

Multi-sensor Fusion for Positioning and Semantic Information Extraction in Indoor Environments

by

Dedong Zhang

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2023

© Dedong Zhang 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: **Dr. Naser El-Sheimy, Professor**
Dept. of Geomatics Engineering
University of Calgary

Supervisor(s): **Dr. Jonathan Li, Professor**
Dept. of Geography and Environmental Management
University of Waterloo

Supervisor(s): **Dr. John Zelek, Associate Professor**
Dept. of Systems Design Engineering
University of Waterloo

Internal Member: **Dr. David Clausi, Professor**
Dept. of Systems Design Engineering
University of Waterloo

Internal Member: **Dr. Alexander Wong, Professor**
Dept. of Systems Design Engineering
University of Waterloo

Internal-External Member: **Dr. Michael A. Chapman, Professor**
Dept. of Civil Engineering
Toronto Metropolitan University

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public

Statement of Contributions

This PhD thesis is accomplished in the manuscript option, following the graduation guidelines administered by the Graduate Program in Systems Design Engineering. It consists of three required manuscripts [1][2][3], which form the main body of Chapters 3, 4 and 5 of this PhD thesis. These manuscripts contain material from three multi-authored papers, where the manuscripts are minorly changed for formatting consistency. As the lead author of these three papers, I was responsible for conceptualizing the studies, writing codes, carrying out main experiments, drafting and submitting the manuscripts. My co-author, Weikai Tan and Lingfei Ma, contributed to the comparison experiments by running compared methods for Papers [1] and [3]. Weikai Tan helped to collect the datasets for Papers [1] and [3]. Zheng Gong helped to collect the datasets and Yiping Chen helped in proofreading for Paper [2]. My supervisors Prof. Jonathan Li and Prof. John Zelek, provided guidance throughout the process and feedback on draft manuscripts. The references for the three papers are provided below:

- [1] **Zhang, Dedong**, Lingfei Ma, Zeng Gong, Weikai Tan, John Zelek and Jonathan Li, 2023. "An Overlap-free Calibration Method for LiDAR-Camera Platforms based on Environmental Perception," IEEE Transactions on Instrumentation and Measurement, doi:10.1109/TIM.2023.3244816.
- [2] **Zhang, Dedong**, Zeng Gong, Yiping Chen, John Zelek and Jonathan Li, 2019. "SLAM-based Multi-Sensor Backpack LiDAR Systems in GNSS-denied Environments," In: Proceedings of IGARSS 2019, pp. 8984-8987, doi: 10.1109/IGARSS.2019.8898669.
- [3] **Zhang, Dedong**, Weikai Tan, Lingfei Ma, John Zelek and Jonathan Li, "Enhanced Indoor LiDAR SLAM with Total Station Measurements for Accurate Trajectory Estimation", IEEE Transactions on Intelligent Transportation Systems, submitted.

I additionally co-authored several papers in related areas of LiDAR point cloud processing and remote sensing during my Ph.D. program, including

- G. Zheng, H. Lin, **Dedong Zhang**, Z. Luo, J. Zelek, Y. Chen, A. Nurunnabi, C. Wang, and J. Li, 2020. "A Frustum-based probabilistic framework for 3D object detection by fusion of

- LiDAR and camera data." ISPRS Journal of Photogrammetry and Remote Sensing vol. 159, pp. 90-100.
- L. Ma, Z. Chen, Y. Li, **Dedong Zhang**, J. Li, and M. A. Chapman, 2019. "Multispectral Airborne Laser Scanning Point-Clouds for Land Cover Classification Using Convolutional Neural Networks," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2-W13, pp. 79–86
 - N. Qin, W. Tan, L. Ma, **Dedong Zhang**, and J. Li, 2021. "OpenGF: An ultra-large-scale ground filtering dataset built upon open ALS point clouds around the world." In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop, pp. 1082-1091.
 - W. Tan, **Dedong Zhang**, L. Ma, L. Wang, N. Qin, Y. Chen, J. Li, 2021."Semantic Segmentation of UAV LiDAR Point Clouds of a Stack Interchange with Deep Neural Networks," In: Proceedings of IGARSS 2021, pp. 582-585, doi: 10.1109/IGARSS47720.2021.9554610.
 - W. Zhu, W. Tan, L. Ma, **Dedong Zhang**, J. Li, and M. A., Chapman, 2021. "A Capsnets Approach to Pavement Crack Detection Using Mobile Laser Scanning Point Clouds." International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 43, pp. 39-44.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis focuses on three significant challenges in multi-sensor data fusion for real-world uses: (1) Lack of multisensory datasets for mapping and positioning datasets in underground environments, (2) inefficient and error-prone of most of the existing multi-sensor calibration methods, and (3) lack of effective methods for semantic segmentation of 3D point clouds in indoor environments, which is an important direction for last-mile autonomous driving and indoor positioning.

This thesis proposes three methods to address the above problem: (1) Using a modular mobile robot platform to collect multi-sensor datasets for underground environments. For various indoor and underground scenarios (e.g., office buildings, underground parking lots), different mobile robots, such as mobile LiDAR backpacks, robotic dogs, and wheeled multi-sensor carts, are used to collect rich 2D-3D data. Accurate trajectory acquisition is achieved by optimized algorithms. (2) Target-free automatic multi-sensor calibration: a target-free automatic self-calibration method is proposed for LiDAR and camera. The proposed method uses the trajectory constraint between LiDAR and camera to optimize the calibration parameters. The graph optimization algorithm is used to solve the problem of camera trajectory drift and finally achieve the purpose of automatic calibration. (3) A novel 3D SLAM algorithm tailored specifically for indoor point cloud scenes consists of a front-end ranging method and a back-end closed-loop detection and optimization algorithm using sub-maps. Based on this, a front- and back-end 3D point cloud segmentation framework is proposed, which includes the PointNet algorithm for creating high-precision point cloud semantic models. And utilizing the fusion information 3D map to achieve environmental semantic information extraction for last-mile autonomous driving.

The above methods are validated using the collected datasets. The results show that (1) for the LiDAR and camera automatic calibration methods, the RMSE can be improved from a maximum of 6.637° to 0.564° , and the RMSE of translation can be improved from a maximum of 0.197 m to 0.564° compared with traditional manual calibration methods. (2) Experimental results of point cloud quality analysis and evaluation for the optimized SLAM algorithm show that the front-end trajectory and attitude estimation (ranging) algorithm, combined with the back-end closed-loop optimization, can provide centimeter-level accuracy for localization and map construction. (3) Results from indoor 3D point cloud data segmentation experiments demonstrate that the PSIF method outperforms others, improving the training efficiency of PointNet. The average semantic modeling accuracy across categories exceeds 84.8%.

Acknowledgments

First of all, I would like to thank my Ph.D. supervisor, Dr. Jonathan Li. You have helped me to open the door to academic research. Whenever I was stagnant, you helped me to open my mind and get encouragement to persevere. You have also been valuable in my life, helping me realize the joy of research and innovation. You have shown your dedication and responsibility to me during my Ph.D., from the advanced hardware equipment for my research to the configuration of the experimental environment. You also gave me a lot of opportunities to learn and exchange. From collaborating with researchers from different schools to going abroad to attend academic conferences, all of them are valuable experiences and assets for me. I will never forget your advice and help.

I would also like to thank my Ph.D. supervisor, Dr. John Zelek. You have been a constant source of academic inspiration and guided me step by step to find my research direction and topics. At the same time, your academic supervision and assistance were the prerequisites for me to be able to complete my scientific research. When I encountered a research problem, you were always the first to help me analyze it and provide a direction for solving it. You always provide me with constructive academic articles and peer research content, so that I can see the shortcomings of my research and discover new research directions. I have seen in you the ability and character of a senior scholar. You will certainly be a model for my future academic career.

I would also like to thank all the members of the doctoral defense committee: Dr. Alexander Wong, Dr. David Clausi, Dr. Michael A. Chapman, Dr. Naser El-Sheimy, Dr. David Del Rey Fernández, Dr. Jonathan Li and Dr. John Zelek. Thank you for your critical questions and advice for my thesis writing and defense, I have learned a lot.

I would also like to thank Dr. Zheng Gong and Dr. Lingfei Ma. They are the first colleagues I met at the University of Waterloo and my academic partners and friends in my life. Because of your help, I quickly adapted to the environment here. Your serious research attitude and excellent academic ability motivated me to monitor myself.

Thanks to my wife Ching-Han Tseng, you are my strongest support and best teammate. You have been behind all my academic efforts and achievements. I am very lucky to have you with me in my life. Thanks to my daughter Bethany, your birth was the best thing that happened during my Ph.D. Study. Because of you, Daddy has the motivation to become a better person. I also want to thank my parents for providing me with ample support in life. You are the ones who have shown me how to be a good partner and a good parent. I would also like to thank my parents-in-law for their support and help.

In addition, I would like to thank my classmates and colleagues in both the Geospatial Intelligence and Mapping Lab and the Vision and Image Processing Lab at the University of Waterloo: Weikai Tan, Ying Li, Hongjie He, Kyle Gao, Ming Liu, Dr. Nannan Qin, Lanying Wang, Hongzhang Xu, William Xu, Dr. Ke Yang, Zihao Yang, Yan Liu, Sarah Fatholahi,

Zilong Zhong. Thank you for your long-time companionship and help. Getting to know you all has made my doctoral studies very exciting.

Finally, I would also like to thank professors in the Department of Systems Design Engineering: Dr. Alexander Wong, Dr. David Clausi, and Dr. Linlin Xu, professor in the Department of Mechanical and Mechatronics Engineering: Dr. Dongpu Cao, and professor in the Department of Geography and Environmental Management: Dr. Peter Deadman. Thank you all for your academic help and guidance.

Table of Contents

Examining Committee Membership	ii
Author's Declaration.....	iii
Statement of Contributions	iv
Abstract	vi
Acknowledgments.....	vii
List of Figures	xiv
List of Tables	xvi
List of Abbreviations	xvii
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 Challenges for Indoor Navigation and Mapping	2
1.1.2 Simultaneous Localization and Mapping	3
1.2 Research Objectives	5
1.3 Thesis Structure.....	9
Chapter 2 Background and Related Studies.....	12
2.1 Multi-line LiDAR and Mobile Scanning Systems	12
2.2 Multi-line Laser Scanner Coordinate System Calibration.....	14

2.3 Camera and LiDAR Calibration.....	15
2.3.1 Target-based Methods	16
2.3.2 Feature-based Methods.....	18
2.3.3 Optimization-based Methods.....	19
2.3.4 Deep Learning-based Methods.....	21
2.4 3D High-precision LiDAR-based SLAM.....	23
2.5 Camera-LiDAR Fusion for 3D Scene Segmentation	24
2.5.1 2D Object Detection/Labeling.....	24
2.5.2 3D Object Detection/Labeling.....	25
2.5.3 2D/3D Joint Estimation	26
2.6 Chapter Summary	27
Chapter 3 WATiMMS: Indoor Multi-Sensor Dataset for Mapping and Positioning	29
3.1 Introduction	29
3.2 Data Collection Platform Setup.....	35
3.3 Sensor Synchronization.....	37
3.4 Multi-sensor Calibration Method	39
3.4.1 Camera Calibration and Distortion Removal	40
3.4.2 External Reference Calibration	43
3.4.3 LiDAR Point Cloud Filtering	45
3.4.4 Point Cloud Downsampling	46
3.4.5 Scan Registration.....	47
3.4.6 Total Station Measurements	47
3.4.7 Graph Optimization.....	49

3.4.8 Fusion of LiDAR Point Clouds with Camera Images	50
3.5 Data Description.....	51
3.5.1 Indoor SLAM Dataset	52
3.5.2 Multi-sensor Calibration Datasets	54
3.5.3 Indoor Segmentation Dataset	56
3.6 Data Analysis	57
3.6.1 Point Cloud and Image Calibration.	57
3.6.2 Semantic Segmentation	60
3.7 Chapter Summary	65
Chapter 4 Target-free LiDAR-Camera Automatic Calibration	67
4.1 Introduction	67
4.2 Point Cloud Calibration.....	68
4.2.1 Coarse Registration	69
4.2.2 Fine-grained Registration	71
4.3 Six-degree-of-freedom Pose Estimation.....	72
4.4 Target-free Automatic Calibration Based on Trajectory Constraints.....	74
4.4.1 Integration of LiDAR Trajectory and Camera Pose	75
4.4.2 Target-based Experiment.....	78
4.4.3 Trajectory-based Experiment	79
4.5 Chapter Summary	83
Chapter 5 Indoor LiDAR-SLAM and Point Cloud Semantic Segmentation.....	84
5.1 Data Acquisition and Coordinate System.....	85
5.2 Six-degree-of-freedom Pose Estimation.....	85

5.3 Loop-closure Estimation and Back-end Optimization Based on Sub-map	87
5.4 3D Point Cloud Semantic Segmentation	90
5.4.1 Dimensional Features	92
5.4.2 The Priori Structural Information Filter	93
5.5 Experimental Results and Analysis	94
5.5.1 Experimental Data and Environment	95
5.5.2 Point Cloud Quality Assessment	96
5.5.3 Evaluation of Semantic Modeling Results	98
5.6 Chapter Summary	103
Chapter 6 Conclusions	106
6.1 Contribution.....	106
6.1.1 WATiMMS Multi-Sensor Dataset.....	107
6.1.2 Target-free LiDAR-Camera Automatic Calibration.....	108
6.1.3 Semantic Segmentation with Optimized SLAM algorithm.....	108
6.2 Recommendations for Future Research.....	110
References	112

List of Figures

Figure 1.1 Thesis structure framework.....	9
Figure 2.1 Mobile scanning systems: (a) Viametris IMS3D, (b)Trimble SR-DLP6, (c)Flyability ELIOS 3, (d)Xiaomi Cyberdog, (e)GeoSLAM ZEB, (f)Waterloo Wat-MMP, (g) Waterloo PLS, and (h) Waterloo Wat-dog.....	13
Figure 3.1 WATiMMS dataset includes (a) large-scale point cloud, (b) trajectory routes, (c) indoor point cloud projection on the image, and (d) indoor point cloud scene segmentation.	33
Figure 3.2 Waterloo modular mobile platform for multi-sensor integration (Wat-MMP).....	36
Figure 3.3 (a) the architecture of the modular mobile platform, (b) Waterloo PLS, (c) Waterloo Wat-MMP, (d) the cart platform, and (e) Waterloo Wat-dog	37
Figure 3.4 Synchronized selection policy.....	38
Figure 3.5 Schematic diagram of the synchronization process.	39
Figure 3.6 Camera coordinate system $O - x - y - z$	40
Figure 3.7 Radial distortion effect.....	43
Figure 3.8 Underground parking portion dataset include: (a) point cloud for underground parking lot, (b) graph-optimized odometers, and (d) reference points on the frame extraction point cloud.	53
Figure 3.9 Position schematic. (a) The mobile platform in the underground parking lot, and (b) LiDAR SLAM point cloud result.	54
Figure 3.10 The calibration data. (a) The point cloud data for calibration, and (b) the image for calibration.....	56
Figure 3.11 Semantic segmentation classification in the dataset.	60
Figure 3.12 (a) Calibration results projection. (b) Translation, rotation, and reprojection error distribution.....	60

Figure 3.13. The training loss for PointNet, PointNet++, and RandLA-Net, respectively.	61
Figure 3.14. Normalized confusion matrices evaluated (a) PointNet, (b) PointNet++, and (c) RandLA-Net.	64
Figure 3.15 Point cloud semantic segmentation (a) Ground truth, (b) PointNet, (c) PointNet++, and (d) RandLA-Net.	65
Figure 4.1 Local feature desc coarse alignment framework.....	70
Figure 4.2 Basic flow of ICP algorithm	72
Figure 4.3 The LiDAR backpack system: (a) 3D LiDAR backpack system exterior, (b) backpack internal structure and sensor settings, and (c) LiDAR and camera external coordinate system.....	73
Figure 4.4. Visual-SLAM trajectory correction and optimization in accordance with LiDAR trajectory.....	75
Figure 4.5. The target-based calibration under (a) image, (b) LiDAR point clouds, and (c) image information registered to the point cloud.	79
Figure 4.6. Trajectory-base calibration results: (a) the LIO-SAM obtains LiDAR trajectory, (b) optimized scaled camera (blue) trajectory and LiDAR (red) trajectory, and (c) the camera (blue) trajectory and the LiDAR (red) trajectory obtained by the graph optimization method.	81
Figure 4.7 proposed method result (a) Images in the underground scene. (b) 10 s LiDAR point cloud. (c) Projection points to the photograph. (d) point cloud data with colour.....	82
Figure 5.1 Topology graph optimization model based on sub-map loop closure estimation.....	89
Figure 5.2 Semantic modeling algorithm flowchart: (1) prior information extraction module, (2) global information mining module, and (3) feature fusion module.....	92
Figure 5.3 Point cloud quality evaluation and analysis: (a) LOAM algorithm, (b) front-end results of SPC-SLAM, (c) closed-loop detection results of SPC-SLAM, and (d) optimization results of total station using reference points.	97
Figure 5.4 Point cloud quality (a) LOAM algorithm, (b) front-end results of SPC-SLAM, and (c) closed-loop detection and optimization results of SPC-SLAM.....	101
Figure 5.5 Semantic segmentation results. (a) The original point cloud, (b)the segmentation results using PointNet, and (c)the segmentation results obtained using the proposed method.....	102
Figure 5.6 The building flyover results. (a) Images of the building flyover. (b) LOAM, (c) PC-SLAM front-end, and (d) SPC-SLAM after optimization.....	103

List of Tables

Table 2.1	Parameter performance of several types of MLS	14
Table 3.1	Comparison of portable LiDAR devices	31
Table 3.2	Existing indoor modeling datasets comparison.	34
Table 3.3	WATiMMS content coverage	52
Table 3.4	Model performance (%) on the WATiMMS testing dataset.	62
Table 4.1	Calibration Errors of Two Methods	82
Table 5.1	Indoor data collected by our mobile LiDAR platform (Wat-MMP)	95
Table 5.2	Error analysis of map building results.....	98
Table 5.3	Six scenarios semantic segmentation results (%).....	100
Table 5.4	Four categories of semantic segmentation IoU results (%).....	101

List of Abbreviations

Acronyms	Description
ALS	Airborne Laser Scanning
AMN	Associative Markov Network
BIM	Building Information Modeling
BNNDR	Bidirectional Nearest Neighbour Distance Ratio
CNN	Convolutional neural networks
DOF	Degree of Freedom
DPM	Deformable Part Model
EKF	Extended Kalman filter
FoV	Fields of View
GNSS	Global Navigation Satellite Systems
ICP	The Iterative Closest Point
IMU	Inertial Measurement Unit
IoT	Internet of Things
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LIO	LiDAR Inertial Odometry
LOAM	LiDAR Odometry and Mapping
MAC	Media Access Control
MLS	Mobile Laser Scanning
MV3D	multi-vision 3D
NDR	Nearest Neighbour Distance Ratio
NN	Nearest Neighbour
PLS	Personal Laser Scanning

Acronyms	Description
PSIF	Prior Structure Information Filter
PTP	Precision Time Protocol
R-CNN	region-based convolutional neural network
RGBD	Red, Green, Blue-Depth
RMSE	Root mean square error
ROS	Robot Operating System
RPN	Regional Proposal Network
SLAM	Simultaneous Localization and Mapping
SOR	statistical outlier removal
SSD	Single Shot Detector
UDP	User Datagram Protocol

Chapter 1

Introduction

1.1 Background

Indoor environments have become increasingly important as people spend a significant amount of their time indoors. With the rapid pace of urbanization and the expansion of indoor spaces, there is a growing demand for precise indoor environment information acquisition and positioning services. This demand arises from various applications, including high-precision indoor navigation, building information modeling (BIM), and positioning services for the Internet of Things (IoT). Autonomous driving technology is one area where precise indoor information is crucial, which has seen significant advancements in recent years. Last-mile automatic parking services heavily rely on accurate locational and semantic information, especially in underground or indoor parking areas. To achieve precise navigation in such environments, vehicle-mounted sensors must analyze environmental data, and a pre-set 3D high-definition indoor semantic map is also needed. Meeting these demands has led to extensive research on theories and algorithms associated with 3D high-precision reconstruction, semantic segmentation, and object extraction in indoor environments. By addressing these challenges, it is possible to develop effective indoor navigation and mapping systems that cater to the needs of various applications and services.

1.1.1 Challenges for Indoor Navigation and Mapping

The vehicle-borne mobile laser scanning (MLS) systems or an airborne laser scanning (ALS) system have been extensively used for effective acquisition of geospatial data (e.g., point clouds, optical images) in outdoor environments. However, such systems rely on both the Global Navigation Satellite Systems (GNSS) and the Inertial Measurement Unit (IMU) for positioning and orientation to generate high-precision vehicle's trajectories to restore 3D point clouds. Once disconnected from the GNSS signals in the GNSS-denied environments (e.g., indoor or underground parking lots), such a MLS system may fail because it cannot find its exact position in the global coordinate system. Under the circumstance that human beings spend much of their time indoors, their demands for precise indoor environment information acquisition and positioning service become enormously urgent with the acceleration of urbanization and increasingly larger indoor space [1]. Such requirements are concerned with high-precision indoor navigation [2], building information modeling (BIM) [3], and positioning service for Internet of Things (IoT) [4], etc. Especially for autonomous driving technology that emerges over the past few years, last-mile automatic parking services associated with underground or indoor parking areas heavily rely on precise locational and semantic information. In this case, not only should environmental data analysis be made by virtue of vehicle-mounted sensors, but also a pre-set 3D high-definition indoor semantic map is needed for accurate navigation [5, 6]. Due to such demands, the most significant importance has been attached to research subjects about theories and algorithms which are associated with 3D high-precision reconstruction, semantic segmentation, and object extraction in an indoor environment.

1.1.2 Simultaneous Localization and Mapping

For high-precision 3D modeling in GNSS-denied environments, one of the most effective solutions to this problem is the use of LiDAR-based Simultaneous Localization and Mapping (SLAM) algorithms [7]. The SLAM algorithm can estimate the position and posture based on its own sensor information for positioning and navigation. At the same time, the estimated position information is used for the construction of the map in early studies, visual SLAM [8, 9] that is based on cameras or underwater cameras attracts the attention of scholars. Thanks to the camera has low hardware cost and stable performance in a texture-rich environment, visual SLAM is widely applied in various robotic systems [9,10]. In an indoor environment, however, cameras are generally vulnerable to illumination. And sometimes the relevant image frames only consist of sparse texture information such as white walls and ceilings. All these defects impair the stability of visual SLAM; Moreover, feature point clouds that are reconstructed based on a visual approach are so sparse as to fail in precisely restoring information required by 3D scenes.

Compared with visual SLAM, that is originated in robotics technology, LiDAR-based Simultaneous Localization and Mapping (LiDAR SLAM), independent of light conditions in an indoor environment, becomes a feasible approach that can be applied in indoor navigation and mapping services. Nevertheless, neither MLS nor ALS outdoor systems fail to scan an indoor scene because of the giant size flexibly. In recent years, Personal Laser Scanning (PLS) systems (e.g., hand-held, backpack, helmet) equipped with either low-cost single or dual LiDAR sensors integrated with panoramic digital cameras have demonstrated the potential to be applied in achieving precise navigation and positioning in multi-story shopping mall and an associated indoor, stacked car park. [11, 12, 13] They feature flexible, fast, and convenient data collection and a rapid and comprehensive environmental scanning

capacity. Advances in relevant hardware and concepts result in many gaps between theories and algorithms. Three significant challenges are summarized in this chapter to satisfy indoor application demands, including the automatic parking of self-driving vehicles.

Challenges from data resources. Indoor navigation and mapping algorithms often require multiple data resources for training and tunneling to achieve accuracy and robustness. However, unlike outdoor environments, indoor environments include underground parking lots and office environments in buildings, underground passageways, survival paths, and other complex and changing environments. Often, no unified and clear reference point or method can provide a global reference coordinate system like GNSS signals. A pioneering multi-sensor ensemble indoor dataset has been developed by Xiamen University [14]. However, this dataset has a single scenario due to device limitations and only provides path reference points collected by smartphones for mobile trajectories. In this case, there is an urgent need for a dataset with a uniform standard and a rich set of scenes combining sensor data. This dataset must satisfy the need for sensor data fusion for multiple scenes and provide annotations and reference point information based on scene information for subsequent training, computational checking, and comparison.

Challenge from data acquisition. In an indoor environment that is narrow and closed, it is impossible for traditional vehicle-mounted/airborne laser scanners to effectively acquire environmental information. Such a gap is filled by backpack laser scanner system, also known as LiDAR backpack. The first backpack prototype was developed in Video and Image Processing Lab of UC Berkeley in 2010 [14]. It is equipped with five 2D laser range finders (LRFs), one 6-DOF (degree of freedom) IMUs and two fisheye cameras. Afterwards, different equipment platforms are raised successively [11,12,13]. Featuring high flexibility and portability, etc., it can be used to collect and calculate data seamlessly in

all corners indoors. Considering that full coverage of 3D environment cannot be achieved through a single 16-lines LiDAR, data obtained by mobile scanning fails in completely restoring such a 3D environment. Moreover, it is easy to block the single LiDAR so as to affect subsequent algorithmic processing. In this context, multi-angle scanning of multi-sensor turns into a mainstream mechanism. Accordingly, automation, speed and calibration precision of multi-sensor are questions that should be urgently settled in the first place.

Challenges from accurate location estimation and high-level semantic information extraction.

GNSS signals cannot be acquired effectively in indoor or underground enclosed environments due to occlusion. Therefore, it is challenging to perform mobile mapping and semantic modeling based on navigation signals and positioning services provided by virtue of GNSS signals. Nowadays, many problems still need to be solved in time to make full use of the scanning information to accomplish high accuracy position estimation and high accuracy modeling. For example, visual SLAM-based localization methods [8,9] are vulnerable to insufficient illumination and texture, and the point clouds for mapping are quite sparse. In LiDAR-based mapping methods, there are still drawbacks such as large cumulative errors and poor robustness of feature extraction. Therefore, in terms of high-precision mapping of indoor environments, how to realize the fusion of sensor data from multi-camera-dual/multi-LiDAR systems to accomplish semantic segmentation, semantic information extraction and object detection is gradually becoming a new trend and challenge.

1.2 Research Objectives

Specifically for the three major challenges facing the theory and algorithms for 3D high-precision reconstruction, semantic segmentation and object extraction in indoor environments, the following three research objectives are proposed here.

The first is multi-sensor data set collection. For indoor environments, a multi-sensor dataset consisting of IMU data, LiDAR point clouds, and image data needs to have synchronized timestamps and a uniform datum for calibration. Unlike outdoor environments with ample lighting and complex 3D textures, indoor environments often consist of large, single-colour walls and smooth, flat roads. This poses a challenge for data collection and large-scale data integration. In addition, how to label and verify the accuracy of the collected data is also an issue to be considered.

Next is the calibration of multiple sensors. In a multi-sensor system, the data obtained by each sensor falls into their own local coordinate system. Here, the coordinates of all sensors must be unified and registered in one coordinate system in order to obtain data with consistent coordinates. The traditional sensor calibration methods are usually performed using camera-LiDAR systems, based on characteristic targets. The goal of calibration is to accurately align the coordinate systems of the LiDAR sensor and the camera, enabling the fusion of data from both sensors for perception and object detection tasks. In this method, a set of predefined targets or patterns are placed in the environment, such as checkerboard patterns or specially designed targets. These targets have known geometric properties and are detectable by both the LiDAR sensor and the camera. The calibration process involves capturing data from both sensors while the targets are visible, and then using this data to estimate the transformation matrix between the LiDAR and camera coordinate systems. This method heavily relies on the presence of detectable targets in the environment. In real-world scenarios where suitable targets are not available or visible, the method may not be applicable. The performance of the method can be affected by environmental conditions such as lighting, occlusions, and reflections, which may make it challenging to accurately detect and localize the targets. At the same time, the calibration process requires capturing data from both the LiDAR sensor and the camera while the targets are visible. This

process can be time-consuming, especially if multiple targets need to be captured from various angles and positions.

The last issue is about 3D point cloud semantic segmentation, which is crucial to the top-level application of point cloud data processing. For example, after solving the point cloud generation problem, how to carry out object detection for massive 3D point clouds. In terms of traditional semantic segmentation based on point clouds, the following problems exist: (1) point clouds do not contain rich texture information, which may lead to geometrically ambiguous judgments; (2) small objects in SLAM data usually contain large noise, so how to effectively fuse camera and LiDAR data to cope with the noise problem and further improve semantic segmentation accuracy becomes critical; (3) since the current depth model-based methods in most cases require a large amount of manual annotation data, so in practical applications, it is difficult to obtain large-scale 3D point clouds and perform manual annotation methods.

In this thesis, the following three main studies were conducted, which are the multi-sensor fusion point cloud dataset, the multi-sensor coordinate system calibration, and the point cloud semantic segmentation.

First, multi-sensor fusion point cloud dataset. To collect multi-sensor high-precision datasets, it is necessary to first build a LiDAR platform that can be applied in indoor environments. The platform can also be modularly transformed to adapt to different indoor environments. In this study, a multi-sensor mobile platform system is proposed, and a dataset for multiple environments is collected according to this system for subsequent use. The dataset also provides datum points that can be used simultaneously in images and point clouds, solving the problem of cross-category data comparison. On top of that, the dataset provides accurate global timestamps and complete annotations, which are able

to be used for simultaneous localization and mapping, 3D environment reconstruction, 3D objects detection, multi-sensor calibration, and other algorithm validation.

Second, target free multi-sensor calibration method. Data acquisition plays an important role in multi-sensor-based computation, and the final processing accuracy of related algorithms depends on the data quality. To address the problems of sparsity and manual labeling errors in multi-sensor systems based on external parameters for multi-line LiDAR calibration, this study proposes a target-free automatic calibration algorithm based on the geometric consistency constraints inherent in the data. Assuming that LiDAR point clouds and camera images can complete environmental measurements in a consistent manner, based on this, the proposed algorithm has the ability to iteratively solve external sensor parameters and finally complete online calibration of LiDAR and cameras automatically. The accuracy can be improved to a great extent by data-based fitting, and the corresponding process is not only automated, but also fast throughout.

