

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

MULTI-SENSOR FUSION BASED
INTELLIGENT PERCEPTION USING UAVS
FOR AUTONOMOUS INFRASTRUCTURE
AND AGRICULTURE INSPECTION

YANG WENYU

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University

Department of Aeronautical and Aviation Engineering

**Multi-Sensor Fusion Based Intelligent
Perception Using UAVs for Autonomous
Infrastructure and Agriculture Inspection**

YANG Wenyu

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

10/2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

YANG Wenyu

Abstract

With the rapid advancement of Artificial Intelligence (AI) and autonomous technologies, intelligent robotics is spearheading a profound technological revolution, comprehensively reshaping our way of life. Significant benefits from technological innovation have already been realized across multiple domains, including autonomous driving, intelligent vehicles for specialized scenarios (factories, mines, farms), and intelligent aerial robots.

Nevertheless, numerous open challenges remain to be addressed in the coming decades. Examples include the ongoing paradigm shift in autonomous driving—from modular design to end-to-end approaches—and the realization of autonomous robot operation across all terrains and throughout the full lifecycle. Amidst these challenges, burgeoning fields such as Embodied AI, humanoid robotics, and aerial manipulation are rapidly emerging.

Despite the existence of diverse research directions in the field, intelligent perception stands as the fundamental cornerstone of intelligent robot systems. Only with accurate and robust intelligent perception capabilities can an intelligent robot system establish a reliable foundation, thereby enabling the seamless execution of downstream tasks such as motion planning and precise control. In this context, our research focuses on intelligent perception tasks driven by multi-sensor fusion, which achieves centimeter-level translation accuracy and sub-degree level rotation accuracy. We further extend our work to the development of a multi-sensor-based Simultaneous Localization and Mapping (SLAM) system, as well as the advancement of LiDAR Bundle Adjustment and hierarchical 3D scene graph techniques. This research work encompasses four primary components:

- Multi-sensor Calibration Algorithms: Development of methods for accurately calibrating diverse sensors.
- SLAM Front-end: Implementation of the multi-sensor fusion SLAM system for on-line pose estimation.
- LiDAR Bundle Adjustment for Accurate Geometric Mapping: Uncertainty-based optimization techniques are applied to LiDAR data to improve the geometric accuracy of the constructed maps.

- Hierarchical 3D Scene Graphs for Metric-Semantic Mapping: Construction of hierarchical representations integrating geometric and semantic information within the environment.

We employ Unmanned Aerial Vehicle (UAV) as the mobile platform, validating the proposed system’s efficacy through two case studies while adapting it to the specific characteristics of distinct application scenarios and tasks. The first application scenario is intelligent vertical farming, and the second is bridge inspection.

In the vertical farming application, we monitor the growth status of crops (strawberries) and perform 3D mapping and counting of strawberry blossoms and fruits. Building upon traditional geometric point cloud maps, we introduce three-dimensional Scene Graph (3DSG) to manage semantic information across different levels of abstraction. By co-optimizing parent and child nodes within the scene graph, our method achieves significantly enhanced fruit counting accuracy compared to conventional approaches.

In the bridge management application, we continue utilizing scene graphs to organize semantic cues critical for bridge inspection and management tasks. Furthermore, tailored to the structural characteristics of bridges, we incorporate corner points, edges, and planar features into the LiDAR Bundle Adjustment (BA) process, which substantially improves the precision of LiDAR mapping.

In conclusion, this study comprehensively demonstrates the considerable potential of UAV-based intelligent inspection systems in agricultural and industrial applications, encompassing sensor fusion, SLAM, semantic detection, 3D object estimation, and high-level organizational structures for management purposes. By integrating AI components, the system significantly enhances the effectiveness of UAVs in infrastructure inspection, providing valuable insights for practitioners in agricultural monitoring and structural health monitoring. Extensive simulations and real-world flight tests validate the system’s efficacy, bringing the vision of intelligent inspection systems closer to reality and laying a solid foundation for downstream tasks.

Keywords: Sensor Fusion, Intelligent Perception, SLAM, UAV Inspection System, 3D Scene Graph, LiDAR Bundle Adjustment.

Publication arising from this thesis

Published/Accepted:
Journal/Conference

- 1 **Wenyu Yang**, Kang Liu*, Zheng Tan, Li-Yu Lo, Yinglun Wang, Ka-Hing Wong, and Chih-Yung Wen, "Hierarchical 3D Scene Graph based Semantic-Metric SLAM for Plant Inspection and Fruit Counting in Intelligent Hydroponics System", *IEEE Internet of Things Journal*, vol. 12, no. 23, pp. 49071-49082, 2025.
- 2 **Wenyu Yang**, Haojun Luo, Kwai-Wa Tse, Haochen Hu, Kang Liu, Boyang Li*, and Chih-Yung Wen*, "Autonomous Targetless Extrinsic Calibration of Thermal, RGB, and LiDAR Sensors", *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-11, 2024.
- 3 **Wenyu Yang**, Haochen Hu, Kwai-Wa Tse, Shengyang Chen, Weisong Wen and Chih-Yung Wen*, "LiDAR Stereo Visual Inertial Pose Estimation Based on Feed-forward and Feedbacks," 2024 International Conference on Unmanned Aircraft Systems (ICUAS), Chania - Crete, Greece, 2024, pp. 1042-1049.
- 4 Kang Liu, **Wenyu Yang***, Jiao Lin*, Zhipeng Yuan and Chih-Yung Wen, "Fast Fixed-Time Distributed Neural Formation Control-based Disturbance Observer for Multiple Quadrotor UAVs Under Unknown Disturbances", in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no.5, pp. 13137-13155, 2025.
- 5 Kwai-Wa Tse, Rendong Pi, **Wenyu Yang**, Xiang Yu and Chih-Yung Wen*, "Advancing UAV-Based Inspection System: The USSA-Net Segmentation Approach to Crack Quantification", *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-14, 2024.
- 6 Yefeng Yang, Tao Huang, Tianqi Wang, **Wenyu Yang**, Han Chen, Boyang Li*, and Chih-Yung Wen, "Sampling efficient rapidly exploring random tree path plan-

ning and improved Actor-Critic based obstacle avoidance for autonomous robots", *Science China: Information Science*, Vol. 67, np. 5, pp: 1-18, 2024.

Submitted/To submit:

- 1 **Wenyu Yang**, Kang Liu, and Chih-Yung Wen, "Metric Semantic SLAM Based on LiDAR BA for Bridge Inspection and Management". (*to submit*)

Awards

- 1 First runner-up award in the UAV competition of 2023 International Conference on Unmanned Aircraft System, Warsaw, Poland.
- 2 First runner-up award in the UAV competition of 2024 International Conference on Unmanned Aircraft System, Chania, Greece.
- 3 First runner-up award in the UAV competition of 2025 International Conference on Unmanned Aircraft System, Charlotte, NC, USA.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my Ph.D. supervisor, Professor Chih-Yung WEN, for his invaluable guidance throughout my doctoral studies. His expertise in robotics, constructive advice, and constant encouragement were essential in helping me navigate challenges and ultimately complete this work. Beyond his steadfast support and clear mentorship, Professor Wen generously provided me with invaluable opportunities to attend international conferences and learn from other distinguished scholars. I am profoundly grateful to have had such an exceptional supervisor; his guidance and support were instrumental in bringing this dissertation to fruition.

Secondly, I would like to thank Prof. Zongyu CHANG, Prof. Hyondong OH, and Prof. Li-Da HSU for serving on my Ph.D. oral examination committee. I am also indebted to many colleagues and friends at The Hong Kong Polytechnic University who have provided valuable comments and encouragement during the preparation of this thesis. Among those who have generously contributed their time and energy are Dr Kang LIU, Dr. Boyang LI, Dr. Kwai-Wa TSE Sky, Dr. Tianqi WANG, Dr. Shengyang CHEN, Dr. Han CHEN, Dr. Weisong WEN, Dr. Yefeng YANG, Mr. Zheng TAN, Mr. Tao HUANG, Mr. Yurong FENG, Mr. Qihang PENG, Mr. Haochen HU, and Mr. Yinglun WANG.

Finally, I extend my deepest gratitude to my family for their unwavering love and support throughout this journey. I could never have come this far without their constant presence and encouragement.

Contents

Certificate of Originality

Abstract	I
----------	---

Publication	III
-------------	-----

Acknowledgements	V
------------------	---

Contents	IX
----------	----

List of Figures	XIV
-----------------	-----

List of Tables	XV
----------------	----

List of Abbreviations	XVII
-----------------------	------

1 Introduction	1
-----------------------	----------

1.1 Motivation and Contribution	3
1.1.1 Autonomous Targetless Extrinsic Calibration for (Thermal) Camera and LiDAR	3
1.1.2 FLiVIS	4
1.1.3 LiDAR Bundle Adjustment	5
1.1.4 Hierarchical 3D Scene Graph based Semantic-Metric SLAM	6
1.2 Thesis Overview	7

2 Literature Review	9
----------------------------	----------

2.1 Autonomous Extrinsic Calibration	9
2.1.1 RGB-RGB and LiDAR-RGB Sensor Calibration	10
2.1.2 RGB(D)-thermal and LiDAR-thermal Sensor Calibration	11
2.2 Online SLAM Systems	12

2.2.1	Visual(-Inertial) Odometry	13
2.2.2	LiDAR(-Inertial) Odometry	13
2.2.3	LiDAR-Visual-Inertial Odometry	14
2.3	Optimized Mapping for Infrastructure Inspection Based on UAVs	15
2.3.1	Geometric SLAM	16
2.3.2	LiDAR BA	17
2.3.3	Autonomous Inspection System Based on UAVs	18
2.4	Hierarchical 3D Scene Graph based Metric-Semantic SLAM System	20
2.4.1	Sensor Fusion for Accurate 3D Object Detection	20
2.4.2	Metric and Semantic SLAM	21
2.4.3	3D Scene Graph for Hierarchical SLAM	22
3	Autonomous Targetless Extrinsic Calibration	25
3.1	Research Background	25
3.2	Preliminaries	28
3.2.1	Basic Notations	29
3.2.2	Lie Group and Lie Algebra	29
3.2.3	Derivatives on Lie Group $SO(3)$	31
3.3	Methodology	31
3.3.1	Edge Detection	32
3.3.2	Feature Matching	35
3.3.3	Extrinsic Calibration	36
3.4	Experiment Results	39
3.4.1	Simulation Results	40
3.4.2	Real-world Experiment	44
3.5	Conclusion	49
4	Feedforward-feedback LiDAR Visual Inertial System	51
4.1	Research Background	51
4.2	Methodology	54
4.2.1	Complementary Filter	54
4.2.2	One Dimensional Complementary Filter from Frequency Domain to Time Domain	56
4.2.3	Complementary Filter on Quaternions	57
4.2.4	Visual and LiDAR measurement model	59
4.3	Experiments	61
4.3.1	KITTI Dataset	61

4.3.2	NTU-Viral Dataset	64
4.4	Conclusion	68
5	Map Optimization Based on LiDAR Bundle Adjustment	69
5.1	Research Background	69
5.2	Methodology	71
5.2.1	LBA without Uncertainties	73
5.2.2	LBA with Uncertainties	75
5.3	Experiment	78
5.3.1	Lidar Bundle Adjustment on Synthesis Data	79
5.3.2	Bridge Mapping with Ground truth (GT) Model	82
5.4	Conclusion	86
6	Hierarchical 3DSG-based Semantic-Metric Mapping	89
6.1	Research Background	89
6.2	Methodology	93
6.2.1	Projection Model and Noise Propagation	93
6.2.2	Depth Estimation Accuracy for Stereo Camera	95
6.2.3	Depth Estimation Accuracy for LiDAR Camera Fusion	97
6.2.4	Absolute and Relative Errors in Hierarchical Structures	99
6.2.5	Hierarchical Scene Graph for Fruit Inspection	100
6.3	Experiment	103
6.3.1	Gazebo Simulation for Fruit Counting	104
6.3.2	TUM RGB-D Dataset	106
6.3.3	ICUAS'24 Competition Dataset	108
6.3.4	Hydroponic Laboratory Test for Strawberry Mapping	111
6.4	Conclusion	113
7	Conclusion and Future Work	115
	Bibliography	117

List of Figures

1.1	Examples of UAVs in inspection tasks. (a) Agriculture inspection. (b) Bridge inspection. (c) Power line inspection. (d) UAV inspection market size prediction.	2
1.2	UAVs with different sizes and sensor configurations	6
1.3	Hierarchical Spatial AI-SLAM framework following [1]’s paradigm, comprising calibration, localization, geometric and semantic mapping modules.	7
1.4	Thesis architecture	8
3.1	The sensor rig assembled on a UAV.	26
3.2	Schematic of the autonomous targetless extrinsic calibration process.	32
3.3	Different edge models.	34
3.4	Thermal edge extraction example. (a) Thermal edge detection with Sobel operator. (b) Thermal edge detection with second-order edge detector. (c) Thermal edge detection with combined edge detector.	35
3.5	Perspective-n-line algorithm schematic.	37
3.6	Example scenes in different modality. (a, d) RGB image, (b, e) thermal image, (e, f) LiDAR point cloud.	39
3.7	Simulation in Gazebo, RGB, depth, thermal image, and LiDAR point clouds are captured. Different temperature is assigned to the ground, different walls, and human model. The house has the same temperature as the gray wall.	41
3.8	LiDAR point cloud and 3D edge extraction in RVIZ.	42
3.9	Calibration result effected by the number of poses. (a) Rotation error. (b) Translation error.	43
3.10	Example calibration scenes. (a) Library corner (b) Library seat. (c) Library window.	44
3.11	Cost comparison using proposed thermal edge detector and Canny edge detector of LiDAR-thermal calibration in different scenes.	45

3.12	Compariosn experiment of calibration accuracy. (a) LiDAR-RGB rotation error. (b) LiDAR-RGB translation error. (c) LiDAR-thermal rotation error. (d) LiDAR-thermal translation error.	46
3.13	Computational Time comparison of different calibration settings	48
3.14	Extrinsic calibration parameters of methods with or without RANSAC (and null Huber Tukey robust loss functions). (a) Euler angles (Roll Pitch Yaw deg) with Yaw marked by diamonds; (b) Translation components (X Y Z, m) with minimal discrepancies.	48
3.15	Image fusion result of RGB image and LiDAR point cloud. (a) Raw RGB image, the FoV of the thermal camera is shown in the red box. (b) RGB image with LiDAR point cloud projection.	49
3.16	Image fusion result of thermal image and LiDAR point cloud. The FoV of the thermal camera is shown in the red box in Fig. 3.15. (a) Thermal image. (b) Thermal image with LiDAR point cloud projection.	50
4.1	System architecture of the proposed pipeline. The inputs are the measurements from the LiDAR, IMU, and camera. The IMU data is processed by the Madgwick filter, and the fusion of IMU estimation and the other two sensors is achieved through feedforward and feedback. The LIO states are maintained by an ESIKF, and VIO mainly depends on feedforward and feedback loops as in FLVIS.	53
4.2	Madgwick filter forms the perspective of the complementary filter.	55
4.3	Visual measurement model, the error is the 2D distance between the 3D waypoint P_i and the 2D image point p_i on the image plane.	60
4.4	Light Detection and Ranging (LiDAR) measurement model, the vector between point P_i (in red) in the current scan and its neighbor point Q_j (in green) is orthogonal to the normal vector of the k-nearest neighbors of P_i	61
4.5	LIO and VIO fusion results. (a) RGB images of the building. (b) Thermal image of the building. (c) Point cloud map colorized by RGB images. (d) Point cloud map colorized by thermal images.	62
4.6	Challenging scene for feature-based Visual Inertial Odometry (VIO), where the majority of an image is occupied by the sky, with feature points concentrated in the middle and lower parts of the image. This situation can easily lead to the failure of feature tracking algorithms, and the pose estimation algorithm may become unstable, or even diverge.	64

