to-model matching based on 3d data," in *2018 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2018, pp. 2480–2485.

[84] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, 2022.

[85] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "Mulls: Versatile lidar slam via multi-metric linear least square," in *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2021, pp. 11 633–11 640.

[86] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2021, pp. 4390–4396.

[87] HKUST-Aerial-Robotics, "Hkust-aerial-robotics/a-loam: Advanced implementation of loam." [Online]. Available: https://github.com/HKUST-Aerial-Robotics/A-LOAM

[88] L. Li, X. Kong, X. Zhao, W. Li, F. Wen, H. Zhang, and Y. Liu, "Sa-loam: Semantic-aided lidar slam with loop closure," in *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2021, pp. 7627–7634.

[89] "Coreslam." [Online]. Available: http://wiki.ros.org/coreslam

[90] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2019, pp. 4530–4537.

[91] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, “Segmap: 3d segment mapping using data-driven descriptors,” *arXiv preprint arXiv:1804.09557*, 2018.

[92] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H.-L. Bloecher, K. Werber, and A. Sailer, “Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding,” in *2016 IEEE Radar Conference (RadarConf)*. IEEE, 2016, pp. 1–6.

[93] T. Zhou, M. Yang, K. Jiang, H. Wong, and D. Yang, “Mmw radar-based technologies in autonomous driving: A review,” *Sensors*, vol. 20, no. 24, p. 7283, 2020.

[94] D. Vivet, F. Gérossier, P. Checchin, L. Trassoudaine, and R. Chapuis, “Mobile ground-based radar sensor for localization and mapping: An evaluation of two approaches,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 8, p. 307, 2013.

[95] S. Cen, “Ego-motion estimation and localization with millimeter-wave scanning radar,” Ph.D. dissertation, University of Oxford, 2019.

[96] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, “Instantaneous ego-motion estimation using multiple doppler radars,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1592–1597.

[97] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A.-A. Agha-Mohammadi, and C. Heckman, “Radar-inertial ego-velocity estimation for visually degraded environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5739–5746.

[98] E. Ward and J. Folkesson, “Vehicle localization with low cost radar sensors,” in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 864–870.

[99] P.-C. Kung, C.-C. Wang, and W.-C. Lin, “A normal distribution transform-based radar odometry designed for scanning and automotive radars,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 417–14 423.

[100] M. Rapp, M. Barjenbruch, M. Hahn, J. Dickmann, and K. Dietmayer, “Probabilistic ego-motion estimation using multiple automotive radar sensors,” *Robotics and Autonomous Systems*, vol. 89, pp. 136–146, 2017.

[101] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901300167X

[102] M. Farsi, K. Ratcliff, J. Johnson, C. Allen, K. Karam, and R. Pawson, "Robot control system for window cleaning," in *Proceedings of 1994 American Control Conference-ACC'94*, vol. 1. IEEE, 1994, pp. 994–995.

[103] M. Ollis and A. Stentz, "Vision-based perception for an automated harvester," in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97*, vol. 3. IEEE, 1997, pp. 1838–1844.

[104] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *The International journal of robotics research*, vol. 22, no. 7-8, pp. 441–466, 2003.

[105] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, "Uniform coverage of automotive surface patches," *The International Journal of Robotics Research*, vol. 24, no. 11, pp. 883–898, 2005.

[106] Z. L. Cao, Y. Huang, and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *Journal of Robotic systems*, vol. 5, no. 2, pp. 87–102, 1988.

[107] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1, pp. 113–126, 2001.

[108] G. Fevgas, T. Lagkas, V. Argyriou, and P. Sarigiannidis, "Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles," *Sensors*, vol. 22, no. 3, p. 1235, 2022.

[109] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, 2021.

[110] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "A survey on multi-robot coverage path planning for model reconstruction and mapping," *SN Applied Sciences*, vol. 1, no. 8, pp. 1–24, 2019.

[111] J. Palacín, T. Palleja, I. Valganón, R. Pernia, and J. Roca, "Measuring coverage performances of a floor cleaning mobile robot using a vision system," in *Proceedings of the 2005 IEEE international conference on robotics and automation.* IEEE, 2005, pp. 4236–4241.

[112] C. Tholen, T. A. El-Mihoub, L. Nolle, and O. Zielinski, "On the robustness of self-adaptive levy-flight," in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO).* IEEE, 2018, pp. 1–5.