Third, indoor environment LiDAR-SLAM and point cloud semantic segmentation. Based on the acquisition of high-quality raw data, this section proposes a probabilistic alignment and estimation framework based on generalized nearest-point iteration for the front-end odometer estimation and back-end optimization for the 3D SLAM problem of multi-line LiDAR. Using this framework and the quadratic matching strategy, the problem of excessive cumulative error of the front-end odometer is improved. Compared with the feature approach, this algorithm can effectively use the global point cloud information to improve the alignment accuracy. In the weak feature environment, this probabilistic framework can effectively solve challenging problems such as symmetry/feature sparsity. And the back-end sub-map matching strategy solves the closed-loop detection problem in 3D point clouds and improves the sub-map-based optimization framework. Experimental results show that comparing the

feature methods, the method proposed performs well in the symmetric environment of underground garage and long corridor, and outperforms the leading algorithms in both trajectory accuracy and point cloud reconstruction accuracy. For the semantic modeling problem, a semantic segmentation framework is proposed for the front and back ends. The idea of combining the front-end module PSIF based on indoor a priori structure information extraction and the back-end deep network PointNet of this module is proposed, which effectively improves the semantic segmentation accuracy of indoor 3D models and improves the training efficiency at the same time. Experiments show that the 3D point cloud semantic modeling method proposed in this chapter can meet the challenges of practical applications of mobile 3D scanning platforms in various indoor scenes.

1.3 Thesis Structure

This thesis consists of six chapters, which are described in detail as follows (see Figure 1.1).

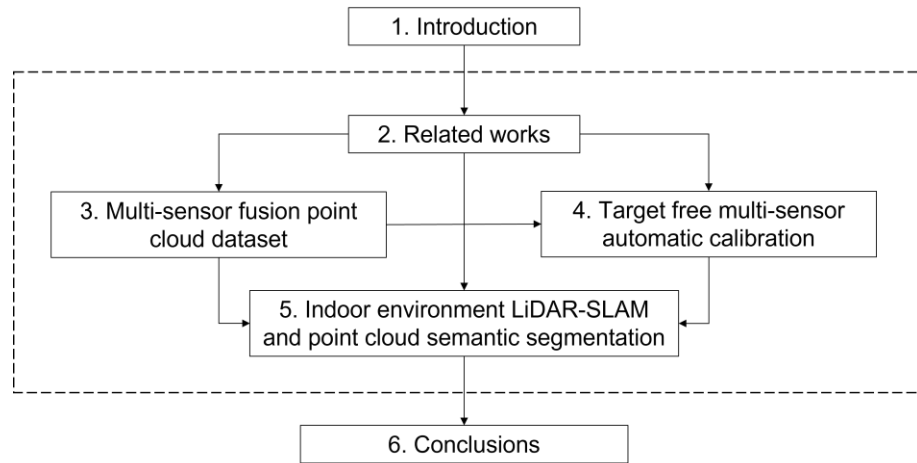


Figure 1.1 Thesis structure framework

In addition to introducing the research background of this topic, Chapter 1 provides an overview of various sensors, point cloud acquisition techniques and their advantages. And to be more specific, the main existing point cloud processing techniques are presented, while the main challenges in the field are also pointed out. In addition, the main research objectives and details of each chapter are presented at the end.

Chapter 2 describes the current state of research on multi-sensor fusion-related technologies, including the introduction of existing backpack laser scanner platforms, multi-line LiDAR calibration, 3D high-precision SLAM problems based on multi-line LiDAR, and 3D object detection with multi-sensor fusion implemented by combining cameras and LiDAR together.

Chapter 3 proposes a platform that can be used to collect multi-sensor data indoors for indoor environments, and modularly explores three multi-sensor robotic devices built on this platform. Then, based on this platform, a dataset for multiple indoor environments is proposed. And a method for detecting the quality of multi-sensor data in indoor environments is proposed. In the experimental section, the proposed use of different sensor data is compared and the application of the dataset to multiple algorithms is explored.

Chapter 4 explores the target-free automatic calibration algorithms for multi-line LiDAR and cameras. Firstly, an iterative calculation method of external parameters based on geometric consistency constraints is proposed; secondly, a consistency determination method is proposed to overcome the problem of symmetric environment registration failure so that external points can be eliminated, and the robustness of the algorithm can be improved. In the experimental study section, the accuracy of the proposed method is compared with the accuracy of manual calibration.

Chapter 5 investigates the high-precision 3D SLAM algorithm based on sub-map construction and the a priori structure information-based extracted target segmentation and semantic modeling algorithms. Firstly, a front-end odometry estimator based on a probabilistic framework is presented. Secondly, a complete optimization framework based on sub-maps is proposed for the sparse characteristics of point clouds. Finally, based on the SLAM modeling data, a 3D target-based a priori structure information extractor is proposed for indoor and underground garage scenarios. The model segmentation and semantic modeling accuracy of the back-end are improved, and experimental tests are conducted in the corridor and underground garage scenes, respectively. The experimental tests are conducted in flyover and underground parking lot scenes.

Chapter 6 presents a comprehensive overview of the entire thesis, while delving into the examination of 3D point cloud reconstruction and semantic models relevant to indoor environments and their associated applications. Some recommendations are made for future research.

Chapter 2

Background and Related Studies

This chapter provides the background of this study and a comprehensive literature review. Section 2.1 describes the mobile LiDAR platforms. Section 2.2 describes multi-line laser scanner coordinate system calibration. After that section 2.3 describes LiDAR and camera calibration method. Section 2.4 describes 3D high-precision SLAM based on LiDAR. And section 2.5 describes 3D scene segmentation. Finally, the entire chapter is concluded in Section 2.6.

2.1 Multi-line LiDAR and Mobile Scanning Systems

A multi-line Light Detection and Ranging (LiDAR) is mainly composed of several high-precision laser distance sensors, an optical-mechanical scanning or electronic scanning unit, and a system control and data output module. Interior construction and scanning principles of multi-line laser scanners are shown in Fig. 2.1. Different from a TLS or an MLS system, a multi-line laser scanner is interiorly equipped with multiple sets (as typical values are found in 16, 32 and 64 sets, they are also known as 16-line, 32-line or 64-line laser scanners respectively) of laser emission/reception modules which are arranged on the same vertical plane at a specific angle. In this way, such modules are able to synchronously trigger TOF ranging at different angles, so that synchronous vertical hold takes place. Such a group of measuring system is fastened on a horizontal rotating platform to perform scanning at 5~20Hz, thereby constituting a 3D multi-line laser scanner. For the past few years, high-frequency and multi-dimensional spatial awareness makes unprecedented development of unmanned systems

come true thanks to the progress in multi-line LiDAR technology. Additionally, mobile laser scanning systems (MLSs) that are based on such a sensor also make gradual progress under the circumstance that power consumption and portability of multi-line laser scanners are significantly improved. In Fig. 2.1, several types of mobile laser scanning devices are presented, including the form of trolley, handheld, backpack, robot dog, and drone-based system, as well as a modular-based mobile platform developed at the University of Waterloo. Moreover, Wat-MMP is a data acquisition platform that is independently developed by the author with an aim to validate the proposed algorithm. As for the basic parameters of diverse devices, they are given in Table. 2.1.

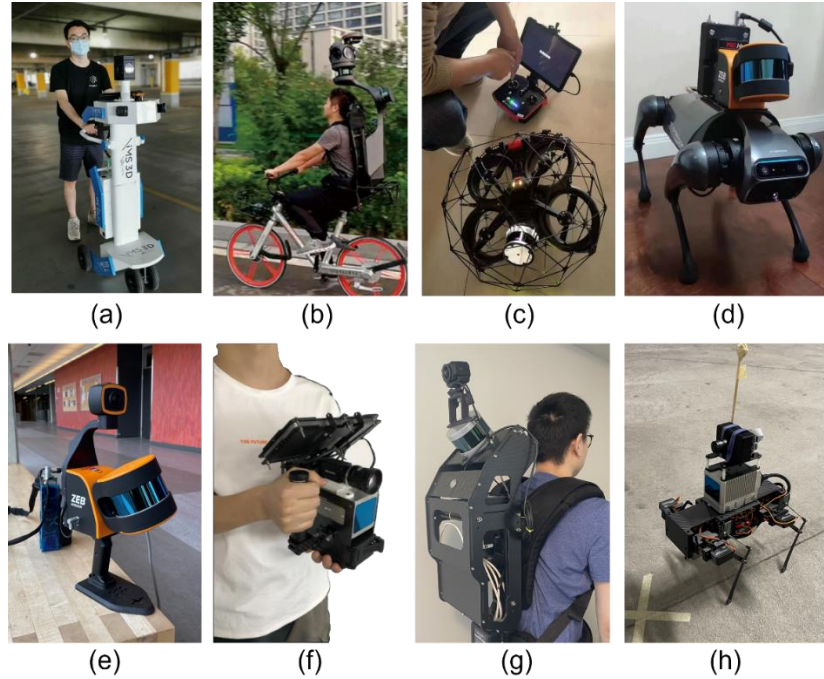


Figure 2.1 Mobile scanning systems: (a) Viamentris IMS3D, (b)Trimble SR-DLP6, (c)Flyability ELIOS 3, (d)Xiaomi Cyberdog, (e)GeoSLAM ZEB, (f)Waterloo Wat-MMP, (g) Waterloo PLS, and (h) Waterloo Wat-dog.

Table 2.1 Parameter performance of several types of MLS

Name	Trimble	Flyability	ViAmetris	Leica	GeoSLAM ZEB	Wat-MMP
Positioning accuracy (cm)	5	3	<10	<10	0.6	<2
Scanning range (m)	100	100	100	200	100	260
Digital camera	360°	monocular	360°	360°	monocular	monocular
IMU module	Not included	included	Not included	included	included	included
LiDAR sensor	Velodyne VLP-16	Ouster OS0-32	Hokuyo UTM-30LX	Velodyne VLP-16	Velodyne VLP-16	Livox-Horizon
Weight (kg)	12	2.5	45+	30	2.9	3.5

2.2 Multi-line Laser Scanner Coordinate System Calibration

According to the number of 3D points clouds to be aligned, the point cloud rigid alignment task can be divided into Multi-view Registration and Pairwise Registration. Although both cloud alignments are point cloud alignment tasks, the core of the problem is not the same. Although both Multi-view Registration and Pairwise Registration are point cloud alignment tasks, the nature of the problem is not the same. Pairwise registration means that two overlapping 3D point clouds are aligned based on the information of the overlapping regions of the two point clouds at a time, while the core of multi-view alignment is to use as many overlapping areas as possible (considered as redundant observations) to simultaneously align multiple partially overlapping 3D point clouds through joint optimization. As an essential branch of the point cloud alignment task, a series of studies have been conducted by researchers in recent years on this problem in both traditional and deep learning-based methods [46,47]

Conventionally, multi-sensor calibration is commonly fulfilled by introducing particular calibration targets into a common view region of multiple sensors [47]. As a standard tool of LiDAR or camera

calibration, targets with particular geometrical shapes or patterns play an essential role in the external parameter calibration of LiDAR. According to Atanacio et al. [48] and Muhammad et al. [49], a multi-line laser scanner self-calibration method is proposed based on feature constraints. A matching algorithm that is based on multiple types of geometrical characteristics such as angular points, lines, and planes is utilized by He et al. [50] to deal with paired LiDAR point cloud calibration problems. Although these methods play a certain part in improving multi-sensor calibration and accuracy, the following defects still exist. First, special targets with special geometrical characteristics or textural features should be designed by researchers to fulfill calibration. Second, it is challenging to identify, from sparse point clouds, features of corresponding target points or angular points in a common view region of sensors during multi-line laser scanner calibration. Especially in the manual selection, it is hard to directly acquire corresponding points, and a common practice is constructing auxiliary lines by means of line fitting or plane fitting so that they can be selected at a point where lines meet. However, such a common practice may give rise to new fitting errors and calculation errors. To overcome the above problems, it is suggested that an energy function-based automatic calibration algorithm previously used for multi-beam laser scanners should be available for the internal parameter calibration of LiDAR. Nevertheless, this algorithm is still infeasible in external parameter calculation subjected to a significant pose error considering that it is an algorithm based on energy functions and involved with a local optimization problem. Moreover, it is also much more likely to generate a great error when used in multi-LiDAR external parameter calibration because of local optimization.

2.3 Camera and LiDAR Calibration

Camera and LiDAR calibration is a critical step in achieving accurate sensor fusion and perception in autonomous vehicles. By aligning the coordinate systems of the camera and LiDAR sensor, the data

from both sensors can be effectively combined to provide a comprehensive understanding of the environment. This section will discuss the importance of camera and LiDAR calibration and various approaches used in this process. Camera and LiDAR calibration aims to estimate the extrinsic parameters (such as translation and rotation) and intrinsic parameters (such as focal length and distortion coefficients) that define the relationship between the camera and LiDAR coordinate systems. The extrinsic parameters describe the spatial alignment between the two sensors, while the intrinsic parameters characterize the internal properties of each sensor. Accurate calibration is crucial because the camera and LiDAR sensor capture data from different perspectives and have inherent differences in their measurement principles. Failure to calibrate correctly can lead to misalignment, resulting in object detection, localization, and tracking errors. Several methods have been developed for camera and LiDAR calibration, including:

2.3.1 Target-based Methods

Related work in camera and LiDAR calibration has extensively explored target-based methods. These methods utilize specially designed targets with known geometric properties to facilitate calibration. Researchers have significantly contributed to advancing target-based camera and LiDAR calibration techniques, improving their accuracy and applicability. One prominent area of research focuses on target design and detection. Various targets have been proposed, such as checkerboard patterns, circular grids, and specially constructed targets with specific geometrical characteristics. Zhang et al. [52] introduced a flexible technique for camera calibration using a planar calibration pattern. This method, estimates the intrinsic and extrinsic parameters of the camera by analyzing the target's feature points. To detect targets, image processing algorithms are employed to locate the target patterns in the camera images. These algorithms can involve techniques like corner detection, edge detection, or template

matching. Once the targets are detected, correspondences between the camera and LiDAR measurements are established based on the overlapping regions in both datasets. The calibration process typically involves capturing data from the camera and LiDAR sensor while the targets are visible in the scene. The acquired data is then used to estimate the transformation matrix that aligns the camera and LiDAR sensor coordinate systems. The calibration parameters, including extrinsic (translation and rotation) and intrinsic (focal length, distortion coefficients, etc.), are computed to establish a precise geometric relationship between the two sensors.

Several studies have addressed the challenges associated with target-based calibration. For example, some researchers have investigated robust target detection algorithms that can handle challenging conditions like varying lighting, occlusions, or partial visibility. Improved corner detection algorithms, such as the Harris corner detector or the Shi-Tomasi corner detector, have been utilized to enhance the accuracy and reliability of target detection [52,58]. Furthermore, efforts have been made to automate the calibration process by developing algorithms that automatically detect and match the target patterns without requiring manual intervention. These algorithms utilize geometric constraints and matching techniques to establish correspondence between the camera and LiDAR data [104,111]. The accuracy of target-based calibration has been a key focus, leading to studies on error analysis and refinement techniques. Some researchers have explored methods to reduce systematic errors caused by lens distortion, nonlinearities, and other factors [52, 56]; Zhang et al. [52] introduced error analysis techniques for camera calibration, providing insights into the sources of calibration errors and suggesting methods for error compensation.

The advancements in target-based camera and LiDAR calibration have contributed to the accurate fusion of data from both sensors in various applications, such as autonomous driving, robotics, and

augmented reality. These methods have been successfully applied in real-world scenarios, demonstrating their effectiveness in improving the perception and spatial understanding capabilities of autonomous systems.

While target-based methods offer significant advantages, they also have limitations. The reliance on detectable targets restricts their applicability in scenarios where suitable targets are not available or visible. Environmental factors, such as challenging lighting conditions or occlusions, can affect target detection accuracy. Additionally, manual intervention in placing and removing targets can introduce errors and labor-intensive efforts.

2.3.2 Feature-based Methods

Feature-based camera and LiDAR calibration methods have been extensively explored in the literature. These methods focus on identifying common features, such as corners or edges, in the camera and LiDAR data to establish correspondence and compute the calibration parameters.

One area of research in feature-based calibration is the development of robust feature detection algorithms. Various algorithms have been proposed to detect corners or distinctive features in the camera and LiDAR data. For example, Yang and Clausi [62] proposed a planar checkerboard corner detection algorithm based on multiscale Harris corner response. This algorithm effectively detects corners in planar targets and can be used in feature-based calibration. Another significant feature-based calibration aspect is the features matching between the camera and LiDAR data. Matching algorithms aim to establish correspondence between the detected features in the two sensor modalities. Zhang et al. [58] proposed a robust technique for matching two uncalibrated images by recovering the unknown polar geometry. This technique effectively establishes correspondence and plays a crucial role in feature-based calibration.

Once correspondences are established, feature-based calibration methods utilize geometric constraints and optimization techniques to estimate the calibration parameters. The geometric constraints include polar constraints, reprojection constraints, and geometrical relationships between the camera and LiDAR data. Optimization-based methods, such as bundle adjustment, can refine the estimated parameters and improve calibration accuracy. Geiger [59] presented a feature-based calibration approach for camera and LiDAR systems in the context of autonomous driving. Their method utilized the KITTI vision benchmark suite and focused on robustly estimating the relative pose between the camera and LiDAR sensors.

Feature-based calibration methods offer flexibility in handling dynamic environments and various sensor configurations. They can adapt to different scenarios without relying on specific targets. However, accurate feature detection and matching are crucial for the success of these methods. Shi and Tomasi [63] proposed the "good features to track" algorithm, which has been widely used in feature-based calibration due to its effectiveness in identifying salient corners. It is important to note that feature-based calibration methods may have limitations in achieving high accuracy compared to target-based methods, especially in complex scenes with limited distinctive features. However, they offer robustness and adaptability, making them suitable for a wide range of applications.

2.3.3 Optimization-based Methods

Optimization-based camera and LiDAR calibration methods provide a flexible framework for calibrating the intrinsic and extrinsic parameters by formulating the calibration problem as an optimization task. These methods aim to refine the initial parameter estimates through an iterative optimization process.

One area of research focuses on formulating the calibration problem as a bundle adjustment task. Bundle adjustment simultaneously estimates the camera and LiDAR parameters by minimizing the reprojection error between the 2D image points and their corresponding 3D LiDAR points. This approach has been widely used in calibration scenarios where a large number of correspondences are available. Sun et al. [65] proposed an automatic extrinsic calibration method for camera-LiDAR systems using geometric features, which utilized an optimization framework based on bundle adjustment to refine the calibration parameters. To solve the optimization problem, different optimization techniques have been employed. Zhang et al. [52] introduced an optimization framework for camera calibration that utilizes the Levenberg-Marquardt algorithm to iteratively refine the calibration parameters. This approach considers both the reprojection error and the constraint on the calibration matrix. Optimization algorithms like Gauss-Newton or stochastic optimization methods can also be used to solve the calibration problem.

Another aspect of optimization-based calibration methods is the incorporation of additional constraints or prior knowledge to improve the accuracy and robustness of the calibration. For example, Wang et al. [61] proposed an optimization framework for LiDAR-camera calibration with unknown extrinsic. Their method included constraints on the rotation matrix and used additional geometric constraints to enhance the calibration accuracy. In addition to the calibration parameters, optimization-based methods can also estimate other parameters like lens distortion coefficients and sensor-specific parameters. By jointly estimating these parameters, the overall calibration accuracy can be improved.

Optimization-based calibration methods offer the advantage of flexibility, as they can handle complex geometric constraints and incorporate additional prior knowledge into the calibration process. However, they can be computationally intensive, particularly with large datasets or complex models.

The convergence and accuracy of these methods can also be sensitive to the initial parameter estimation, and there is a risk of converging to local optima, leading to suboptimal calibration results. Despite these challenges, optimization-based calibration methods have been successfully applied in various applications, including robotics, autonomous driving, and augmented reality, where precise sensor fusion and accurate spatial understanding are crucial.

2.3.4 Deep Learning-based Methods

Deep learning-based camera and LiDAR calibration methods leverage the power of neural networks to learn the intrinsic and extrinsic parameters directly from the data. These methods aim to automatically discover the underlying patterns and relationships between the camera and LiDAR measurements, enabling accurate calibration without relying on explicit geometric models.

One area of research focuses on deep learning architectures for joint camera and LiDAR calibration. Convolutional neural networks (CNNs) have been widely used to extract features from camera images, while LiDAR point cloud data can be processed using various techniques such as voxelization or point-based networks. The extracted features are then fed into a neural network that predicts the calibration parameters. Ahmed et al. [67] proposed a deep learning-based method for camera and LiDAR calibration that utilized a CNN to learn the mapping between the camera images and LiDAR point clouds. The network was trained on a large dataset with ground truth calibration parameters, enabling accurate parameter estimation.

Another aspect of deep learning-based calibration methods is generating synthetic data for training purposes. Synthetic data can be generated with known ground truth calibration parameters, allowing the neural network to learn the calibration mapping effectively. Dong et al. [68] proposed a synthetic data generation framework for Camera-LiDAR calibration, where a virtual LiDAR sensor model was

combined with a virtual camera model to generate paired data for training a deep neural network. In addition to direct parameter estimation, deep learning-based methods can also be used for feature extraction and correspondence matching. These methods leverage the representation learning capabilities of deep neural networks to extract discriminative features and establish correspondence between the camera and LiDAR data. This, in turn, can aid in the calibration process.

Deep learning-based camera and LiDAR calibration methods offer several advantages, including their ability to handle complex data and model highly nonlinear relationships between the camera and LiDAR measurements. They can also adapt to different sensor configurations and imaging modalities. However, these methods require large amounts of annotated training data to achieve good performance, and the calibration quality heavily relies on the quality and diversity of the training dataset. Despite these challenges, deep learning-based calibration methods have shown promising results in various applications, including autonomous driving, robotics, and augmented reality. They have the potential to enhance the accuracy and efficiency of the camera and LiDAR calibration in real-world scenarios.

Each method has its advantages and limitations, and the choice depends on factors such as the available sensor configurations, computational requirements, and desired accuracy.

Camera and LiDAR calibration is an active area of research, and ongoing advancements aim to improve the accuracy, robustness, and efficiency of calibration techniques. Moreover, the integration of additional sensors, such as radar or IMU, into the calibration process further enhances the perception capabilities of autonomous vehicles.

2.4 3D High-precision LiDAR-based SLAM

In early studies, two classical algorithms used for SLAM pose estimations were Kalman Filter [52] and Particle Filter [53]. Both EKF and its modified filter are selected for SLAM by Choset et al. [10]. In recent years, various advanced algorithms and their applications have been proposed by many researchers for various sensors as high-performance 3D sensors emerge, including RGBD-SLAM [9], LiDAR-based SLAM [22], and other methods [56,57]. Among them, loop closure estimation [25,58] and back-end optimization [59,60] have become the mainstream frameworks of SLAM. Based on the SLAM algorithm based on Velodyne data, a front-end odometer algorithm is raised by Zhang et al. [21] on the basis of feature selection. It is an algorithm with the highest positioning accuracy among KITTI benchmarking test sets in 2021 [61]. As of May 2023, the best record on the KITTI benchmarking test sets was achieved by SOFT2, a visual-SLAM method based on a stereo camera [62].

In terms of hardware, the first LiDAR backpack prototype was developed in Video and Image Processing Lab of UC Berkeley in 2010. It is equipped with five 2D laser range finders (LRFs), one 6-DOF (6 degrees of freedom) IMUs and two fisheye cameras [14]. In 2013, an “Akhka” LiDAR knapsack system developed in Excellence Laser Scanning Research Centre by Geological Survey of Finland was not only provided with a FARO Photon-120 laser scanner, but also integrated with a GNSS-IMU navigation system and a data recording computer [11]. In University of Wuerzburg (Germany), a LiDAR backpack system with a SICK LMS 100 laser scanner and a Riegl VZ-400 3D high-end laser scanner were presented by relevant researchers [12,13]. During this study, the 6-DOF-SLAM algorithm was utilized to construct a 3D map. Researchers in Swedish University of Agricultural Sciences investigate and develop a backpack equipped with Velodyne VLP-16, a GPS positioning system and a data logging computer [62] to carry out forest resource surveys. Such a system fulfills

positioning depending on commercial stereo vision solutions. Besides, there are also some LiDAR backpack systems for commercial purposes, such as Leica Pegasus backpack [54] which contains two Velodyne VLP-16 laser radars. A LiDAR system equipped with Velodyne HDL-32 and developed by Glenie et al. [63] makes use of GPS/GNSS and IMU to estimate poses accurately. Three 2D laser scanners are selected in the Laboratory of Sensing and Computing for Smart City (SCSC) in Xiamen University to construct a LiDAR backpack [64]. With the help of extended Kalman filter (EKF) based 2D laser scanners and inertial sensor data fusion, the backpack fulfills positioning and map construction by means of 6-DOF pose tracking.

2.5 Camera-LiDAR Fusion for 3D Scene Segmentation

2.5.1 2D Object Detection/Labeling

Conventionally, 2D object detection is performed based on manually designed feature descriptors. In combination with multi-scale features, a Viola-Jones detector is utilized by Viola et al. [65] to match similarities of 2D objects based on sliding window detection. A histogram of oriented gradients is used as the feature descriptor of the 2D object detector by Dalal and Triggs [66], which is of great significance in pedestrian detection. As for a Deformable Part Model (DPM) developed by Felzenszwalb et al. [67], object detection is carried out using manually designed features and a divide-and-conquer strategy based on packages, so detection speed and accuracy are tremendously elevated. Object feature descriptiveness of the existing methods is still far from satisfactory, although acceptable results can be traditionally obtained. For the past few years, a region-based convolutional neural network (R-CNN) and a Faster R-CNN have been developed by Girshick et al. [68] to carry out object detection in deep learning. In [69], the average object detection precision mAP) is further improved. However, the computational efficiency of the above framework is relatively low due to network

architecture limitations. To settle the above problem, Liu et al. [70] put forward a Single Shot Detector (SSD), and thus, a convolution kernel can be adopted on a feature map to predict class scores and offset, of a series of candidate's detection boxes. Redmon et al. [71] propose a YOLO detector and R-CNN-based YOLOv3 [72]. All these investigations use the feature explanatory power of the deep neural network to enhance the efficiency of the object detection framework and accelerate relevant calculations.

2.5.2 3D Object Detection/Labeling

Among point clouds-based 3D object detection studies, Munoz et al. [73] comes up with a function gradient algorithm to learn Associative Markov Network (AMN) model (raised by Taskar et al. [74]); and this algorithm is applied in 3D point cloud semantic segmentation. Based on R-CNN, Gupta et al. [75] detect and segment 3D objects from different angles. Chen et al. [76] presents a new method based on which different energy equations are provided for different features of a 3D object and these equations are also correspondingly optimized to acquire segmentation results. In addition to a sliding shape detection method based on 3D CNN deep networks, Song et al. [77] also presents a multi-scale 3D Regional Proposal Network (RPN) framework which can detect small objects from multiple scales. Although the above methods can generate high detection efficiency in an ideal environment, they fail to overcome challenges raised by noise and shelters, etc. in a practical condition. After elaborating on these problems, Deng and Latecki [78] design a proper 3D model function expression specific for RGB-D data on one hand; and, on the other hand, a regional proposal algorithm of a bounding box model and a faster R-CNN regression algorithm are also adopted by them to solve problems caused by noise and shelters to a certain extent. In terms of the PointNet model presented by Qi et al. [79], symmetric functions are used to integrate global information and further overcome the problem of point expression

disorder. From single point feature extraction based on MLP to global feature integration by means of Max Pooling, brand-new point cloud feature expressions are eventually obtained.

2.5.3 2D/3D Joint Estimation

3D point clouds are usually figured out by the SLAM algorithm. Considering that point clouds subjected to this algorithm commonly have some problems such as sparsity, noise and ambiguity, a 3D feature descriptor hardly works well in this case. It is much more likely to fail if 3D object detection is conducted from the perspective of 3D data with noise. Contrary to methods that take advantage of 3D feature descriptors, the present 2D object detection approaches perform rather well thanks to more powerful CNN models (meticulously designed and pre-trained by ImageNet) and more intensive and complete image data used here [77]. Therefore, a combination of 2D image data and 3D point cloud data is becoming a current research tendency. A 3D object detection framework based on a multi-vision 3D (MV3D) network has been developed by Chen et al. [80] to carry out object detection in a context of multi-vision/multi-sensor fusion. As regards Ku et al. [81], LiDAR point clouds and RGB images are combined together to generate features; afterwards, 3D candidate boxes are formed for object detection by virtue of different character patterns; then, these features are applied in RPN and a Phase II object detection network to predict object range and object pose direction; in the end, the 3D object is detected. In another study similar to this paper, a view frustum-based method [82] is put forward for object detection. According to this method, the 2D object detection approach is firstly used to project, by virtue of a view frustum, the detected region in an image perspective to acquire single frame point clouds related to the 3D object; subsequently, PointNet is selected to segment and extract point clouds in a view frustum search box. In this way, the problem caused by shelters is effectively resolved so that

not only can large-scale search of target point clouds be significantly reduced, but object detection efficiency can be improved.

Analysis of previous studies indicates that 3D object detection, instead of depending on a traditional method of manual feature descriptor design, has been developed into a process based on deep learning. Through literature search, however, it is found that there is only a limited number of investigations that can settle the problem of performing 3D object detection based on point cloud data featuring sparsity, strong noise and ambiguity, which is especially the case in a high-dynamic SLAM environment. Inspired by the view frustum based PointNet [82], a probabilistic position prediction and detection algorithm is designed here based on SLAM trajectories and projection for 3D objects. In this framework, CNN should be selected to execute object detection in a 2D image and probabilistic mesh projection is also adopted to locate and detect objects in 3D point clouds. In addition, the proposed framework also predicts object positions and object correlation in 3D space from the information of previous and post frames, which significantly enhances detection algorithm stability but lowers computation complexity.

2.6 Chapter Summary

Several key problems in the field of multi-sensor fusion and computing are comprehensively elaborated in this chapter; and background introduction and analytical investigation are also conducted in detail. To be specific, this chapter involves four major parts: multi-line laser radar coordinate system calibration, 3D high-precision LiDAR-based SLAM, and Camera-LiDAR fusion-based 3D object detection. Except for the research status of such critical issues, problems existing in currently advanced algorithms together with challenges with which solutions in an actual scene are confronted are also analyzed meticulously.

First, foundations for multi-sensor applications are concerned with the following aspect of sensor calibration. The typical scheme of manually designing targets and selecting corresponding points can lead to high errors and performs poorly. The automatic calibration method only applies to internal parameter calibration for sensors and cannot be used to calibrate external parameters of wildly varying coordinates. Secondly, as for 3D high-precision LiDAR-based SLAM, the current algorithms fail to solve the problem of point cloud sparsity effectively; and the feature-based method can quickly get stuck in local optimization in a single environment. Finally, these algorithms can still not cope with problems, such as noise, shelters, and geometrical ambiguity related to point clouds as far as multi-sensor object detection is concerned.

Chapter 3

WATiMMS: Indoor Multi-Sensor Dataset for Mapping and Positioning

This chapter focuses on the multi-sensor integration of indoor environmental datasets. Due to the limitations of GNSS-denied indoor environments, there is a need to provide a set of hardware platforms that can be modularly migrated with different environments. In Section 3.1, existing mature datasets and their limitations are discussed. Section 3.2 proposes an overall modular mobile platform, and a modular variant of this mobile platform is analyzed. In Section 3.3, the proposed mobile platform is mapped out regarding the features that should be presented in the collected dataset and the detection scheme, according to which the dataset and the experimental detection are completed. Section 3.4 describes the calibration method for the camera and LiDAR in the system and the results achieved using this method. Using the above methods with the collected datasets, a series of experiments and validations were conducted in Section 3.5 to demonstrate the dataset results and assess the data quality. Finally, the corresponding chapter summary is made in Section 3.6.