4.7	Qualitative comparison on KITTI dataset. (a) FLVIS result, and (b) FLiVIS result.	65
4.8	Trajectory of FLiVIS on the NTU-Viral dataset. (a) sequence eee_03, and (b) sequence sbs_03.	66
4.9	Mapping result of School of Biological Sciences, (a) LiDAR mapping result, and (b) top-down view of LiDAR mapping aligns with Google map.	67
5.1	Bridge inspection system architecture, from sensor fusion to perception, planning, and control	70
5.2	Visual bundle adjustment illustration.	71
5.3	LiDAR bundle adjustment illustration. (a) Raw LiDAR point cloud, red for the first frame, green for the second frame, and blue for the third frame. (b) Point cloud map after LiDAR Bundle Adjustment (LBA)	72
5.4	Cube feature point cloud before and after LiDAR Bundle Adjustment (BA). (a) Before LiDAR BA. (a) After LiDAR BA.	80
5.5	Different types of features for LiDAR BA in the simulation. (a) Plane features. (b) Edge features. (c) Corner features.	80
5.6	Comparison of accuracy and computational efficiency with benchmarks under different experimental configurations.(a)–(c): Fixed point cloud noise (5 cm) and LiDAR pose count (15), feature number 4–256; (a) rotational accuracy, (b) translational accuracy, (c) computational time.(d)–(f): Fixed point cloud noise and feature number (256), LiDAR pose count 2–17; (d) rotational accuracy, (e) translational accuracy, (f) computational time.(g)–(i): Fixed feature number (256) and LiDAR pose count (15), point cloud noise 1–8 cm; (g) rotational accuracy, (h) translational accuracy, (i) computational time.	81
5.7	The Entrepreneurship Overpass in BaoAn district, Shenzhen, China. (a) Satellite map, and (b) Computer Aided Design (CAD) model.	83
5.8	Data collection platform, a Livox Mid-360 LiDAR and a Realsense D435 RGB-D camera are mounted on a 15-inch Unmanned Aerial Vehicle (UAV).	83
5.9	The UAV is flying around the bridge piers and deck for coverage inspection task.	84
5.10	LiDAR point cloud mapping visualization results of the southwest corner of the Overpass.	87
5.11	Comparative analysis of point cloud-to-CAD structural deviations between our method and BALM2 [2].	87

6.1	Schematic diagram of the dual-subsystem architecture for agricultural inspection and harvesting.	90
6.2	System architecture of the proposed framework. Functional blocks are grouped into front-end (low-level perception) and back-end (high-level perception) layers, organized by increasing computational complexity and abstraction level. Data flows are indicated by: green for raw sensor data, red for semantic information, and blue for system-level communication.	91
6.3	Projection model and uncertainty propagation.	100
6.4	Plant and fruit reconstruction position error comparison in centimeters. . .	105
6.5	TUM RGB-D dataset test results. (a) 2D object detection in fr2_office. (b) 2D object detection in fr3_office. (c) three dimensional Scene Graph (3DSG) in fr2_office. (d) 3DSG in fr3_office.	107
6.6	ICUAS'24 dataset collection scenario. (a) The dataset collection orchard environment. (b) The UAV configuration.	108
6.7	Tree Segmentation in the Orchard.	109
6.8	The ICUAS'24 orchard mapping results with fruit landmarks and point cloud mapping.	109
6.9	System architecture of the intelligent hydroponic laboratory with embedded sensors and UAV.	110
6.10	Ripe and unripe strawberry detection by YOLOv8.	111
6.11	3D strawberry mapping in the hydroponic platform	112
6.12	Hydropionic platform strawberry 3D mapping accuracy comparison, (a) static mapping accuracy, (b) dynamic mapping accuracy.	113

List of Tables

2.1	Comparison of Different Multi-modal Extrinsic Calibration Methods	12
3.1	Sensor Specifications	40
3.2	Mean (and Std Dev) of Linear(e_t) and Angular(e_r) Calibration Errors in Different Pose	42
3.3	Calibration Accuracy Comparison using Cameras with Different Resolution in Gazebo Simulation	43
3.4	Extrinsic Calibration Comparison	47
4.1	Quantitative Results on KITTI Dataset	63
4.2	Absolute Trajectory Error in (m) on NTU-Viral Dataset	66
4.3	Computational Time (ms) on i7-8565U, 8GB RAM evaluated on NTU-Viral Dataset	68
5.1	Voxel Count Comparison Experiment	85
6.1	Experiment Datasets Configuration	103
6.2	Fruit Counting Accuracy in Gazebo Simulation Setup Priveded by ICUAS'24 Simulation Phase	105
6.3	Object Counting Accuracy in TUM RGB-D Dataset and Diomond Dataset	106
6.4	Object Counting Accuracy in ICUAS'24 Dataset	108
6.5	Strawberry Counting Accuracy in hydroponic Platform	112

List of Abbreviations

3DSG three dimensional Scene Graph

AI Artificial Intelligence

BA Bundle Adjustment

CAD Computer Aided Design

CNN Convolutional Neural Network

CPP Coverage Path Planning

CV Computer Vision

EKF Extended Kalman Filter

ENU East, North, and Up

FLiVIS Feedforward-feedback LiDAR Visual Inertial System

GPS Global Positioning System

GT Ground truth

HPF High Pass Filter

ICP Iterative Closest Point

IIR Infinite impulse response

IMU Inertial Measurement Unit

IoT Internet of Things

IoU Intersection over Union

KNN K Nearest Neighbors

LBA LiDAR Bundle Adjustment

LiDAR Light Detection and Ranging

LIO LiDAR Inertial Odometry

LLM Large Language Model

LPF Low Pass Filter

LVIO LiDAR Visual Inertial Odometry

MPC Model Predictive Control

NDT Normal Distribution Transformation

PnL Perspective-N-Line

ROS Robot Operating System

SLAM Simultaneous Localization and Mapping

SVIO Stereo Visual Inertial Odometry

TSP Travelling Salesman Problem

UAV Unmanned Aerial Vehicle

VO Visual Odometry

VIO Visual Inertial Odometry

VLM Visual Language Model

VPP View Point Planning

YOLO You Only Look Once

Chapter 1

Introduction

UAVs have become a transformative inspection platform, outperforming ground vehicles and legged robots in maneuverability, spatial accessibility, and efficiency. Capable of navigating complex, hard-to-reach environments (e.g., agricultural [3], bridges [4], and power grids [5] inspection, depicted in Fig. 1.1), UAVs provide an ideal solution for large-scale, high-frequency non-destructive inspection, and the UAV inspection market is growing steadily [6], as depicted in Fig. 1.1 (d). As the primary data acquisition tool in modern inspection workflows, they seamlessly capture visual, thermal, and other multi-sensory data, laying a solid foundation for subsequent data analysis. Advances in UAV hardware (e.g., lightweight actuators and long-endurance batteries) and flight control systems have further expanded their inspection applicability, making them indispensable for enhancing detection accuracy while cutting labor costs and mitigating safety risks.

In the UAV inspection stack, multi-sensor fusion is one of the bases for improving the reliability and comprehensiveness of UAV inspection data. Integrating complementary data from diverse sensors (RGB camera, thermal camera, LiDAR, etc.), Multi-sensor fusion offsets single-sensor limitations like environmental interference, incomplete data, and low accuracy. In inspections, it fuses spatial, spectral, and temporal information—e.g., merging LiDAR point clouds with RGB images for high-precision 3D models, or thermal and visible light data to detect hidden defects. This process enriches data semantics and boosts UAV robustness in complex environments (low light, harsh weather, clutter).

The rapid advancement of Simultaneous Localization and Mapping (SLAM) technologies underpins the autonomy of UAV inspection systems in GPS-denied scenarios, with LiDAR Visual Inertial Odometry (LVIO) systems emerging as a pivotal technical breakthrough. Unlike standalone SLAM such as VIO or LiDAR Inertial Odometry (LIO), LVIO synergistically fuses LiDAR geometric data, RGB visual features, and IMU motion mea-

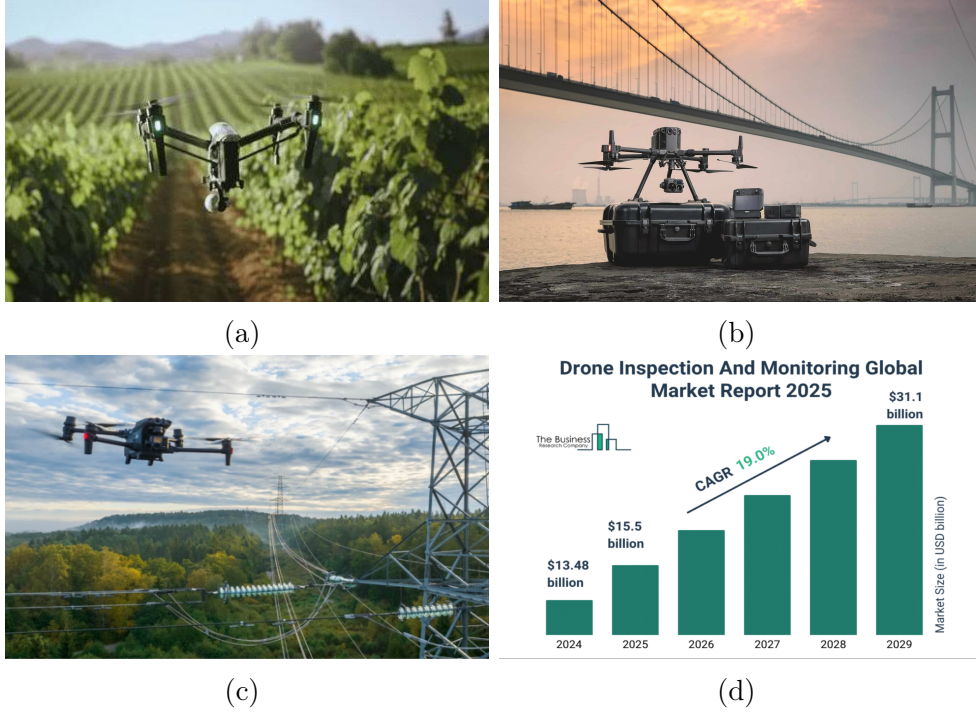


Figure 1.1: Examples of UAVs in inspection tasks. (a) Agriculture inspection. (b) Bridge inspection. (c) Power line inspection. (d) UAV inspection market size prediction.

surements to compensate for the inherent limitations of individual sensors—mitigating LiDAR’s point cloud sparsity, overcoming visual SLAM’s susceptibility to extreme lighting, and reducing IMU’s drift accumulation. For UAV inspections, SLAM enables robust onboard real-time localization and dense 3D mapping in unstructured environments (e.g., orchards), laying the foundation for autonomous path planning and target positioning.

The development of Spatial AI has further elevated the autonomy and intelligent operation level of inspection UAVs. A closed-loop Spatial AI paradigm was proposed in [1], ranging from an inner-loop SLAM module for motion control, through a medium-loop module for path and trajectory planning, to an outer-loop semantic mapping module for task planning. This integrated, holistic system is also applicable to UAV inspection tasks. Under the core Spatial AI framework, diverse functional modules are unified into an organic whole. Specifically, the SLAM module provides odometry information and achieves geometric understanding of scenes, while the Spatial AI module, based on SLAM outputs, endows the system with multi-dimensional scene understanding capabilities that cover semantics, affordances, dynamics, and beyond. This synergistic mechanism forms a closed loop with high-level decision-making modules (e.g., task planning modules), enabling users to issue high-level task commands to UAVs. The framework significantly enhances the autonomy and intelligent decision-making capacity of inspection operations,

serving as the core hardware-software bridge that translates fused multi-sensor data into an actionable decision-making basis for high-performance UAV inspections.

Building upon this basis, this research commences with multi-sensor fusion in Chapter 3 and establishes a multi-sensor SLAM in Chapter 4. Building upon the front-end localization, we developed LBA technology to optimize LiDAR mapping results in Chapter 5. Finally, we further extended the geometric mapping layer by introducing a 3DSG in Chapter 6 to manage semantic objects at different levels of abstraction. The LiDAR Bundle Adjustment (BA) technique was validated within the bridge inspection and mapping application. And the 3DSG technology was employed for fruit counting and mapping, validated within the intelligent vertical farming application. Finally, Chapter 7 concludes the thesis by summarizing key lessons learned and outlining future research directions.

1.1 Motivation and Contribution

The pursuit of intelligent robotic perception in complex, unstructured environments is driven by two complementary yet equally critical challenges. First, while individual sensors excel at measuring specific physical properties—such as geometry, color, or temperature—true environmental understanding demands multimodal integration, akin to biological perception in nature. This motivates our work on autonomous extrinsic calibration and multimodal sensor fusion, enabling robust perception under diverse and challenging conditions. Second, mere geometric or low-level semantic data (e.g., isolated object detections) remains insufficient for advanced robotic tasks such as manipulation or task planning. As AI advances—particularly in open-world detection and vision-language models—the need emerges for structured, hierarchical semantic representations that encode not only objects but also their relationships. Accordingly, we introduce scene graphs as a foundational semantic mapping framework, tailored for agricultural and other dynamic settings, to bridge the gap between low-level perception and high-level reasoning.

1.1.1 Autonomous Targetless Extrinsic Calibration for (Thermal) Camera and LiDAR

Sensor fusion combining RGBD cameras, LiDAR, and thermal cameras significantly

enhances robot perception, localization, and decision-making by providing complementary visible light, depth, and temperature data. However, achieving effective fusion relies critically on precise extrinsic calibration to align data from these diverse modalities. While LiDAR-RGB and RGB-thermal calibration methods are relatively established, LiDAR-thermal calibration remains particularly challenging and underexplored, hindering the full utilization of thermal cameras in applications like service robots.

The core challenges stem from the difficulty in finding cross-modal correspondences due to the fundamentally different data domains (visible light, specific IR wavelengths for LiDAR, thermal radiation) and the complementary nature of the sensors (texture, precise distance, temperature). Traditional target-based calibration is cumbersome and impractical for potential recalibration needs caused by sensor movement. Furthermore, the complexity of thermal imaging and the scarcity of dedicated robotics algorithms have resulted in a severe lack of robust autonomous, targetless calibration methods specifically for thermal cameras, especially for LiDAR-thermal pairs.

To address these challenges, we propose an autonomous and targetless calibration framework for both LiDAR-RGB and LiDAR-thermal extrinsic calibration. This approach, believed to be the first of its kind for directly obtaining LiDAR-thermal extrinsics without targets or manual intervention, leverages an analysis of thermal imaging principles to develop novel thermal edge detection and cross-modal edge-matching algorithms. The framework’s effectiveness was evaluated across various scenes for convergence, precision, and accuracy.