[113] Z. J. Butler, A. A. Rizzi, and R. L. Hollis, "Contact sensor-based coverage of rectilinear environments," in *Proceedings of the 1999 IEEE international symposium on intelligent control intelligent systems and semiotics (Cat. No. 99CH37014).* IEEE, 1999, pp. 266–271.

[114] C.-h. Li, Y. Song, F.-y. Wang, Z.-q. Wang, and Y.-b. Li, "A chaotic coverage path planner for the mobile robot based on the chebyshev map for special missions," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 9, pp. 1305–1319, 2017.

[115] C. Gao, Y. Kou, Z. Li, A. Xu, Y. Li, and Y. Chang, "Optimal multirobot coverage path planning: ideal-shaped spanning tree," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[116] J. Yan, Y. Li, Z. Song, and Q. Zhang, "'research on uav coverage path planning algorithm based on improved artificial potential field method," *Oper. Res. Fuzziology*, vol. 9, pp. 264–270, 2019.

[117] R. Shivgan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR).* IEEE, 2020, pp. 1–6.

[118] X. Chen and Y. Li, "Smooth path planning of a mobile robot using stochastic particle swarm optimization," in *2006 International conference on mechatronics and automation.* IEEE, 2006, pp. 1722–1727.

[119] Z. Chibin, W. Xingsong, and D. Yong, "Complete coverage path planning based on ant colony algorithm," in *2008 15th International Conference on Mechatronics and Machine Vision in Practice.* IEEE, 2008, pp. 357–361.

[120] K. Sandamurthy and K. Ramanujam, "A hybrid weed optimized coverage path planning technique for autonomous harvesting in cashew orchards," *Information Processing in Agriculture*, vol. 7, no. 1, pp. 152–164, 2020.

[121] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan," *Algorithmic and Computational Robotics*, pp. 303–307, 2001.

[122] W. Jing, J. Polden, P. Y. Tao, C. F. Goh, W. Lin, and K. Shimada, "Model-based coverage motion planning for industrial 3d shape inspection applications," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE).* IEEE, 2017, pp. 1293–1300.

[123] X. Zhou, Z. Yi, Y. Liu, K. Huang, and H. Huang, "Survey on path and view planning for uavs," *Virtual Reality & Intelligent Hardware*, vol. 2, no. 1, pp. 56–69, 2020.

[124] S. Lindner, C. Garbe, and K. Mombaur, “Optimization based multi-view coverage path planning for autonomous structure from motion recordings,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3278–3285, 2019.

[125] A. Maoudj and A. Hentout, “Optimal path planning approach based on q-learning algorithm for mobile robots,” *Applied Soft Computing*, vol. 97, p. 106796, 2020.

[126] W. Li, T. Zhao, and S. Dian, “Multirobot coverage path planning based on deep q-network in unknown environment,” *Journal of Robotics*, vol. 2022, 2022.

[127] A. V. Le, P. Veerajagadheswar, P. Thiha Kyaw, M. R. Elara, and N. H. K. Nhan, “Coverage path planning using reinforcement learning-based tsp for htetran—a polyabolo-inspired self-reconfigurable tiling robot,” *Sensors*, vol. 21, no. 8, p. 2577, 2021.

[128] A. K. Lakshmanan, R. E. Mohan, B. Ramalingam, A. V. Le, P. Veerajagadeshwar, K. Tiwari, and M. Ilyas, “Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot,” *Automation in Construction*, vol. 112, p. 103078, 2020.

[129] K. G. S. Apuroop, A. V. Le, M. R. Elara, and B. J. Sheu, “Reinforcement learning-based complete area coverage path planning for a modified htrihex robot,” *Sensors*, vol. 21, no. 4, p. 1067, 2021.

[130] S. Ragi and E. K. Chong, “Uav path planning in a dynamic environment via partially observable markov decision process,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2397–2412, 2013.

[131] H. Choset and P. Pignon, “Coverage path planning: The boustrophedon cellular decomposition,” in *Field and service robotics.* Springer, 1998, pp. 203–209.

[132] S. G. Tzafestas, *Introduction to mobile robot control.* Elsevier, 2013.