3.1 Introduction

3D reconstruction, navigation, trajectory tracking, and object recognition of indoor environment provide more possibilities for human-computer interaction and geographic information services [101], whether it is traffic planning [15], emergency rescue [82], logistics services, or medical care [84], all need indoor location information as the basis in the future. Therefore, a suitable indoor data model and

a guiding indoor environment algorithm become the key to solving the problems related to indoor 3D environments. However, indoor environments have a more complex layout structure compared to outdoor environments. Moreover, limited light conditions and blocking coverage are common problems. Especially in underground parking lots, tunnels, escape routes, and other weakly textured areas (areas where 3D textures are lacking and 2D colours are monotonous), sensor data collection causes great difficulties [91]. In addition, outdoor data collection equipment is too costly, large in form, and too heavy to be used in indoor environments.

As shown in Table 3.1, three types of portable LiDAR devices are used in indoor scenarios. For each type, we have selected one product as a representative to make a comparison: the first is the use of vehicle-mounted or UAV-mounted devices migrated to portable mobile platforms (mainly in the form of backpacks). The advantage of such devices is their high accuracy, but the disadvantages are their high cost and heavy weight. The second is using mainstream outdoor LiDAR and other sensors represented by Velodyne. LiDAR mounted on a cart, backpack, and other mobile platform equipment. This type of equipment has a lower price than surveying-grade LiDAR and has relatively high-quality data accuracy. However, the disadvantage is the low robustness of these LiDAR sensors and the low scanning range for indoor environments, which has a significant disadvantage for factories, warehouses, and even narrower tunnels. The third uses non-repetitive scanning LiDAR at lower cost than mainstream outdoor LiDAR, using its small and stable characteristics, which can be achieved by mounting on handheld devices, small robots (robot dogs), carts, or backpacks. Compared to the previous two devices, this type of LiDAR has a clear price advantage and is lighter weight; compared to Velodyne also has better robustness and longer scanning distance. But the disadvantage is also more

apparent: this kind of LiDAR display view is limited. In the case of the same distance, the point cloud density per unit area is also relatively low.

Table 3.1 Comparison of portable LiDAR devices

Equipment name	6T3 RIEGL Backpack	GreenValley LiBackpack DGC50	Wat-MMP (Ours)
LiDAR Sensor Models	RIEGL miniVUX	Velodyne Puck	Livox Horizon
LiDAR Sensor Price (USD)	60,000	4,000	999
Equipment weight (kg)	>10	8.6	3.2 (without RTK)
Other sensors	RTK, IMU	RTK	RTK , IMU, Camera
Point density (pts/m ²) at 3m/s	500	2000	350
Scanner range (m)	500	100	260
Range accuracy (cm)	1	5-10	2

Establishing a low-cost class migration data collection platform and developing a set of rules to use the platform can be flexible data collection modes have become an urgent challenge to solve. Therefore, the goal of this section is not simply to propose a data collection device and a data set per se. Instead, it is to propose an innovative solution to the problem of data collection using a portable, modular platform. It also develops an open indoor dataset that different research groups can freely supplement in the future.

Object recognition, map reconstruction, and road planning in GNSS-denied environments (e.g., indoor, or underground) are increasing research topics for fully automated and robotic autonomous driving [86]. The characteristics of indoor or underground environments, such as insufficient lighting conditions, missing 3D textures, and weak or inaccessible GNSS signals, also pose challenges for indoor localization and mapping tasks [91]. Accordingly, there are various approaches to solving the above problems, and the existing approaches are divided into three main categories: geometry

modeling, semantic modeling, and topology modeling [89], all of which have in common the need for a complete and valid indoor dataset as a data source. Table 3.2 shows some of the current representative datasets. Indoor scene datasets are usually data scanned from different angles of a room. Most of the datasets have a moderate amount of point clouds per frame with high density and uniform distribution of point clouds. However, such scenes have the following problems: (1) more planar structures, leading to more false matches and less efficient methods based on point matching, (2) more self-obscuring severe phenomenon, which leads to serious missing of some local structures, and (3) average data quality and more noise, leading to local structures being corrupted. Thus, to tackle these issues, we present an indoor multi-sensor dataset collected using modular mobile devices in this section. This dataset can be used for multi-sensor calibration, SLAM algorithm detection, scene segmentation, and other algorithms that need to validate indoor localization and 3D reconstruction. Figure 3.1 shows some of the characteristics of this dataset.

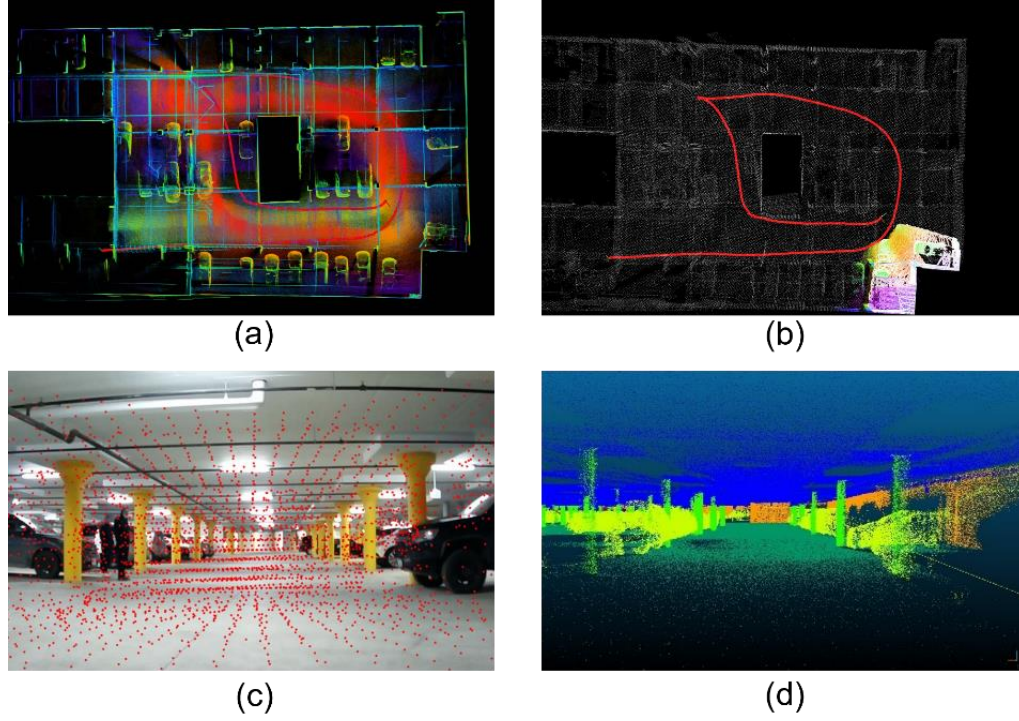


Figure 3.1 WATiMMS dataset includes (a) large-scale point cloud, (b) trajectory routes, (c) indoor point cloud projection on the image, and (d) indoor point cloud scene segmentation.

Table 3.2 summarizes the current mainstream indoor datasets. As shown in the table, most of the datasets are based on RGB-D cameras or single LiDAR sensors and function with a certain reference control as the destination. For example, ICL-NUIM [87], TUM [95] and EuRoc [83] provide the basis for 3D reconstruction and trajectory localization, but do not label the objects in space, so they cannot be used for object recognition and scene segmentation. SUN RGB-D [94], ScanNet [85], and NYU Depth [92,93] are labeled for 3D data, but since they are mainly used for object recognition and scene segmentation, they do not provide benchmarks for sensor trajectory determination and indoor frame line extraction. MiMAP [14], which is currently a more compatible dataset, can perform object recognition, scene segmentation and frame line extraction for a series of tasks of validation, but its sensors lack the use of IMU and only use smartphones to provide trajectory routes, which makes the

accuracy of trajectory routes low. In this section, we propose a method for collecting indoor data using a modular mobile platform, and we collect and organize the dataset according to this method. This dataset can compensate for existing datasets' deficiencies and is suitable for novel algorithm validation.

Table 3.2 Existing indoor modeling datasets comparison.

Dataset	Type	Sensor raw data	Benchmark data	Annotation	Size (GB)
ICL-NUIM	Geometric	RGB-D, IMU	Trajectory, Frame	N/A	Living Room 9.9 Office 8.7
TUM	Geometric	RGB-D	Trajectory	N/A	Indoor 18.89 Calibration 12.7
EuRoc	Geometric	Camera, IMU	Trajectory	N/A	Indoor 18.5 Calibration 3.6
MiMAP	Geometric, Semantic	LiDAR, Camera	Trajectory, Frame	2D, 3D	4.7
ISPRS Benchmark	Geometric	LiDAR	Trajectory, Frame	N/A	3.7
SUN RGB-D	Semantic	RGB-D	N/A	2D, 3D	6.4
ScanNet	Semantic	RGB-D	N/A	3D	Video 1.3TB
NYU Depth	Semantic	RGB-D	N/A	3D	Raw Image 428
Stanford 2D-3D Semantic	Geometric, Semantic	RGB-D	Frame	2D, 3D	766
Matterport3D	Geometric, Semantic	RGB-D	Frame	2D,3D	>2.2 TB
Ours	Geometric, Semantic	LiDAR, IMU, Camera	Reference point, Frame, Segment	2D, 3D	Indoor 40.9 Calibration 1.82

The dataset makes innovative use of a modular mobile platform for data collection. The platform is highly replicable and has been validated in different scenarios migrating to backpack, cart, handheld

and robot dog forms. The modular mobile platform and the various application forms are described below.

3.2 Data Collection Platform Setup

Given the characteristics of various sensors and devices presented in the previous section, we believe using the cheaper Livox and Velodyne LiDARs in indoor environments, especially the non-repetitive scanning LiDAR (Livox Horizon), is more reasonable. Because of the variety of indoor scenarios, a single mobile device (LiDAR cart or backpack) is unsuitable for all scenarios. For example, a LiDAR backpack form factor will not work in relatively narrow scenes such as underground walkways or office corridors that lack 3D textures. It should be replaced with a combination of LiDAR, camera, IMU sensors, and a cart form to collect the desired data smoothly. We believe that a modular, lower-cost, mobile platform that can integrate multiple sensors is significantly more suitable for indoor scenarios than the commercial indoor/outdoor collection devices on the market today. The problem of sparse point cloud density and small viewing angles of LiDAR like Livox Horizon can be solved by using multiple sensors and repeatedly scanning the same area. At the same time, data upsampling and artificial intelligence algorithms can also be used to increase the point cloud density to solve this problem in future work.

On this basis, we propose a modular indoor scanning system with the multi-sensor combination. Figure 3.2 shows this section's modular multi-sensor system platform for collecting indoor data. This platform can add or remove sensors according to the changes of the carrier. This system platform includes a non-repetitive scanning 32-line equivalent 3D LiDAR with a horizontal field of view (FoV) of 81.7° and a vertical field of view of 25.1° . The point cloud $P(x, y, z)$ coordinate system acquired by the LiDAR is the same as the local coordinate system of the LiDAR itself.

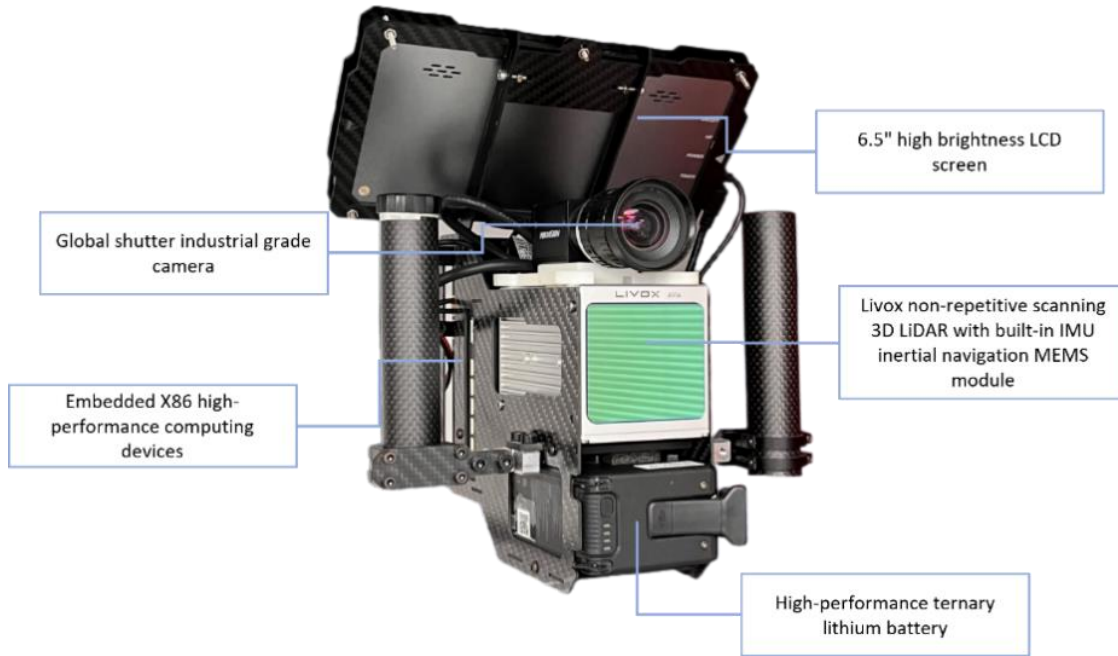


Figure 3.2 Waterloo modular mobile platform for multi-sensor integration (Wat-MMP).

For different scenarios in indoor environments, this modular platform is extended in four different forms. Figure 3.3 (a) shows the structural design of the mobile platform, on which different modules can be switched as needed. Figure 3.3 (b) shows that in the backpack form, the LiDAR can be replaced from a non-repetitive scanning 3D LiDAR to a 360°mechanically scanning 16-line LiDAR, and a 360°camera is selected to replace the fisheye camera for environmental data collection in buildings, corridors, underground walkways and so on. As shown in Figure 3.3 (d), a larger size battery is added and placed on a cart for data collection in large-scale indoor scenarios such as airport parking buildings, underground parking lots, and other large-scale indoor environments. Or, as shown in Figure 3.3 (e), add a mechanical motion structure to place the module on a robot dog for data collection.

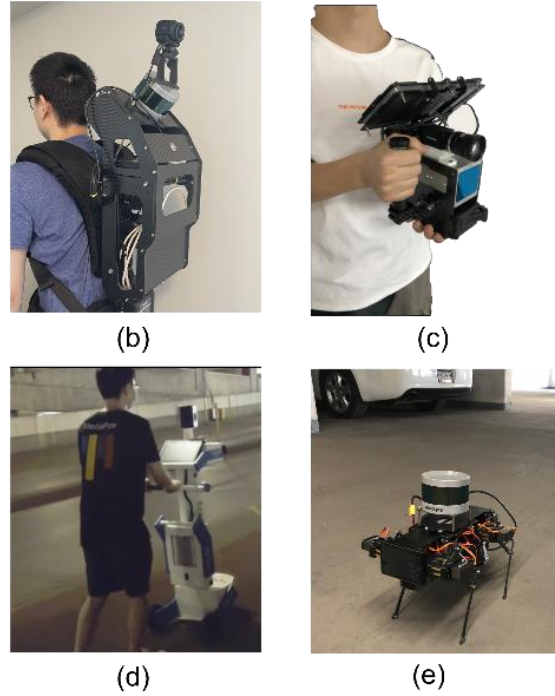
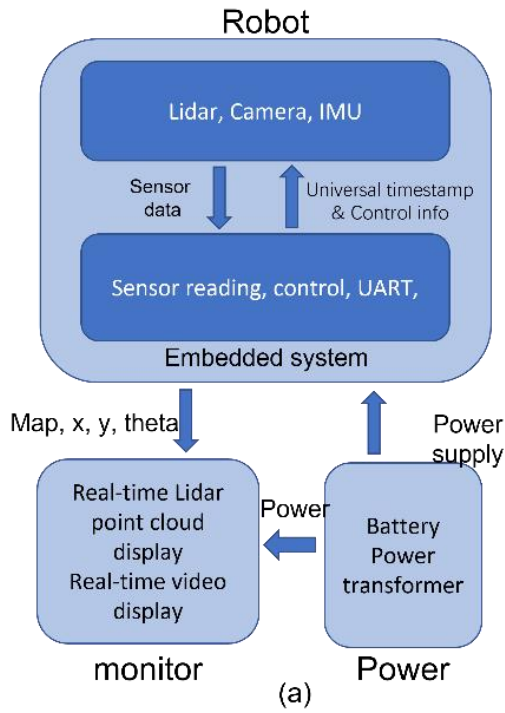


Figure 3.3 (a) the architecture of the modular mobile platform, (b) Waterloo PLS, (c) Waterloo Wat-MMP, (d) the cart platform, and (e) Waterloo Wat-dog

3.3 Sensor Synchronization

In order for features received by multiple sensors to benefit the system, the frequency of information collection between sensors needs to be adjusted and there needs to be a way to aggregate the information received at a given time for different sensors so that the information is considered valid. Two principles need to be observed for sensor synchronization: (1) The same set of signals needs to be acquired by different sensors at the same time (or within a concise time difference). (2) The time information between sensors must have a constant clock source. Based on these two principles, we have developed a time synchronization scheme in our system that can be used for multiple sensors. This solution can be used with the ARM architecture embedded system with Sony IMX477 sensor, DJI Livox Horizon LiDAR, HIKROBOT MV-CE060-10UC, and X86 architecture PC systems.

Network-based high-precision time synchronization protocol PTP (Precision Time Protocol, 1588 V2), synchronization accuracy can reach a sub-microsecond level. Meanwhile, PTPv2 is an active time synchronization system, and the use of hardware timestamps can significantly reduce software processing time. Meanwhile, PTP can run at the L2 layer (MAC layer) without going through the UDP protocol stack, thus largely reducing the protocol stack residence time and further improving the time synchronization accuracy, which is a very friendly architecture for indoor systems.

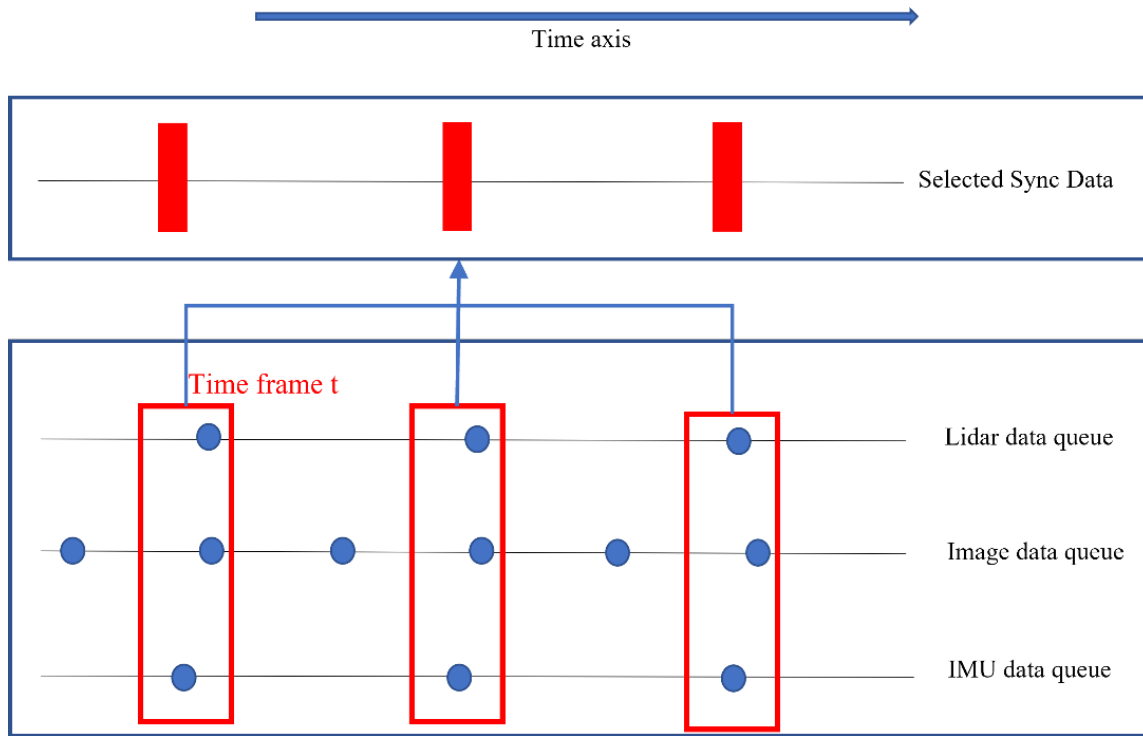


Figure 3.4 Synchronized selection policy.

Figure 3.4 shows the selection rule of the time synchronization system. The system will take the lowest frequency sensor as the reference base, based on which the other sensor data with the closest time is selected as the same group of data identified. The benefit of this setup is for modular mobile platforms. You can freely switch between sensor brands and connection methods. In the case of the

handheld platform, we are using a Livox horizon LiDAR sensor and a HIKROBOT MV-CE060-10UC camera. The frequency of the LiDAR is 10 FPS and the camera is 20 FPS. According to this method, the point cloud collected by the LiDAR is used as the basis for timestamp synchronization, which means that the point cloud resolution is not degraded by sensor synchronization alignment. For the camera, the coverage of a single photo is sufficient to include the area covered by two or more point clouds, which means that the colouring of the point cloud is not affected. Since not all sensors can be triggered by hardware (e.g., cameras), we use the ROS message-filter TimeSynchronizer in the system design (Figure 3.5). The ROS package obtains sensor data and timestamp information topics subscribed, and this part of data is then passed from the node to the ROS synchronizer, which processes the data by the above synchronization policy to achieve time synchronization.

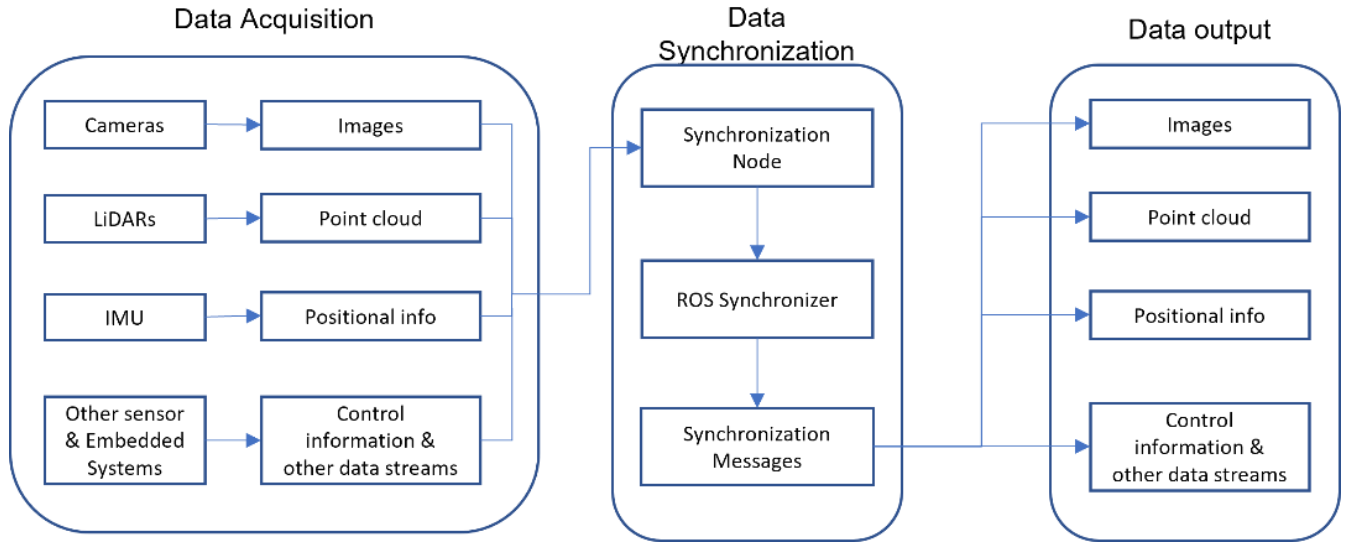


Figure 3.5 Schematic diagram of the synchronization process.

3.4 Multi-sensor Calibration Method

Among multi-sensor mobile platform devices, LiDAR and camera have their advantages and disadvantages, respectively. The advantages of cameras are that they are inexpensive, more people are

using them for algorithm development, and the technology is relatively mature. The disadvantages of cameras are that it is challenging to obtain accurate 3D information (monocular cameras are almost impossible, and some people propose binocular or trinocular cameras to do it); another disadvantage is that it is more restricted by ambient light. The advantage of LiDAR is that its detection distance is long, and it can accurately obtain 3D information of the object; in addition, it is quite stable and robust. However, the cost of LiDAR is currently high, and the product's final form has not yet been determined. The Livox LiDAR we use has an equivalent point cloud density of 32 lines, but due to its limited view angle, it does not display complete 360-degree environment information in a single frame of data.

3.4.1 Camera Calibration and Distortion Removal

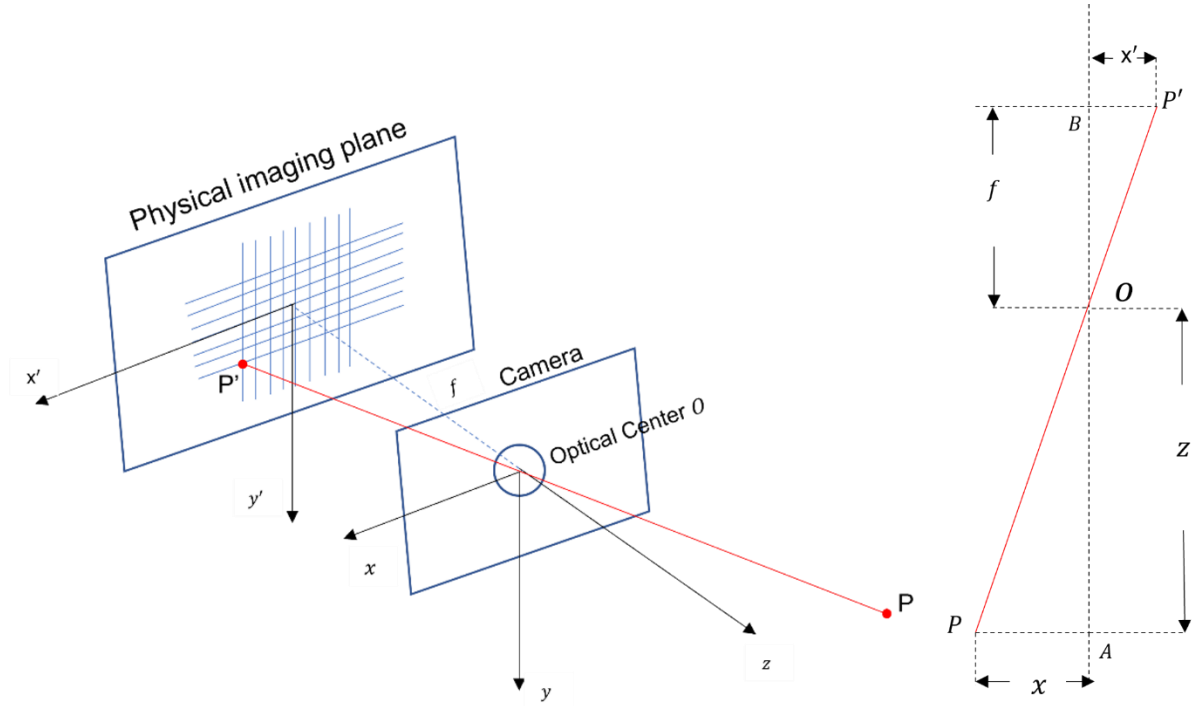


Figure 3.6 Camera coordinate system $O-x-y-z$.

Figure 3.6 shows the camera imaging model. In this model, let $O-x-y-z$ be the camera coordinate system, the z -axis pointing to the front of the camera, x to the right, y to the bottom, and O the camera's

optical center. After passing through the small aperture O projection, the real-world spatial point P falls on the physical imaging plane O'-x'-y'-z', and the imaging point is P'. Let the coordinates of P be $[X, Y, Z]^T$, coordinates of P' be $[X', Y', Z']^T$. Let the distance from the physical imaging plane to the small aperture be f (focal length). Then, according to the triangle similarity relation, there are

$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'} \quad (3.1)$$

Where the negative sign indicates that the imaged image is upside down, to simplify the model, the imaging plane can be symmetrical to the front of the camera, and three-dimensional space points together on the same side of the camera coordinate system, the formula is simplified as follows.

$$\begin{aligned} X' &= f \frac{X}{Z} \\ Y' &= f \frac{Y}{Z} \end{aligned} \quad (3.2)$$

The above equation describes the spatial relationship between a point P and its image. However, in a camera, what is finally obtained is pixel by pixel, which requires sampling and quantizing the image on the imaging plane. To describe the process by which the sensor converts the perceived light into image pixels, a pixel plane o - u - v is fixed in the physical imaging plane. the pixel coordinates of P' are obtained in the pixel plane: $[u, v]^T$. The origin o' is located in the upper left corner of the image, u-axis is parallel to the x-axis to the right, and the v-axis is parallel to the y-axis to the down. The pixel coordinate system differs from the imaging plane by scaling and a translation of the origin. Let the pixel coordinates abreacted α times on the u-axis and β times on v-axis. Also, the origin is translated $[cx, cy]^T$. Then, the coordinates of the point P' and the pixel coordinates $[u, v]^T$ are related as follows:

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \quad (3.3)$$

After optimized became:

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases} \quad (3.4)$$

Writing the equation in matrix form, then we get:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq \frac{1}{Z} KP \quad (3.5)$$

where the internal parameter matrix of the camera is K.

To obtain a good imaging effect, a lens is added in front of the camera. The addition of the lens will have a new impact on the propagation of light during imaging: on the one hand, the shape of the lens itself on the propagation of light, and on the other hand, the lens and the imaging plane cannot be perfectly parallel during mechanical assembly. This also causes a change in the position of the light when it is projected through the lens onto the imaging plane. The distortion caused by the shape of the lens are called radial distortion. They are mainly divided into two categories, barrel distortion and pincushion distortion, as shown in Figure 3.7.