1.1.2 FLiVIS

SLAM is essential for robot operation in unknown environments, such as UAV search and rescue. Common systems utilize exteroceptive sensors (LiDAR or cameras) fused with Inertial Measurement Unit (IMU) data, implemented as VIO, Stereo Visual Inertial Odometry (SVIO), or LIO. However, single exteroceptive sensors often lack robustness in challenging conditions (e.g., VIO fails under lighting changes, LIO degrades with poor geometry). While recent LVIO approaches fuse cameras and LiDAR, filter-based or optimization-based frameworks face high computational demands as data volume increases, posing challenges for resource-constrained platforms like UAVs [7].

To address this, we propose a feedforward-feedback framework building on prior work [8]. It leverages the complementary nature of IMU sensors: gyroscopes (accurate short-term, low-frequency noise) and accelerometers (noisy short-term, accurate long-term). A

complementary filter jointly processes IMU data for rotation prediction, while feedforward and feedback mechanisms integrate LiDAR and visual measurements. This structure combines Kalman filter advantages with decoupled models, providing better initial estimates for faster convergence and improved accuracy in LiDAR-visual-inertial fusion [9].

The resulting system, Feedforward-feedback LiDAR Visual Inertial System (FLiVIS), integrates a VIO subsystem (based on FLVIS [8]) and a LIO subsystem (based on Fast-LIO2 [9]). Evaluated on the KITTI [10] and NTU VIRAL [11] datasets for autonomous driving and UAV inspection, FLiVIS achieves accuracy comparable to state-of-the-art systems while enabling real-time performance on resource-limited hardware.

1.1.3 LiDAR Bundle Adjustment

Contemporary robotic automation systems significantly enhance workplace safety and efficiency in critical sectors like infrastructure inspection and urban security. Robots perform hazardous tasks (e.g., 24/7 patrols) and UAVs enable visual assessments, reducing human risk and operational costs [12, 13]. These systems follow a standardized architecture with four core modules: SLAM (typically LiDAR-inertial-camera fusion [14]), motion control (e.g., Model Predictive Control (MPC) [15]), path planning (sampling-based algorithms [16]), and environmental perception (deep learning for sensor fusion [17], segmentation [18]).

Despite progress, key limitations persist: (1) Noisy LiDAR mapping lacks precision for applications like bridge management; (2) Geometric maps lack semantic layers for functional understanding of specialized components; (3) Existing sim2real pipelines are unidirectional, lacking reverse optimization. These deficiencies hinder high-precision structural inspection and semantic-aware operation.

To address these challenges, we propose a hierarchical semantic SLAM framework and bidirectional sim2real system. Key contributions include: (1) Structural Feature-Enhanced LiDAR Bundle Adjustment incorporating primitives (corners/edges/planes) to improve map consistency; (2) A Three-Layer Semantic Architecture: geometric (topology-aware point clouds), structural (parametric component models via CAD matching), and semantic (attribute fusion via 3DSG); (3) A Bidirectional Sim2Real Framework enabling dual validation in Gazebo and physical bridge environments for full-coverage inspection.

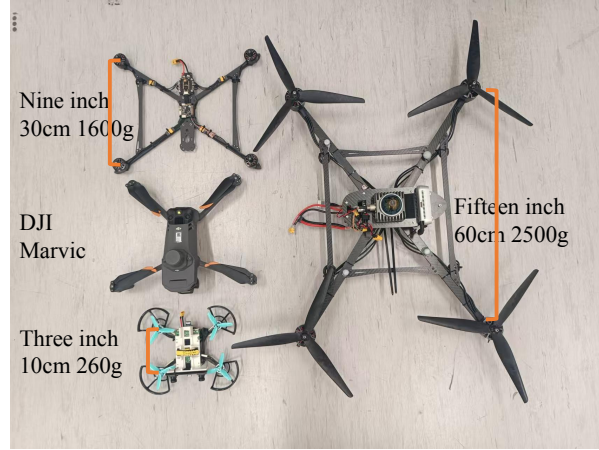


Figure 1.2: UAVs with different sizes and sensor configurations

1.1.4 Hierarchical 3D Scene Graph based Semantic-Metric SLAM

Under dual pressures of population growth and climate change, smart agriculture - empowered by multimodal technologies (Large Language Model (LLM), robotics, Internet of Things (IoT))—is reshaping production systems. This approach establishes a closed-loop "perception-analysis-decision-execution" system, overcoming traditional agriculture's resource inefficiency while supporting sustainable development aligned with UN Sustainable Development Goals [19]. Our research focuses on autonomous intelligent vertical farming, utilizing abundant urban structures (e.g., high-rises) in dense regions like Hong Kong and Singapore, where systems integrate UAVs to autonomously regulate growth parameters through perceptual feedback [20].

UAVs have evolved beyond operational tools into intelligent agricultural hubs, addressing infrastructure gaps via a "device-edge-cloud" architecture [21] that bridges farm-digital connectivity [22]. They perform diverse tasks (seeding, mapping) using multimodal sensors, with edge computing enabling agricultural knowledge graphs for AI decision-making [23]. However, perception limitations hinder autonomy: SLAM generates semantically deficient maps; 2D-focused detection fails to model 3D growth dynamics; and current methods neglect biological-light coupling while struggling with orchard complexities [24, 25].

To address these challenges, we propose a metric-semantic SLAM framework using 3DSG for plant inspection. This enables interpretable growth analysis and traceable monitoring through multidimensional spatiotemporal representation. We systematically analyze object estimation uncertainties across sensors (stereo/RGB-D/camera-LiDAR) and model error propagation, validating effectiveness in simulations and real-world tests.

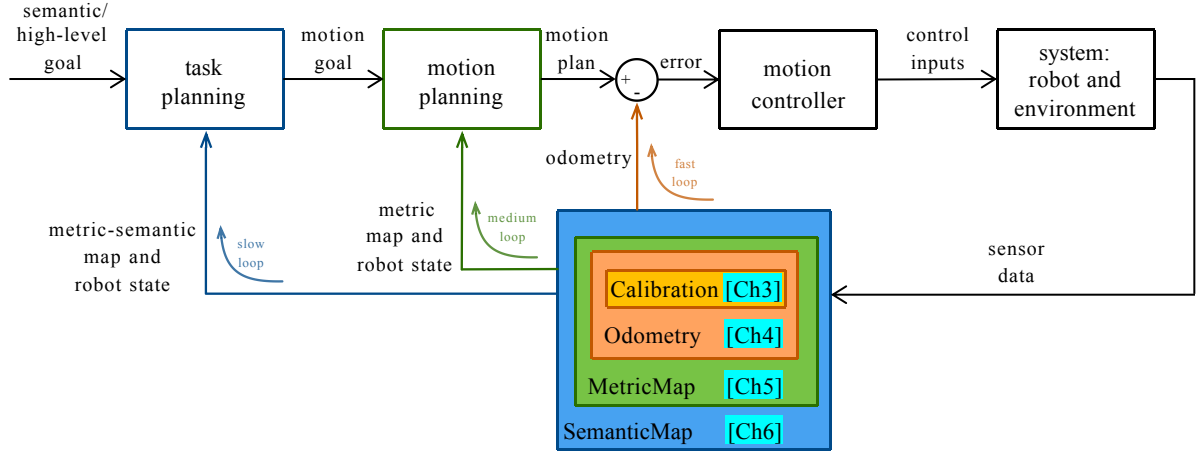


Figure 1.3: Hierarchical Spatial AI-SLAM framework following [1]’s paradigm, comprising calibration, localization, geometric and semantic mapping modules.

The framework integrates IoT-based mobile robotic perception with automated environmental control, establishing a closed-loop smart vertical farming system for real-time crop regulation. We also developed several UAV platforms with varying specifications to meet the requirements of different application scenarios. As shown in Fig. 1.2, the drones range in size from 3 inches to 15 inches, with takeoff weights between 250 grams and 2500 grams. These platforms are equipped with different sensors for different purposes.

1.2 Thesis Overview

From the perspective of the entire autonomous intelligent system, spatial AI extends the geometric reasoning capability of SLAM by endowing it with concurrent semantic and physical reasoning abilities [1]. Aligned with the paradigm established in The SLAM Handbook, a seminal work in the field, our framework is structured hierarchically across four core modules, as depicted in Fig.1.3. First, the calibration module in Chapter 3 furnishes essential extrinsic and intrinsic parameters for multi-sensor fusion within the SLAM pipeline. Second, the SLAM module in Chapter 4, leveraging odometric measurements as input, achieves low-level geometric understanding of the operating environment. Third, building upon the geometric map generated by SLAM and LiDAR BA in Chapter 5, the Spatial AI module integrates a suite of scene understanding components in Chapter 6, encompassing semantic attributes and functional properties of environmental entities. Finally, these multi-modal scene representations enable higher-level decision-making modules (e.g., task planning and motion control) to form a closed-loop system,

Chapter 2

Literature Review

In this chapter, we will conduct a comprehensive review of the relevant literature, following the same structural sequence as presented in Chapter 1. First, Section 2.1 provides an overview of various sensors and the core concepts of sensor fusion technology. Second, Section 2.2 offers a focused review on the topic of simultaneous localization and mapping (SLAM). Next, Section 2.3 first introduces LiDAR BA for pointcloud map optimization, then shifting to a holistic perspective to delve into the application research of UAV inspection systems. Finally, Section 2.4 summarizes the metric and semantic mapping technologies associated with fruit counting and orchard management. This review framework allows us to grasp the application background and inherent interconnections of each technical module from a top-down perspective, rather than confining our focus to specific techniques such as SLAM and path planning. However, this does not imply that an in-depth examination of the technical details of these modules is trivial. On the contrary, we will elaborate on the technical principles and research progress of each module in subsequent subsections. This balanced approach ensures a thorough understanding of both the overarching framework and the intricate technical details of the research content.

2.1 Autonomous Extrinsic Calibration

Extrinsic calibration can be categorized based on various factors or characteristics. Two factors, in particular, significantly influence the applicability and cost considerations [26]: 1) autonomous versus manual calibration; 2) calibration with or without a target [27]. Initially, the manual target-based approach was the most basic and commonly employed method. However, the autonomous calibration method without the need for target pairs

is highly sought after due to its cost-effectiveness and broader scene coverage, making it the preferred choice. In this section, we conduct a literature review on sensor extrinsic calibration from these two dimensions. In Section 2.1.1, we first review the calibration of RGB-RGB and the calibration of LiDAR-RGB sensor calibration. In Section 2.1.2 we review RGB-thermal and LiDAR-thermal sensor calibration.

2.1.1 RGB-RGB and LiDAR-RGB Sensor Calibration

As the most widely used and most similar sensor to human vision, the RGB camera is intensively studied and best understood. For camera calibration, the most widely applied method is the one developed by Zhang [28]. This method uses a checkerboard with known size as a target and manually extracts features to obtain responses between real-world and image points. Subsequent research has largely focused on improving and extending Zhang’s method. Some researchers have provided more user-friendly GUIs [29] to label the key points manually, while others have tried to develop better algorithms for automatic checkerboard corner detection [30].

With the rapid development of autonomous driving in the past decade, the price of LiDAR sensors has also dropped significantly [31]. As a 3D sensor, LiDAR is naturally complementary to cameras, so there has been a lot of research on LiDAR-RGB calibration [32]. The development of LiDAR-RGB is also similar to camera calibration. In the early days, target-based and manual methods were predominant [33]. Then some researchers developed autonomous feature detection [34] feature matching [35]. Most recently, autonomous targetless extrinsic calibration methods have drawn the attention of the research community [36, 37].

The main breakthroughs and innovations in calibration methods are in two areas:

1. towards autonomous feature detection;
2. towards generic scenes.

In the first domain, with the advancement of computer vision technology, automatic feature detection algorithms have continuously evolved, thereby simplifying the calibration process. Simultaneously, these automatic detection algorithms have also enhanced the precision when utilizing fixed calibration patterns [38]. In the second domain, feature detection methods have expanded beyond the confines of checkerboards. Automatic detection approaches for objects such as boxes [35], spheres [39], and other entities have

been developed. Furthermore, methods for detecting people, skylines, or edge discontinuities have also emerged [40]. Overall, the progress in this field is geared towards more versatile features and increasingly automated and accurate feature detection. However, these two factors somewhat impose constraints on each other since more general features may result in reduced accuracy. To enhance accuracy and automation, the utilization of more specific calibration targets becomes necessary.

In summary, the core process of these methods is:

1. Feature detection;
2. Feature matching;
3. Extrinsic solving.

Although there are also other schools of thought, such as information theory-based methods [41], the most widely used, mature, and reliable technique, offering the highest accuracy, remains the aforementioned three-step process. From Zhang’s pioneering method [28] to the present day, the variations among different techniques lie primarily in whether features are manually or automatically extracted and whether these features are obtained from artificially created targets or extracted from the surrounding environment. As the pursuit of greater autonomy and universality in extrinsic calibration methods intensifies, so does the challenge. Consequently, we speculate that the main direction of development lies in autonomous-targetless calibration methods.

2.1.2 RGB(D)-thermal and LiDAR-thermal Sensor Calibration

There exists a scarcity of studies focused on the extrinsic calibration between thermal cameras and other sensors; however, the calibration process bears similarities to that of RGB cameras [42]. The use of calibration checkerboards is common in earlier researchers [43]. Nonetheless, the creation of a checkerboard for thermal cameras poses challenges due to the requirement of introducing a temperature difference onto the checkerboard, which subsequently induces heat transfer, while the colored pattern remains fixed [44]. Numerous endeavors have been undertaken to devise reliable calibration patterns [45]. Nevertheless, the manufacturing of such patterns proves exceedingly laborious, thereby impeding the widespread adoption of thermal cameras and their fusion with other types of sensors.

Following the standard pipeline of 1) feature extraction, 2) feature matching, and 3) extrinsic solving, one of the main difficulties is to align thermal and visible or geometric features by finding features in the environment. Lussier *et al.* [46] used the human body as a target because the temperature of the human body is usually significantly different from the ambient temperature. Recently, Fu *et al.* [47] used buildings to extract edges and then used these edges for calibration.

While several attempts have been made to calibrate thermal cameras and LiDAR sensors directly, no successful outcomes have been reported thus far. Fu *et al.* [47] have made notable progress towards this objective, although the presence of relatively large errors in their method can be attributed to the inherent blurriness of thermal images when compared to colored images [48] and LiDAR point clouds are sparse. These factors present challenges for optimization algorithms, hindering their convergence and the attainment of accurate results. In our research, we thoroughly analyze these factors and propose solutions to mitigate their impact, potentially establishing the first autonomous and targetless calibration method capable of directly calibrating thermal cameras, RGB cameras, and LiDAR sensors. Table 2.1 provides an overview of the sensor modalities and algorithmic limitations associated with various approaches for multi-modal extrinsic parameter calibration.

Table 2.1: Comparison of Different Multi-modal Extrinsic Calibration Methods

Method	LiDAR-RGB	LiDAR-thermal	auto	targetless
Beltran’s [34]	✓	×	✓	×
Yuan’s [36]	✓	×	✓	✓
Fu’s [47]	✓	×	✓	✓
Zhang’s [49]	✓	✓	✓	×
Ours	✓	✓	✓	✓

2.2 Online SLAM Systems

SLAM is the core technology for autonomous mobile systems to navigate unknown environments, realizing real-time ego-pose estimation and local mapping. To compensate for single-sensor limitations, sensor-fusion variants like VIO and LIO have been developed. VIO fuses camera and IMU data, offering a cost-effective solution for texture-rich sce-

narios such as AR/VR devices and lightweight UAVs. LIO integrates LiDAR geometric data with IMU motion constraints, enhancing robustness against extreme lighting and dynamic obstacles, thus suiting high-precision tasks like autonomous driving and field robotics (e.g., orchard UAVs).