[133] J. H. Holland, “Genetic algorithms,” *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[134] L. Qiu, “Research on hierarchical cooperative algorithm based on genetic algorithm and particle swarm optimization,” in *International Symposium on Intelligence Computation and Applications.* Springer, 2017, pp. 16–25.

[135] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[136] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[137] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L. Xiao, "A review of algorithms for filtering the 3d point cloud," *Signal Processing: Image Communication*, vol. 57, pp. 103–112, 2017.

[138] C. Chen, J. Guo, H. Wu, Y. Li, and B. Shi, "Performance comparison of filtering algorithms for high-density airborne lidar point clouds over complex landscapes," *Remote Sensing*, vol. 13, no. 14, p. 2663, 2021.

[139] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 339, 2017.

[140] "Point cloud processing." [Online]. Available: https://ww2.mathworks.cn/help/vision/point-cloud-processing.html

[141] R. P. de Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "Multi-object detection and pose estimation in 3d point clouds: A fast grid-based bayesian filter," in *2013 IEEE International Conference on Robotics and Automation.* IEEE, 2013, pp. 4250–4255.

[142] L. Qin, W. Wu, Y. Tian, and W. Xu, "Lidar filtering of urban areas with region growing based on moving-window weighted iterative least-squares fitting," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 6, pp. 841–845, 2017.

[143] H. Chen, S. Liu, W. Chen, H. Li, and R. Hill, "Equivariant point network for 3d point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 514–14 523.

[144] O. Schall, A. Belyaev, and H.-P. Seidel, "Feature-preserving non-local denoising of static and time-varying range data," in *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, 2007, pp. 217–222.

[145] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271).* IEEE, 1998, pp. 839–846.

[146] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," in *ACM SIGGRAPH 2009 papers*, 2009, pp. 1–9.

[147] C. Dinesh, G. Cheung, and I. V. Bajić, "Point cloud denoising via feature graph laplacian regularization," *IEEE Transactions on Image Processing*, vol. 29, pp. 4143–4158, 2020.

[148] Y. Xiong, M. Ren, R. Liao, K. Wong, and R. Urtasun, "Deformable filter convolution for point cloud reasoning," *arXiv preprint arXiv:1907.13079*, 2019.

[149] A. Sampath and J. Shan, "Clustering based planar roof extraction from lidar data," in *American Society for Photogrammetry and Remote Sensing Annual Conference, Reno, Nevada, May*, 2006, pp. 1–6.

[150] T. Melzer, "Non-parametric segmentation of als point clouds using mean shift," 2007.

[151] A. Ferraz, F. Bretar, S. Jacquemoud, G. Gonçalves, and L. Pereira, "3d segmentation of forest structure using a mean-shift based algorithm," in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 1413–1416.

[152] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds," *IEEE Transactions on geoscience and remote sensing*, vol. 48, no. 3, pp. 1554–1567, 2009.

[153] P. V. Hough, "Method and means for recognizing complex patterns," *US patent*, vol. 3, no. 6, 1962.

[154] A. Kaiser, J. A. Ybanez Zepeda, and T. Boubekeur, "A survey of simple geometric primitives detection methods for captured 3d data," in *Computer Graphics Forum*, vol. 38, no. 1. Wiley Online Library, 2019, pp. 167–196.

[155] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A probabilistic hough transform," *Pattern recognition*, vol. 24, no. 4, pp. 303–316, 1991.

[156] A. Yla-Jaaski and N. Kiryati, "Adaptive termination of voting in the probabilistic circular hough transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 911–915, 1994.

[157] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic hough transform for line detection," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 1. IEEE, 1999, pp. 554–560.

[158] L. A. Fernandes and M. M. Oliveira, "Real-time line detection through an improved hough transform voting scheme," *Pattern recognition*, vol. 41, no. 1, pp. 299–314, 2008.

[159] M. Camurri, R. Vezzani, and R. Cucchiara, "3d hough transform for sphere recognition on point clouds," *Machine vision and applications*, vol. 25, no. 7, pp. 1877–1891, 2014.

[160] S. Choi, T. Kim, and W. Yu, "Performance evaluation of ransac family," *Journal of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[161] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," in *European conference on computer vision*. Springer, 2008, pp. 500–513.

[162] A. Adam, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "H-ransac: A hybrid point cloud segmentation combining 2d and 3d data," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 4, no. 2, pp. 1–8, 2018.

[163] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells," *Remote Sensing*, vol. 9, no. 5, p. 433, 2017.