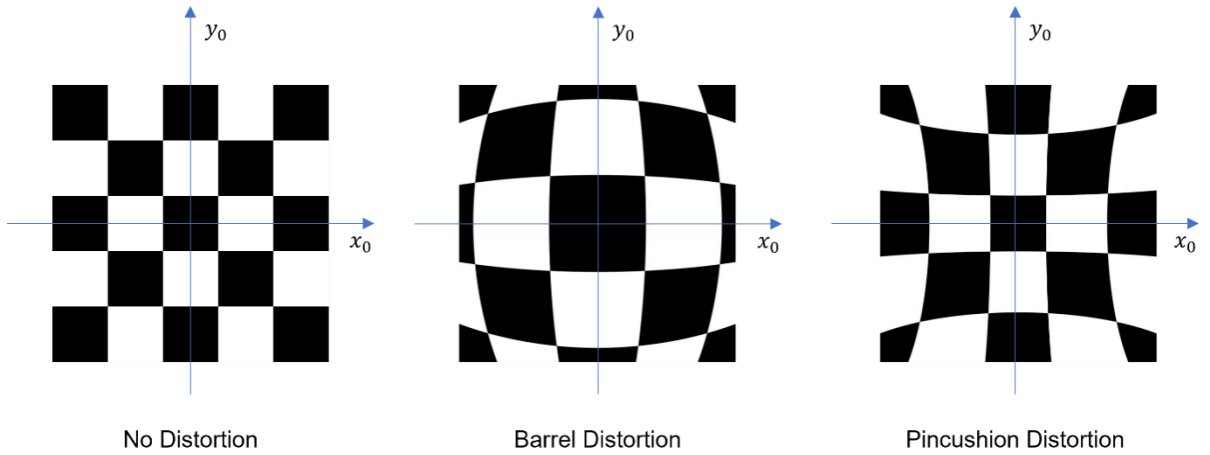


Figure 3.7 Radial distortion effect.

Any point p in the plane can be expressed in cartesian coordinates as $[x, y]^T$, or written in the form of polar coordinates as $[r, \theta]^T$, where r denotes the distance of the point p from the origin point and θ is the angle with the horizontal axis. For the radial distortion, the change in coordinates before and after the distortion can be described by a polynomial function:

$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{cases} \quad (3.6)$$

where $[x, y]^T$ are the coordinates of the uncorrected points and $[x_{corrected}, y_{corrected}]^T$ are the coordinates of the corrected points. Radial and tangential distortion correction is performed for points on the normalized plane:

$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_2xy + p_1(r^2 + 2y^2) \end{cases} \quad (3.7)$$

The corrected points are projected to the pixel plane through the internal parameter matrix to get the correct position of the changed points on the image.

$$\begin{cases} u = f_x x_{corrected} + c_x \\ v = f_y y_{corrected} + c_y \end{cases} \quad (3.8)$$

3.4.2 External Reference Calibration

The external reference of the camera and the LiDAR is the correspondence between the coordinates of the image and the point cloud in the same frame. The LiDAR and the camera are considered rigidly connected structures and the internal reference results of the camera have been obtained. Then the external reference solution can be considered a Perspective-n-Point (PnP) problem [90]. In this case, the transformation relationship between the camera and the image pixel points is:

$$\begin{bmatrix} u & v & 1 \end{bmatrix}^T = \frac{1}{Z_{camera}} M \begin{bmatrix} X_{camera} & Y_{camera} & Z_{camera} \end{bmatrix}^T \quad (3.9)$$

Where M is the camera's internal reference. $(X_{camera} \ Y_{camera} \ Z_{camera})$ are the coordinates in the camera coordinate system. The conversion relationship between the coordinates of the camera coordinate system and the coordinates of the radar coordinate system is:

$$\begin{bmatrix} X_{camera} & Y_{camera} & Z_{camera} \end{bmatrix}^T = [R_t \ t_t] \begin{bmatrix} X_{lidar} & Y_{lidar} & Z_{lidar} \end{bmatrix}^T \quad (3.10)$$

$(X_{lidar} \ Y_{lidar} \ Z_{lidar})$ are the position coordinates in the LiDAR coordinate system. R_t denotes the 3x3 rotation matrix of the point cloud in the LiDAR coordinate system to the image pixels in the camera coordinate system. t_t denotes the 3D translation vector.

When using the calibration board to solve for the parameters in the external parameter (R_t, t_t) , it is necessary to have more than 3 different sets of position data for multiple sets of pixels and point clouds corresponding to the key points of the calibration board (which is $N \geq 3$). Then solve for the external parameter according to Eqs. (9), (10). Because the normal vector $n^{(lidar)}$ of the plane P is transformed by the rotation matrix R_t to obtain $n^{(lidar, camera)}$, which is $n^{(lidar, camera)} = R_t n^{(lidar)}$. Then in the ideal case, the dot product of $n^{(lidar, camera)}$ with the image plane vector should be 0, but due to the existence of systematic measurement errors, there will be an average dot product error e_d . Similarly, the calibration error of $n^{(lidar, camera)}$ with the image plane normal vector $n^{(camera)}$ is e_r , which can be obtained as follows:

$$\begin{cases} e_d = \frac{1}{N} \sum_{i=1}^N \left[(o_{1,0}^{camera} - o_{0,i}^{camera}) \cdot n_i^{(lidar, camera)} \right]^2 \\ e_r = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum \left(n_i^{(lidar, camera)} - n_i^{(camera)} \right)^2} \end{cases} \quad (3.11)$$

Then the optimal rotation matrix $\widehat{R}_t$ can be found:

$$\widehat{R}_t = \min_{R_t} e_d + e_r \widehat{t}_t \quad (3.12)$$

Where $\widehat{t}_t$ is the optimal translation vector, when the point cloud keypoints are projected onto the image, the average error can be obtained as the average distance between the image keypoints and the point cloud keypoints as follows.

$$e_t = \frac{1}{3N} \sum_{i=1}^N \sum_{j=0}^2 \sqrt{\sum \left(o_j^{(camera,i)} - o_j^{(lidar,camera,i)} \right)^2} \quad (3.13)$$

where $o_j^{(camera,i)}$ is the image keypoint j on the i pose and $o_j^{(lidar,camera,i)}$ is the projection of the point cloud keypoint j on the i pose.

3.4.3 LiDAR Point Cloud Filtering

To remove noise and outliers, a statistical outlier removal (SOR) filter is applied to the raw LiDAR point cloud data. This filter computes the mean distance and standard deviation of each point to its k -nearest neighbors. Suppose a point's distance to the mean distance is greater than a user-defined threshold (usually set to a multiple of the standard deviation). In that case, it is considered an outlier and thus removed from the point cloud.

Let $P = \{p_1, p_2, \dots, p_n\}$ be a set of points in the raw point cloud, and k_i be a set of k -nearest neighbors of point p_i . The mean distance μ_i and standard deviation σ_i of point p_i to its k -nearest neighbors are computed as follows:

$$\mu_i = \frac{1}{k} \sum_{j=1}^k dist(p_i, p_{k_i,j}) \quad (3.14)$$

$$\sigma_i = \left| \frac{1}{k} \sum_{j=1}^k \text{dist}(p_i, p_{k_{i,j}}) - \mu_i \right| \quad (3.15)$$

Where $\text{dist}(p_i, p_{k_{i,j}})$ is the Euclidean distance between points p_i and $p_{k_{i,j}}$, If the distance from p_i to the mean distance μ_i is more significant than a user-defined threshold:

$$T = \alpha * \sigma_i \quad (3.16)$$

where α is a constant factor, the point p_i is considered an outlier and accordingly removed from the raw point cloud.

3.4.4 Point Cloud Downsampling

To reduce the computational complexity of the proposed method, the filtered point cloud is downsampled using a voxel grid filter. The voxel grid filter divides the point cloud into a regular grid of 3D voxels, and each voxel's points are replaced by their centroid. This process results in a lower-resolution point cloud that retains the overall structure of the environment, while reducing the amount of data to be processed.

Let $V = \{v_1, v_2, \dots, v_m\}$ be a set of non-empty voxels in the grid, and $Q = \{q_1, q_2, \dots, q_m\}$ be a set of downsampled points. The centroid q_i of each non-empty voxel v_i is computed as follows:

$$q_i = \frac{1}{|v_i|} \sum_{p \in v_i} p \quad (3.17)$$

Where $|v_i|$ is the number of points in voxel v_i , and p is a point in voxel v_i . The resulting downsampled point cloud Q is used as the input for the subsequent processing steps.

By applying filtering and downsampling in the data preprocessing step, the proposed method effectively reduces noise, outliers, and computational complexity, allowing for more efficient and accurate scan registration and graph optimization.

3.4.5 Scan Registration

Scan registration is the process of aligning consecutive LiDAR scans to estimate the relative motion between them. In this section we employ the ICP algorithm [7] to register consecutive point cloud scans. The ICP algorithm iteratively refines the alignment by minimizing the point-to-point distance between the adjacent two scans.

Given two point clouds $P = \{p_1, p_2, \dots, p_n\}$ and $Q = \{q_1, q_2, \dots, q_m\}$. The objective of the ICP algorithm is to find the optimal transformation matrix T (composed of a rotation matrix R and a translation vector t). The method to minimize the mean squared error between these two point clouds is to use the following equation:

$$E(T) = \frac{1}{n} \sum_{i=1}^n \|T(p_i) - q_i'\|^2 \quad (3.18)$$

Where $T(p_i)$ represents the transformed point p_i , and q_i' is the corresponding point in point cloud Q .

3.4.6 Total Station Measurements

Total station measurements provide highly accurate positional data of key points during data acquisition. These measurements are integrated into the proposed method to improve the accuracy of the estimated trajectory. The key points measured by the total station are used as landmarks, and their positions are incorporated into the SLAM system as additional constraints.

Let $K = \{k_1, k_2, \dots, k_l\}$ be a set of key points measured by the total station, with l being the total number of key points. The total station provides the highly accurate 3D coordinates (x, y, z) of each key point k_i in a fixed reference frame (usually a global or local coordinate system). The goal is to integrate these measurements into the SLAM algorithm to refine the robot's trajectory. To this end we first need to find the corresponding LiDAR points for each key point k_i . Let $L = \{l_1, l_2, \dots, l_n\}$ be a set of LiDAR points, with n being the total number of these LiDAR points. The task is to find the LiDAR point l_i that corresponds to each key point k_i . For each key point k_i , we search for the nearest LiDAR point l_j using a nearest-neighbor search algorithm as follows:

$$l_j = \arg \min_{l \in L} \|k_i - l\| \quad (3.19)$$

Where $\|k_i - l\|$ is the Euclidean distance between the key point k_i and the LiDAR point l .

Once the corresponding LiDAR points are identified, the total station measurements are incorporated into the SLAM system as additional constraints. These constraints are formulated as the error between the key point k_i and its corresponding LiDAR point l_j , given the current estimated robot pose e_i as follows:

$$e_i = k_i - T(p_i) * l_j \quad (3.20)$$

Where $T(p_i)$ is the transformation matrix representing the robot pose at the time when the LiDAR point l_j was observed, and $*$ denotes the transformation operation.

The proposed method effectively integrates the total station measurements into the SLAM system by minimizing the sum of squared errors between the key points and their corresponding LiDAR points. And also providing additional constraints that help refine the robot's trajectory.

3.4.7 Graph Optimization

Graph optimization is used to refine the robot's trajectory by minimizing the errors between the estimated poses and the constraints from scan registration and total station measurements. The graph optimization problem is formulated as a pose graph, where nodes represent robot poses and edges depict constraints between two poses.

Let $G = (V, E)$ be a pose graph, where $V = \{v_1, v_2, \dots, v_n\}$ is a set of nodes representing the robot poses, and $E = \{e_1, e_2, \dots, e_m\}$ is a set of edges representing the constraints between the poses. Each edge e_{ij} is associated with a transformation matrix T_{ij} and an information matrix Ω_{ij} , which represents the inverse of the covariance matrix of the constraint. In this section, we employ two types of constraints in the pose graph: odometry constraints and total station constraints. Odometry constraints are derived from the relative pose transformations T_i obtained from the scan registration process, which connect consecutive poses and enforce the consistency between the estimated poses and the relative motion estimated from the scan registration. Total station constraints are derived from the total station measurements, which connect the robot poses to the key points and enforce the consistency between the estimated poses and the absolute positions of the key points.

The objective of the graph optimization is to find the optimal robot poses that minimize the total error in the pose graph using the following equation:

$$E(V) = \sum_{i,j \in E} e_{ij}^T * \Omega_{ij} * e_{ij} \quad (3.21)$$

Where e_{ij} is the error between the connected poses v_i and v_j , and Ω_{ij} is the information matrix.

We employ the Gauss-Newton algorithm to solve the graph optimization problem, which iteratively refines the robot poses by minimizing the total error in the pose graph. The Gauss-Newton algorithm consists of the following steps:

Algorithm: Gauss-Newton Optimization
Input: initial robot poses V , pose graph constraints C Output: refined robot poses V Parameters: maximum iterations MAX_ITER , convergence threshold $THRESHOLD$
<pre> 1. Set V = initial robot poses 2. for iter = 1 to MAX_ITER do 3. e = compute_errors(V, C) 4. if sum(e) < $THRESHOLD$ then 5. break 6. end if 7. J = compute_jacobian(V, C) 8. r = transpose(e) 9. ΔV = solve_linear_system(J, r) 10. $V = V + \Delta V$ 11. end for 12. return V </pre>

The output of the graph optimization process is a refined set of robot poses that minimize the total error in the pose graph, effectively correcting the drift and trajectory offset in the SLAM system.

3.4.8 Fusion of LiDAR Point Clouds with Camera Images

As mentioned earlier, the image data captured by the camera is represented by (U, V) . And the 3D point cloud captured by the LiDAR is represented by (X, Y, Z) . The goal is to build a transformation matrix M that maps the 3D points (x, y, z) to the 2D points (u, v) :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.22)$$

Where m_{ij} is the element of matrix M. The matrices (f_u, f_v, u_0, v_0) are the camera parameters. f_u and f_v are the scale factors (effective focal lengths in the horizontal and vertical directions) in the XY-axis direction, u_0, v_0 are the centre points of the image plane, also known as the principal point coordinates. R is the rotation matrix and t is the translation vector. According to the calibration board plane at different attitudes, a series of linear equations can be obtained to solve for the calibration parameters.

$$\begin{cases} u = \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \\ v = \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \end{cases} \quad (3.23)$$

3.5 Data Description

Based on the modular mobile platform, we propose an indoor dataset with multi-sensor data named WATiMMS. The unique feature of this dataset is that, compared to the dataset presented in Table 3.3, our dataset encompasses the complete full process from multi-sensor calibration, 3D reconstruction verification, 2D/3D object recognition, etc. And the dataset covers a wide range of scenarios, from underground parking lots, airport parking buildings, indoor offices, underground pedestrian tunnels, etc. And the modular device for collecting data is designed as open source with clear standards so other research teams can replicate it and can be added to existing datasets to fill in the gaps in the dataset. Table 3.3 presents the content coverage of the dataset, which also includes the accuracy metrics and uses included in the dataset. As the table shows, the WATiMMS dataset contains three components, each described in detail below.

Table 3.3 WATiMMS content coverage

Data name	Scene category	Data size (GB)	Data type	Description
Parking_01#1 Parking_01#2 Parking_02#1 Parking_02#2	Underground parking lot	19.07	Image(.JPG), Point Cloud (.pcd), LiDAR Raw data (.lvx), IMU data(.bag), Label data & Parameters (.txt)	Cart form platform for single plane scene data collection. The scene is rich in the number of vehicles and no pedestrians walking around. The scene ground is flat and rich in 3D textures.
Parking_03#1	Airport parking building	8.23	Image(.JPG), Point Cloud (.pcd), LiDAR Raw data (.pcap), Label data & Parameters (.txt)	Backpack form platform for single plane scene data collection. The scene is sparsely populated with vehicles and a few pedestrians walking around. The scene ground is flat and rich in 3D textures.
Stiar_01#1 Stair_01#2	building stairs	2.94	Image(.JPG), Point Cloud (.pcd), LiDAR Raw data (.lvx), IMU data (.bag), Label data (.txt)	Handheld form mobile platform for data collection on a four-level planar scene. The scene has many pedestrians walking around. The data can do loopback detection and move back to the same location via the starting point.
Calib_01#1 Calib_02#1	Indoor hallway Underground parking lot	1.82	Image(.JPG), Point Cloud (.pcd), intrinsic parameters & LiDAR to camera rigid transformation parameters (.txt)	In both underground parking and indoor environments, the handheld calibration board is collected within the field of view without moving the equipment.
Tunnel_01#1 Tunnel_02#1	Underground pedestrian tunnel	4.17	Image(.JPG), Point Cloud (.pcd), LiDAR Raw data (.lvx), IMU data(.bag), Label data & Parameters (.txt)	Cart form platform for underground planar scene data collection. The scene has no pedestrian walking, smooth walls and sparse 3D textures, accompanied by a large number of 2D images.
Ofiice_01#1 Ofiice_01#2	Open office space	6.45	Image(.JPG), Point Cloud (.pcd), LiDAR Raw data (.lvx), IMU data(.bag),	More occlusion challenges than parking lots and underground tunnels, can be used as a cloud stitching test and upsampling for large scale scenes.

3.5.1 Indoor SLAM Dataset

This part of the data consists of four scenarios: underground parking, airport parking building, single-floor indoor office, and underground pedestrian tunnel. As an indoor dataset, this part of the data can usually be used for the validation of indoor 3D reconstruction algorithms. A single frame of signal data that has been synchronized is provided for each scene in the dataset. At the same time this data also

provides a large-scale 3D point cloud file stitched using LIO-SAM algorithm [121] as a reference for 3D reconstruction. The reason why LIO-SAM algorithm is chosen as the reference for 3D reconstruction is because the trajectory of LIO-SAM algorithm is closer to the ground truth's odometer after comparison with LOAM, FLOAM, and Livox SLAM. The reference file has three forms of 3D reconstruction, which are the stitching of the complete point cloud, the extraction of the frame lines of the 3D environment, and a 3D point cloud with reference points labeled. Figure 3.8 shows the three forms of data provided by the underground parking portion of the dataset.

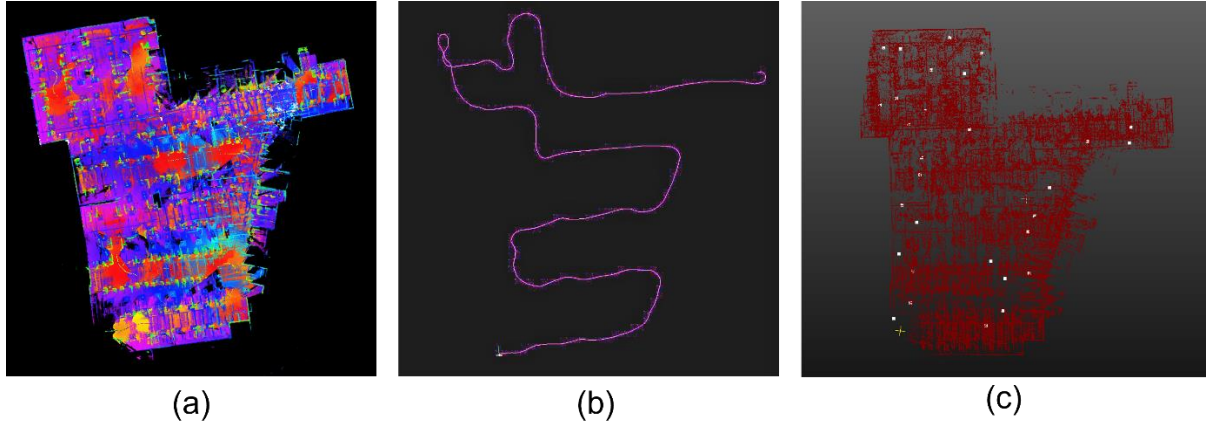


Figure 3.8 Underground parking portion dataset include: (a) point cloud for underground parking lot, (b) graph-optimized odometers, and (d) reference points on the frame extraction point cloud.

On this basis, multiple cones were placed in each indoor environment where data were collected to increase the number of observable reference points. Based on the collected data, the relative distances between all cones were checked using a Leica Total Station instrument for error analysis. Note that the labeled reference point in Figure 3.9 is a schematic representation of the position of the reference point in the point cloud in the algorithm synthesis, and this position cannot be used as the ground truth value of the relative position of the reference point, which is obtained manually by the Leica MS50 total station in the field environment, with a relative distance accuracy of $2\text{mm}+2\text{ ppm}$. Figure 3.10 shows

the 3D positions of the reference points in the dataset and the method of their relative position acquisition. Considering that this dataset can be added by the modular platform with new point cloud data, the local coordinate system of the dataset will be changed accordingly. Therefore, for the reference points on this basis, the relative distance between the point cloud points is chosen to replace the position of the points in the local coordinate system as the criterion for judging the point cloud stitching.

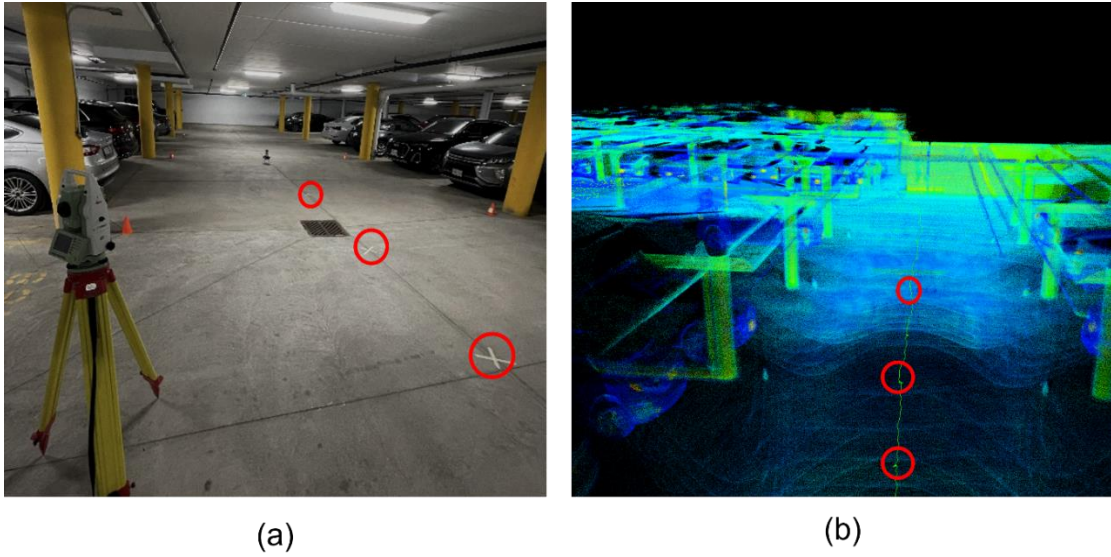


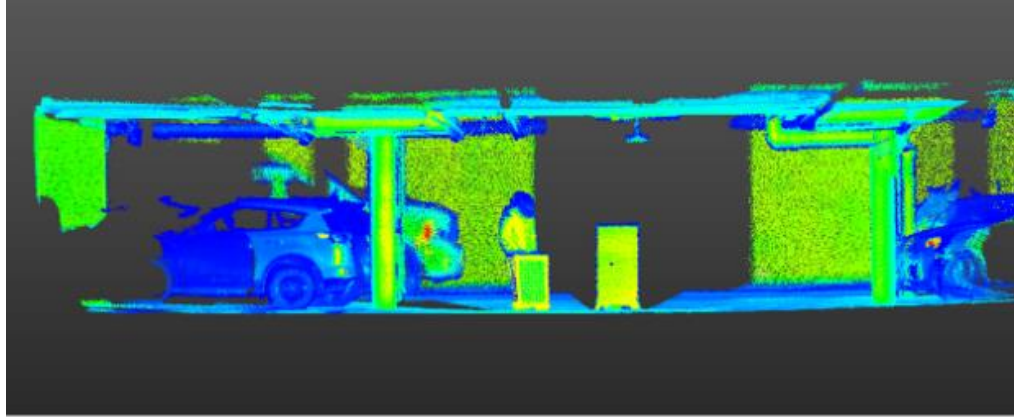
Figure 3.9 Position schematic. (a) The mobile platform in the underground parking lot, and (b) LiDAR SLAM point cloud result.

Taking parking_01#1 as an example, there are 30 reference points in this dataset. 5 points are selected as a group, and any point in each group is chosen as the base point. Also, the graph-optimized trajectory can be used as a ground truth reference for the platform movement odometer. The accuracy of the result is expressed by the spatial distance error and spatial angle error.

3.5.2 Multi-sensor Calibration Datasets

Multi-sensor integration, one of the key strengths of this dataset, provides a calibration dataset for both image and point cloud data, which has calibration data in multiple scenes. The data including images,

point clouds and IMU data are synchronized with timestamps according to Precision Time Protocol (PTPv2), based on which accurate annotation and alignment are performed. The labeled camera inline functions, rigid transformation matrix, rotation matrix and translation matrix are also included. Figure 8 shows the image and point cloud data for a labeled scene. The number of point clouds in this data is 5,592,000. The resolution of the image is 1280×720 pixels. Since the purpose of this dataset is to verify the fusion of point clouds with images, with reference to the mainstream application scenarios of indoor and outdoor point cloud image fusion [88], we define that the acceptable fusion range in this dataset is a target within 50 meters whose edges of the point cloud coincide with the edges of the image, or the error range of the calibration result is less than 10 pixels, then the accuracy of the calibration result is proved to be as expected.



(a)



(b)

Figure 3.10 The calibration data. (a) The point cloud data for calibration, and (b) the image for calibration.

3.5.3 Indoor Segmentation Dataset

The part of the dataset used for scene segmentation and object recognition is used as another focus of this dataset. The point cloud data is generated by manual annotation through ClockCompare software. After completion of the labeling process, the labeled data is saved in a format that includes both the point coordinates and the assigned labels. It should be noted that maintaining consistency and accuracy in labeling is paramount, as it directly impacts the performance of the resulting machine-learning model. Figure 3.11 shows the comparison of partial semantic segmentation classification in the dataset.

Six common scenes for indoor scenes, especially in underground parking environments, were classified into six categories, namely:

- (1) Ground (Label 1): Ground with speed bump rubber and manholes.
- (2) Ceiling (Label 2). Ceilings and beams at the top of parking lots, and pipes.
- (3) Pillars (Label 3). All bollards and poles.
- (4) Vehicles (Label 4). Vehicles including cars, SUVs, and trucks.
- (5) Walls (Label 5). The wall is located at the boundary of the parking lot and the dividing wall in the middle of the parking lot.
- (6) Unclassified (Label 6). Unidentifiable objects and other noises.

3.6 Data Analysis

This dataset covers three sets of evaluation metrics: for point cloud stitching and 3D scene reconstruction, the dataset provides indoor reference points as ground truth. The relative position relationship between reference points can be used to compare the relative position relationship of 3D models. For camera and LiDAR calibration, the dataset provides fully overlapping images with point cloud internal participation external reference, which is generated by manual calibration, and point cloud to image mapping results in less than 10 pixels within 50 m. In the semantics segmentation section, point cloud annotation is performed for the five most occurring indoor semantics, and annotation rules are provided in the file for obtaining more annotated data from other researchers after the dataset is opened.

3.6.1 Point Cloud and Image Calibration.

In this experiment, the LiDAR- Camera Calibrator provided by MATLAB was used as a tool for automatic calibration, and eight sets of calibration boards in motion were selected as validation data.

The tool uses an automatic extrinsic method based on the line and plane correspondences. The results in Figure 3.12 can be obtained by automatic calibration method. The errors of the ten sets of results are also show in the figure. For the point cloud reprojection onto the image the errors are within a reasonable range of 10 pixels defined. It can also be seen that the alignment of the camera and the LiDAR can be achieved based on the colouring of its point cloud.

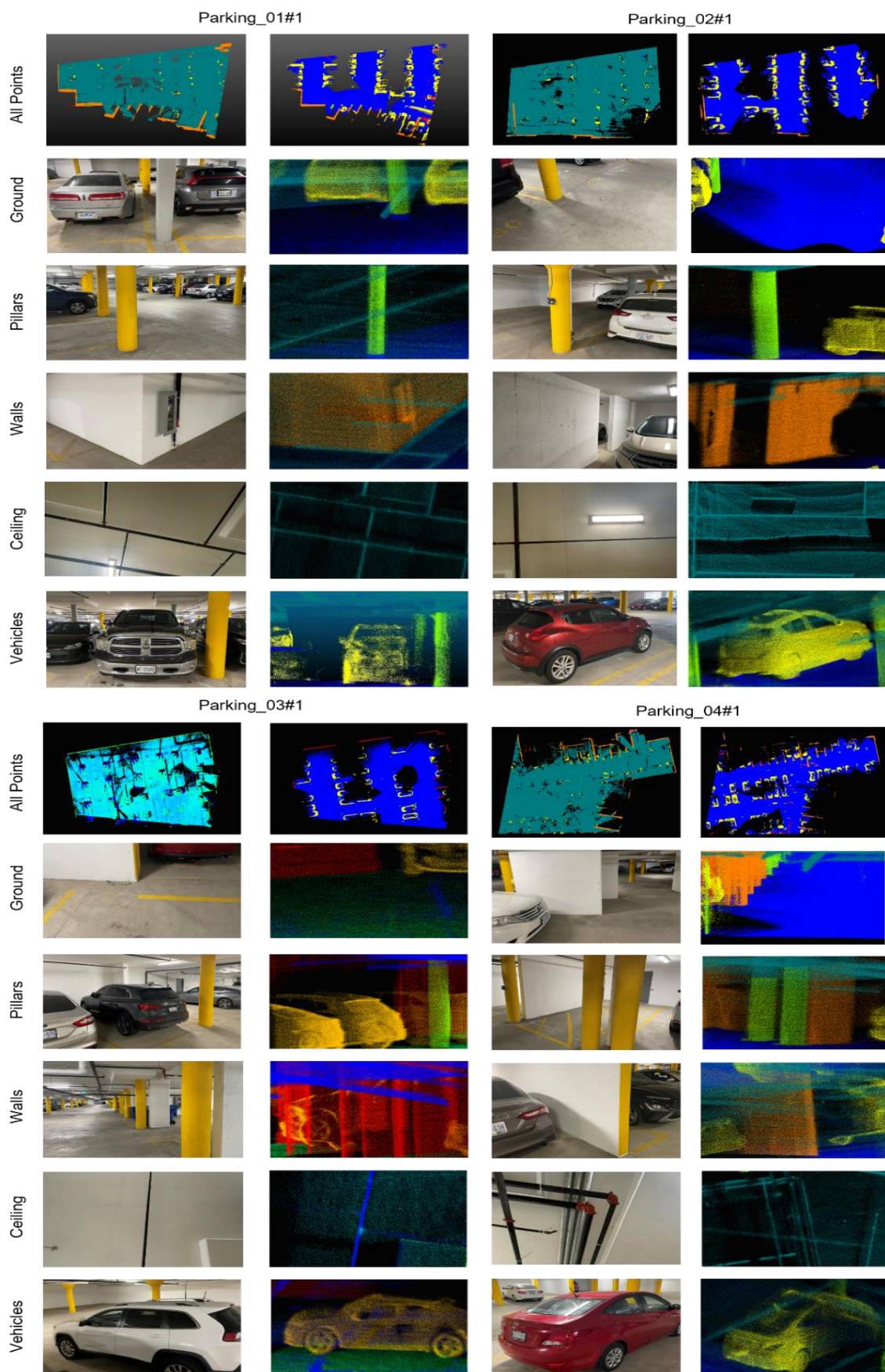


Figure 3.11 Semantic segmentation classification in the dataset.