2.2.1 Visual(-Inertial) Odometry

Visual Simultaneous Localization and Mapping (V-SLAM) [50] has been widely applied in robotics and AR/VR applications. With regard to different sensor properties, it has evolved significantly with variations including Mono(monocular), stereo, and RGBD V-SLAM. Notable systems such as PTAM [51], and ORB-SLAM [52] and DSO [53] and SVO [54] have been developed in the past decade. Based on the extraction of feature points, V-SLAM methodologies can be categorized into two distinct categories: feature-based and direct methods. The feature-based approach [55] extracts salient feature points, a process that can be computationally intensive and time-consuming. Conversely, the direct method [56] circumvents the feature extraction step, focusing solely on the computation of pixel gradients, thereby offering a more expedient solution. However, this efficiency comes at the cost of increased susceptibility to failure.

A common enhancement across these systems is the fusion with IMU [57], which has been observed to substantially improve the robustness of the SLAM system. This improvement has been particularly beneficial for UAV navigation due to its lightweight and robust nature, thereby broadening the scope of its applications. Significant VIO systems are the Vins series [58], Openvins [59], MSCKF [60], and FLVIS [8].

2.2.2 LiDAR(-Inertial) Odometry

Previously, due to factors such as large size and high cost, LiDAR was primarily used in vehicle autonomous driving, with less application in UAVs. However, in recent years, with the advancement of LiDAR hardware technology [61], its size has been continuously reduced and the price has also decreased. Consequently, there are now examples of LiDAR being used in UAVs [62]. This demonstrates the expanding applicability of LiDAR technology in the field of UAVs.

The majority of existing LiDAR odometry methodologies encompass two primary components: scan registration and mapping [63]. Scan registration is facilitated by the

Iterative Closest Point (ICP) [64] or normal distribution transformation(NDT) [65] process. The ICP process iteratively searches the K Nearest Neighbors (KNN) [66] to accomplish the registration of point cloud scans between consecutive frames, while the NDT method directly matches the probability density function of the current scan to the last scan or the map. Different from the scan registration step, mapping operates at a lower frequency [67], registering newly added point clouds into the local map. This two-step structure forms the backbone of contemporary LiDAR odometry systems, enabling efficient spatial recognition and navigation.

The integration of IMU can significantly enhance the accuracy and robustness of LiDAR odometry [9]. This is achieved by compensating for motion distortion in a LiDAR scan and providing an optimal initial pose required for the ICP process. Recent advancements in LiDAR-inertial fusion have led to the development of more tightly coupled systems that perform odometry within a compact local map [68], composed of a fixed number of recent LiDAR scans or keyframes. This scan-to-local-map registration typically yields higher accuracy than scan-to-scan registration by utilizing more recent information. For instance, LIO-SAM [69] requires a nine-axis IMU to generate attitude measurements as a prior for scan registration within a small local map. Similarly, LINS [70] incorporates a tightly coupled iterated Kalman filter and a robocentric formula into the LiDAR pose optimization process. However, due to the small size of the local map used to achieve real-time performance, the odometry tends to drift rapidly. This necessitates a low-rate mapping process, such as map refining (as in LINS), sliding window joint optimization (as in [71]), and factor graph smoothing (as in LIO-SAM [69]).

2.2.3 LiDAR-Visual-Inertial Odometry

The utilization of multiple sensors in LiDAR-Visual-Inertial SLAM systems enhances their capability to navigate various challenging environments, even in instances where one sensor fails or is partially degraded [72]. This advantage has motivated the development of several LiDAR-Visual-Inertial SLAM systems within the research community, demonstrating the adaptability and resilience of these systems in diverse environmental conditions. For instance, LIMO [73] utilizes depth information extracted from LiDAR measurements for image features and estimates motion among keyframes using bundle adjustment. In contrast, R2LIVE [72] employs a different approach, extracting features from both LiDAR and images and then conducting re-projection error within the framework of an Iterated Error-State Kalman Filter. Notably, R2LIVE provides a tightly-coupled sys-

tem for fusing LiDAR-Visual-Inertial sensors. In LVI-SAM [74], the accuracy of the VIO is enhanced by extracting depth information of visual features using LiDAR measurement data. Correspondingly, the LIO utilizes the estimates from VIS as initial values to aid in scan matching. Loop closure is initially identified by VIO and further improved by LIO. These diverse systems underscore the dynamic evolution and application of LiDAR-Visual Inertial SLAM methodologies in the field.

Similar to VIO and LIO, most LVIO fuses data from different sensors by Kalman filter-based [60] approach or optimization-based [75] approach. The Markov assumption in the Kalman filter-based method makes it prone to failure in noisy data, while the optimization-based methods are usually computationally costly [76].

In this work, we exploit the complementarity of IMU measurement [77] and fuse the information from different sensors by feedforward and feedbacks. A better initial value is fed into visual or LiDAR pose estimation in the pipeline, thus the system is lightweight and robust. The main contributions of this work are listed as follows:

- A novel pose estimation framework that fuses the LiDAR visual and inertial measurement through designed feedforward and feedbacks.
- Performance evaluation on both the KITTI autonomous driving dataset and NTU-Viral UAV dataset. The benchmark evaluation results are comparable to the top-ranked existing algorithms.

2.3 Optimizaed Mapping for Infrastructure Inspection Based on UAVs

Traditional inspection tasks have relied heavily on manual labor, where damage to the surfaces of buildings, bridges, or aircraft is assessed through visual observation [78]. This process often involves accessing hard-to-reach areas, requiring ladders or workers to climb ladders, which not only heightens the risk factor for workers but also escalates costs.

The advent and maturation of UAVs present a viable solution to these challenges, allowing for the recording of hard-to-reach places, effectively replacing human eyes. At this stage, UAVs used in inspection tasks are typically pilot-driven and serve as auxiliary tools in inspection tasks [79]. However, with technological advancements, drones are now capable of detecting targets based on preset objectives within a known environment, marking the emergence of semi-autonomous inspection systems. Considering the potential real-world applications, such as bridge inspection [80], building inspection [81], and

aircraft inspection [82], one representative case is selected from each scenario for detailed analysis. This approach ensures a comprehensive understanding of the capabilities and limitations of UAVs in various inspection tasks.

2.3.1 Geometric SLAM

SLAM has been developing for several decades [13], and benefits from both hardware and software improvements. In terms of hardware, it ranges from, IMU [58], LiDAR [83], and Global Positioning System (GPS) to multi-sensor fusion [84]. The software is divided into the front-end and the back-end. The front-end includes the processing of raw sensor data, while the back-end optimizes the data processed by the front-end at a lower rate to obtain better results. Kalman filter [85] and factor graph optimization [86] are common technologies for back-end optimization. Overall, the advancements in both hardware and software of metric SLAM have contributed to its increasing positioning accuracy and widespread adoption.

Visual SLAM employs feature-based [87] or direct methods [88] for localization, with monocular [58], stereo [8], and RGBD [89] variants. Feature-based approaches rely on computationally intensive keypoint extraction, while direct methods prioritize efficiency via pixel gradients but sacrifice robustness. Fusion with IMUs enhances reliability, enabling VIO systems like VINS-Mono [58] and OpenVINS [59], which is crucial for UAV navigation. Similarly, LiDAR odometry utilizes scan registration (ICP/Normal Distribution Transformation (NDT)) and mapping, where IMU integration mitigates motion distortion and initializes pose estimation. Systems like LIO-SAM [90] leverage local maps and factor graphs to address drift, though rapid error accumulation remains a challenge. LVIO systems leverage cross-modal fusion to mitigate sensor degradation. R3LIVE++ [91] integrates LiDAR-visual data via Kalman filtering, while LVI-SAM [92] couples VIO with LiDAR depth and LIO with visual constraints. These approaches—whether filter-based (prone to Markov noise sensitivity) or optimization-driven (accuracy-computation trade-offs)—highlight the robustness-efficiency balance in complex environments, advancing resilient autonomy through heterogeneous sensor fusion.

2.3.2 LiDAR BA

The BA algorithm has a long history in geodesy, Computer Vision (CV), and visual-SLAM, and it has been reinvented by researchers in different fields. We adopt the definition from [93]: the problem of refining a 3D reconstruction to produce jointly optimal 3D structure and viewing parameter (sensor pose) estimates. BA plays a fundamental role in many vital tasks, such as Visual Odometry (VO) [94] [87], VIO [59] [8], and 3D reconstruction [95]. Even though visual BA theory is mature, visual sensors are often affected by environmental conditions, making it difficult to obtain accurate results. However, thanks to recent advancements in hardware technology and the decreasing cost of LiDAR sensors, LBA research has gradually gained attention [96]. Nevertheless, LBA is much less mature compared to visual BA.

The inherent sparsity of LiDAR point clouds imposes structural limitations on LBA, precluding feature extraction mechanisms and cross-frame point-to-point correspondence establishment—a foundational capability of visual BA. Consequently, established visual BA paradigms exhibit limited applicability to LBA frameworks due to this fundamental disparity in data modality constraints. In the literature, multi-frame ICP [97], which is the most similar to LBA, can be mainly divided into two approaches. One approach uses loop closure constraints to simultaneously optimize the poses of all frames, thereby avoiding error accumulation and drift. This method is also used in LOAM systems, such as [92], but it does not consider mapping consistency. Another approach continuously estimates a virtual frame as a reference using the EM algorithm, methods in this category include [97] and [98]. However, due to the computational cost, these methods cannot be applied to large-scale LiDAR point clouds.

To enhance computational efficiency and improve feature matching fidelity, researchers commonly restrict LBA to planar or edge features within point clouds, these features are adopted by [99], BALM [100], and BALM2 [96]. A critical component in these methods is the point cluster abstraction. While effective in structured environments, such feature selection strategies degrade markedly in unstructured settings. To address this limitation, recent advances focus on enforcing mapping consistency through point cloud curve fitting, relaxing rigid geometric constraints to incorporate generalized curved surfaces (PSS-BA [101]), and semantic cues are also applied for better feature association [102]. However, most existing methods rely on implicit cost function minimization, seldom incorporating explicit uncertainty-aware feature representation. While sufficient for geometric mapping fidelity, this paradigm inadequately addresses higher-level autonomy requirements and poses integration challenges for semantic-aware systems tasked with

infrastructure inspection and management.

2.3.3 Autonomous Inspection System Based on UAVs

Autonomous Building Inspection System Based on UAVs

Gao *et al.* [103] have addressed the tasks of indoor inspection and scene reconstruction in two steps. They proposed an efficient and safe "explore-then-exploit" system that can be easily deployed in any scenario without any initial understanding of the environment. They introduced a novel two-step SLAM method for the explore-then-exploit system, where LiDAR SLAM Fast-lio2 [9] and visual SLAM VINS [58] are integrated, providing robust positioning and 3D map information for two types of drones. They also proposed a coverage path planning algorithm for inspection tasks, specifically the furthest point sampling algorithm is used to generate viewpoints. This guarantees the coverage of the target area and the safety of the viewpoint while considering flight efficiency to achieve the shortest flight path.

The entire system can operate fully automatically without the need for human intervention. However, their selection settings for the detected targets are somewhat naive, merely setting bounding boxes at different coordinates, and the coordinates of the bounding boxes also need to be known in advance. This assumption is reasonable for known buildings, but it is not applicable for targets such as **bridges** [80] and **power lines** [104] that are difficult to fix within a bounding box, or for **aircraft** [82] of different sizes in a hangar.

Autonomous Bridge Inspection System Based on UAVs

The application scenario of this paper is infrastructure inspection. From a system-wide perspective, aerial inspection systems are widely used in scenarios such as bridges [105], power lines [106], buildings [107], and industrial facilities [108]. The commonality among these scenarios is that they are all difficult or hazardous areas that require remote inspection. Important modules include SLAM [13], full coverage path planning [109], and perception, such as defect detection [17] and semantic segmentation [105]. This paper mainly focuses on the SLAM and perception subsystems. We first introduce works that focus on geometric SLAM, including different sensors such as IMU, RGB, and LiDAR sensors [84], and different algorithms such as various feature-based front-ends [110], and filter-based or optimization-based back-ends [8]. In this section, we introduce related

work about traditional geometric SLAM in 2.3.1, and subsequently discuss LiDAR BA for better mapping results in 2.3.2.

Autonomous Aircraft Inspection System Based on UAVs

The traditional method of aircraft inspection involves manual labor, where inspectors walk around the aircraft, use work ladders, and climb ropes to approach the inspection area. The main method of inspection is visual, with handheld devices used for key areas. The disadvantages of this method are its high difficulty, slow speed, low coverage, poor positioning, and identification accuracy, difficulty in collecting data at the optimal distance and viewing angle of the sensor, difficulty in integrating multiple detection methods, and inability to implement dynamic warnings [111].

The use of UAVs for inspection overcomes many of these challenges. UAVs can approach different inspection areas and rely on various sensors such as visible light, infrared, and ultrasound. The advantages of this method include lower difficulty, faster speed, full coverage, high positioning, and identification accuracy, ensuring the optimal distance and viewing angle of the sensor for data collection, the ability to integrate multiple detection methods, and the ability to establish a database for dynamic warnings.

A TUM team proposes an extension to the LOAM framework that enables reference object-based trajectory and map optimization. The approach assumes that the location and geometry of a large reference object are known, such as a CAD model from BIM or a previously captured dense point cloud model [111]. The method uses the poses of the LOAM algorithm as an initial guess to refine them with scan-to-model alignment. An Extended Kalman Filter (EKF) based motion prior filtering step is employed to evaluate if the alignment was accurate. The past trajectory is optimized by adding the model-aligned pose as a pose graph constraint and the map of the LOAM algorithm is corrected to improve future localization and mapping. The approach was evaluated with data captured in a visual airplane inspection scenario inside an aircraft hangar¹. The results show that enabling the proposed extension yields a reduction in Absolute Pose Error compared to conventional LOAM [112] and R-LOAM [82], while being able to run online. This reduces drift and improves map quality.

The difference between aircraft inspection and building inspection, bridge inspection, or power line inspection is that virtually all aircraft have accurate CAD models. Therefore, most of the work in path planning is based on existing CAD models, without the need to focus on point cloud mapping or 3D reconstruction. Currently, there is no comprehensive work that includes mapping and coverage path planning for aircraft inspection. Therefore, we can also work in this area, which would give us a head start. In addition to the

aforementioned solutions, another possible method is to collect a point cloud map for the aircraft and align it with the CAD model. Based on this alignment, coverage path planning for aircraft inspection can then be carried out.

2.4 Hierarchical 3D Scene Graph based Metric-Semantic SLAM System

Despite rapid advancements in robotic perception technologies, a significant cognitive gap persists between computer algorithm paradigms and human cognition [113]: low-level perception focuses on geometric localization (e.g., spatial XYZ coordinates), while computer vision-based high-level abstract perception, despite breakthroughs in object detection, still suffers from fragmented multi-level representations in algorithms like robotic SLAM. The inherent incompatibility between heterogeneous information, such as geometric point clouds and semantic objects leads to fragmented environmental representations, ultimately constraining downstream task effectiveness like precise manipulation. To address these challenges, this paper systematically reviews research progress through three dimensions: sensor fusion (2.4.1) for multimodal data synergy enabling 3D object detection, metric-semantic mapping (2.4.2) bridging geometric-semantic divides, and hierarchical scene graphs (2.4.3) constructing logically connected structured environmental representations.