[164] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3d: A review of point cloud semantic segmentation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 38–59, 2020.

[165] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, "A review of deep learning-based semantic segmentation for point cloud," *IEEE Access*, vol. 7, pp. 179 118–179 133, 2019.

[166] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, vol. 19, no. 19, p. 4188, 2019.

[167] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," *Automation in construction*, vol. 19, no. 7, pp. 829–843, 2010.

[168] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 166–210, 2001.

[169] "Revit software," Dec 2022. [Online]. Available: https://www.autodesk.com/products/revit/overview?term=1-YEAR&amp;tab=subscription

[170] "3d-4d building information modeling," Apr 2022. [Online]. Available: https://www.gsa.gov/real-estate/design-and-construction/3d4d-building-information-modeling?gsaredirect=bim

[171] E. Hoffman, "Specifying laser scanning services," *Chemical engineering progress*, vol. 101, no. 5, pp. 34–38, 2005.

[172] H. E. Goldberg, "State of the aec industry: Bim implementation slow, but inevitable," *Revista CAD alystmaio*, 2005.

[173] I. Panushev and J. Brandt, "3d imaging pilot projects: Three case studies," *Harvard Design School: Boston, MA, USA*, 2007.

[174] "Netfabb," Jun 2022. [Online]. Available: https://www.autodesk.com/products/netfabb/free-trial

[175] "Meshlab." [Online]. Available: https://www.meshlab.net/

[176] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun, "Using em to learn 3d models of indoor environments with mobile robots," in *ICML*, vol. 1, 2001, pp. 329–336.

[177] A. Nuchter, H. Surmann, and J. Hertzberg, "Automatic model refinement for 3d reconstruction with mobile robots," in *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, 2003, pp. 394–401.

[178] L. Ding, W. Jiang, Y. Zhou, C. Zhou, and S. Liu, "Bim-based task-level planning for robotic brick assembly through image-based 3d modeling," *Advanced Engineering Informatics*, vol. 43, p. 100993, 2020.

[179] S. Kim, M. Peavy, P.-C. Huang, and K. Kim, "Development of bim-integrated construction robot task planning and simulation system," *Automation in Construction*, vol. 127, p. 103720, 2021.

[180] "Construction solutions." [Online]. Available: https://www.bostondynamics.com/solutions/construction

[181] "Leica blk2fly - flying laser scanner." [Online]. Available: https://shop.leica-geosystems.com/leica-blk/blk2fly

[182] F. SA, "Elios 2 - indoor drone for confined space inspections." [Online]. Available: https://www.flyability.com/elios-2

[183] "Quicabot." [Online]. Available: https://www.transformarobotics.com/quicabot

[184] A. Ibrahim, A. Sabet, and M. Golparvar-Fard, "Bim-driven mission planning and navigation for automatic indoor construction progress detection using robotic ground platform," in *EC3 Conference 2019*, vol. 1. University College Dublin, 2019, pp. 182–189.

[185] A. Adán, B. Quintana, S. Prieto, and F. Bosché, "An autonomous robotic platform for automatic extraction of detailed semantic models of buildings," *Automation in construction*, vol. 109, p. 102963, 2020.

[186] R. Kurazume, S. Oshima, S. Nagakura, Y. Jeong, and Y. Iwashita, "Automatic large-scale three dimensional modeling using cooperative multiple robots," *Computer Vision and Image Understanding*, vol. 157, pp. 25–42, 2017.

[187] D. Borrmann, A. Nüchter, M. Đakulović, I. Maurović, I. Petrović, D. Osmanković, and J. Velagić, "A mobile robot based system for fully automated thermal 3d mapping," *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 425–440, 2014.

[188] P. Kim, J. Chen, and Y. K. Cho, "Slam-driven robotic mapping and registration of 3d point clouds," *Automation in Construction*, vol. 89, pp. 38–48, 2018.

[189] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "3d lidar-imu calibration based on upsampled preintegrated measurements for motion distortion correction," in *2018 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2018, pp. 2149–2155.