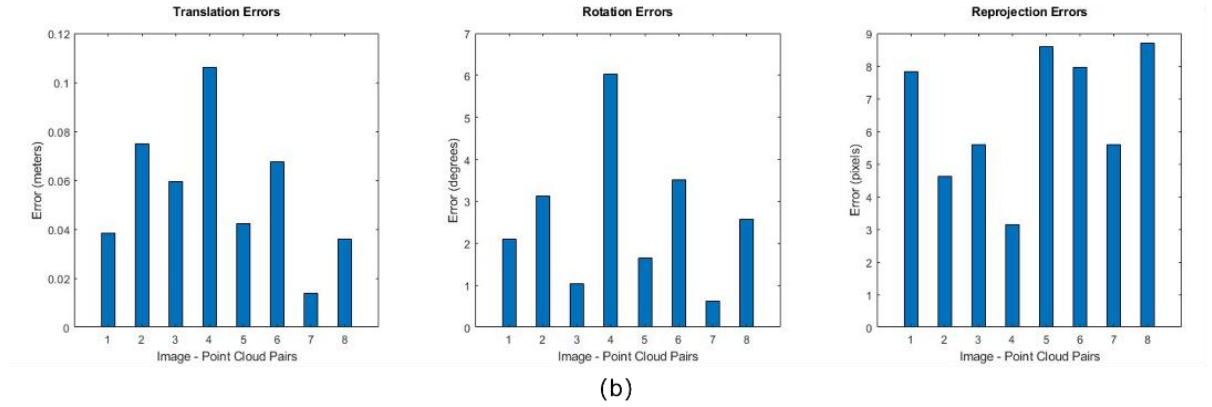
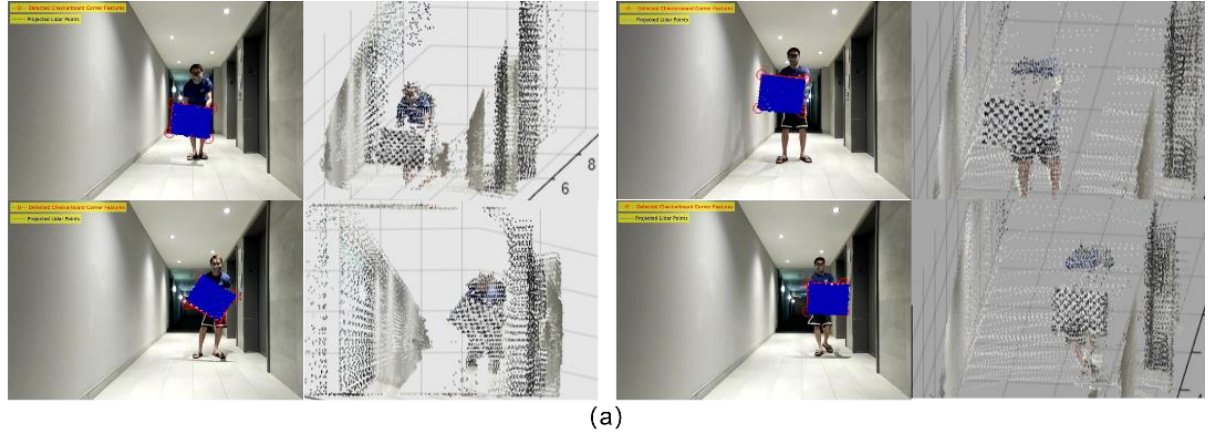


Figure 3.12 (a) Calibration results projection. (b) Translation, rotation, and reprojection error distribution.

3.6.2 Semantic Segmentation

Then the semantic segmentation part of the dataset was experimented. The purpose of this experiment was to examine the performance of the dataset according to the three mainstream classical semantic segmentation networks (PointNet, PointNet++, and RandLA-Net) and the comparison of the segmentation of these three networks.

Figure 3.13 displays the training loss convergence of PointNet, PointNet++, and RandLA-Net. It can be concluded that all three deep learning methods converge after 60 epochs. Table 3.44 shows the quantitative model performance for each class obtained from the testing dataset. It indicates that the mIoU, precision, recall, and F₁-score for PointNet is 67.5%, 74.7%, 75.5%, and 75.0%, respectively, 69.0%, 80.4%, 76.6%, and 77.8% for PointNet++, respectively, and 73.0%, 76.9%, 78.6%, and 77.6% for RandLA-Net, respectively. It demonstrates that RandLA-Net has better semantic segmentation performance compared with both PointNet and PointNet++.

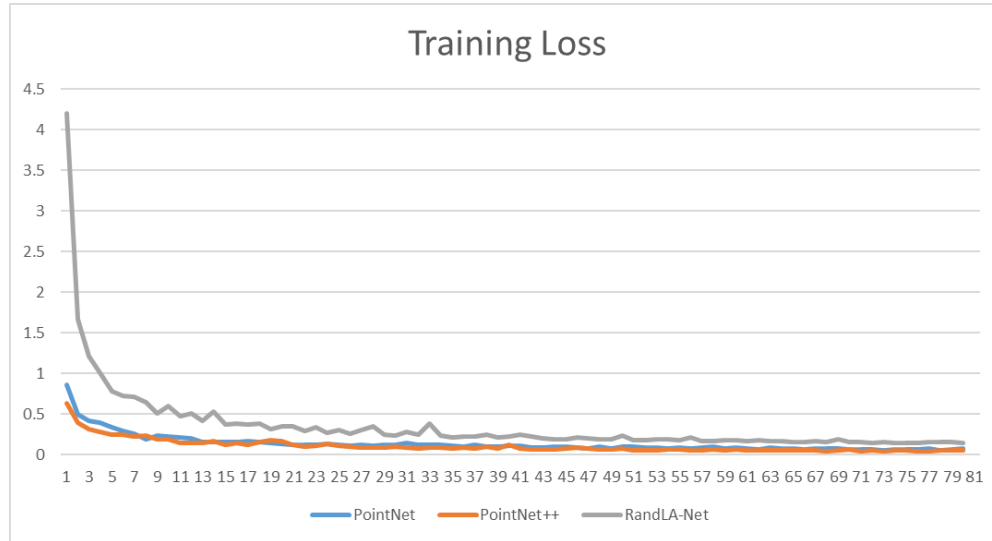


Figure 3.13. The training loss for PointNet, PointNet++, and RandLA-Net, respectively.

Table 3.4 Model performance (%) on the WATiMMS testing dataset.

Network	Metrics	Ceiling	Floor	Pillars	Car	Wall	Unclassified	Average
PointNet	IoU	93.8	95.0	64.2	79.7	68.1	4.1	67.5
	Precision	97.2	99.4	75.1	87.5	78.9	10.3	74.7
	Recall	96.4	95.5	81.5	90.0	83.3	6.3	75.5
	F ₁ -score	96.8	97.4	78.2	88.7	81.0	7.8	75.0
PointNet++	IoU	92.0	97.3	57.9	88.7	60.9	17.0	69.0
	Precision	95.9	98.2	72.7	92.3	77.9	45.1	80.4
	Recall	95.8	99.1	73.9	95.8	73.6	21.4	76.6
	F ₁ -score	95.8	98.6	73.3	94.0	75.7	29.1	77.8
RandLA-Net	IoU	97.2	86.7	94.3	89.3	70.7	0	73.0
	Precision	98.7	97.0	96.3	91.5	77.8	0	76.9
	Recall	98.4	89.1	97.9	97.4	88.5	0	78.6
	F ₁ -score	98.6	92.9	97.1	94.3	82.8	0	77.6

Figure 3.14 presents the normalized confusion matrices for PointNet, PointNet++, and RandLA-Net on the testing dataset. Such confusion matrices provide a reference for the segmentation pattern with more details. It shows that PointNet and PointNet++ are prone to confusing two pairs of objects, i.e., pillars and walls, and vehicles and walls, respectively. Regarding PointNet, about 12% of the points of pillars and walls are confused, and this ratio is 16% for PointNet++. The results of RandLA-Net

show that ground and vehicle are likely to be the confused pair of objects. Besides, the confusion might result from insufficient points due to occlusions, labeling mistakes, and point density variances between classes.



(a)



(b)



(c)

Figure 3.14. Normalized confusion matrices evaluated (a) PointNet, (b) PointNet++, and (c) RandLA-Net.

Figure 3.15 indicates the semantic segmentation results on the testing dataset. All these models could achieve satisfactory segmentation accuracy in the underground parking lot scene. However, there are still some misclassified points. The wall class tends to be misclassified as pillars and vehicles. These missegmentation results of PointNet, PointNet++, and RandLA-Net may be due to the proximity between the two objects in this dataset. In addition, some portions of walls and pillars are connected, which also leads to misclassification. Furthermore, as spatial information encoding was completed within a small sphere, the pillars could present similar curvature structures as the walls, resulting in extracting similar geometric information. On the whole, all these three models could deliver accurate point-based segmentation results in a GNSS-denied environment, while RandLA-Net preserves the best estimation of the point labels.

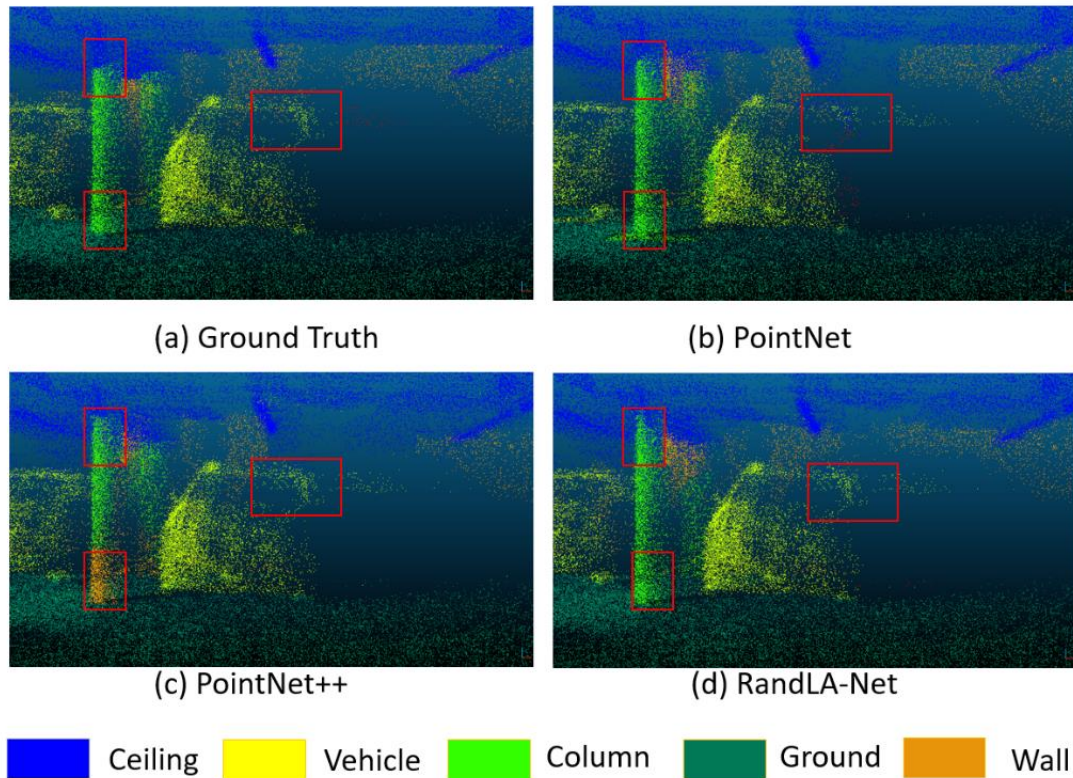


Figure 3.15 Point cloud semantic segmentation (a) Ground truth, (b) PointNet, (c) PointNet++, and (d) RandLA-Net.

3.7 Chapter Summary

This chapter presents WATiMMS, a dataset that can be used for sensor fusion, 3D reconstruction, trajectory localization and semantic segmentation. The proposed dataset addresses the current lack of a dataset in the indoor data domain that can be used to unify and validate multi-sensor calibration algorithms, scene segmentation algorithms, and trajectory validation algorithms. The dataset has two kinds of LiDAR data, three kinds of camera data, and IMU data. The manually labeled scene segmentation data and the motion odometry data of the platform are also provided.

First, based on the characteristics of indoor scenes, this chapter proposes a modular mobile platform for data collection. The platform uses a collection of multiple sensors to summarize a data normalization model for different forms of devices. On this basis, the data collected by the platform is temporally synchronized and the calibration method for multiple sensors is presented. Next, the collected data is further optimized by a keypoint graph optimization method. The method uses a total station to collect keypoints and optimizes the motion trajectory obtained by the SLAM algorithm with the keypoints to obtain a motion odometer that approximates the ground truth. Finally, the collected indoor data are manually calibrated so that the data can meet the requirements for validating the indoor scene segmentation algorithm. By testing and analyzing the dataset, the results show that this dataset, using automatic segmentation algorithms such as PointNet, PointNet++, and RandLA-Net, can obtain high scene segmentation accuracy. The experimental results show that RandLA-Net has the highest correct rate of 73.0% in the mIoU result, PointNet++ has the highest correct rate of 80.4% in the Precision result, RandLA-Net has the highest correct rate of 78.6% in the Recall result, and PointNet++ has the highest correct rate in the F_1 -score. PointNet++ has the highest correct rate of 77.8% in F_1 -score.

Chapter 4

Target-free LiDAR-Camera Automatic Calibration

This chapter introduces a new approach to reaching automatic multi-sensor calibration. Accurate sensor calibration can provide a vital data foundation for processing multiple sensors data. This chapter focuses on the target-free automatic calibration algorithm and the matching failure detection and error elimination algorithm. Specifically, the theoretical concepts of point cloud data calibration, data acquisition, and coordinate systems are discussed in Section 4.1. The target-free automatic calibration based on the same trajectory constraints is described in Section 4.2. The source of the calibration error and the solution are discussed in Section 4.3. Section 4.4 summarizes the chapter.

4.1 Introduction

Compared to outdoor environments, indoor environments lack GNSS signal reception and have complex structural layouts, which pose significant challenges for both 2D cameras and 3D LiDAR sensors in indoor mobile mapping [83]. Limited illumination conditions and occlusion coverage are common problems. Especially, in underground parking lots, tunnels, escape routes and other weakly textured areas (areas lacking 3D textures and monotone 2D colours), single LiDAR sensor data acquisition can cause significant difficulties for 3D reconstruction [84]. Cao et al. [85] used a low-channel LiDAR, a camera, and a brushless motor with a crank-link system to replace the traditional LiDAR for 3D object detection. Yin et al. [86] used 3D LiDAR with a monocular camera and an inertial measurement unit (IMU) for a SLAM algorithm. Ye et al. [87] and Xie et al. [88] proposed a target-

based calibration method using LiDARs and cameras together. Also, multi-sensor indoor mapping platforms have limitations on the device size and weight, so there is a move towards low-cost and fast data collection [89].

In contrast to the single-sensor method, the multi-sensor algorithms require external calibration of the sensors. Existing approaches focus on two main categories: target-based calibration and target-free calibration. The target-based calibration uses the same objects or building structures in different sensors' fields of view (FoVs) to extract the external parameters between sensors [90]. The targetless calibration obtains the coordinate conversion relationship by satisfying certain constraint relationships with the change of pose after sensor motion [91][92]. However, in the actual calibration process, the illumination of indoor environments, the occlusion of indoor objects, and even the data accuracy of the camera and LiDAR sensors can affect the calibration quality. Most current calibration algorithms are designed based on 360° LiDAR with a specific friendly environment (e.g., no occlusion, sufficient light, and clear 3D texture). However, the high price of 360°LiDAR hinders its large-scale applications in indoor scenarios. Solid-state LiDARs are cheaper and have a denser point cloud within their limited FoVs. It is an ideal platform setup for data collection in indoor environments. Some of the challenges currently encountered in the external calibration of narrow FoV LiDAR and cameras.

4.2 Point Cloud Calibration

Traditional point cloud calibration methods can be generally divided into two categories: Coarse Calibration and Fine Calibration. Coarse calibration is to find a transformation matrix that aligns the overlapping regions of two partially overlapping point clouds when the initial relative positions of the two clouds differ significantly. Fine calibration is to find a transformation matrix that aligns the

overlapping areas of two point clouds when the initial positions are better. The exact calibration of the two 3D point clouds requires a good initial position.

4.2.1 Coarse Registration

In the coarse calibration stage, the point cloud coarse calibration methods are classified according to the features used to determine the correspondence. The coarse calibration method can be divided into coarse calibration based on local feature description and coarse calibration based on spatial geometric relationship constraints. In the following, we will use the following sections to introduce them systematically in Sections 4.2.1.1. and 4.2.1.2.

4.2.1.1 Coarse Calibration Based on Local Feature Description.

The coarse calibration method based on a local feature description is divided into the following three steps: (1) design local feature descriptors with rigid body transformation invariance and strong descriptive capability, high robustness, high timelines, and high compactness; (2) calculate the feature similarity based on the feature descriptors and perform feature matching based on the similarity to establish the correspondence between the local feature descriptors of the source and target point clouds, thus establishing the correspondence between points; and (3) reject the wrong matching pairs and solve the transformation parameters. Usually, when the volume of point cloud data is large, detecting a number of sparse key points with strong distinguishing abilities in the initial point cloud is a widely used preprocessing scheme. Since the detection of keypoints is not the focus of this chapter, it will not be discussed further.

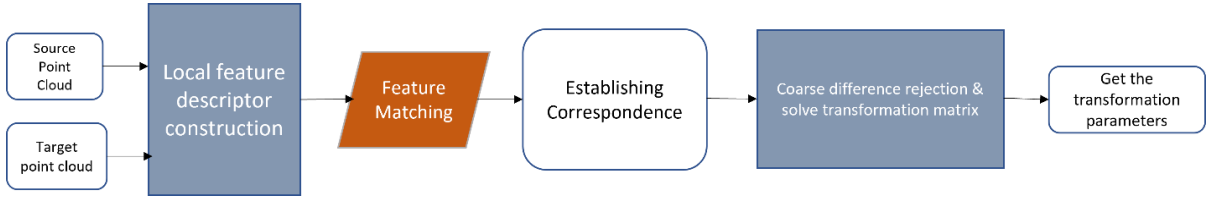


Figure 4.1 Local feature desc coarse alignment framework.

Feature matching, also known as feature descriptor matching. Its primary goal is to establish correspondence between feature points. This is achieved by matching the feature descriptors in both source and target point clouds. The matching is based on the similarity of the feature descriptors. Hence, a relationship is created between the descriptors and corresponding feature points. Compared with local feature descriptors, there is a lack of research on feature-matching strategies. In general, the existing feature matching methods mainly contain the following four basic strategies: (1) threshold matching, which directly selects the matching pair whose descriptor similarity is greater than the threshold; (2) Nearest Neighbour (NN) based matching, which directly selects the matching pair with the greatest descriptor similarity; (3) Nearest Neighbour Distance Ratio (NDR) based Neighbour Distance Ratio (NNDR) matching strategy, i.e., selecting the matching pair whose ratio between the maximum similarity of the descriptors and the second maximum similarity is less than a set threshold; and (4) Bidirectional Nearest Neighbour Distance Ratio (BNNDR) matching strategy, i.e., selecting the matching pair with the largest nearest neighbour distance ratio for both the forward and reverse descriptors.

4.2.1.2 Coarse Calibration Based on Spatial Geometric Relationship Constraints

The method of coarse calibration based on spatial geometric relationship constraints is divided into three key steps. The first step involves the design of spatial geometric features that exhibit invariance to rigid body transformations and possess strong descriptive ability, high robustness, high timeliness,

and compactness. The second step calculates the feature similarity using these spatial geometric features and performs feature matching based on the calculated similarity, thereby establishing correspondence between the spatial geometric features of the source and target point clouds. This, in turn, establishes point correspondences. The third step then involves the rejection of incorrect matching pairs and the resolution of transformation parameters.

When handling large volumes of point cloud data, a prevalent pre-processing scheme often involves detecting sparse key points with strong distinguishing ability in the initial point cloud. However, as the detection of these key points is not the focal point of this section, it will not be elaborated further.

4.2.2 Fine-grained Registration

The Iterative Closest Point (ICP) algorithm proposed by Besl and Mckay proposed in 1992 is one of the well-known algorithms to solve the rigid fine alignment problem. As shown in Figure 4.2, it updates the transformation matrix (R, t) by iteratively performing the matching point finding and least squares solution to minimize the error function $E(R, t)$ of the source and target point clouds P and Q (Eq. 4.1).

$$E(R, t) = \frac{1}{n} \sum_{i=1}^n \| q_j - (Rp_j + t) \|^2 \quad (4.1)$$

where n is the number of nearest point pairs, p_i is a point in the source point cloud P , q_j is the nearest point corresponding to p_i in the target point cloud Q , R is the rotation matrix, and t is the translation vector.

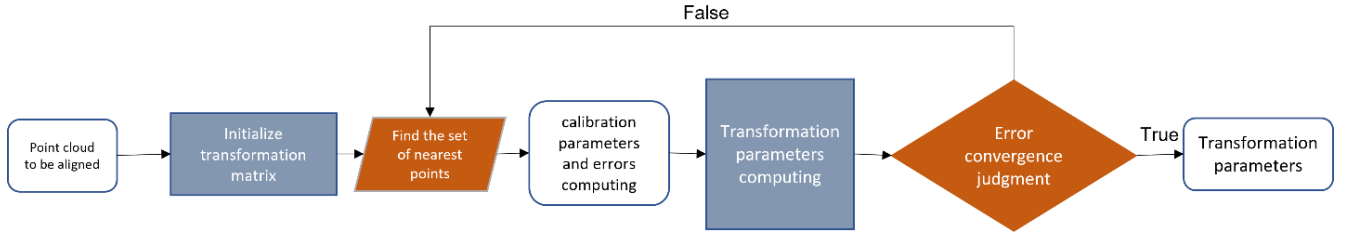


Figure 4.2 Basic flow of ICP algorithm

4.3 Six-degree-of-freedom Pose Estimation

As shown in Fig. 4.3 (a), the LiDAR backpack system used in this chapter is composed of a Velodyne VLP-16 [6] and a panoramic camera. Specially, 16 sets of distance detection data are captured in each LiDAR at the increment of 2° with the vertical horizon of LiDAR of 30° (from -15° to $+15^\circ$) (Fig. 3.1(c)). The LiDAR serves as the core of the point cloud generation, and this work converts the collected data according to the coordinate system of the LiDAR placed horizontally on top. So that the data from both the 45° obliquely placed LiDAR and the 360° camera are converted into the coordinate system of that horizontal LiDAR. This means that the local coordinate system, in this case, is considered to be the coordinate system of the LiDAR. The calculation process of the point cloud in the local LiDAR coordinate system is presented in Eq. 4.2.

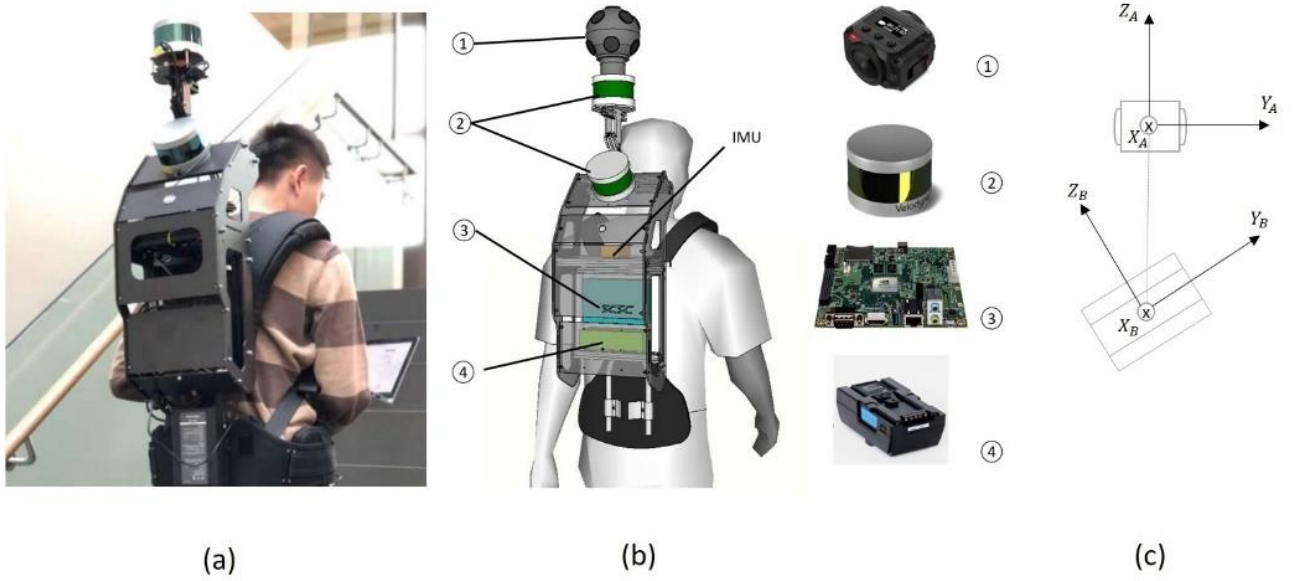


Figure 4.3 The LiDAR backpack system: (a) 3D LiDAR backpack system exterior, (b) backpack internal structure and sensor settings, and (c) LiDAR and camera external coordinate system.

The LiDAR-camera coordinate system is shown in Figure.4.3(b). The camera is mounted vertically above the LiDAR A. And LiDAR B is mounted below LiDAR A at an angle of 45°. The point cloud calibration process of LiDAR in the local Cartesian coordinates is presented in Eq. 4.2.

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} R * \cos(\omega) * \sin(\alpha) \\ R * \cos(\omega) * \cos(\alpha) \\ R * \sin(\alpha) \end{bmatrix} \quad (4.2)$$

$$P_{global} = P_{LaserA} + T_{cali} * P_{LaserB} \quad (4.3)$$

we seek the coordinate relationship between the camera and the LiDAR B and transforming the coordinate system of the LiDAR B into the camera coordinate system via the coordinate relationship matrix T_{cali} to unify the coordinate systems of both. In this way, a complete set of 3D data can be provided for subsequent data processing.

4.4 Target-free Automatic Calibration Based on Trajectory Constraints

Target free automatic calibration is proposed on the basis of the same trajectory constraints. This finding refers to the corresponding transformation matrix through minimizing the objective function with the same trajectory assumption after data synchronization of the camera and LiDAR data as the objective function constraints. This method utilizes trajectories as a reference for the inter-data optimization to avoid the unnecessary pre-processing step of target-based calibration. Specifically, the camera-to-LiDAR registration relationship is obtained by exploiting the features that the LiDAR and the camera trajectories are similar under a rigid connection structure. Moreover, it is necessary to have relatively stable two LiDAR and camera trajectories. The main contribution of this chapter also lies in solving the problem of optimizing the two trajectories in the case of large errors.

With the optimization, the trajectory of the camera motion as well as the LiDAR motion trajectory, can be obtained. The camera motion R_{Camera}^i , t_{Camera}^i and LiDAR motion R_{Lidar}^i , t_{Lidar}^i at any given moment satisfy:

$$R_{\text{Camera}}^i = R R_{\text{Lidar}}^i \quad (4.4)$$

$$R_{\text{Camera}}^i t = R t_{\text{Lidar}}^i + t \quad (4.5)$$

The minimization nonlinear optimization cost function is introduced, and the mathematical model can be expressed as:

$$R = \arg \min_R \sum_i |R_{\text{Camera}}^i - R R_{\text{Lidar}}^i| \quad (4.6)$$

The optimized R and t can be found by a least-squares method. The above serves as a general solution to the calibration problem, and the following section explains specifically if the optimization of trajectories is achieved.

4.4.1 Integration of LiDAR Trajectory and Camera Pose

Firstly, the LiDAR-IMU tightly coupled trajectory estimation and map construction based on a LiDAR-SLAM (LIO-SAM) framework is implemented using the built-in IMU and point cloud data [96]. LiDAR obtains a more accurate sub-map M with the assistance of IMU [97]. The LiDAR trajectory obtained by this algorithm is not equivalent to the ground truth. Still, it can be approximated as an accurate trajectory due to the negligible small cumulative error over a short period. Then, the camera image is used separately for the visual SLAM calculation to obtain the camera trajectory information [98]. Hence, the probabilistic roadmap in the visual SLAM is set up using the feature that the LiDAR trajectory approximates the ground truth. The probabilistic roadmap is not a precise position, but it ensures that the visual SLAM trajectory drift is within a controllable range. It can effectively cope with the long-time drift of visual SLAM, which cannot achieve loop closure detection for a period of time. The pose diagram after adding the probabilistic roadmap of the LiDAR trajectory is shown in Figure. 4.4.

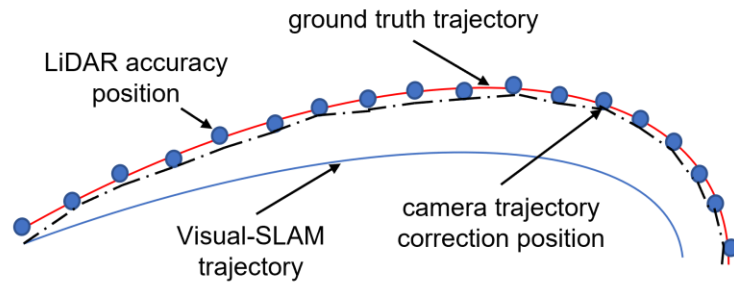


Figure 4.4. Visual-SLAM trajectory correction and optimization in accordance with LiDAR trajectory.