On the other hand, the research on smart and sustainable agriculture faces a dual mission: it should address the limitations of traditional agricultural production models in terms of resource efficiency and environmental adaptability, while simultaneously establishing a closed-loop scientific system to address resource-population conflicts [12]. Its technical framework comprises four core modules: perception, analysis, decision-making, and execution. The perception module, implemented through IoT networks, enables data acquisition and execution feedback, serving as the data foundation of the closed-loop system that directly determines its performance ceiling.

2.4.1 Sensor Fusion for Accurate 3D Object Detection

To establish a comprehensive environmental perception system, multiple core computer vision technologies, including object detection and object tracking, have been widely in-

tegrated into perception systems. Among these, 3D object detection technology plays a foundational role by intelligently predicting the spatial coordinates, dimensional parameters, and category attributes of critical objects (e.g., crops or machinery) surrounding agricultural vehicles. This provides object-level information support for downstream tasks such as branch pruning [114], fruit picking [115], and other labor-intensive operations. Current mainstream methodologies in this field can be categorized into three sensor-driven technical approaches: vision camera-based, LiDAR-based, and multi-sensor fusion solutions [14].

While current research on 3D object detection predominantly focuses on autonomous driving and domestic service robots [25], exploration in agricultural scenarios remains in its infancy. Compared to highly structured urban environments [10], agricultural settings present three core challenges [116]: complex spatial topologies formed by intertwined foliage, high-density information distribution, and dynamic lighting variations, which render generic perception algorithms and datasets inadequate for precision agriculture demands [117]. Existing agricultural semantic datasets exhibit significant limitations: early datasets [118] provided only 2D RGB images lacking depth information; recent efforts like BUP20 [119] (which includes RGB-D data for greenhouse sweet pepper instance segmentation [120]) still suffer from narrow scene coverage and insufficient sample diversity. Although the Cyrill team’s multimodal apple orchard dataset achieves tree-level instance segmentation [121], it fails to overcome the technical barrier of fruit-level fine-grained detection. To address these gaps, this study innovatively proposes a fruit-level object detection algorithm based on multimodal data fusion. By establishing a cross-scale feature correlation mechanism, the proposed method achieves the first synergistic optimization of spatial localization and semantic recognition for fruits in agricultural environments, effectively bridging the research void in fine-grained perception models for this domain.

2.4.2 Metric and Semantic SLAM

Through 3D object detection algorithms, we can obtain an object’s coordinates in the body frame (i.e., its position relative to the robot). However, to derive the object’s world coordinates, knowledge of the robot’s position relative to the world frame is required, which necessitates SLAM technology. Metric SLAM has evolved over decades [13], driven by advancements in both hardware and software [122] [123]. Hardware innovations have spanned sensors such as IMU [58], LiDAR [83], GPS, and multi-sensor fusion systems [124] [125]. Software advancements are typically divided into front-end [126] and

back-end [86] components. The front-end processes raw sensor data, progressing from traditional feature detection techniques to deep learning-based feature extraction [11]. The back-end optimizes front-end outputs at lower frequencies, evolving from Kalman filters [127] to factor graph optimization [86]. Collectively, these developments have enhanced positioning accuracy and enabled metric SLAM’s broad adoption across domains.

Although metric SLAM has achieved significant progress, the world, for us, is not merely a collection of geometric primitives (points [128], lines [129], surfaces [100]) or isolated pixels/LiDAR point clouds. These map elements inherently carry higher-level semantic information, motivating extensive research on integrating semantics into SLAM systems. In semantic perception, transformative progress has been made over the past decade, beginning with the ImageNet dataset and AlexNet [130]. Technologies such as object detection and semantic segmentation [105] have matured significantly. Recently, the rise of foundation models [131] has ushered in a new era for Artificial Intelligence (AI), further expanding the horizons of semantic understanding.

2.4.3 3D Scene Graph for Hierarchical SLAM

Semantic SLAM, as an intersection of SLAM and CV, includes a wide range of studies that integrate higher-level semantic information into purely metric SLAM maps [132]. We categorize this body of work as metric-semantic SLAM, which includes various approaches [133]. However, most of these works maintain flat metric-semantic maps, where elements of different abstraction levels, such as basic landmark points and objects, are indiscriminately stored in the same map database [134]. Typically, attributes distinguish whether a point is a landmark or other object (e.g., a bottle) [135]. This fragmented combination fails to reflect the relationships between elements in the map and does not provide the robot with a holistic view of its surroundings.

To achieve a more complete and holistic perception, Armeni *et al.* [136] first proposed the hierarchical 3DSG to organize semantic-metric elements at different levels of abstraction. A 3DSG is a layered graph where nodes represent spatial concepts at multiple levels of abstraction (from low-level geometry to objects, places, rooms, buildings, etc.) and edges represent relations between these concepts. In this direction, Kimera [24] and Hydra [113] proposed 3DSG, which is a multi-resolution hierarchical graph structure to organize these semantic elements at different levels of abstraction. However, these works primarily focus on an ideal simulation setting that produces precise semantic segmentation outcomes. As reported in Hydra [113], even the GT-Trajectory configuration does

not achieve 100% accuracy for “% Found” and “% Correct”. Achieving entirely flawless and all-encompassing semantic segmentation results in real-world scenarios is impractical, highlighting the necessity for further improvements in object mapping accuracy under noise interference.

In this work, we address the limitations of existing metric-semantic SLAM systems, which primarily focus on calculating the coordinates of objects without considering their orientation or the hierarchical relationships between different levels of abstraction. These simplifications may be acceptable in an ideal simulation environment, but are not feasible in the real world, which is characterized by noise and bias. To address these issues, Chapter 6 presents a metric-semantic SLAM framework, as shown in Fig. 6.2, for fruit inspection based on 3DSG. By leveraging the top-down structure of the scene graph, our work significantly improves the accuracy of fruit counting by reducing the impact of various errors on object pose estimation. The systematic analysis of error sources and the hierarchical approach proposed in this paper represent important advancements in the field of metric-semantic SLAM.

Chapter 3

Autonomous Targetless Extrinsic Calibration

3.1 Research Background

In computer vision and robotics applications, sensor fusion enhances various functionalities such as robot perception, localization, SLAM, and decision-making. The combination of RGBD cameras, LiDAR, and thermal cameras has demonstrated synergistic benefits that surpass the output of individual sensors. By merging information from these diverse sensors, a robot’s perception system can acquire data encompassing visible light, ambient temperature, and depth. Fig. 3.1 illustrates an example use case where an aerial robot equipped with LiDAR, RGBD camera, and thermal camera efficiently performs multi-modal inspection tasks. However, sensor fusion also presents challenges, with precise alignment of sensor data becoming crucial. External calibration is a key step to establish associations between sensors and ensure accurate alignment for effective fusion.

Extrinsic calibration entails establishing correspondences between data obtained from two different sensors, encompassing combinations such as LiDAR-RGB, RGB-thermal, LiDAR-thermal, and others. Our primary focus centers on multi-modal extrinsic calibration, specifically LiDAR-RGB, and LiDAR-thermal calibrations. Among these, LiDAR-RGB calibration has been extensively studied [28] [34], followed by RGB-thermal calibration [45], while LiDAR-thermal calibration remains relatively underexplored [137].

Various calibration methods can be broadly categorized as target-based or targetless, and manual or autonomous. Early approaches predominantly employed target-based and manual techniques. Autonomous and targetless calibration is most desired, though this poses the greatest challenge. Currently, autonomous and targetless methods are preva-

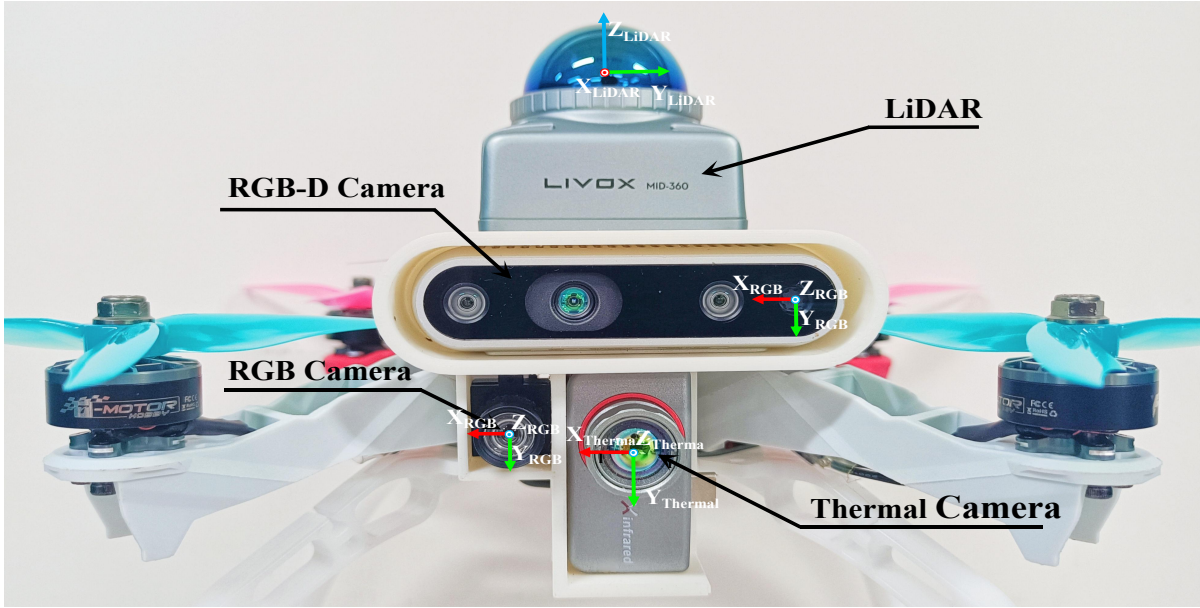


Figure 3.1: The sensor rig assembled on a UAV.

lent for RGB-LiDAR calibration. While LiDAR-thermal calibration methods are virtually non-existent. Many existing extrinsic calibration methods that involve a thermal camera still rely on target-based approaches. This lack of research and development in LiDAR-thermal calibration hampers the full utilization of thermal cameras' advantages, particularly in the context of service robots. The absence of robust calibration techniques impedes the advancement of multi-sensor fusion and downstream algorithms, preventing the realization of their full potential.

Extrinsic calibration encounters its first major challenge in achieving cross-modal information correspondence, which proves difficult even when performed manually. Each sensor captures data from different domains: thermal cameras capture infrared (IR) waves, RGB cameras capture visible light, and LiDAR relies on IR but with a different wavelength, each representing unique and complementary information. RGB cameras are adept at capturing detailed texture information, whereas LiDAR provides precise distance measurements despite having sparser data points. On the other hand, thermal cameras excel in perceiving temperature, making them particularly suitable for tasks like pedestrian detection and avoidance in service robots and autonomous driving. The complementary nature of these sensors enhances their effectiveness when used together.

However, their complementarity presents challenges in achieving extrinsic calibration, as establishing relevant correspondences between sensors to solve for extrinsic parameters becomes increasingly difficult. Traditional calibration methods involve creating calibration patterns capable of generating temperature information and possessing associated

features like texture and specific geometric shapes. These common features are then extracted manually or automatically. However, the complexity and bulkiness of such calibration patterns make their manufacturing arduous. Moreover, if the extrinsic parameters of the sensors on the robot change due to vibrations or collisions, the original calibration becomes invalid, further calibration and intervention will be required or false alignment will be caused.

The second challenge arises from the intricate nature of the thermal imaging process and the dearth of robotics algorithms designed to address the autonomous-targetless calibration problem. The complexity inherent in thermal imaging presents significant challenges for researchers, resulting in a relative scarcity of research focused on thermal camera calibration compared to other sensor types. As a result, most existing calibration methods for thermal cameras tend to rely on target-based or manual approaches. The development of automatic calibration algorithms specifically tailored for thermal cameras is limited, and even fewer options exist for autonomous-targetless calibration methods. This lack of automated calibration techniques for thermal cameras impedes the full utilization of their capabilities within multi-sensor fusion systems, hindering a comprehensive understanding of their potential benefits across various applications.

We tackle the above two problems by investigating the thermal imaging principle and a cross-modal common feature extraction method is developed. A LiDAR-RGB, and LiDAR-thermal calibration framework is proposed. The main contributions of this work are as follows:

- We propose an autonomous and targetless calibration framework for LiDAR-RGB and LiDAR-thermal sensors calibration, which is the first autonomous targetless calibration method that can directly obtain the extrinsic parameters between the thermal camera and LiDAR sensors based on our knowledge;
- Based on the analysis of thermal edge properties, we propose thermal edge detection and edge-matching algorithms, and evaluated them in various scenes in terms of convergence precision, and accuracy;
- The source codes is released ¹ to benefit the community.

The structure of this article is organized as follows. We review the related works of multisensor fusion and autonomous targetless calibration in Section II. Section III provides some preliminaries. Section IV details the extrinsic calibration methodology.

¹https://github.com/allenthreee/thermal_range_calib

Simulation and real-world experiment results are presented in section V. Finally, we draw the conclusion in section VI.

In computer vision and robotics applications, sensor fusion enhances various functionalities such as robot perception, localization, simultaneous localization and mapping (SLAM), and decision-making. The combination of RGBD cameras, LiDAR, and thermal cameras has demonstrated synergistic benefits that surpass the output of individual sensors. By merging information from these diverse sensors, a robot’s perception system can acquire data encompassing visible light, ambient temperature, and depth. Fig. 3.1 illustrates an example use case where an aerial robot equipped with LiDAR, RGBD camera, and thermal camera efficiently performs multi-modal inspection tasks. However, sensor fusion also presents challenges, with precise alignment of sensor data becoming crucial. External calibration is a key step to establish associations between sensors and ensure accurate alignment for effective fusion.

The most widely used method for extrinsic calibration can be divided into three steps: 1) feature extraction, 2) feature matching, and 3) extrinsic parameter estimation, as shown in Fig. 3.2. Common features include point features, line features, and surface features. Feature matching can be divided into two types: 1) descriptor-based matching, which calculates the similarity of features using descriptors, and 2) spatial-based matching, which calculates the spatial distance of features and selects matched features with similar spatial distances. Point feature matching is usually achieved by descriptors and is used more in camera images, while line feature matching often uses spatial distance, and line features are commonly used in cross-modal calibration. In this paper, we extract line features and match them using spatial distance.

3.2 Preliminaries

In this paper, we minimize the reprojection error of edge features to solve for the extrinsic parameters (i.e., relative pose including rotation and translation) between different sensors. The translation vector can be represented by a 3-dimensional vector, while there are multiple choices for representing rotation, such as Euler angles, quaternions, rotation matrices, and rotation vectors. To facilitate optimization without introducing new constraints (quadratic constraint of unit quaternions), we use rotation matrices (Lie Group on $SO(3)$) and rotation vectors (Lie Algebra of the Lie Group $SO(3)$) to represent rotation and optimize using rotation vectors that are living on smooth manifolds. We minimize the cost function using nonlinear optimization on a smooth manifold. To clearly state the

problem, we provide a brief review of some useful concepts.