[190] Hku-Mars, "Hkumars/livox_camera_calib: This repository is used for automatic calibration between high resolution lidar and camera in targetless scenes." [Online]. Available: https://github.com/hku-mars/livox_camera_calib

[191] E. Barbieri, "An intro to real-time linux with ubuntu," Jun 2022. [Online]. Available: https://ubuntu.com/blog/real-time-linux-qa

[192] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

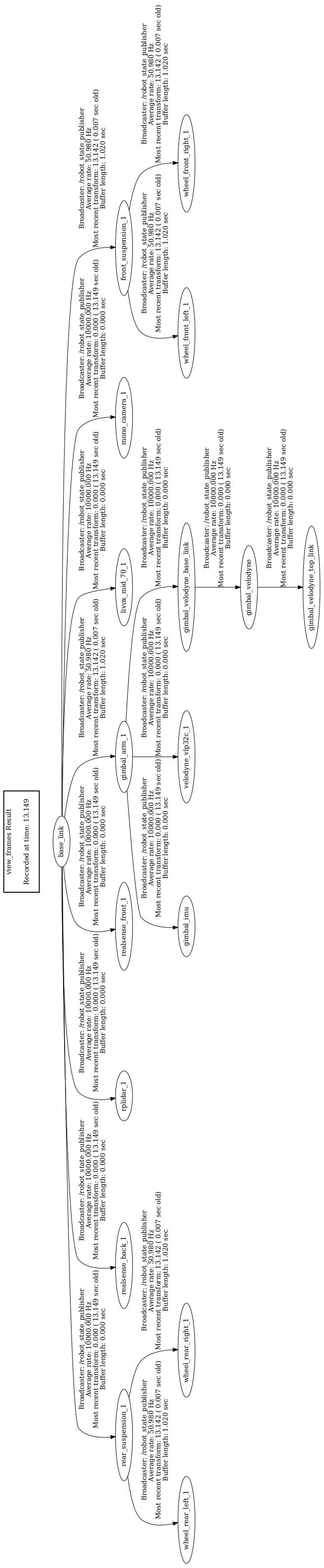
[193] "Cmake is an open-source, cross-platform family of tools designed to build, test and package software." [Online]. Available: https://cmake.org/

[194] "sbpl_lattice_planner." [Online]. Available: http://wiki.ros.org/sbpl_lattice_planner

[195] A. Ayala, F. Cruz, D. Campos, R. Rubio, B. Fernandes, and R. Dazeley, "A comparison of humanoid robot simulators: A quantitative approach," in *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob).* IEEE, 2020, pp. 1–6.

[196] A. Farley, J. Wang, and J. A. Marshall, "How to pick a mobile robot simulator: A quantitative comparison of coppeliasim, gazebo, morse and webots with a focus on accuracy of motion," *Simulation Modelling Practice and Theory*, vol. 120, p. 102629, 2022.

Appendix



view_frames Result
Recorded at time: 13.149
base_link
Broadcaster: /robot_state_publisher
Average rate: 50.980 Hz
Most recent transform: 13.142 (0.007 sec old)
Buffer length: 1.020 sec
front_suspension_1
Broadcaster: /robot_state_publisher
Average rate: 50.980 Hz
Most recent transform: 13.142 (0.007 sec old)
Buffer length: 1.020 sec
wheel_front_right_1
Broadcaster: /robot_state_publisher
Average rate: 50.980 Hz
Most recent transform: 13.142 (0.007 sec old)
Buffer length: 1.020 sec
wheel_front_left_1
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
mono_camera_1
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
livox_mid_70_1
Broadcaster: /robot_state_publisher
Average rate: 50.980 Hz
Most recent transform: 13.142 (0.007 sec old)
Buffer length: 1.020 sec
gimbal_arm_1
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
gimbal_velodyne_base_link
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
gimbal_velodyne
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
gimbal_velodyne_top_link
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
velodyne_vlp32c_1
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
gimbal_imu
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
realsense_front_1
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
rplidar_1
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
realsense_back_1
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (13.149 sec old)
Buffer length: 0.000 sec
rear_suspension_1
Broadcaster: /robot_state_publisher
Average rate: 50.980 Hz
Most recent transform: 13.142 (0.007 sec old)
Buffer length: 1.020 sec
wheel_rear_right_1
Broadcaster: /robot_state_publisher
Average rate: 50.980 Hz
Most recent transform: 13.142 (0.007 sec old)
Buffer length: 1.020 sec
wheel_rear_left_1

Figure 1: The detailed sensors coordinate conversion relationships of *Robo-BIM* automatic scanning robot.