This method uses a 6D vector $\xi \in \mathbb{R}^{1 \times 6}$ to express the camera poses, and the transformation between any two poses (e.g., positional edges) in the set of poses $\varepsilon = \{\xi_1, \xi_2, \dots, \xi_n\}$ can be expressed as:

$$\Delta \xi_{ij} = \xi_i^{-1} \cdot \xi_j^{-1} = \ln[\exp(-\widehat{\xi}_i) \exp(-\widehat{\xi}_j)]^\vee \quad (4.7)$$

where $\wedge$ is the vector antisymmetric matrixing operation and $\vee$ is the anti-symmetric matrix-to-vector operation. $\Delta \xi_{ij}$ denotes the motion between neighboring poses motion, which the transformation matrix can be also described by:

$$T_{ij} = T_i^{-1} T_j \quad (4.8)$$

where T_i and T_j denote the corresponding positional transformation matrix of nodes i and j , respectively. The existence of errors e_{ij} makes the Eq. (4.7) or (4.8) not hold exactly in practice. To solve this problem, the idea of positional optimization is used, which is to construct the least squares error to optimize the variables and minimize the sum of squared errors.

$$\begin{aligned} e_{ij} &= \ln(T_{ij}^{-1} T_i^{-1} T_j)^\vee \\ &= \ln\{\exp[(-\widehat{\xi}_{ij})'] \exp(-\widehat{\xi}_i) \exp(-\widehat{\xi}_j)\}^\vee \end{aligned} \quad (4.9)$$

After the derivation of the error function with respect to the optimization variables, the positional optimization algorithm can be expressed as follows:

$$\xi_{\text{new}} = \min_{\xi} \frac{1}{2} \sum_{i,j \in \varepsilon} e_{ij}^T \Sigma_{ij}^{-1} e_{ij} \quad (4.10)$$

where ξ_{new} is the optimized pose, where Σ_{ij}^{-1} is the information matrix. It is the inverse of covariance matrix for the positional estimation variables. The main work in this section is to use the LiDAR trajectory pose data ξ_i brought from Eq. 4.7 to Eq. 4.10 to obtain the optimal pose estimation for visual SLAM.

After the camera's initial time and position results are aligned with the LiDAR time, the camera position point ξ_i and its corresponding LiDAR trajectory data observation value g are obtained after N poses. The alignment error is estimated first. Then, the LiDAR local coordinate system position data and the corresponding poses are integrated into this estimate to control the overall error within a specific range. It is assumed that the estimation accuracy of the image alignment position is linearly related to the camera movement speed and conforms to the Gaussian distribution model. The estimation accuracy is inversely proportional to the system movement speed. Specifically, when the system movement is too violent, the difference between the distance or rotation of the two images becomes more significant over time, decreasing the image pose estimation accuracy.

Assume that the system moving speed is $v = S/t$, where t is the image alignment sampling period constant. It can also be taken as being the frequency of the camera in the system. and S is the position estimation of two points of the total distance. Let Σ_{r0} be the camera position estimation covariance matrix when the velocity is 0 and define $\Sigma_{rv} = v\Sigma_{r0}/k$ as the image position estimation covariance matrix in the case of velocity v , where k is the image position estimation error gain constant. In practice, only observations with error $S' = S + \delta S$ can be obtained, and assuming that δS increases linearly with S , the point position estimation covariance matrix for the case of velocity v is defined as:

$$\Sigma_{rv} = \frac{(v + \lambda)\Sigma_{r0}}{k} \quad (4.11)$$

where λ is the error estimated of the position covariance at velocity v . According to Eq. (4.11), the point covariance matrix Σ_{ir} ($i=1, 2, \dots, n$) of the position pose at any moment can be approximated (i is the system position sequence). The accuracy estimation matrix for the integration of arbitrary camera pose and LiDAR trajectory position data can be expressed as:

$$A = \sum_{i=1}^N \Sigma_{ir}^{-1} + \Sigma_{gj}^{-1} \quad (4.12)$$

where Σ_{ir} is the covariance matrix of each pose node, Σ_{gj} is the covariance matrix of the current LiDAR trajectory position data. A is the data integration accuracy estimation matrix. Then, the current pose node position parameters of the image and the LiDAR trajectory position after the integration of the pose position parameters are:

$$\xi_{iT} = A^{-1} \Sigma_{ir}^{-1} T_r + A^{-1} \Sigma_{gj}^{-1} g_j \quad (4.13)$$

where T_r is the current system position data. This part of LiDAR trajectory and camera pose data integration uses the Kalman estimation for the image trajectory optimization model construction to achieve the purpose of LiDAR trajectory to camera pose optimization.

4.4.2 Target-based Experiment

The target-based calibration approach uses a calibration board of size 650×500 mm, with each frame of the board measuring 50×50 mm (see Figure 4.5). The board was placed in a room at a distance of 3 meters from the data collection device. To minimize the impact of manual annotation results on LiDAR point cloud density and image clarity, 10s of manual calibration data was considered a set. Accordingly, a total of 10 sets of experiments were conducted in two scenes (i.e., the office walkway and the underground parking garage.) to ensure the consistency of the results of the comparison

experiments. The camera's internal parameters were calculated from the same calibration plate at a distance of 1 m from 20 sets of photos with different angles and positions. The same internal parameters were also used in the experiments of the trajectory-based calibration algorithm. The experimental results were uniformly compared using images and point clouds with calibration plates in both scenes.

In each data set, the vertices of the target are $P1 \sim P4$ and $P1' \sim P4'$. The error of the calibration matrix was analyzed according to the obtained calibration results. According to the calibration matrix, the mean distance and the total minimum variance between the corresponding points can be obtained. The data and projection results are shown in Figure 4.5, and the error analysis results are listed in Table 4.1

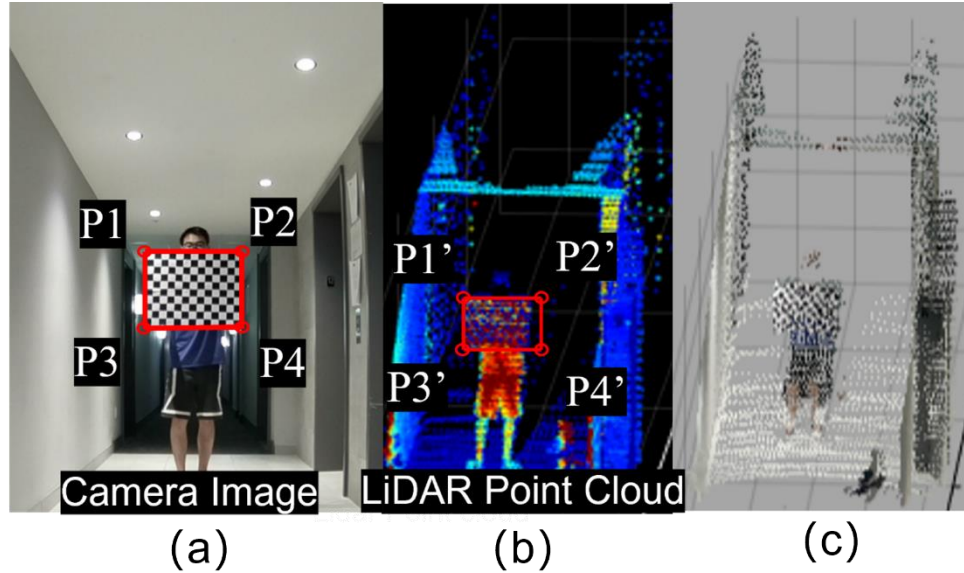


Figure 4.5. The target-based calibration under (a) image, (b) LiDAR point clouds, and (c) image information registered to the point cloud.

4.4.3 Trajectory-based Experiment

Two data sets were acquired using a multi-sensor mobile mapping platform in two scenes. Each data set contains 530 frames of the point cloud, IMU, and image data. The data of each set has been time-

synchronized by the method introduced in the previous chapter. The frequency of the synchronized data is 10 FPS. The LIO-SAM algorithm was used to compute the LiDAR data stitching while obtaining the moving trajectory and pose data. Considering that the speed of data collection may affect the accuracy of the data. Before starting the official experiment, we measured three sets of data within the normal walking range of a pedestrian, approximately 1.0m/s, 1.2m/s, and 1.5m/s (the average speed was obtained from the measured distance and time spent). The obtained results show that the walking speed is negligible to the results at the current LIDAR and camera frequencies.

Furthermore, according to the ORB-SLAM method [99], the camera also gets the running trajectory. Since the sensors are rigidly connected, the camera trajectory results can be scaled by the length of the two trajectories. The camera trajectory was iteratively optimized using the graph optimization algorithm presented above. Figure 4.6 (a) shows the results obtained in the underground parking lot scenario. Optimizing the camera trajectory keeps the camera and LiDAR trajectories in a stable relative position. The position relationship between the camera and the LiDAR was found using the trajectory. Hence, external rotation and translation matrices were obtained. As seen by the results in Figure 4.6(b) and 4.6(c), the optimized two trajectories achieve considerable improvements and the relative position relationships remain stable over the time that the system platform is moved. Through experimentation, it was discovered that in a well-lit office walkway environment, which has rich textures, it is possible to obtain high-quality trajectory data using the SLAM results. The camera trajectories and LiDAR trajectories match almost perfectly after scaling, and the proposed method is not effective for such minor improvements. This is because the scene has enough textures, allowing for accurate tracking and mapping. In contrast, if the environment lacks sufficient textures, it may be more difficult to obtain

high-quality trajectory data using Visual-SLAM. Moreover, it is in this environment that the proposed algorithms gain the greatest advantage.

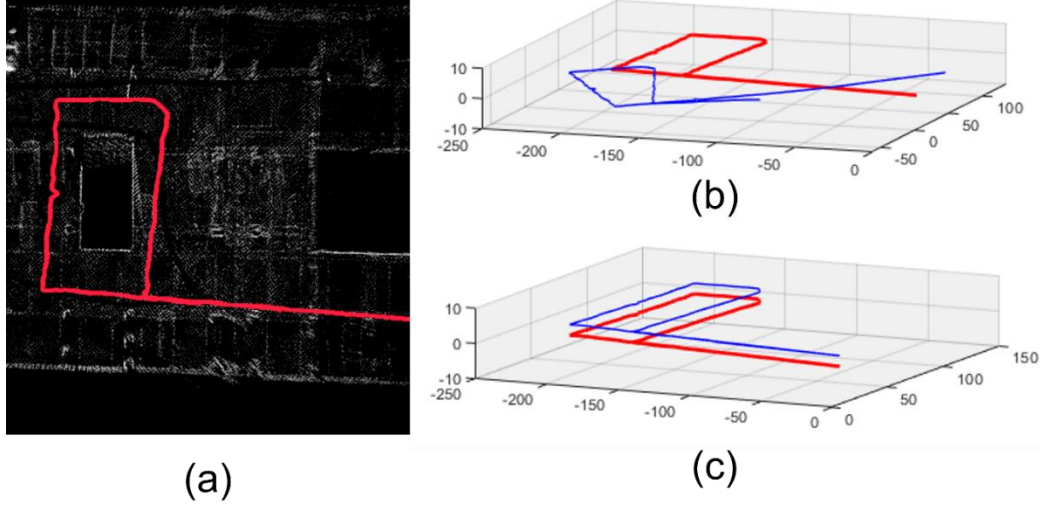


Figure 4.6. Trajectory-base calibration results: (a) the LIO-SAM obtains LiDAR trajectory, (b) optimized scaled camera (blue) trajectory and LiDAR (red) trajectory, and (c) the camera (blue) trajectory and the LiDAR (red) trajectory obtained by the graph optimization method.

Figure. 4.7 illustrates the projection of the results obtained from calibrating a set of data collected from an underground parking lot using the proposed method. The projected data are accurately registered, particularly in regard to the pillars, vehicles, and the transition area of the platform where the pillars and the ceiling meet. It is worth mentioning that a part of the wall as a background is blank because the point cloud in Figure 4.7(b) is obscured by a column. This missing data can be observed clearly in Figure. 4.7(d). This issue can be resolved by performing multiple scans of the data from various angles. The obtained external parameters are calculated and brought into the target-based data in the previous section. A set of calibration mapping data obtained according to the proposed method can be obtained again. Then, using the same calculation method, the mean distance and the total

minimum variance between the corresponding points can be obtained. Table 4.1 shows that our method has a clear advantage over the target-based calibration methods in indoor environments with low brightness and high occlusion.

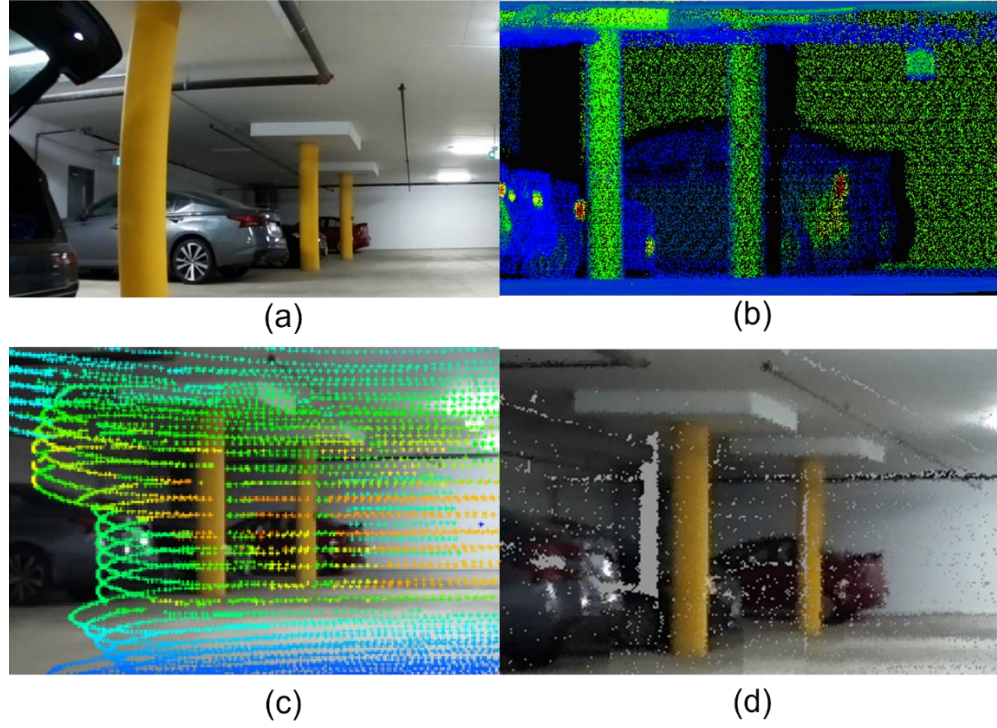


Figure 4.7 proposed method result (a) Images in the underground scene. (b) 10 s LiDAR point cloud. (c) Projection points to the photograph. (d) point cloud data with colour.

Table 4.1 Calibration Errors of Two Methods

Error	Target-based in Scenario A	Ours in Scenario A	Target-based in Scenario B	Ours in Scenario B
x (m)	0.076	0.025	0.248	0.015
y (m)	0.203	0.045	0.083	0.046
z (m)	0.055	0.027	0.116	0.085
Roll (deg)	6.672	0.615	2.323	0.539

Error	Target-based in Scenario A	Ours in Scenario A	Target-based in Scenario B	Ours in Scenario B
<i>Yall (deg)</i>	4.587	0.238	1.665	0.410
Pitch (<i>deg</i>)	2.230	0.475	9.183	0.115
RMS_T (<i>m</i>)	0.116	0.055	0.197	0.082
<i>RMS_R (deg)</i>	4.319	0.435	6.637	0.564

4.5 Chapter Summary

In this chapter, a novel automatic calibration method is proposed to achieve high accuracy data fusion in indoor environments. Since the trajectory generation in the SLAM algorithm does not depend on a single target as a reference, the method using the trajectory as the basis of calibration achieves stable acquisition of extrinsic parameters in an indoor environment with insufficient light and object occlusion. Meanwhile, graphical optimization solves the problem that the visual SLAM trajectory offset cannot determine the stable extrinsic. During the data acquisition process, our algorithm acquires the calculated trajectory information in real time and completes the external calibration of the sensor based on the optimized trajectory information. This method has less manual involvement, a high degree of automation, and does not require special targets are needed, which solves the problem of accurate and fast calibration in multi-LiDAR systems. According to the experimental results, the RMSE can be improved from a maximum of 6.637° to 0.564° , and the RMSE of translation can be improved from a maximum of 0.197 m to 0.082 m. These results show that the calibration accuracy is significantly improved compared to other target-based methods.

Chapter 5

Indoor LiDAR-SLAM and Point Cloud Semantic Segmentation

High-precision geolocation has revolutionized transportation, commercial logistics and personal trips. These applications have benefited from the advancement of GNSS. It becomes crucial to have stable positioning in a GPS/GNSS environment with the acceleration of urbanization and the rapid development of automatic driving technology. However, GPS/GNSS signals cannot be stably acquired in a variety of conditions. For example, GPS signals are susceptible to interference, seriously degrading the positioning accuracy, due to the multi-path effect of urban high-rise buildings. It is almost impossible to receive stable GPS/GNSS signals due to severe occlusion indoors or in an underground facility. How to perform self-positioning of pedestrians and vehicles in an indoor/underground garage environment without GNSS signals (or the GNSS-Denied environment) is a challenging research problem. In particular, the underground garage parking application in the unmanned driving system not only relies on the sensors mounted on the vehicle for environmental data analysis, but also requires the HD indoor 3D semantic map for accurate navigation [100,101].

This chapter introduces a high-precision 3D SLAM and point cloud semantic segmentation of LiDAR, including a front-end odometry algorithm using twice-matching and the back-end optimization algorithm based on the sub-map and a prior pre-processing deep network for semantic segmentation. To be specific, data acquisition and the coordinate system are briefed in 6.1; the six-degree-of-freedom attitude estimation is illustrated in 6.2; and the closed-loop detection and back-end optimization

algorithm based on the sub-map is introduced in 6.3. Finally, the indoor 3D point cloud semantic segmentation is described in 6.4.

5.1 Data Acquisition and Coordinate System

Chapter 4 introduced the 3D PLS and its coordinate system. Two Velodyne VLP-16 and a panoramic camera are adopted in the system. Each LiDAR establishes 16 sets of distance detection data at the increments of 2° . The automatic calibration algorithm discussed in Chapter 4 is utilized in this chapter to calibrate multi-sensor data.

5.2 Six-degree-of-freedom Pose Estimation

Once data calibration is complete, calibrated data is fused and synchronized before being sent to the front-end framework for the 6DoF pose estimation of sensors and the incremental construction of a sub-map. The LOAM algorithm proposed by Zhang et al. [102] use the feature method for pose estimation. Firstly, the current point cloud frame is linearly scanned for a rough extraction of features using the three-dimensional curvature. Secondly, after unstable feature points are eliminated with a screening strategy, stable feature points left behind are used for triangulation and the estimation of the three-dimensional position feature points. Finally, the pose of the sensor can be inversely calculated with positions of these feature points. Although this method works well in scenes with a wealth of geometric features, it might be subject to a local minimum due to the degradation of features once the scene feature becomes sparse. Since the LiDAR backpack system is operated indoors, the LOAM algorithm might fail because of failing to find out feature points in the geometric feature environment such as narrow tunnels, and symmetric long corridors, etc. Inspired by the g-ICP algorithm, the front-end scheme based on the probability matching framework for odometry estimation can be used in this chapter, so as to solve the robust SLAM problem in the weak-feature environment. The core idea of the

method is to use the surface variance of the point cloud as the calculation weight for matching. Therefore, weighted distribution is conducted on the matching through calculating covariance of 20 neighbors corresponding to specific points in the searching process of neighborhood points, so as to improve the registration accuracy of the sparse point cloud and enhance the success rate of registration in the environment with weak features.

$$f_i^0 = T_{i-1} \cdot f_i \quad (5.1)$$

$$\Delta T_i = \underset{\Delta T_i}{\operatorname{argmin}} \sum_{j=1}^n d_j^{(\Delta T_i)^T} \left(C_j^{f_i^0} + \Delta T_i C_j^{f_{i-1}^0} \Delta T_i^T \right)^{-1} d_j^{(\Delta T_i)} \quad (5.2)$$

$$\begin{bmatrix} f_i^0 \\ T_i' \end{bmatrix} = \Delta T_i \cdot \begin{bmatrix} f_i^0 \\ T_{i-1}' \end{bmatrix} \quad (5.3)$$

Inter-frame registration: The specific steps of inter-frame registration are given in Eq. 5.1, 5.2, and 5.3. Two adjacent frames of calibrated data, f_i , and f_{i-1} from bi-static LiDAR are adopted for matching the point clouds in the front and back frames. And the matched transformation matrix T_i' can be used for initial rough odometry estimation. Firstly, the point cloud f_i of the current frame is transformed from the LiDAR local coordinate system to f_i^0 in the world coordinate system via the global trajectory data T_i' of the previous frame based on the eq. 5.1. Secondly, the probability model framework is used as the matching calculation weight through the surface variance of the point cloud in accordance with Eq. 5.2. Of which, $C_j^{f_i^0}$ and $C_j^{f_{i-1}^0}$ are co-variances on the surface of the point cloud of the corresponding data frame. $d_j^{(\Delta T_i)}$ is the eigenvector matrix. In the neighborhood search process, the covariance of 20 neighbours corresponding to a specific point is calculated, and the weights are

assigned to the matching, and the registration of the front and rear frame point clouds is performed to calculate the registration matrix ΔT_i . The global trajectory data T_{i-1} at the last moment is updated to T'_i with the estimated rotation matrix ΔT_i . Meanwhile, the current point cloud frame f_i^0 is fine-tuned, as shown in Eq. (5.3). And the trajectory is transformed as an initial estimation for the next fine registration.

$$\Delta T'_i = \underset{\Delta T'_i}{\operatorname{argmin}} \sum_{j=1}^n d_j^{(\Delta T'_i)^T} \left(C_j^{f_i^0} + \Delta T'_i C_j^{f_i^0} \Delta T'^T_i \right)^{-1} d_j^{(\Delta T'_i)} \quad (5.4)$$

$$\begin{bmatrix} f_i^0 \\ T_i \end{bmatrix} = \Delta T'_i \cdot \begin{bmatrix} f_{i-1}^0 \\ T'_{i-1} \end{bmatrix} \quad (5.5)$$

$$\begin{aligned} f_{i-1}^0 &= f_i^0 \\ M'_i &= M'_{i-1} + f_i^0 \end{aligned} \quad (5.6)$$

5.3 Loop-closure Estimation and Back-end Optimization Based on Sub-map

Accumulating data errors brought by environmental noise, and sensor noise, etc. and inevitable numerical calculation errors of the algorithm for a long time might become a core problem based on odometry estimation. As the running time of the system becomes longer, the problem will be more severe. Specifically, cumulative errors are usually presented in higher-order growth, resulting in large-area deviations in the map. In turn, the map with deviations has a significant impact on the front-end algorithm based on the map matching strategy. Therefore, how to effectively reduce this error and build a globally-consistent map has become an important research challenge in SLAM. In recent years, there has been an effective means of combining closed-loop detection and back-end optimization for solving the above problem. Thus, accurate and robust closed-loop detection has become a necessary

prerequisite for correct back-end optimization. However, the closed-loop detection based on a single frame for the multi-LiDAR sensors mentioned in this chapter fails easily due to the sparse characteristics of single-frame data and feature ambiguity, resulting in failed optimization. A closed-loop detection and optimization framework constructed by multiple frames of the sub-map is proposed with the following specific steps:

First of all, the framework is optimized according to the classic map of the SLAM system. The optimization vector P of the graph topology is given in Eq. (5.7). The vector P is composed of six parameters to be optimized $[(\phi_i, \theta_i, \psi_i), (x_i, y_i, z_i)]$. A generalized graph optimization model framework is defined in Eq. 5.8. Where, z_{i-j} refer to the edges of the graph, presenting the measured value between the nodes i , and j ; Ω represents the error weight matrix; and $e(P_i, P_j, z_{i,j})$ is the error equation with the role of calculating the similarity between (P_i, P_j) and $z_{i,j}$.

$$P_i = [R_i, T_i] = [(\phi_i, \theta_i, \psi_i)^T, (x_i, y_i, z_i)^T]^T \quad (5.7)$$

$$\begin{aligned} F(P) &= \sum_{\langle i,j \rangle \in E} e(P_i, P_j, z_{i,j})^T \Omega_{i,j} e(P_i, P_j, z_{i,j}) \\ P^* &= \arg \min_P F(P) \end{aligned} \quad (5.8)$$

According to [103], the error equation is defined as follows:

$$e(P_i, P_j, z_{i,j}) = t2v(Z_{i,j}^{-1}, (P_i^{-1}, P_j)) \quad (5.9)$$

The above is an optimization model for constructing a loop closure based on single-frame data. $t2v$ is an operation that converts a matrix into a vector. Nevertheless, it's difficult for the single-frame data

to seek loop closure for matching in its vicinity due to sparse features. What's worse, each frame should be searched in the global map for each loop closure estimation, which is computationally expensive. In this section, an optimization topology map construction and calculating framework based on the sub-map is proposed for the sparse characteristics of single frame point cloud. First of all, the LiDAR trajectory attitude and its corresponding sub-map are assumed as rigid constraints in this chapter. Based on this assumption, a sub-map is matched in the loop closure estimation position; and the obtained transformation matrix is used as the new measurement side information of the topological attitude map for adding to a new constraint matrix upon global optimization. A new constraint is provided by the information for subsequent optimization. The process is shown in Fig. 5.1 The rigid constraint is assumed as $\Delta T_{[Sub_0, Sub_1]} = \Delta T_{[P_i, P_{i-t}]}$. As shown in the equation, the sub-map relative to the positional transformation of the global map is related to the trajectory relative to the transformation rigidity of the global trajectory.

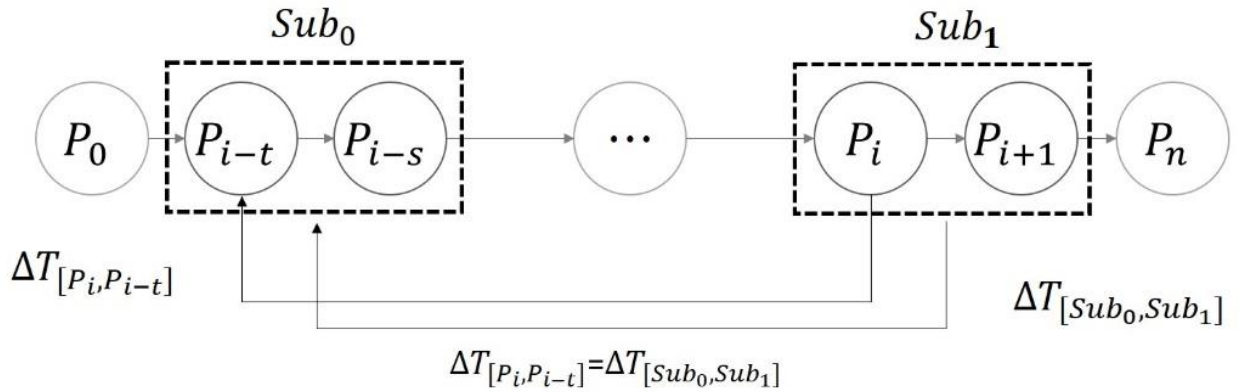


Figure 5.1 Topology graph optimization model based on sub-map loop closure estimation.

First of all, the sub-map M'_n , ($M'_n = M_{sub}^A$) and the corresponding trajectory T_n can be obtained from the front-end odometry algorithm. Then, the distance between the trajectories is used as an activation condition for the loop closure detection. In other words, once the distance between the current trajectory

attitude and the past historical trajectory attitude being less than the set threshold δ is detected, the point collection T_c of the nearest trajectory can be extracted from the closed-loop point. And then, LiDAR data corresponding to the point collection T_c can be used for constructing a sub-map, M_{sub}^B for the closed-loop detection, that is, the submaps, M_{sub}^A and M_{sub}^B are registered. The trajectory point set T_c is all the trajectory points from the beginning to the end of this closed loop point. The construction of the sub-map starting from T_c can be seen as a sub-map construction using the same method as the overall map construction. There is no difference in the construction algorithm from the overall map. The purpose of doing so is to add more information about the topological map edges, that is, the updated measurements derived from the attitude and the closed-loop points.

Secondly, the score of registering M_{sub}^A and M_{sub}^B are utilized as the basis for the successful loop closure estimation. If the score exceeds the set threshold, the loop closure estimation is considered auspicious. At this time, new measuring values $z_{i,j}$ can be calculated by the new transform H_{icp} generated by the registration together with the current attitude, P_j and loop closure point P_i . The measured value is optimized as new side information of the topology map. Finally, the global consistency point cloud map is reconstructed with the optimized attitude results.

5.4 3D Point Cloud Semantic Segmentation

Since the PointNet algorithm has an excellent effect on the 3D scene semantic segmentation, it's considered as a basic module of the back-end segmentation, as shown in Fig. 4.2. The 3D point cloud is regarded as input in PointNet. Each point is marked through the results of its grid training. Also, the marking result of each point is output. However, like other DNN-based methods, priori information of the model is scarcely utilized by PointNet. Therefore, taking into account a great deal of prior geometrical information indoors or in the underground garage, such as wall, ceilings, and cylinders, etc., a Prior Structure Information Filter (PSIF) is proposed to guide the training of PointNet.

By using the PSIF module and the basic PointNet module, this chapter gives a semantic partitioning model with front and back end, the priori structural information based PointNet model (PPM) is proposed. As shown in Fig. 4.2, PPM consists of three key components: the a priori information extraction module, the global information mining module, and the feature fusion module. The a priori module provides rich priori 3D shape information, so it can be regarded as an accelerated training module to guide the network. More importantly, the a priori module can provide supplementary information to improve the performance of the model. Moreover, the basis of the global feature extraction module is PointNet, which has been proven to perform complete point cloud global scene segmentation. Finally, the functional fusion module obtains the final fusion through combining global features and previous features, which can be characterized via the operational element multiplication.

Preliminary inference of geometric structure plays an important role in the 3D point cloud data processing. For example, the rough geometric structure can be used for depicting the contour of a 3D shape to initially confirm the classification of 3D objects. As the 3D point cloud consists of a set of point collection, its geometric structure is composed of the relationship between the points. Therefore, the shape of the 3D point cloud is determined by the connection between the points. The relationship between the point set and the 3D structure is given in this section. The most typical relationship is the object dimension feature, or the relationship between the property of the point set and the geometric features of the 3D point cloud.