3.2.1 Basic Notations

We use $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ to denote 3D rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ to denote 3D translation vector and $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$ to denote 3D rigid body transformation. Considering two transformation matrices $\mathbf{T}_1 = \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{0} & 1 \end{bmatrix}$ and $\mathbf{T}_2 = \begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$, where $SE(3)$ represents the Special Euclidean Group in three dimensions, we define the transformation as $\mathbf{T}_1 \cdot \mathbf{T}_2 = \begin{bmatrix} \mathbf{R}_1 \mathbf{R}_2 & \mathbf{t}_1 + \mathbf{R}_1 \mathbf{t}_2 \\ \mathbf{0} & 1 \end{bmatrix}$. We use T_{21} to denote the transformation matrix from coordinate 1 to coordinate 2. For a specific scenario, we aim to add two vectors r_v^{pq} and r_w^{qi} . Here, the vector r_w^{qi} points from point q to point i located in the world coordinate system, while r_v^{pq} points from point p to point q in the vehicle coordinate system. To achieve this, we first transform the vector r_v^{pq} to the world coordinate system using T_{vw} . Subsequently, we add this transformed vector to r_w^{qi} . The complete transformation operation can be described as follows:

$$r_w^{pi} = T_{vw} r_v^{pq} + r_w^{qi}. \quad (3.1)$$

3.2.2 Lie Group and Lie Algebra

The Special Orthogonal Group (the Rotation Group) describes the group of rotation matrices and it is formally defined as $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} | \mathbf{R}\mathbf{R}^\top = \mathbf{I}, \det(\mathbf{R}) = 1\}$, and the Special Euclidean Group describes the group of rigid motion in 3D, which is the semi-direct product of $SO(3)$ and $\mathbb{R}^3$. And it is defined as $SE(3) = \{\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} | \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3\}$.

The Group $SO(3)$ forms a smooth manifold, and in its tangent space of the identity is the Lie Algebra $\mathfrak{so}(3)$ (which can be identified with the rotation vectors) of $SO(3)$. The Lie Algebra $\mathfrak{so}(3)$ consists of all skew-symmetric matrices $\mathbf{R} \in \mathbb{R}^{3 \times 3}$. This means every skew-symmetric matrix in $\mathbb{R}^{3 \times 3}$ can be mapped by a vector $w \in \mathbb{R}^3$ by the *hat* operator

$(\cdot)^\wedge$:

$$\mathbf{w}^\wedge = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \in \mathfrak{so}(3). \quad (3.2)$$

For the 3D rotation group $SO(3)$, we have its Lie algebra:

$$\mathfrak{so}(3) = \{\phi \in \mathbb{R}^3, \Phi = \phi^\wedge \in \mathbb{R}^{3 \times 3}\}. \quad (3.3)$$

And for the 3D transformation group $SE(3)$, we have its Lie algebra:

$$\mathfrak{se}(3) = \{\xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6, \xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}\}, \quad (3.4)$$

where $\rho \in \mathbb{R}^{3 \times 3}$, $\phi \in \mathfrak{so}(3)$. We can map a skew-symmetric matrix to a vector $\mathbf{w} \in \mathbb{R}^3$ using the *vee* operator $(\cdot)^\vee$.

The exponential map (at the identity) $\exp: \mathfrak{so}(3) \rightarrow SO(3)$ maps an element of the Lie Algebra to its Lie Group (the rotation matrix) and coincides with standard matrix exponential (Rodrigues' formula):

$$\exp(\phi^\wedge) = \mathbf{I} + \frac{\sin(\|\phi\|)}{\|\phi\|} \phi^\wedge + \frac{1 - \cos(\|\phi\|)}{\|\phi\|^2} (\phi^\wedge)^2. \quad (3.5)$$

The logarithm map (at the identity) associates a matrix $\mathbf{R} \neq \mathbf{I}$ in $SO(3)$ to a skew-symmetric matrix:

$$\log(\mathbf{R}) = \frac{\varphi \cdot (\mathbf{R} - \mathbf{R}^\top)}{2\sin\varphi} \text{ with } \varphi = \cos^{-1} \frac{\text{tr}(\mathbf{R}) - 1}{2}. \quad (3.6)$$

The capitalized are convenient shortcuts to map vector elements. For notational convenience, we adopt capitalized Exp and Log maps to map vector elements to the Rotation matrix and vice versa:

$$\text{Exp} : \mathbb{R}^3 \rightarrow SO(3) \quad ; \quad \phi \mapsto \exp(\phi)^\wedge, \quad (3.7)$$

$$\text{Log} : SO(3) \rightarrow \mathbb{R}^3 \quad ; \quad \mathbf{R} \mapsto \log(\mathbf{R})^\vee, \quad (3.8)$$

which operate directly on vectors, rather than on skew-symmetric matrices in $\mathfrak{se}(3)$.

3.2.3 Derivatives on Lie Group $SO(3)$

Following [138] we define plus and minus operators to introduce increments between elements on the manifold of $SO(3)$, which are expressed on the tangent space of the manifold. In our notation, we denote the plus and minus operators as $\oplus$ and $\ominus$, respectively. The $\oplus$ adds an incremental element in the Lie algebra $\mathfrak{so}(3)$ to an element within the Lie group $SO(3)$. The $\ominus$ subtracts two elements within the Lie group $SO(3)$ and yields an element within Lie algebra $\mathfrak{so}(3)$.

Giving a 3D rotation $f : SO(3) \rightarrow \mathbb{R}^3$ of a 3D point $\mathbf{p}$; $f(\mathbf{R}) = \mathbf{R}\mathbf{p}$, we have:

$$\begin{aligned}
 \frac{d\mathbf{R}\mathbf{p}}{\mathbf{p}} &= \lim_{\boldsymbol{\theta} \rightarrow 0} \frac{(\mathbf{R} \oplus \boldsymbol{\theta})\mathbf{p} \ominus \mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} = \lim_{\boldsymbol{\theta} \rightarrow 0} \frac{(\mathbf{R}\text{Exp}(\boldsymbol{\theta})\mathbf{p} - \mathbf{R}\mathbf{p})}{\boldsymbol{\theta}} \\
 &= \lim_{\boldsymbol{\theta} \rightarrow 0} \frac{\mathbf{R}(\mathbf{I} + [\boldsymbol{\theta}]_{\times})\mathbf{p} - \mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} = \lim_{\boldsymbol{\theta} \rightarrow 0} \frac{\mathbf{R}[\boldsymbol{\theta}]_{\times}\mathbf{p}}{\boldsymbol{\theta}} \\
 &= -\mathbf{R}[\mathbf{p}]_{\times} \in \mathbb{R}^{3 \times 3}.
 \end{aligned} \tag{3.9}$$

3.3 Methodology

The most widely used method for extrinsic calibration can be divided into three steps: 1) feature extraction, 2) feature matching, and 3) extrinsic parameter estimation, as shown in Fig. 3.2. Common features include point features, line features, and surface features. Feature matching can be divided into two types: 1) descriptor-based matching, which calculates the similarity of features using descriptors, and 2) spatial-based matching, which calculates the spatial distance of features and selects matched features with similar spatial distances. Point feature matching is usually achieved by descriptors and is used more in camera images, while line feature matching often uses spatial distance, and line features are commonly used in cross-modal calibration. In this paper, we extract line features and match them using spatial distance.

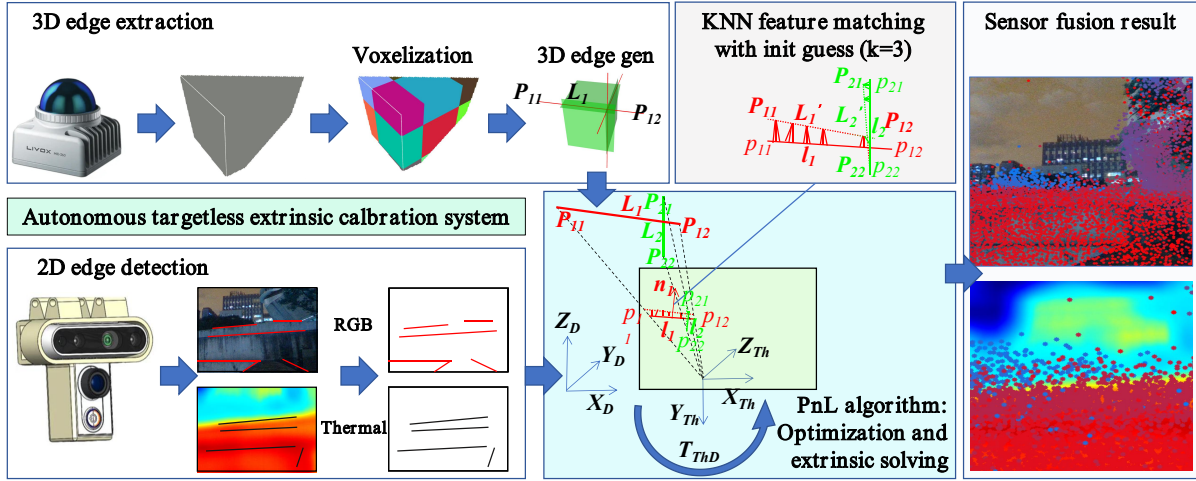


Figure 3.2: Schematic of the autonomous targetless extrinsic calibration process.

3.3.1 Edge Detection

LiDAR Edge Detection

LiDAR generates 3D point clouds, capturing 3D spatial information without being affected by changes in lighting. Extensive research has been conducted on extracting features from LiDAR point clouds, typically treating discontinuities in the 3D point cloud as 3D edges. However, the point clouds generated by LiDAR have bleeding edges, and extracting all edges would introduce significant errors and reduce the accuracy of the calibration algorithm. To avoid such error source, connected edges are preferred. In this paper, we primarily use the method proposed by Yuan et al. [36] to extract 3D edges. Specifically, as shown in the top-left of Fig. 3.2, the RANSAC algorithm is used to fit planes on dense point clouds. Then, the intersection lines of the planes are extracted as connected edges and prepared for subsequent calibration. For solid-state LiDAR such as Livox series, dense point clouds can be obtained by accumulating point clouds at the same location. For mechanical (rotating) LiDARs, the LiDAR needs to be moved around and algorithms such as Fast-LIO [139] are run to generate dense point clouds.

Thermal Edge Detection

The key distinctions between thermal images and RGB images can be summarized in two aspects. Firstly, most RGB images represent steady-state conditions, whereas thermal images capture transient states. Secondly, thermal edges tend to be blurrier compared to RGB edges. In most scenarios, the color and shape of an object remain unchanged, while

its temperature undergoes changes, flowing from high to low. RGB images rely on visible light to capture information related to color and geometry, while thermal images capture temperature differences. Consequently, the dynamic nature of temperature variations introduces the difference between transient and steady states in thermal images (whereas RGB images can be considered as representing steady states), and it also contributes to the blurred appearance of thermal edges. The widely used Canny Edge algorithm, which has been optimized over the years for RGB images, is not specifically tailored for thermal images. Therefore, we have developed a novel edge detection algorithm to address the aforementioned issues.

Transient and steady-state temperature fields are influenced by the principles of Fourier's law, which governs the flow of heat transfer from high-temperature objects to low-temperature objects. In the absence of an external heat source, the temperature field of an isolated system with a temperature difference will evolve over time until the temperature becomes homogeneous. In a thermal image, black tends to "flow" towards white. In contrast, in an RGB image, there is no inherent "flow" from black to white based on temperature. Hence, we distinguish between transient temperature differences and steady-state temperature differences in thermal images based on whether the temperature changes over time (i.e., the presence or absence of an external heat source maintaining the temperature difference). We classify a stable environment that can be maintained during the period of data collection for calibration (approximately 30 minutes) as a steady state. On the other hand, scenes that need to be heated multiple times such as passive calibration patterns are categorized as transient.

In the absence of a stable heat source, artificially created temperature differences for calibration purposes are typically transient. A checkerboard pattern heated by a heat gun is an example of a transient state. In these artificially created transient temperature difference scenarios, due to their transient nature and the gradual reduction of temperature differences, the clarity of boundaries is limited, leading to increasingly blurred thermal edges during the calibration process.

Stable temperature differences in an environment are generated by heat sources such as human body temperature [46], and active calibration patterns that have heating resistors. When the heat dissipation from the air and the heat generated by these sources reach equilibrium, steady-state temperature differences occur, leading to local stability. Observing that the temperature variation of building contours is relatively slow, within the data collection period of several tens of minutes, it can be considered as a steady-state condition. Therefore, structured environments capable of generating temperature differences can be utilized as calibration scenes. This approach eliminates the need for

external heat sources or active calibration patterns, reducing complexity and cost while still providing a suitable environment for calibration.

However, before utilizing building contours for calibration, it is necessary to address the issue of detecting blurry edges in thermal images. Heat transfer occurs at the boundaries between objects with different temperatures, resulting in a diffuse phenomenon at the edges, making thermal images typically blurrier compared to RGB images. As shown in Fig. 3.3, there are four common image edge models: step, ramp, pulse, and ridge. In thermal images, the temperature cannot exhibit a sudden change (or maintain a sudden change continuously) because heat conducts and flows. Unlike color and geometry, which can remain constant at a pixel, temperature values cannot. Therefore, step and pulse models, which are common in RGB images, are almost non-existent in thermal images. The remaining ramp and ridge edge models are more prevalent in thermal images.

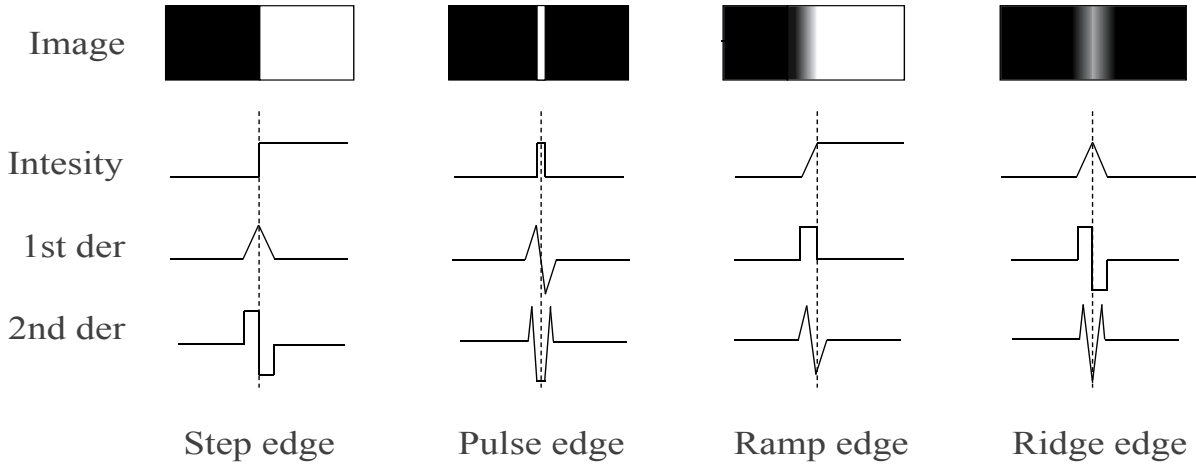


Figure 3.3: Different edge models.

Most edge detection algorithms are developed for the common step and pulse models observed in RGB images, where the first-order gradient is used to detect edges. These algorithms are unable to effectively detect edges corresponding to the ramp and ridge models commonly found in thermal images, as shown in Fig. 3.3, the first-order derivative for ramp and ridge edge models is not local maximum. While the second-order derivative for the ridge model is the local maximum.