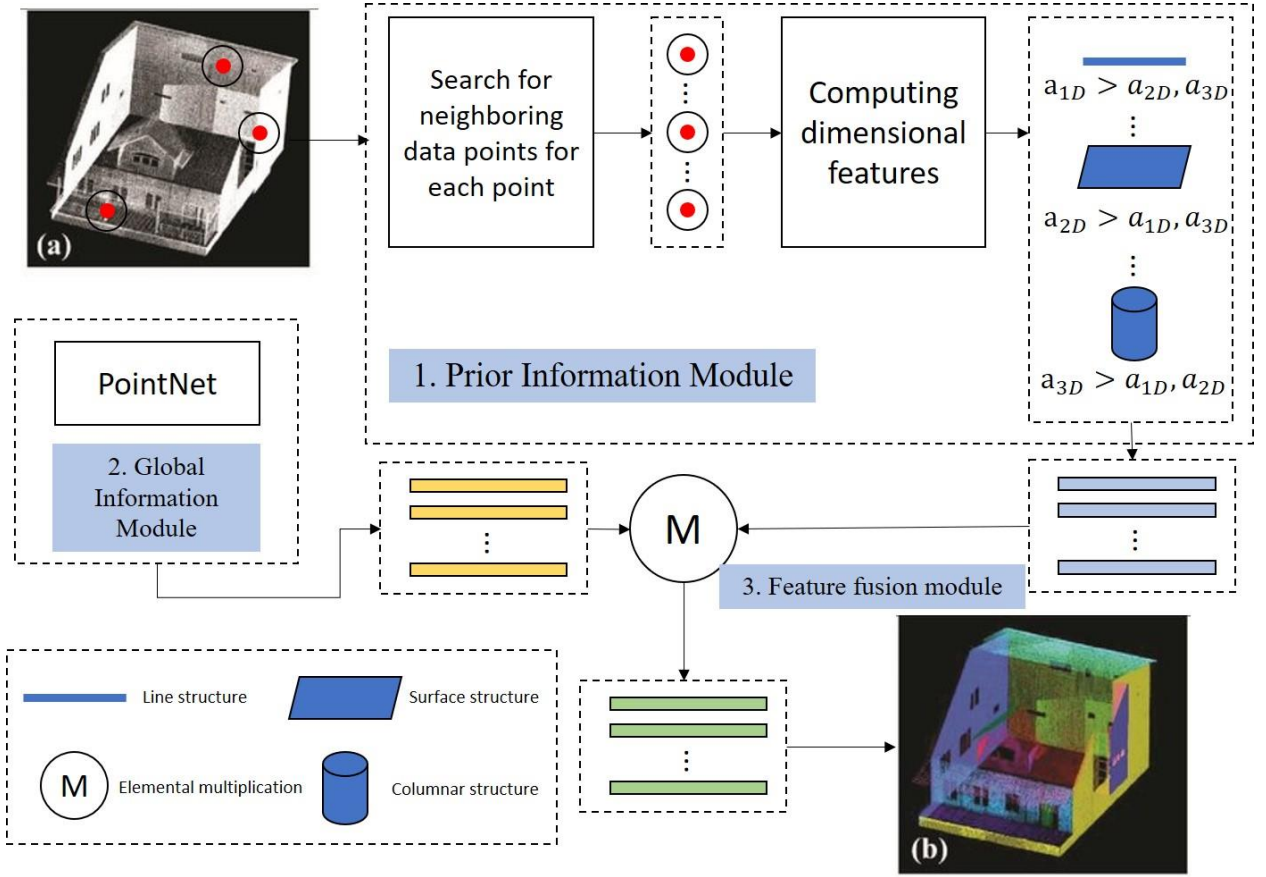


Figure 5.2 Semantic modeling algorithm flowchart: (1) prior information extraction module, (2) global information mining module, and (3) feature fusion module.

5.4.1 Dimensional Features

As for $P = \{p_1, p_2, \dots, p_n\}$, n is the number of the point. using principal components analysis method to get three eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$. If $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, and three eigenvectors are $(\vec{v}_1, \vec{v}_2, \vec{v}_3)$. The standard deviation σ_i for the corresponding eigenvector i can be expressed as follows:

$$\forall i \in [1, 3], \sigma_i = \sqrt{\lambda_i} \quad (5.10)$$

After that, the three-dimensional shape of the point cloud is presented by a directional ellipsoid. Three geometric features [104] are introduced in this chapter to describe geometric characteristics of the dispersion point (a_{3D}), the line (a_{1D}), and the plane (a_{2D}):

$$a_{1D} = \frac{\sigma_1 - \sigma_2}{\sigma_1}, a_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1}, a_{3D} = \frac{\sigma_3}{\sigma_1} \quad (5.11)$$

According to the relationship between the distribution of points and the three eigenvalues [103], if $\sigma_1 \geq \sigma_2$ and σ_3 , it can be assumed that a_{1D} is larger than a_{2D} and a_{3D} , when the point set P denotes a linear structure. If $\sigma_1 \approx \sigma_2 \geq \sigma_3$, a_{1D} is maximal and then the point set P denotes the planar structure. Finally, $\sigma_1 \approx \sigma_2 \approx \sigma_3$, which means that a_{1D} is larger than a_{2D} and a_{3D} , when the point set P denotes the scattered point structure.

In addition to the above dimensional features, this chapter also introduces the feature of full variance O , which is also derived from the standard deviation which is shown as follows.

$$O = \prod_{i \in [1,3]} \sigma_i \quad (5.12)$$

The full variance feature O proportional to the ellipsoidal volume allows to characterize the shape of the neighborhood and, in particular, to enhance one or two salient eigenvalues.

5.4.2 The Prior Structural Information Filter

For a given set of points $P = \{p_1, p_2, \dots, p_n\}$, combining the above 3D features and in the full variance feature, this chapter defines the prior module as a filter, which is a score vector of prior structure probability. $V^P = \{V_1^P, V_2^P, \dots, V_m^P\}$, where m is the number of categories and V_i^P is the prior structure probability score of the i -th category. The score vector V^P , which is not the probability itself, can be considered as the weight of each principal component. It is assumed that for 3D point clouds belonging

to the same category, the distribution of probability scores follows a normal distribution. This assumption is consistent with the real situation. And then, the vector V^p is calculated by the transformation F as follows:

$$F(a_{1D}, a_{2D}, a_{3D}, O) = V^p = \{V_1^p, V_2^p, \dots, V_m^p\} \quad (5.13)$$

And

$$V_i^p = 1 + f_i^D(x_i) + f_i^O(O) = 1 + \frac{\exp\left(\frac{(x_i - \mu_i^D)^2}{2(\delta_i^D)^2}\right)}{\delta_i^D \sqrt{2\pi}} + \frac{\exp\left(\frac{(x_i - \mu_i^O)^2}{2(\delta_i^O)^2}\right)}{\delta_i^O \sqrt{2\pi}} \quad (5.14)$$

where $x_i \in (a_{1D}, a_{2D}, a_{3D})$, with $\mu_i^D, \mu_i^O, \delta_i^D$ and δ_i^O are the necessary estimation parameters. Notice that x_i is determined by the i^{th} category of a priori knowledge. For example, the structure of the wall is represented by the planar structure, so that $x = a_{1D}$.

After obtaining the a priori structural probability score vector V_p , it is inserted into the result $V_{PointNet}$ which was generated by PointNet. The elemental multiplication method is applied to V_p and $V_{PointNet}$ to obtain the final score vector V_{final} after fusion:

$$V_{final} = V_{PointNet} \dot{V}_p \quad (5.15)$$

5.5 Experimental Results and Analysis

In order to quantitatively analyze the method consisting of three modules presented in this chapter. This chapter will first introduce the experimental environment and data, followed by two evaluation methods for the mapping algorithm, namely, point cloud quality assessment, and trajectory error analysis. Finally, the evaluation criteria for quantitative analysis will be given for semantic segmentation. The accuracy evaluation and the intersection over union (IoU) evaluation will be given respectively. The

definition of each evaluation method and the corresponding experimental results are described in detail below. The experimental environment and the number set are described as follows.

5.5.1 Experimental Data and Environment

Table 5.1 Indoor data collected by our mobile LiDAR platform (Wat-MMP)

Dataset	Size (m)	Scanning frames	Travel distance (m)
Parking Lot #1	$350 \times 270 \times 7$	1.96×10^6	1817.4
Parking Lot #2	$330 \times 500 \times 7$	7.53×10^6	2427.1
Underground tunnel #1	$200 \times 3 \times 7$	2741	131.5
Underground tunnel #2	$244 \times 3 \times 3$	2529	230.7
Building flyover #1	$170 \times 3 \times 3$	1867	118.1
Building flyover #2	$291 \times 3 \times 3$	2177	236.3

In the first set of control experiments, a handheld mobile platform is used to capture 3D point clouds of flyovers, underground tunnels and underground parking lots in multi-story buildings. For the back-end requirements of closed-loop, the trajectories are captured using "8" shaped movements (underground garage) or "0" shaped movements (flyover and tunnel), so that the trajectories form at least one closed loop. Figure 5.6 shows a representative map of the indoor environment reconstructed by the algorithm in this study: a flyover structure with a narrow corridor. Figures 5.5(c) show the point cloud map details and trajectories generated by SPC-SLAM. The data are synchronized by a synchronization trigger and pre-calibrated using the auto-calibration algorithm in Chapter 4. The detailed data and its attributes are given in Table 5.1.

In this experiment, two sets of control experiments are set in this chapter. The first group of experiments is the current excellent 3D mapping algorithm LOAM, which is a feature matching-based 3D mapping algorithm that can provide high accuracy front-end mapping but does not have back-end closed-loop detection and optimization. The second set of experiments is conducted by Livox 3D laser scanning equipment in the same environment, while the ground points of interest are measured with a total station. The global trajectory is then optimized using the method in Chapter 3, so that the generated point cloud is used as a ground truth to evaluate the trajectory accuracy and mapping quality of this chapter's method and the LOAM method.

5.5.2 Point Cloud Quality Assessment

In the first set of evaluation experiments, 20 points of interest for reference are selected from the point clouds obtained by the algorithm in this chapter, and the point clouds obtained by the control LOAM algorithm, respectively, $Q = [q_1, q_2, \dots, q_{20}]$. Also, the same position reference points obtained based on the total station measurements are $P, P = [p_1, p_2, \dots, p_{20}]$. In this chapter, we adopt manual alignment of point pairs and calculate the distance between corresponding points. Based on the assumption of rigid invariance of trajectory and corresponding point cloud, this section takes the distance of interest point on the point cloud data and the distance of reference point as the trajectory error. The specific calculation formula is as follows:

$$e_t(p_i, q_i) = \frac{\sum_{i=0, j=0, i < j}^n \|p_i - p_j\|^2}{\sum_{i=0, j=0, i < j}^n \|q_i - q_j\|^2} \quad (5.16)$$

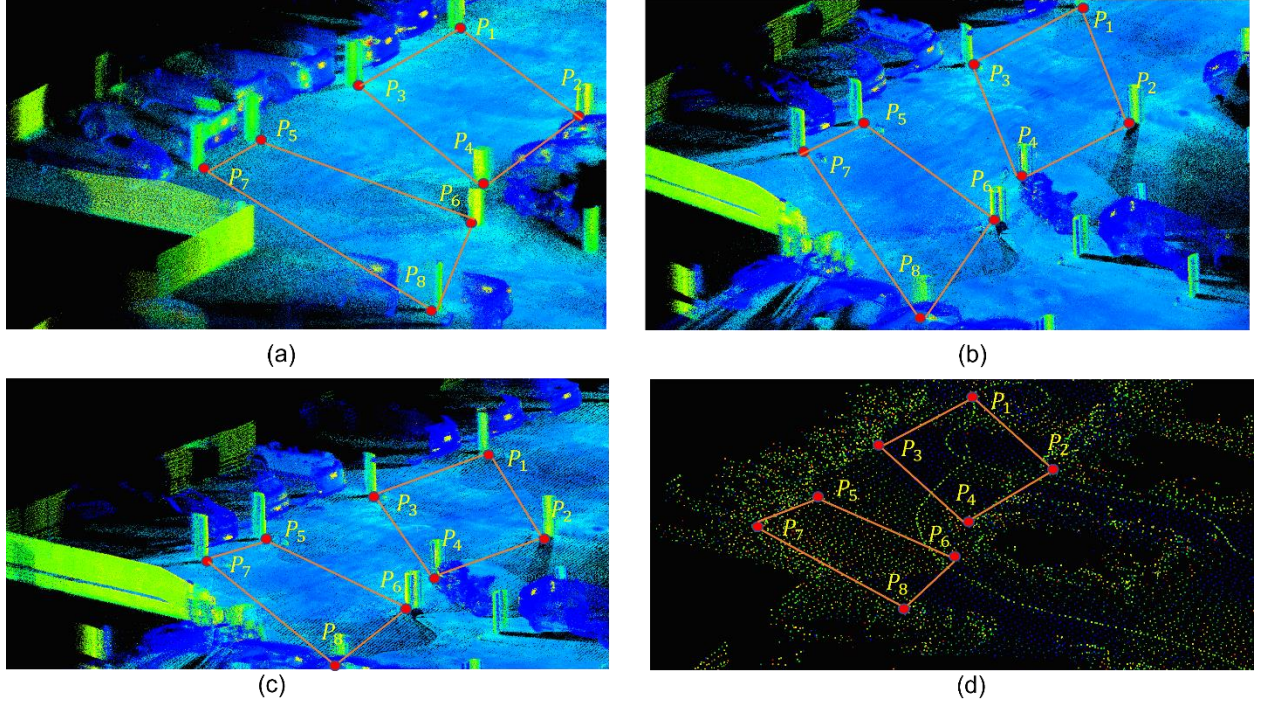


Figure 5.3 Point cloud quality evaluation and analysis: (a) LOAM algorithm, (b) front-end results of SPC-SLAM, (c) closed-loop detection results of SPC-SLAM, and (d) optimization results of total station using reference points.

Figure 5.3 shows the interest point and reference point selection of the static reference data and the corresponding results method compared with the LOAM method, respectively. It can be seen that the method of this chapter is more accurate than the LOAM method. The distance between the interest points and the reference points is also closer. Table 5.2 shows the results of error calculation between proposed method (front-end odometer and back-end closed-loop optimization) and LOAM method for map building using control points and measurement points, indicated by err_{cp} . err_{cp} is the overall error for every group of four closed-loop points. From the error data analysis, the accuracy of the results of this chapter's method is better than LOAM. especially for the parking lot data with high occlusion and obvious features, the accuracy of the trajectory of this section proposed algorithm is significantly better than LOAM. The proposed algorithm outperforms the LOAM algorithm based on feature filtering for

symmetric, and weak feature indoor environments. In particular, the proposed algorithm achieves 0.09% localization accuracy (236m travel distance, 0.17m cumulative error) in the data of flyover #2. Figure 5.5 shows the different 3D building results of the SPC-SLAM algorithm and the LOAM algorithm on the scanned data of the underground garage, and it is obvious that the LOAM algorithm results show a serious drift after entering the steering movement and heavily occluded area. However, the SPC-SLAM algorithm outperforms the LOAM algorithm for this environment.

Table 5.2 Error analysis of map building results

Dataset	LOAM			SPC-SLAM (only front-end)			SPC-SLAM(Optimized)		
	err_{cp1}	err_{cp2}	err_rate	err_{cp1}	err_{cp2}	err_rate	err_{cp1}	err_{cp2}	err_rate
Parking Lot #1	1.6643	2.8256	5.47%	0.1764	0.1648	0.49%	0.0572	0.0411	0.19%
Parking Lot #2	0.8246	1.3273	2.71%	0.3284	0.1648	0.58%	0.1123	0.0856	0.32%
Underground tunnel #1	4.0204	3.8976	12.6%	0.2461	0.1758	0.82%	0.0494	0.105	0.15%
Underground tunnel #2	1.0564	1.0804	3.11%	0.1056	0.0658	0.74%	0.0136	0.0156	0.12%
Building flyover #1	8.9918	8.9946	27.03%	0.0653	0.1102	0.34%	0.0294	0.065	0.24%
Building flyover #2	2.0516	2.3076	7.65%	0.2944	0.1482	1.39%	0.0053	0.0297	0.09%

5.5.3 Evaluation of Semantic Modeling Results

This subsection evaluates the performance of the proposed priori module PSIF in scene segmentation of an underground parking lot. A dataset containing scans from two regions of the package using Livox LiDAR is included, obtained through the SPC-SLAM method presented in section 5.2. Each point in

the scan was annotated using one of the four categories of semantic labels: ceiling, floor, wall, and other categories.

Similar to PointNet, the proposed method also predicts each point class in the point cloud scene. In the experiments of this chapter, each point is represented by a 3-dimensional vector XYZ. Accuracy and IoU are used as evaluation criteria and the methods in this chapter are compared with the basic module PointNet. The accuracy and IoU are calculated based on the point set, it can be defined as:

$$Accuracy = \frac{TP}{TP + FN} \quad (5.17)$$

$$IoU = \frac{DR \cap G}{DR \cup G}$$

Where TP, FN represent the number of true positives and false negatives, respectively, DR is the detection result value, and G is the true value.

Table 5.3 shows the results of semantic segmentation in six scenarios, where the proposed improved version of PointNet (PPM) significantly outperforms PointNet. In particular, it improves about 5% over PointNet in scenario 2 and scenario 5. In addition, the proposed method also outperforms PointNet++ [106] and DGCNN [107] in scenes 1, 2 and 5. The accuracy of the proposed method is also about 3% better than the other three methods according to the average of the overall results. These results show the significant effectiveness of the proposed a priori module PSIF. It is shown that the a priori module in this section provides important structural information that makes the segmentation algorithm more effective. The IoU results for each category are shown in Table 5.4. The PPM method proposed in this chapter has better performance than the basic model for each category. In particular, the IoU of the wall

category generated by PPM is about 9% higher than that generated by the basic model. All these results show that PSIF is capable of scene semantic segmentation.

Table 5.3 Six scenarios semantic segmentation results (%)

Methods	S1	S2	S3	S4	S5	S6	Average
PointNet	83.1	78.3	81.6	81.0	75.3	87.3	81.4
PointNet++	84.0	81.6	82.3	84.5	79.3	88.1	83.3
DGCNN	86.5	82.8	83.5	84.2	79.6	88.7	84.2
PPM(Ours)	87.6	83.2	83.8	84.9	80.6	88.5	84.8

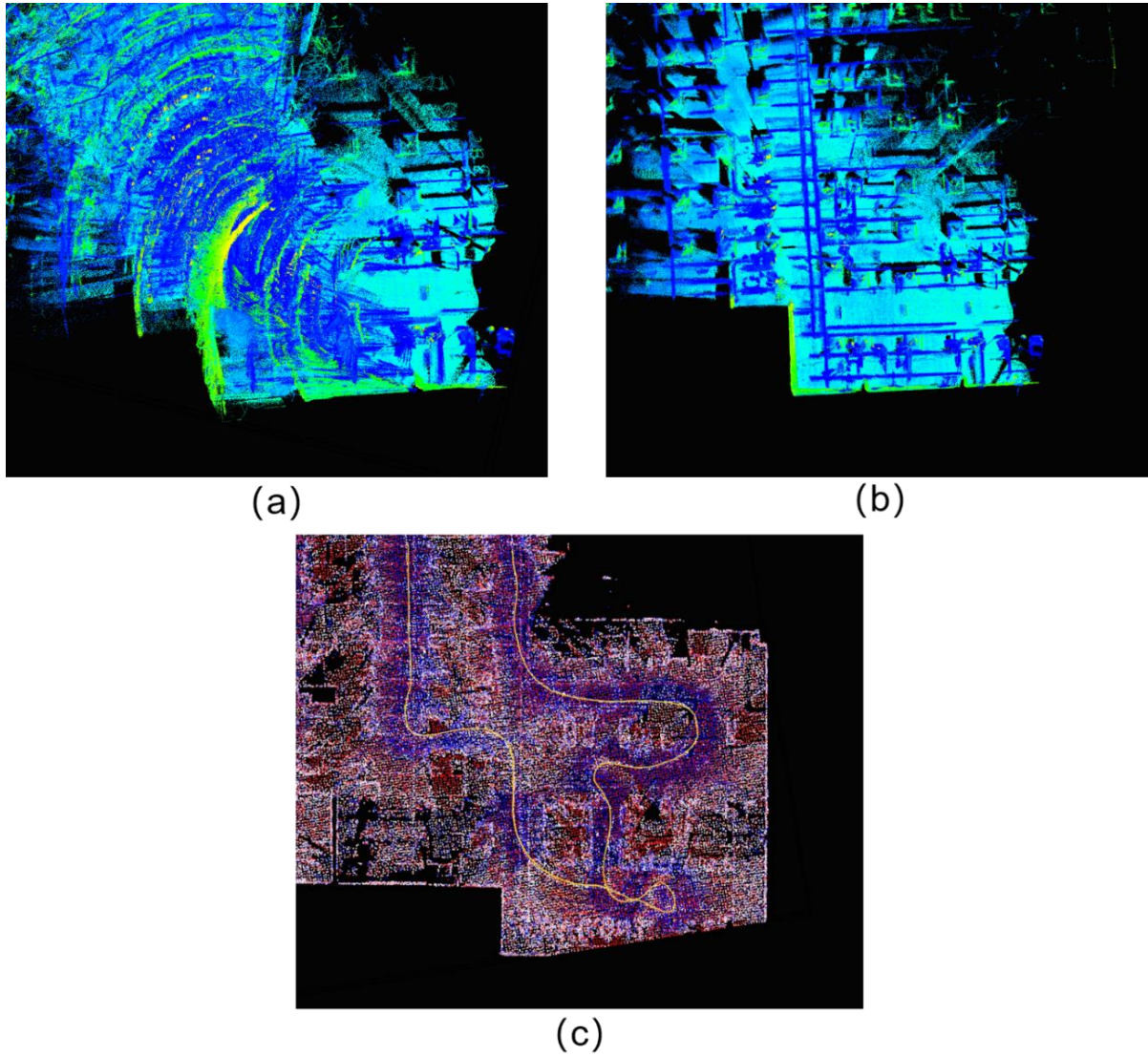


Figure 5.4 Point cloud quality (a) LOAM algorithm, (b) front-end results of SPC-SLAM, and (c) closed-loop detection and optimization results of SPC-SLAM

Table 5.4 Four categories of semantic segmentation IoU results (%)

Methods	Ceiling	Floor	Wall	Other	Average
PointNet	77.0	94.0	34.9	42.0	62.0
PointNet++	78.2	94.3	35.7	44.3	63.1

DGCNN	79.0	94.8	40.2	47.5	65.3
PPM(Ours)	79.2	95.5	43.4	48.7	66.7

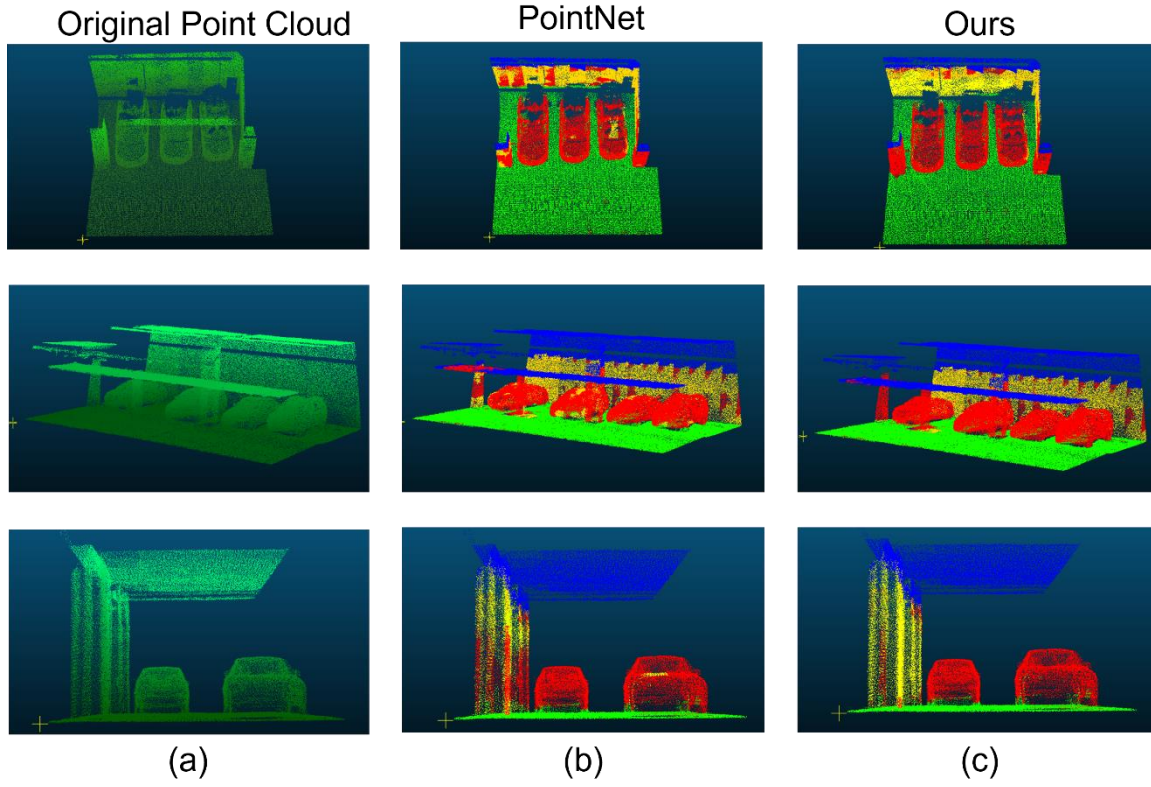


Figure 5.5 Semantic segmentation results. (a) The original point cloud, (b) the segmentation results using PointNet, and (c) the segmentation results obtained using the proposed method.



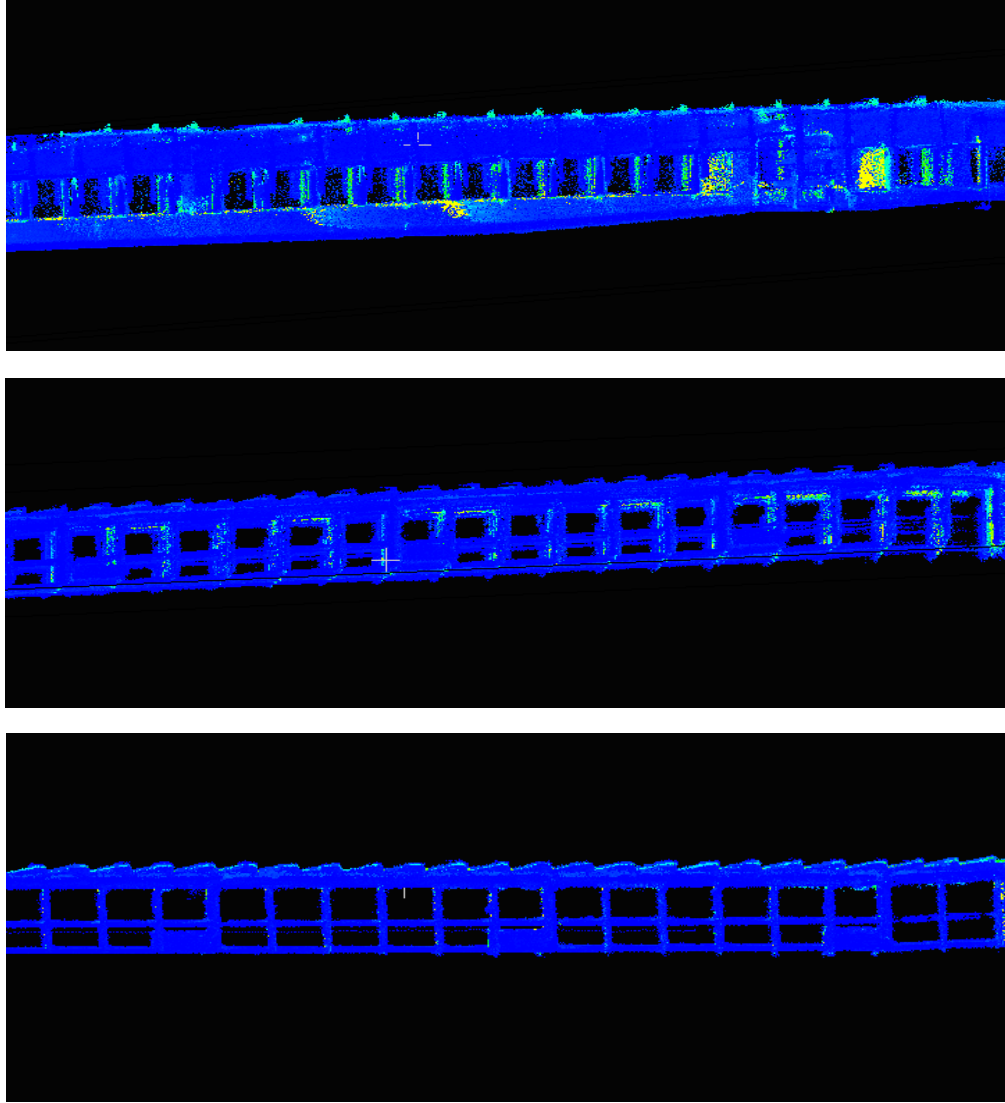


Figure 5.6 The building flyover results. (a) Images of the building flyover. (b) LOAM, (c) PC-SLAM front-end, and (d) SPC-SLAM after optimization.

5.6 Chapter Summary

This chapter introduces a 3D SLAM algorithm designed explicitly for sparse point cloud scenes, comprising of a front-end odometry approach and a back end closed-loop detection and optimization algorithm utilizing sub-maps. The algorithm effectively tackles the challenges of positioning and 3D

mapping in environments such as indoors and underground parking lots. Furthermore, utilizing mapping results, this chapter presents a framework for front- and back-end 3D point cloud segmentation, integrating the PointNet algorithm to construct high-precision point cloud semantic models with efficiency.

To address the challenge of odometry calculation, the front-end method employs a probabilistic matching algorithm based on point cloud surface covariance weight matching and a quadratic matching strategy. The results of the experiments demonstrate that this approach, combined with the quadratic matching strategy, effectively minimizes cumulative error, and yields accurate results even in weak feature environments. To tackle the issue of global trajectory optimization, this chapter introduces an optimization algorithm based on post-terminal maps that uses a sub-map matching strategy to construct topological maps while preserving the rigidity of corresponding trajectories. Finally, to address the problem of constructing discrete high-precision point cloud semantic models, this chapter presents the PSIF front-end segmentation algorithm, which leverages a priori information extraction of 3D semantic structures to provide fundamental information.

Experimental results from point cloud quality analysis and evaluation indicate that the front-end trajectory and attitude estimation (odometry) algorithm, combined with back-end closed-loop optimization, provides centimeter-level localization and map building accuracy. It is important to note that this centimeter-level accuracy measure is not typical for evaluating trajectories and maps. The algorithm demonstrates higher robustness compared to the LOAM algorithm based on feature filtering, especially in indoor environments that are symmetrical and heavily occluded. Additionally, results from indoor 3D point cloud data segmentation experiments demonstrate that the PSIF method outperforms

others, improving the training efficiency of PointNet. Finally, experiments reveal that the average semantic modeling accuracy across categories exceeds 84.8%.

Chapter 6

Conclusions

With technological advancements, LiDAR has evolved towards higher accuracy, miniaturization, and affordability. It has found widespread use in various industries, including smart mapping, autonomous driving, and robotics, demonstrating its immense value and promising commercial prospects. However, single LiDAR struggles to meet the demands of mapping and target identification in real-world applications, given its sparse point clouds and small data volume. Hence, the trend is shifting towards computation utilizing multi-camera/multi-LiDAR fusion of sensor data. Currently, algorithms relying on GNSS signal localization and point cloud reconstruction have proven successful. However, in GNSS-denied indoor and underground environments, high-precision mapping and 3D reconstruction are challenging due to the lack of localization signals. To overcome these obstacles in practical applications, this thesis focuses on comprehensive studies of multi-sensor data in indoor environments, including automatic multi-sensor calibration, 3D point cloud data reconstruction, and semantic model construction.