Ramp edges typically occur at the boundary of different mediums, such as the boundary of solid and air, which are also bleeding edges for LiDAR edges. On the other hand, ridge edges typically occur at the junctions of different materials or where there are geometric changes of the same material, such as corners of walls. Hence ridge edges in thermal images correspond to connecting edges in the 3D space, which are preferred for

LiDAR extrinsic calibration. Therefore, this study proposes a method that combines the first-order and second-order derivatives for thermal image edge detection. Traditional Canny edge detection [140] is used for RGB image edge detection. An example of the proposed edge detection method and Canny edge detection is shown in Fig. 3.4.

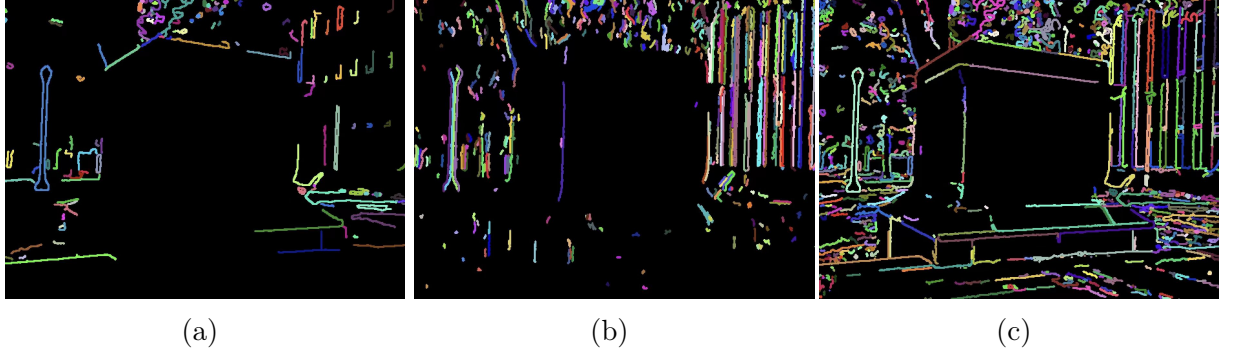


Figure 3.4: Thermal edge extraction example. (a) Thermal edge detection with Sobel operator. (b) Thermal edge detection with second-order edge detector. (c) Thermal edge detection with combined edge detector.

3.3.2 Feature Matching

According to the summary of Section II, feature matching is mainly divided into two types: the former is based on feature descriptor, which is often used for point features; and the latter is based on spatial relationship, which is mainly used for edge or line features. In this paper, we propose a novel edge-matching method based on spatial relationships. The edge matching process is summarized in Algorithm 1. This method utilizes geometric constraints and edge orientation to achieve robust and accurate results. By incorporating the spatial information of edges, it can handle complex scenes with occlusions and cluttered backgrounds. The experimental results demonstrate that our method outperforms the state-of-the-art methods in accuracy and robustness.

Assume M 3D edges are extracted from the 3D point cloud and N 2D edges are extracted from the RGB image or thermal image. Denote the set of edge points in 3D point cloud as $\mathbf{P} = \{P_i^j; \quad i = 1, \dots, M; \quad j = 1, \dots, \text{edge}[i].\text{size}()\}$, and the set of edge points in 2D image plane as $\mathbf{p} = \{p_i^j; \quad i = 1, \dots, N; \quad j = 1, \dots, \text{edge}[i].\text{size}()\}$. To build the matching relationship between 2D and 3D edges, we first randomly sample multiple points in every 3D edge, then project this 3D edge point cloud to 2D image plane according to the initial extrinsic and search in the neighbor area of the projected points. Lastly,

the image coordinate of the projected point is adjusted to the mean value of its k nearest neighbors. After all the sampled points have been matched, we can match the 3D edge with the 2D edges.

Algorithm 1 2D and 3D Edge Matching Algorithm

Input: 2D edges l , 3D edges L , cam intrinsic K , init extrinsic T_{init} , k , ϵ_d , ϵ_n , α , n

Output: Matching pairs $\{L, l\}$

```

1: for  $i = 1$  to  $3D\_edges.size()$  do
2:   Sample  $n$  points  $P_i^1, \dots, P_i^n$  on  $L_i$ ;  $c = 0$ 
3:   for  $j = 1$  to  $n$  do
4:      $p_i^j = K \cdot T_{\text{init}} \cdot P_i^j$  (project 3D to 2D)
5:     Search  $k$  nearest neighbors of  $p_i^j$ :  $q_i^{j,1}, \dots, q_i^{j,k}$ 
6:      $q_i = \frac{1}{k} \sum_{pt=1}^k q_i^{j,pt}$  (refined projection)
7:      $\mathbf{n}_i =$  normal vector of 2D edge at  $q_i$  (quantify edge direction)
8:     Find closest 2D edge  $p_i$  to  $q_i$ 
9:     if  $\text{Dist}(p_i, q_i) \leq \epsilon_d$  and  $\text{AngleDist}(L_i, \mathbf{n}_i) \leq \epsilon_n$  then
10:        $c = c + 1$ ;  $l_{\text{cand}} = l_m$ 
11:     end if
12:   end for
13:   if  $c/n \geq \alpha$  then
14:     Append  $\{L_i, l_{\text{cand}}\}$  to pairs
15:   end if
16: end for
17: return edge pairs  $\{L, l\}$ 

```

3.3.3 Extrinsic Calibration

Having edges extracted and matched, we will address the problem of extrinsic calibration of a camera **C** or a thermal camera **Th** and a LiDAR **LiDAR**. Let $\mathbf{T}_{CLiDAR}$ be the relative pose between the reference frame from the LiDAR to the camera and $\mathbf{T}_{ThLiDAR}$ be the relative pose from the LiDAR to the thermal camera. To simplify notation, we denote $T = [R|t] \in \mathbb{SE}(3)$ as the relative pose from the LiDAR to the other two 2D sensors.

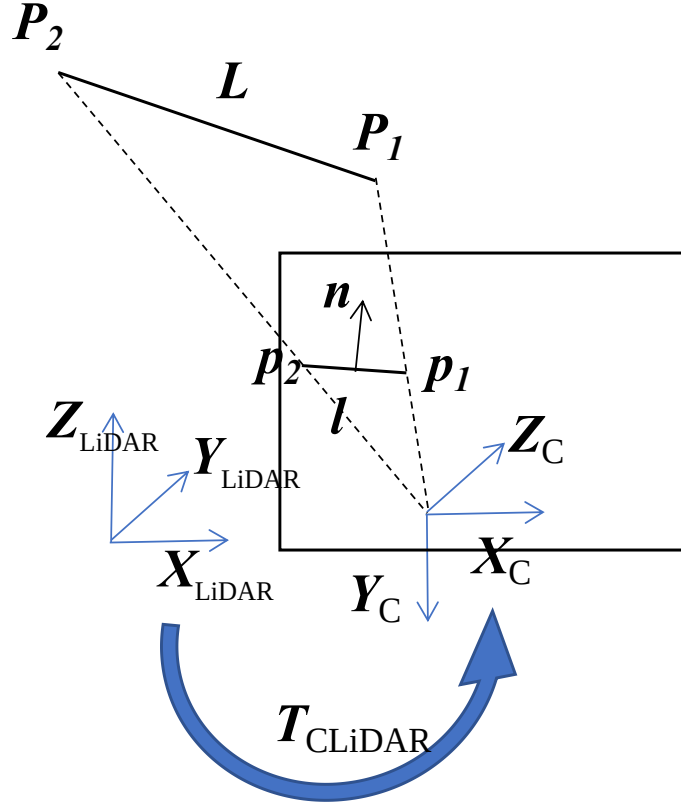


Figure 3.5: Perspective-n-line algorithm schematic.

Perspective-n-Line

As shown in Fig. 3.5, given a pair of 2D-to-3D line correspondence, we have 2 constraints on the camera pose [141]. Formulating the extrinsic solving as a Perspective-N-Line (PnL) problem, the relative pose could be solved by at least three line correspondences. Most scenes hold more than 3 edge correspondences, and the extrinsic parameter could be solved by at least one shot.

Assume the intrinsic matrix of the camera is K_C , and the homogeneous coordinate of a 2D line l_i^j is $\mathbf{l}_i^j$. The back-projection plane of l_i^j is $[K_C \mathbf{l}_i^j; 0]$ [141], and the normal of the 2D line on the back-projection plane l_i^j is $\mathbf{n}_i^j$. A straight line could be represented by a unit direction vector $\mathbf{v}$ and a point $\mathbf{p}$ which is closest to the origin of the line. Together with the camera center, the 3D line forms a norm vector in the camera frame $\mathbf{n}_C = (n_C^x, n_C^y, n_C^z)^\top$.

Considering one line correspondence in 3D and 2D space $L_i^j \leftrightarrow l_i^j$, the 2D line in the thermal image is the intersection of the 3D line with the thermal image plane. The 2D line, the 3D line, and the camera center lie on the same plane which can be parameterized

by the camera center and a normal vector $\mathbf{n}$. As in (3.1), we can transform the normal vector from the LiDAR coordinate to the camera coordinate as follows:

$$\mathbf{n}_C = \mathbf{T}_{CLiDAR} \cdot \mathbf{n}^{LiDAR}. \quad (3.10)$$

For any 3D point $\mathbf{P} = (x, y, z)^\top$ point on the 3D line and a 2D point $p = (u, v, 1)^\top$ on its corresponding 2D line, we have the constraint with the norm vector formed by the 3D line and the camera center. After transforming the normal vector into the thermal camera coordinate, $\mathbf{n}_C$ can form a constraint with any point on the corresponding thermal edge.

$$0 = \mathbf{n}_C (\mathbf{K}_C \mathbf{T}_{CLiDAR} \mathbf{P} - \mathbf{p})^\top. \quad (3.11)$$

(3.11) can be viewed as the cost function of the extrinsic parameter. The extrinsic parameter can be obtained by solving the least-mean-square error problem:

$$\mathbf{T}_{CLiDAR} = \arg \min_{\mathbf{T}_{CLiDAR}} \frac{1}{2} \left\| \sum_{i=1}^n \mathbf{n}_C (\mathbf{K}_C \mathbf{T}_{CLiDAR} \mathbf{P}_i - \mathbf{p}_i)^\top \right\|_2^2. \quad (3.12)$$

Since the main re-projection error is caused by noise orthogonal to the direction of the line, and is unrelated to the line length. A scalar len , which is the length of the line, is applied to the distance function to make longer and shorter edges contribute equally to the cost function.

In order to obtain an unconstrained optimization form, we solve (3.12) on the tangent space of the manifold. As in (3.4), we use $\boldsymbol{\xi} \in \mathbb{R}^6$ to represent the relative transformation between the thermal camera and the range sensor. Then the error term can be expressed as:

$$e(\boldsymbol{\xi}) = \mathbf{n}_C (\mathbf{K}_C \exp(\boldsymbol{\xi}^\wedge) \mathbf{P}_i - \mathbf{p}_i)^\top. \quad (3.13)$$

The derivative can be derived using the $\oplus$ and $\ominus$ operator defined in Section III-C.

$$\frac{\partial e}{\partial \delta \boldsymbol{\xi}} = \lim_{\boldsymbol{\xi} \rightarrow 0} \frac{e(\delta \boldsymbol{\xi} \oplus \boldsymbol{\xi}) - e(\boldsymbol{\xi})}{\delta \boldsymbol{\xi}} = \frac{\partial e}{\partial \mathbf{T}_{CLiDAR} \mathbf{P}} \frac{\partial \mathbf{T}_{CLiDAR} \mathbf{P}}{\partial \delta \boldsymbol{\xi}}. \quad (3.14)$$

Substituting (3.9) into (3.14), we have the gradient of the line point re-projection error:

$$\begin{aligned} \frac{\partial e}{\partial \delta \boldsymbol{\xi}} = & \\ - \mathbf{n}_i^\top & \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x XY}{Z'^2} & f_x + \frac{f_x X^2}{Z'^2} & -\frac{f_x Y}{Z'} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} & -f_y - \frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X}{Z} \end{bmatrix}. \end{aligned} \quad (3.15)$$

This non-linear optimization problem is solved using Levenberg-Maquardt method [142].

3.4 Experiment Results

In this section, we evaluate our method by both simulation and real-world datasets. Intel Realsense D435 RGBD camera, iRay T3S thermal camera, and Livox Mid-360 LiDAR mounted on a Totem-250 UAV frame are used to generate the datasets, shown in Fig. 3.1. The LiDAR is installed on the top of the UAV platform, while the thermal camera and RGBD camera are assembled inside a 3D printed rig which is mounted in the front of the UAV. The LiDAR, stereo camera, and thermal camera run at 10, 25, and 30 Hz respectively. Some examples of the data captured by different sensors are shown in Fig. 3.6.

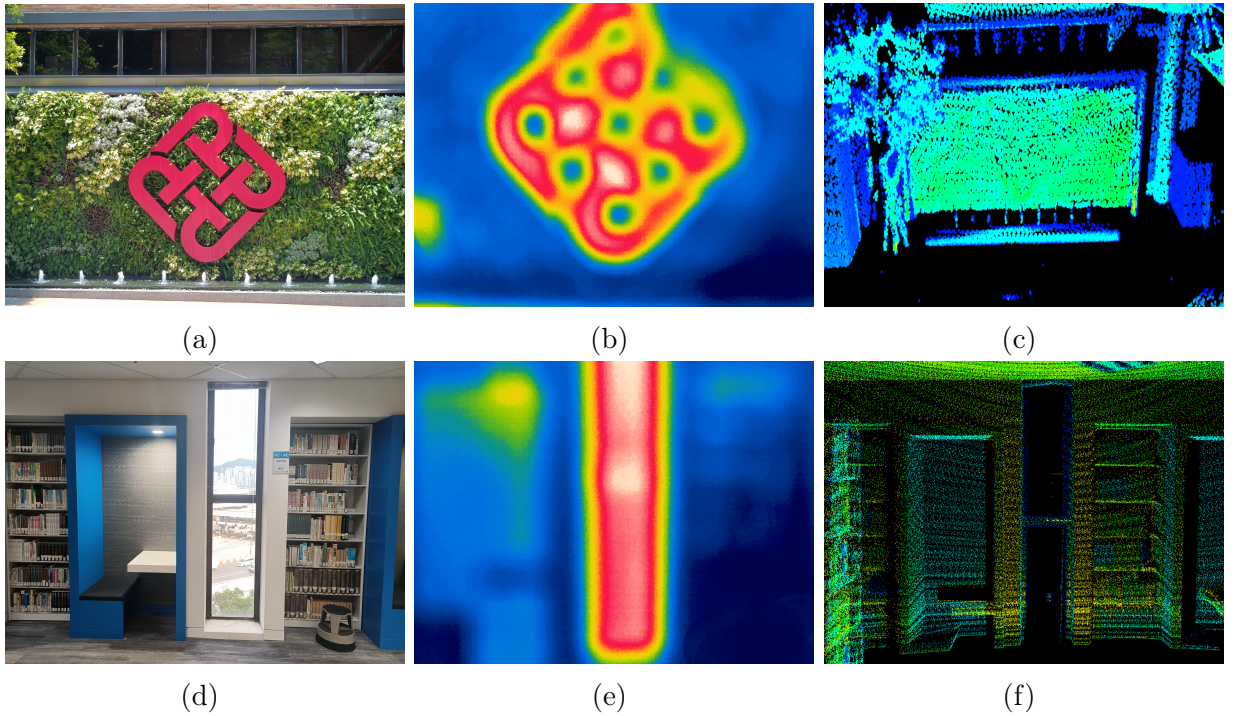


Figure 3.6: Example scenes in different modality. (a, d) RGB image, (b, e) thermal image, (e, f) LiDAR point cloud.