6.1 Contribution

In the realm of multi-sensor scanning and data collection for indoor environments, several challenges require novel and efficient solutions. This thesis presents three contributions to address these challenges, with the goal of improving the accuracy and efficiency of data collection, sensor calibration,

and semantic segmentation in indoor environments. The following is a summary of the three contributions.

6.1.1 WATiMMS Multi-Sensor Dataset

In Chapter 3, this thesis introduces WATiMMS, a dataset that can be used for sensor fusion, 3D reconstruction, trajectory localization, and semantic segmentation. The proposed dataset addresses the current lack of datasets in the indoor data domain that can be used to unify and validate multi-sensor calibration algorithms, scene segmentation algorithms, and trajectory validation algorithms. The dataset has two types of LiDAR data, three types of camera data, and IMU data. Manually labeled scene segmentation data and motion trajectory data of the platform are also provided.

Based on the characteristics of indoor scenes, a modular mobile platform for data acquisition is proposed in Chapter 3. The platform uses a collection of multiple sensors to summarize a data normalization model for different forms of devices. Based on this, the data collected by the platform are temporally synchronized and a multi-sensor calibration method is proposed. Next, the collected data are further optimized by the optimization method of key point diagram. The method uses a total station to collect key points and optimizes the motion trajectory obtained by the key point optimization SLAM algorithm to obtain a motion odometer close to the ground truth. Finally, the collected indoor data are manually calibrated so that the data can meet the requirements for validating the indoor scene segmentation algorithm. Through testing and analysis of the dataset, the results show that this dataset, using automatic segmentation algorithms such as PointNet, PointNet++ and RandLA-Net, can obtain high scene segmentation accuracy. The experimental results show that RandLA-Net has the highest correct rate in the mIoU results with 73.0%, PointNet++ has the highest correct rate in the accuracy results with 80.4%, RandLA-Net has the highest correct rate in the recall results with 78.6%, and

PointNet++ has the highest correct rate in the F1 scores. pointNet++ has the highest correct rate in the F1 score had the highest correct rate of 77.8%.

6.1.2 Target-free LiDAR-Camera Automatic Calibration

Chapter 4 of the thesis proposes a novel automatic calibration method to achieve high accuracy data fusion in indoor environments. Since the trajectory generation in the SLAM algorithm does not depend on a single target as a reference, the method using the trajectory as the calibration base achieves stable acquisition of extrinsic parameters in indoor environments with insufficient light and object occlusion. Meanwhile, graphical optimization solves the problem that the visual SLAM trajectory offset cannot determine the stable extrinsic. During the data acquisition process, our algorithm acquires the calculated trajectory information in real time and completes the external calibration of the sensor based on the optimized trajectory information. This method has less manual involvement, a high degree of automation, and does not require special targets, which solves the problem of accurate and fast calibration in multi-LIDAR systems. According to the experimental results, the RMSE can be improved from a maximum of 6.637° to 0.564° , and the RMSE of translation can be improved from a maximum of 0.197 m to 0.082 m. These results show that the calibration accuracy is significantly improved compared with other target-based methods.

6.1.3 Semantic Segmentation with Optimized SLAM algorithm

In Chapter 5, a 3D SLAM algorithm explicitly designed for sparse point cloud scenes is presented, including a front-end ranging method and a back-end loop closure detection and optimization algorithm using sub-maps. The algorithm effectively addresses the challenges of localization and 3D mapping in GNSS-poor environments such as indoor and underground parking lots. In addition, using the mapping results, Chapter 5 also presents a framework for front- and back-end 3D point cloud segmentation that

integrates the PointNet algorithm to efficiently construct highly accurate semantic models of point clouds.

To solve the challenge of mileage calculation, the front-end method adopts a probabilistic matching algorithm based on point cloud surface covariance weight matching and quadratic matching strategy. The experimental results show that this method combined with the quadratic matching strategy can effectively minimize the cumulative error and obtain accurate results even in a weak feature environment. To address the problem of global trajectory optimization, the chapter introduces a post-terminal map-based optimization algorithm that uses a sub-map matching strategy to construct topological maps while preserving the rigidity of the corresponding trajectories. Finally, to solve the problem of constructing discrete high-precision semantic models of point clouds, Chapter 5 introduces the PSIF front-end segmentation algorithm, which uses a priori information extraction of 3D semantic structures to provide basic information.

Experimental results of point cloud quality analysis and evaluation show that the front-end trajectory and pose estimation (odometry) algorithm, combined with the back-end closed-loop optimization, can provide centimeter-level localization and map construction accuracy. It is important to note that this centimeter-level accuracy measure is not typical when evaluating trajectories and maps. Compared to the feature filtering-based LOAM algorithm, this algorithm exhibits higher robustness, especially in symmetric and heavily occluded indoor environments. In addition, the results of indoor 3D point cloud data segmentation experiments show that the PSIF method outperforms other methods and improves the training efficiency of PointNet. Finally, the experiments show that the average semantic modeling accuracy of different classes exceeds 84.8%.

6.2 Recommendations for Future Research

In the automatic calibration algorithm, although the algorithm can automatically complete the effective calibration of camera and radar, it cannot complete the calibration of other sensors such as IMU and millimeter wave radar. Extending our theory to a generalized multi-sensor chain calibration is a possible future research direction. For example, using two-by-two calibration and mutual verification model to improve the robustness of calibration, and the use of multi-objective optimization algorithms and consistency constraints is one way to improve the final calibration results. The algorithms are used in the automatic calibration process. In the automatic calibration process, the input data of the algorithm depends on the front-end attitude and position estimation algorithm. Invalid data or misaligned trajectory data with failed matching can seriously degrade the calibration accuracy. Therefore, the input confidence prediction is also a topic worthy of further study.

Second, for the study of LiDAR-based high-precision 3D SLAM and semantic segmentation, in which the front-end module of the SLAM algorithm may still fail under more than a certain degree of weak features, for example, a long corridor with too much symmetry and lack of 3D information. The front-end module of SLAM algorithm may still fail under symmetric and weak features, e.g., symmetric environment. The long corridor problem may cause the model matching to fall into local optimum and fail. How to combine the visual feature texture information of camera data and For the SLAM back-end, how to combine the visual feature texture information of camera data and IMU inertial navigation information to further improve the robustness of front-end odometry is a topic that needs further research. For the back-end of SLAM, how to effectively introduce visual information, such as bag-of-words model matching, to improve the accuracy and robustness of closed-loop detection is a topic that needs further study. The accuracy and robustness of closed-loop detection is a possible research

direction. On the other hand, in the part of semantic model reconstruction, only four types of segmentation are currently done, which is not enough for practical applications. How to add more semantic types without degrading the quality of segmentation accuracy and semantic reconstruction needs to be further investigated.

Finally, once a high-quality semantic segmentation point cloud is obtained, the focus can shift to real-time 3D object recognition, especially for objects in indoor environments where full views may be obstructed due to occlusion.

References

- [1] L. Mainetti, L. Patrono, and I. Sergi. 2014, "A survey on indoor positioning systems, " In: Proceedings of the 22nd International Conference on Software, Telecommunications and Computer Networks, pp. 111–120.
- [2] L. Ran, S. Helal, and S. Moore. 2004, "Drishti: an integrated indoor/outdoor blind navigation system and service," In: Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications, pp. 23–30.
- [3] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle. 2010, "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.
- [4] L. Atzori, A. Iera, and G. Morabito. 2010, "The Internet of Things: A survey. " Computer Networks, vol. 54, no. 15, pp. 2787–2805.
- [5] H. Wang and W. He. 2011, "A reservation-based smart parking system," In: Proceedings of the 2011 IEEE Conference on Computer Communications Workshops, pp. 690–695.
- [6] Y. Hu, J. Gong, Y. Jiang, L. Liu, G. Xiong, and H. Chen. 2013, "Hybrid map-based navigation method for unmanned ground vehicle in urban scenario," Remote Sensing, vol. 5, no. 8, pp. 3662–3680.
- [7] H. Durrant-Whyte and T. Bailey. 2006, "Simultaneous localization and mapping: Part I," IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99–110.
- [8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. 2015, "ORB-SLAM: A versatile and accurate monocular SLAM system," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163.
- [9] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard. 2012, "An evaluation of the RGB-D SLAM system." In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1691–1696.
- [10] H.-P. Li, D.-M. Xu, F.-B. Zhang, and Y. Yao. 2009, "Consistency analysis of EKF-based SLAM by measurement noise and observation times," Acta Automatica Sinica, vol. 35, no. 9, pp. 1177–1184.
- [11] Y. Wang, X. Liang, C. Flener, A. Kukko, H. Kaartinen, M. Kurkela, M. Vaaja, H. Hyypä, P. Alho. 2013, "3D modeling of coarse fluvial sediments based on mobile laser scanning data." Remote Sensing, vol. 5, no. 9, pp. 4571–4592.
- [12] A. Nüchter, D. Borrmann, P. Koch, M. Kühn, and S. May. 2015, "A man-portable, imu-free mobile mapping system," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. II-3-W5, pp. 17–23.
- [13] H. A. Lauterbach, D. Borrmann, R. Heß, D. Eck, K. Schilling, and A. Nüchter. 2015, "Evaluation of a backpack-mounted 3D mobile scanning system," Remote Sensing, vol. 7, no. 10, pp. 13753–13781.

- [14] Wang, C., Dai, Y., Elsheimy, N., Wen, C., Retscher, G., Kang, Z., and Lingua, A. 2020, "ISPRS benchmark on multisensory indoor mapping and positioning," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 117–123.
- [15] N. Corso, A. Zakhor. 2013, "Indoor localization algorithms for an ambulatory human operated 3D mobile mapping system." *Remote Sensing*, vol.5, no. 12, pp. 6611–6646.
- [16] J. Behley, V. Steinhage, A. B. Cremers. 2012, "Performance of histogram descriptors for the classification of 3D laser range data in urban environments." In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4391–4398.
- [17] K. Klasing, D. Wollherr, M. Buss. 2018, "A clustering method for efficient segmentation of 3D laser data." In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4043–4048.
- [18] A. Anand, H. S. Koppula, T. Joachims, A. Saxena. 2018, "Contextually guided semantic labeling and search for three-dimensional point clouds." *International Journal of Robotics Research*, vol. 32, no.1, pp. 19–34.
- [19] G. Elbaz, T. Avraham, and A. Fischer. 2017, "3D point cloud registration for localization using a deep neural network auto-encoder," In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481.
- [20] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. 1998, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the Association for Computing Machinery*, vol. 45, no. 6, pp. 891–923.
- [21] M. Muja, D.G. Lowe. 2014, "Scalable nearest neighbor algorithms for high dimensional data." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2227–2240.
- [22] J. Zhang, S. Singh. 2014, "LOAM: LiDAR odometry and mapping in real-time." *Robotics: Science and Systems*, DOI: 10.15607/RSS.2014.X.007
- [23] F. Moosmann, C. Stiller. 2011, "Velodyne SLAM." In: *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium*, pp. 393–398.
- [24] Y. Chen, G. Medioni. 1992, "Object modelling by registration of multiple range images." *Image and Vision Computing*, vol.10, no. 3, pp. 145–155.
- [25] S. Thrun, W. Burgard, and D. Fox. 2000, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," In: *Proceedings of the IEEE International Conference on Robotics and Automation*. vol.1, pp. 321–328.
- [26] M. Labbé, F. Michaud. 2013, "Appearance-based loop closure detection for online large-scale and long-term operation." *IEEE Transactions on Robotics*, vol. 29, no.3, pp. 734–745.
- [27] A. Andoni and P. Indyk. 2006, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," In: *Proceedings of the 2006 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 459–468.
- [28] P. Indyk and R. Motwani. 1998, "Approximate nearest neighbors: towards removing the curse of dimensionality," In: *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pp. 604–613.
- [29] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. 2007, "Multi-probe LSH: Efficient indexing for high-dimensional similarity search." In: *Proceedings of the 33rd International Conference on Very Large Databases*, pp. 950–961.
- [30] M. Bawa, T. Condie, P. Ganesan. 2005, "LSH forest: Self-tuning indexes for similarity search." In: *Proceedings of the International World Wide Web Conferences*, pp. 651–660.

- [31] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, L. Schmidt. 2015, "Practical and optimal LSH for angular distance." In: Proceedings of the Neural Information Processing Systems, pp. 1225–1233.
- [32] R. A. Finkel and J. L. Bentley. 1974, "Quad trees a data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, no. 1, pp. 1–9.
- [33] D. J. Meagher. 1980 "Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-D objects by computer." Department of Electrical and Systems Engineering, Rensselaer Polytechnic Institute.
- [34] J. L. Bentley. 1975, "Multidimensional binary search trees used for associative searching." *Communications of the Association for Computing Machinery*, vol. 18, no. 9, pp. 509–517.
- [35] A. Guttman. 1984, "R-trees: a dynamic index structure for spatial searching," *Association for Computing Machinery*, vol. 14, no. 2, pp. 47–57.
- [36] S. M. Omohundro. 1989, "Five Balltree Construction Algorithms." Technical Report, International Computer Science Institute, UC Berkeley, 562p.
- [37] A. Nuchter, K. Lingemann, and J. Hertzberg. 2007, "Cached k-d tree search for ICP algorithms," In: Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling, pp. 419–426.
- [38] C. Silpa-Anan and R. Hartley. 2008, "Optimised KD-trees for fast image descriptor matching," In: Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, DOI: 10.1109/CVPR.2008.4587638.
- [39] C. Faloutsos, S. Roseman. 1989, "Fractals for secondary key retrieval." In: Proceedings of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 247–252.
- [40] G.M. Morton. 1966, "A computer oriented geodetic database and a new technique in file sequencing." *Physics of Plasmas*, vol. 24, no. 7, pp. 159–173.
- [41] H. Tropf, H. Herzog. 1981, "Multidimensional range search in dynamically balanced trees." *Angewandte Informatik*, vol. 23, no. 2, pp. 71–77.
- [42] J. A. Orenstein, T. H. Merrett. 1984, "A class of data structures for associative searching." In: Proceedings of the 3rd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, pp. 181–190.
- [43] M. Bern. 1993, "Approximate closest-point queries in high dimensions," *Information Processing Letters*, vol. 45, no. 2, pp. 95–99.
- [44] M. Connor, P. Kumar. 2010, "Fast construction of k-nearest neighbor graphs for point clouds." *IEEE Transactions on Visualization & Computer Graphics*, vol.16, no. 4, pp. 599–608.
- [45] J. K. Lawder, P. J. H. King. 2001, "Querying multi-dimensional data indexed using the Hilbert spacefilling curve." *ACM SIGMOD Record*, vol. 30, no. 1, pp. 19–24.
- [46] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp. 2019, "A deep learning-based radar and camera sensor fusion architecture for object detection," In: Proceedings of the 2019 Sensor Data Fusion: Trends, Solutions, Applications, pp. 1–7.
- [47] A. Kloukinotis, A. Papandreou, C. Anagnostopoulos, A. Lalos, P. Kapsalas, D. Nguyen, K. Moustakas. 2022, "CarlaScenes: A synthetic dataset for odometry in autonomous driving," In: Proceedings of the 2022 IEEE Conference on Computer Vision and Pattern Recognition, pp. 4520–4528.
- [48] S. Liao, M. A. Lopez, and S. T. Leutenegger. 2001, "High dimensional similarity search with space filling curves," In: Proceedings of the 17th International Conference on Data Engineering, pp. 615–622.

- [49] M. T. Goodrich and J. A. Simons. 2011, "Fully retroactive approximate range and nearest neighbor searching," In: *Proceedings of the Algorithms and Computation*, pp. 292–301.
- [50] Y. Zhuang, F. Yan, H. Hu. 2014, "Automatic extrinsic self-calibration for fusing data from monocular vision and 3-D laser scanner." *IEEE Transactions on Instrumentation and Measurement*, vol.63, no. 7, pp. 1874–1876.
- [51] G. Atanacio-Jiménez, J.-J. González-Barbosa, J. B. Hurtado-Ramos, F. J. Ornelas-Rodríguez, H. Jiménez-Hernández, T. García-Ramírez, R. González-Barbosa. 2011, "LiDAR Velodyne HDL-64e calibration using pattern planes." *International Journal of Advanced Robotic Systems*, vol.8, no. 5, pp. 70-82.
- [52] Z. Zhang. 2000, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334.
- [53] M. Li, S. Cheng, and G. Wang. 2010, "A camera-laser calibration method for stereo vision based on a planar calibration pattern," *Journal of Visual Communication and Image Representation*, vol. 21, no. 4, pp. 324-331.
- [54] J. Valentin, P. Sturm, and R. Siegwart. 2012, "Automated intrinsic and extrinsic calibration of a multi-layer laser scanner and a camera mounted on a pan-tilt unit," *Journal of Field Robotics*, vol. 29, no. 2, pp. 259-274.
- [55] C. Sun, J. Wang, and R. Huang. 2019, "Automatic extrinsic calibration of camera-LiDAR system using geometric features," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 6, pp. 2170-2181.
- [56] H. Yi, B. Liu, B. Zhao, and E. Liu. 2022, "Extrinsic calibration for LiDAR-camera systems using direct 3D-2D correspondences," *Remote Sensing*, doi: 10.3390/rs14236082.
- [57] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. 2013, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 44, no. 1-2, pp. 139-155.
- [58] A. Geiger, P. Lenz, and R. Urtasun. 2012, "Are we ready for autonomous driving? The KITTI vision benchmark suite," In: *Proceedings of the 2014 IEEE Computer Vision and Pattern Recognition*, pp. 3354-3361
- [59] J. Valentin, P. Sturm, and R. Siegwart. 2012, "Automated intrinsic and extrinsic calibration of a multi-layer laser scanner and a camera mounted on a pan-tilt unit," *Journal of Field Robotics*, vol. 29, no. 2, pp. 259-274.
- [60] S. Wang, R. Yang, and C. Stiller. 2015, "An optimization framework for LiDAR-camera calibration with unknown extrinsics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3391-3402.
- [61] I. Cvišić, I. Marković, and I. Petrović. 2023, "SOFT2: Stereo visual odometry for road vehicles based on a point-to-epipolar-line metric," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 273–288.
- [62] Q. Yang and D. A. Clausi. 2017, "Planar checkerboard corner detection based on multiscale Harris corner response," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1865-1871.
- [63] J. Shi and C. Tomasi. 1994, "Good features to track," In: *Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600.
- [64] C. Sun, J. Wang, and R. Huang. 2019, "Automatic extrinsic calibration of camera-LiDAR system using geometric features," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 6, pp. 2170-2181.

- [65] N. Muhammad, S. Lacroix. 2010, "Calibration of a rotating multi-beam LiDAR." In: Proceedings of the 2010 IEEE International Conference on Intelligent Robots and Systems, pp. 5648–5653.
- [66] M. He, H. Zhao, F. Davoine, J. Cui, H. Zha. 2013, "Pairwise LiDAR calibration using multi-type 3D geometric features in natural scene." In: Proceedings of the 2013 IEEE International Conference on Intelligent Robots and Systems, pp. 1828–1835.
- [67] H. Nouraa, J. E. Deschaud, and F. Goulettea. 2016, "Point cloud refinement with a target-free intrinsic calibration of a mobile multi-beam LiDAR system," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLI-B3, pp. 359–366.
- [68] M. S. Montemerlo. 2002, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association." In: Proceedings of the AAAI National Conference on Artificial Intelligence, pp. 593–598.
- [69] M. Montemerlo, S. Thrun, D. Roller, and B. Wegbreit. 2003, "FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges," In: Proceedings of the 18th international Joint Conference on Artificial intelligence, pp. 1151–1156.
- [70] V. V. Lehtola, H. Kaartinen, A. Nüchter, R. Kaijaluoto, A. Kukko, P. Litkey, E. Honkavaara, T. Rosnell, M. T. Vaaja, J.-P. Virtanen. 2017, "Comparison of the selected state-of-the-art 3d indoor scanning and point cloud generation methods." Remote Sensing, doi:10.3390/rs9080796.
- [71] M. Draelos, Q. Qiu, A. Bronstein, G. Sapiro. 2015, "Intel realsense= real low cost gaze." 2015 In: Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), pp. 2520–2524.
- [72] Y. Liu, W. Gao, and Z. Hu. 2018, "Geometrically stable tracking for depth images based 3D reconstruction on mobile devices," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 143, pp. 222–232.
- [73] X. Qu, B. Soheilian, and N. Paparoditis. 2018, "Landmark based localization in urban environment," ISPRS Journal of Photogrammetry and Remote Sensing, Jun. 2018, vol. 140, pp. 90–103.
- [74] A. Nüchter, M. Bleier, J. Schauer, and P. Janotta. 2017, "Improving Google's Cartographer 3D mapping by continuous-time slam," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2-W3, pp. 543–549.
- [75] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. 2011, "G2O: A general framework for graph optimization," In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, pp. 3607–3613.
- [76] A. Kukko, R. Kaijaluoto, H. Kaartinen, V. V. Lehtola, A. Jaakkola, and J. Hyypä, 2017, "Graph SLAM correction for single scanner MLS forest data under boreal forest canopy," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 132, pp. 199–209.
- [77] A. Geiger, P. Lenz, and R. Urtasun. 2012, "Are we ready for autonomous driving? The KITTI vision benchmark suite," In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361.
- [78] J. Holmgren, H. Tulldahl, J. Nordlöf, M. Nyström, K. Olofsson, J. Rydell, E. Willén. 2017, "Estimation of tree position and stem diameter using simultaneous localization and mapping with data from a backpack-mounted laser scanner." International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-3-W3, pp. 59–63.
- [79] C. Glennie, B. Brooks, T. Ericksen, D. Hauser, K. Hudnut, J. Foster, J. Avery. 2013, "Compact multipurpose mobile laser scanning system—Initial tests and results." Remote Sensing, vol. 5, no. 2, pp. 521–538.

- [80] A. A. Ahmed, M. Al-Shaboti, and A. Al-Zubairi. 2015, "An indoor emergency guidance algorithm based on wireless sensor networks," In: Proceedings of the 2015 International Conference on Cloud Computing, pp. 1–5.
- [81] M. Burri J. Nikolic, P. Gohl, T. Schneider, J. Rehder. 2016, "The EuRoC micro aerial vehicle datasets," The International Journal of Robotics Research, vol. 35, no. 10, pp. 1157–1163.
- [82] B. De Vos, J. Wolterink, P. Jong, T. Leiner, M. Viergever, and I. Išgum. 2017, "ConvNet-based localization of anatomical structures in 3D medical images," IEEE Transactions on Medical Imaging, vol. 36, no. 7, pp. 1470–1481.
- [83] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. 2017, "ScanNet: richly-annotated 3d reconstructions of indoor scenes," In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2432–2443.
- [84] S. Dasgupta, K. Fang, K. Chen, and S. Savarese. 2016, "DeLay: Robust spatial layout estimation for cluttered indoor scenes," In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 616–624.
- [85] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, 2014, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," In: Proceedings of the 2014 IEEE International Conference on Robotics and Automation, pp. 1524–1531.
- [86] X. Hu, Z. Zheng, D. Chen, X. Zhang, and J. Sun. 2022, "Processing, assessing, and enhancing the Waymo autonomous vehicle open dataset for driving behavior research," Transportation Research Part C: Emerging Technologies, doi: 10.1016/j.trc.2021.103490.
- [87] Z. Kang, J. Yang, Z. Yang, and S. Cheng. 2020, "A review of techniques for 3D reconstruction of indoor environments," ISPRS International Journal of Geo-Information, doi: 10.3390/ijgi9050330.
- [88] E. Kim and S.-Y. Park. 2020, "Extrinsic calibration between camera and LiDAR sensors by matching multiple 3D planes," Sensors, doi:10.3390/s20010052.
- [89] V. Sequeira, J. G. M. Goncalves, and M. I. Ribeiro. 1996, "3D reconstruction of indoor environments," In: Proceedings of the 3rd IEEE International Conference on Image Processing, vol.2, pp. 405–408.
- [90] N. Silberman and R. Fergus. 2011, "Indoor scene segmentation using a structured light sensor," In: Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops, pp. 601–608.
- [91] N. Silberman and R. Fergus. 2011, "Indoor scene segmentation using a structured light sensor," In: Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops, pp. 601–608.
- [92] S. Song, S. P. Lichtenberg, and J. Xiao. 2015, "SUN RGB-D: A RGB-D scene understanding benchmark suite," In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp. 567–576.
- [93] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. 2012, "A benchmark for the evaluation of RGB-D SLAM systems," In: Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 573–580.
- [94] F. Zhu, X. Xu, L. Ma, D. Guo, X. Cui, and Q. Kong. 2020, "Autonomous driving vehicle control auto-calibration system: An industry-level, data-driven and learning-based vehicle longitudinal dynamic calibrating algorithm," In: Proceedings of the 2020 IEEE Intelligent Vehicles Symposium, pp. 391–397.

- [95] C. Wen, S. Pan, C. Wang, J. Li. 2016, "An indoor backpack system for 2D and 3D mapping of building interiors." *IEEE Geoscience & Remote Sensing Letters*, vol. 13, no.7, pp. 992–996.
- [96] P. Viola, M. J. Jones, D. Snow. 2005, "Detecting pedes trains using patterns of motion and appearance." *International Journal of Computer Vision*. vol. 63, no. 2, pp. 153–161.
- [97] N. Dalal, B. Triggs. 2005, "Histograms of oriented gradients for human detection." In: *Proceedings of the 2005 IEEE International Conference on Computer Vision & Pattern Recognition*, pp. 886–893.
- [98] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan. 2009, "Object detection with discriminatively trained part-based models." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645.
- [99] S. Ren, K. He, R. Girshick, J. Sun. 2015, "Faster R-CNN: Towards real-time object detection with region proposal networks." In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 91–99.
- [100] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell. 2014, "Caffe: Convolutional architecture for fast feature embedding." In: *Poceedings of the 22nd ACM International Conference on Multimedia*, pp. 675–678.
- [101] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg. 2016, "SSD: Single shot multibox detector." In: *Proceedings of the European conference on computer vision*, pp. 21–37.
- [102] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. 2016, "You only look once: Unified, real-time object detection." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788.
- [103] J. Redmon, A. Farhadi. 2018, "YOLOv3: An incremental improvement." *arXiv: 1804.02767*.
- [104] D. Munoz, J. A. Bagnell, N. Vandapel, M. Hebert, 2009, "Contextual classification with functional max-margin Markov networks." In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 975–982.
- [105] B. Taskar, C. Guestrin, D. Koller. 2004, "Max-margin Markov networks." In *Proceedings of Advances in Neural Information Processing Systems*, pp. 25–32.
- [106] S. Gupta, R. Girshick, P. Arbeláez, J. Malik. 2014, "Learning rich features from RGB-D images for object detection and segmentation." In: *Proceedings of the European Conference on Computer Vision*, pp. 345–360
- [107] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, R. Urtasun. 2015, "3D object proposals for accurate object class detection." In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 424–432.
- [108] S. Song and J. Xiao. 2016, "Deep Sliding shapes for amodal 3D object detection in RGB-D images," In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 808–816.
- [109] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. 2017, "PointNet: Deep learning on point sets for 3D classification and segmentation," In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 77–85.
- [110] X. Chen, H. Ma, J. Wan, B. Li, T. Xia. 2017, "Multi-view 3D object detection network for autonomous driving. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*," pp. 1907–1915.

- [111] J. Ku, M. Mozifian, J. Lee, A. Harakeh, S. L. Waslander. 2018, “Joint 3D proposal generation and object detection from view aggregation.” In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 1–8.
- [112] C. R. Qi, W. Liu, C. Wu, H. Su, L. J. Guibas. 2018, “Frustum PointNets for 3D object detection from RGB-D data.” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 918–927.
- [113] Z. Kang, J. Yang, Z. Yang, and S. Cheng. 2020, “A review of techniques for 3D reconstruction of indoor environments,” *ISPRS International Journal of Geo-Information*, doi:0.3390/ijgi9050330.
- [114] V. Sequeira, J. G. M. Goncalves, and M. I. Ribeiro. 1996, “3D reconstruction of indoor environments,” In: Proceedings of the 3rd IEEE International Conference on Image Processing, vol. 2, pp. 405–408.
- [115] M. Cao, P. Su, H. Chen, S. Tang, and Y. Liu. 2021, “3-D dense rangefinder sensor with a low-cost scanning mechanism,” *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2020.3016415.
- [116] J. Yin, D. Luo, F. Yan, and Y. Zhuang. 2022, “A novel LiDAR-assisted monocular visual SLAM framework for mobile robots in outdoor environments,” *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2022.3190031.
- [117] C. Ye, H. Pan, and H. Gao. 2022, “Keypoint-based LiDAR-camera online calibration with robust geometric network,” *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2021.3129882.
- [118] S. Xie, D. Yang, K. Jiang, and Y. Zhong. 2019, “Pixels and 3-D points alignment method for the fusion of camera and LiDAR data,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3661–3676.
- [119] X. Liu, C. Yuan, and F. Zhang. 2022, “Targetless extrinsic calibration of multiple small FoV LiDARs and cameras using adaptive voxelization,” *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2022.3176889.
- [120] G. Koo, J. Kang, B. Jang, and N. Doh. 2022, “Precise camera-LiDAR extrinsic calibration based on a weighting strategy using analytic plane covariances,” *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2022.3202549.
- [121] T. Fu, H. Yu, W. Yang, Y. Hu, and S. Scherer. 2022, “Targetless extrinsic calibration of stereo, thermal, and laser sensors in structured environments,” *IEEE Transactions on Instrumentation and Measurement*, doi: 10.1109/TIM.2022.3204338.
- [122] M. Shu, G. Chen, and Z. Zhang. 2022, “Efficient image-based indoor localization with MEMS aid on the mobile device,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 185, pp. 85–110.
- [123] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus. 2020, “LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping,” In: Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5135–5142.
- [124] IEEE precision clock synchronization protocol, 2008, IEEE Standard 1588.
- [125] E. Kim and S.-Y. Park. 2020, “Extrinsic calibration between camera and LiDAR sensors by matching multiple 3D planes,” *Sensors*, doi:10.3390/s20010052.
- [126] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. 2015, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163.

- [127] H. Wang and W. He. 2011, “A reservation-based smart parking system,” In: Proceedings of the 2011 IEEE Conference on Computer Communications, pp. 690–695.
- [128] Y. Hu, J. Gong, Y. Jiang, L. Liu, G. Xiong, and H. Chen. 2013, “Hybrid map-based navigation method for unmanned ground vehicle in urban scenario,” *Remote Sensing*, vol. 5, no. 8, pp. 3662–3680.
- [129] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. 2011, “G2O: A general framework for graph optimization,” In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, pp. 3607–3613.
- [130] J. Demantké, C. Mallet, N. David, and B. Vallet, 2012, “Dimensionality based scale selection in 3D LiDAR point clouds,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-5-W12, pp. 97–102.