Section 3.4.1 introduces the simulation environment and results, followed by the introduction of real-world experiments in Section 3.4.2. Table 3.1 gives the sensor specifications:

Table 3.1: Sensor Specifications

Device	Modality	Resolution	FOV
iRay T3S	Thermal	384×288	28.2° × 21.3°
Intel Realsense D435	RGB	1920×1080	69.4° × 42.5°
Livox Mid-360	LiDAR	1.3 × 1.3	360° × 59°

3.4.1 Simulation Results

A common problem for quantitatively assessing extrinsic calibration results is the unavailability of the exact ground truth [34]. Due to the lack of ground truth, many earlier efforts used manually annotated results as a benchmark and evaluated the calibration results qualitatively by contrasting the discontinuity in data fusion. Qualitative analysis only provides a general comparison and obscures the specific factors that influence the results. In order to fully describe the effectiveness of the suggested strategy and the impact of various variables, some researchers focused on simulation verification.

Benefiting from recent developments, we built a simulation suite that contains a thermal camera in the Gazebo platform. The simulation suite consists of a thermal camera, a RGBD camera, a LiDAR, and the calibration scene. The sensor properties, including resolution, field of view, and accuracy, are replicated in the simulation suite. As shown in Fig. 3.7, a thermal camera, an Intel Realsense D435 RGBD camera, and a LiDAR are modeled in the simulation suite. RGB images are captured by the left camera of the RGBD camera, and thermal images are captured by the thermal camera. To obtain dense LiDAR point cloud, the LiDAR is slightly moved during the point cloud record process. Afterward, the LiDAR-inertial odometry Fast-lio [143] is used to track LiDAR motion and register all points to the LiDAR’s initial pose. We build a calibration scene using an apartment model as the background and several walls with different temperatures as the foreground to generate edge features in the simulation environment. The point cloud generated by the LiDAR in the simulation is shown in Fig. 3.8.

In the simulation environment, all parameters are under control, so we assume perfect feature detection and matching. Under this assumption, the factors that may influence the calibration results could be categorized into 2 folds, namely 1) the number of frames captured at different relative poses between the sensor rig and the calibration scene, and 2) the noise level added to the data captured by the different sensors. Since the FOV of the thermal camera is relatively small, the thermal camera is placed right under the

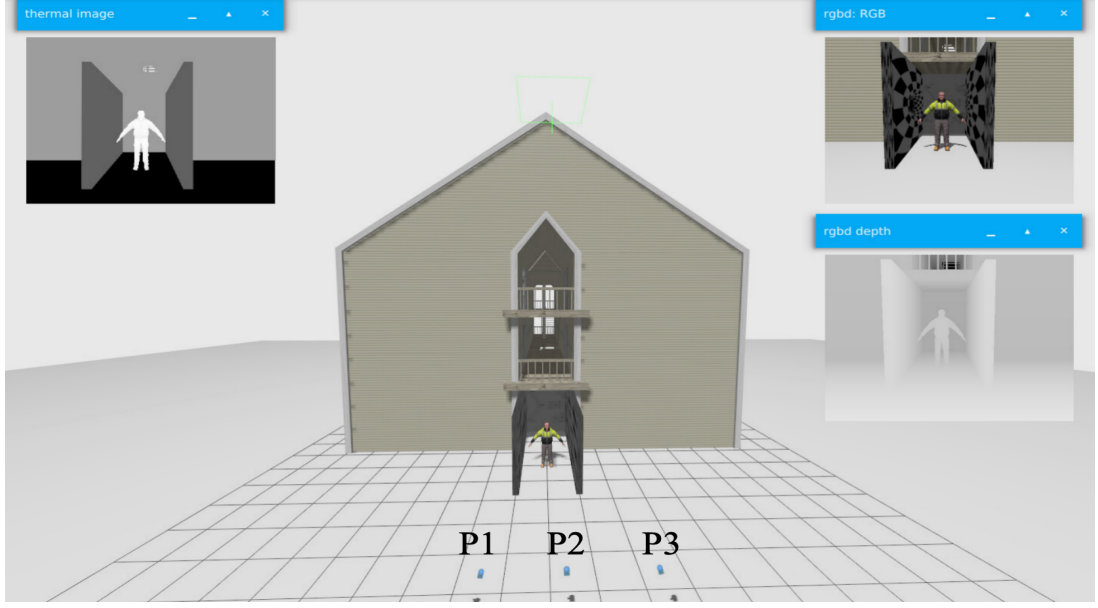


Figure 3.7: Simulation in Gazebo, RGB, depth, thermal image, and LiDAR point clouds are captured. Different temperature is assigned to the ground, different walls, and human model. The house has the same temperature as the gray wall.

RGBD camera and the LiDAR. All sensors heading in the same direction, so that the image and point cloud captured by the three different sensors can be fused in an effective manner.

Single Pose Experiments

In this subsection, we feed only one frame into the pipeline to demonstrate the feasibility of single-frame calibration. To evaluate the impact of acquiring data from different relative positions and angles on the calibration results, we collect data at three distinct points: $P1$, $P2$, and $P3$ shown in Fig. 3.7. Given the ground truth transformation $\mathbf{T}_{gt}$ and the estimated transformation $\hat{\mathbf{T}}$, we compute the transformation error in terms of rotation error e_r and translation error e_t as follows:

$$e_r = \|\text{Log}(\mathbf{R}_{gt}^\top \mathbf{R})\| = \|\text{Log}(\delta \mathbf{R})\| \quad (3.16)$$

$$e_t = \|\mathbf{t}_{gt} - \hat{\mathbf{t}}\| = \|\delta \mathbf{t}\|, \quad (3.17)$$

where $\mathbf{R}$ is the rotation matrix, $\text{Log}()$ operation refers to (3.7), and $\mathbf{t}$ is the translation vector. $\mathbf{R}$ and $\mathbf{t}$ together form the transformation matrix as in (3.4). The single pose calibration results are shown in Table 6.2.

From the table, it can be observed that overall, the algorithm provides more accurate

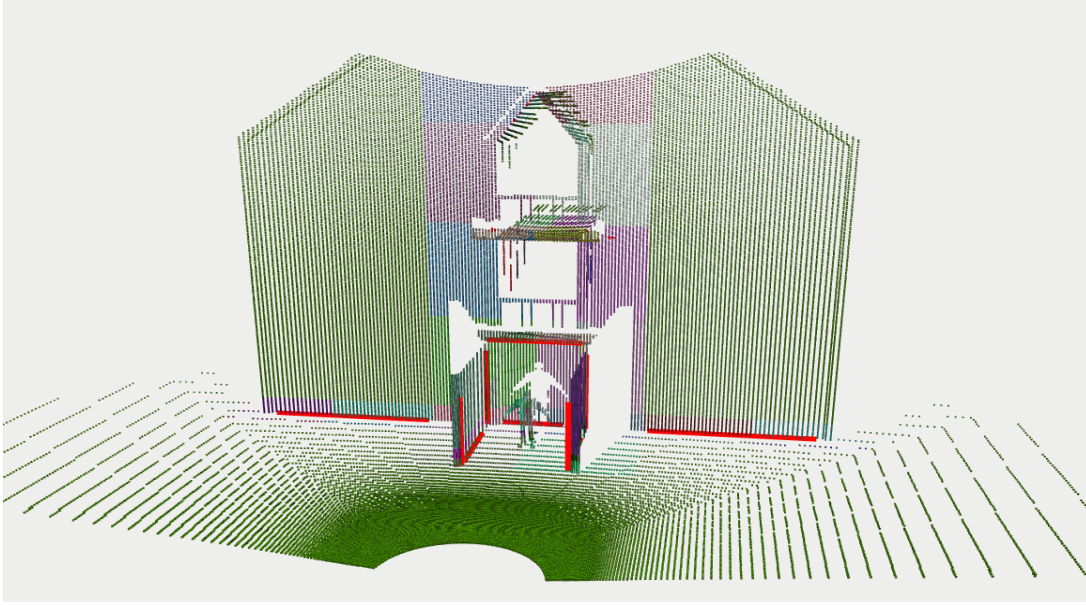


Figure 3.8: LiDAR point cloud and 3D edge extraction in RVIZ.

Table 3.2: Mean (and Std Dev) of Linear(e_t) and Angular(e_r) Calibration Errors in Different Pose

	Error	P1	P2	P3
LiDAR-thermal	e_t (cm)	8.31(1.329)	7.89(1.326)	8.44(1.458)
	e_r (degree)	5.72(0.718)	5.39(1.078)	5.98(0.583)
LiDAR-RGB	e_t (cm)	3.37(1.102)	3.52(1.287)	3.58(1.541)
	e_r (degree)	3.69(1.219)	3.85(1.320)	3.32(1.491)

and robust estimates for rotation compared to translation. Both rotation and translation accuracy vary among different sensor combinations, with the LiDAR-RGBD combination being the more precise and the LiDAR-thermal combination showing the lowest accuracy. We also set identical camera parameters for both thermal and RGB cameras with resolutions of 320×240 and 640×480 in the Gazebo simulation. The FoV of all cameras is set as 60×47 . Table 3.3 presents the experimental results. It is evident from the table that the reprojection error remains consistent across all control groups, as the PnL algorithm minimizes the reprojection error in pixel values. However, due to the difference in camera resolution, with the same reprojection error the translation error is nearly twice as large for small-resolution cameras.

In conclusion, the results obtained from single-frame calibration demonstrate the effectiveness of the algorithm and lay the foundation for multi-frame calibration. Additionally,

the comparison between different resolutions indicates that, with the same reprojection error, the absolute error is influenced by the camera resolution. The higher the resolution, the more precise the absolute transformation accuracy that can be achieved.

Table 3.3: Calibration Accuracy Comparison using Cameras with Different Resolution in Gazebo Simulation

Sensor	Resolution	$e_t(\text{cm})$	$e_r(\text{deg})$	$e_{res}(\text{pix})$
Thermal	320×240	8.42	4.43	1.2
Thermal	640×480	3.78	3.26	1.3
RGB	320×240	8.26	4.39	1.1
RGB	640×480	3.85	3.01	1.2

Multi Pose Experiment

The accuracy of the calibration results in the context of multi-pose experiments can be influenced by various sources of noise inherent to the method, such as sensor noise and feature detection errors. When the sensor rig is posed at different positions and data is collected using diverse approaches, including features with varying depths and non-coplanar lines, additional constraints can be obtained by providing new poses. This process has the potential to decrease ambiguity and enhance accuracy.

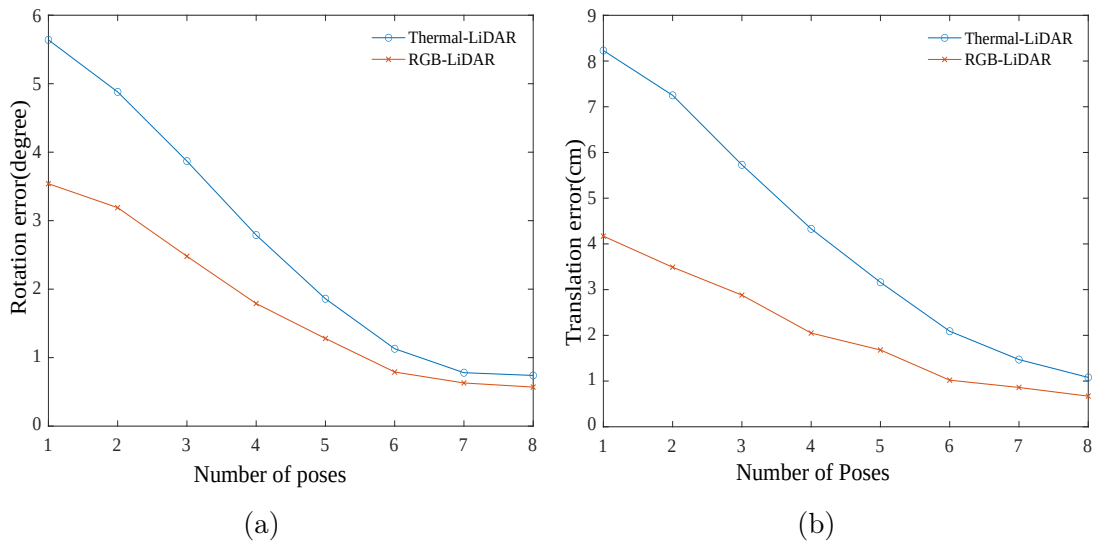


Figure 3.9: Calibration result effected by the number of poses. (a) Rotation error. (b) Translation error.

The rotation and translation error decreased with the increase of the frames used in the calibration process as shown in Fig. 3.9. As can be seen, the rotation error reaches a relatively stable state when the number of input frames is more than 6. While translation estimation is more prone to noise and reaches a high accuracy when the number of input frames exceeds 7.

3.4.2 Real-world Experiment

In this section, our system is evaluated in real-world experiments, both quantitatively and qualitatively. The ground truth extrinsic parameters are obtained by manually selecting corresponding feature points between 10 frames and applying the PnP algorithm based on these correspondences.

We test our method in a variety of scenes, as depicted in Fig. 3.10. Data collection is guided by the analysis in Section 3.3.1, considering the thermal properties and geometric edge distributions to ensure alignment between thermal edges and geometric edges.

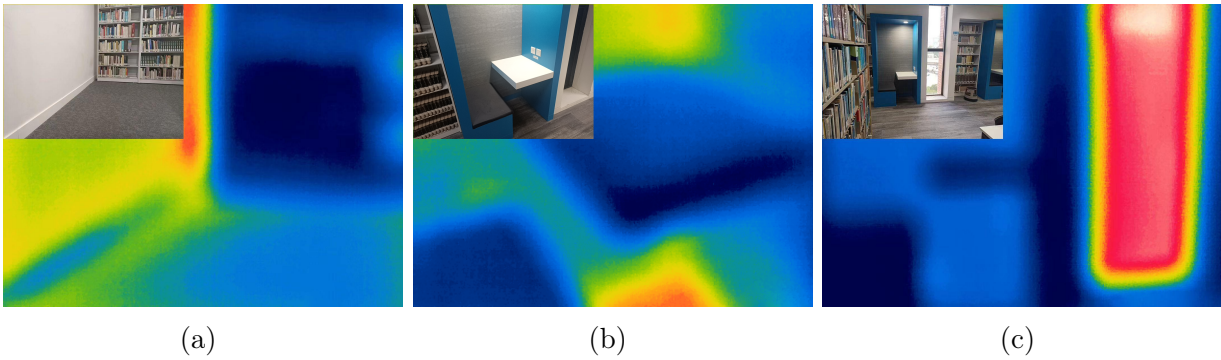


Figure 3.10: Example calibration scenes. (a) Library corner (b) Library seat. (c) Library window.

Convergence Validation

To assess the convergence of the proposed method for estimating extrinsic parameters, experiments are conducted under three distinct scenarios as depicted in Fig. 3.10. A randomized initialization approach is employed to ensure robustness, with initial values sampled within a range of plus or minus 5 degrees in rotation and 5 cm in translation. Each scenario is executed 20 times to yield statistically significant results. The cost function is evaluated for each run, and the statistical analysis of the results is presented in Fig. 3.11. Our findings demonstrate successful convergence of the proposed algorithm

in all scenarios, with final estimates exhibiting minimal cost. These results indicate the reliability and effectiveness of our method in accurately estimating extrinsic parameters across diverse scenes.

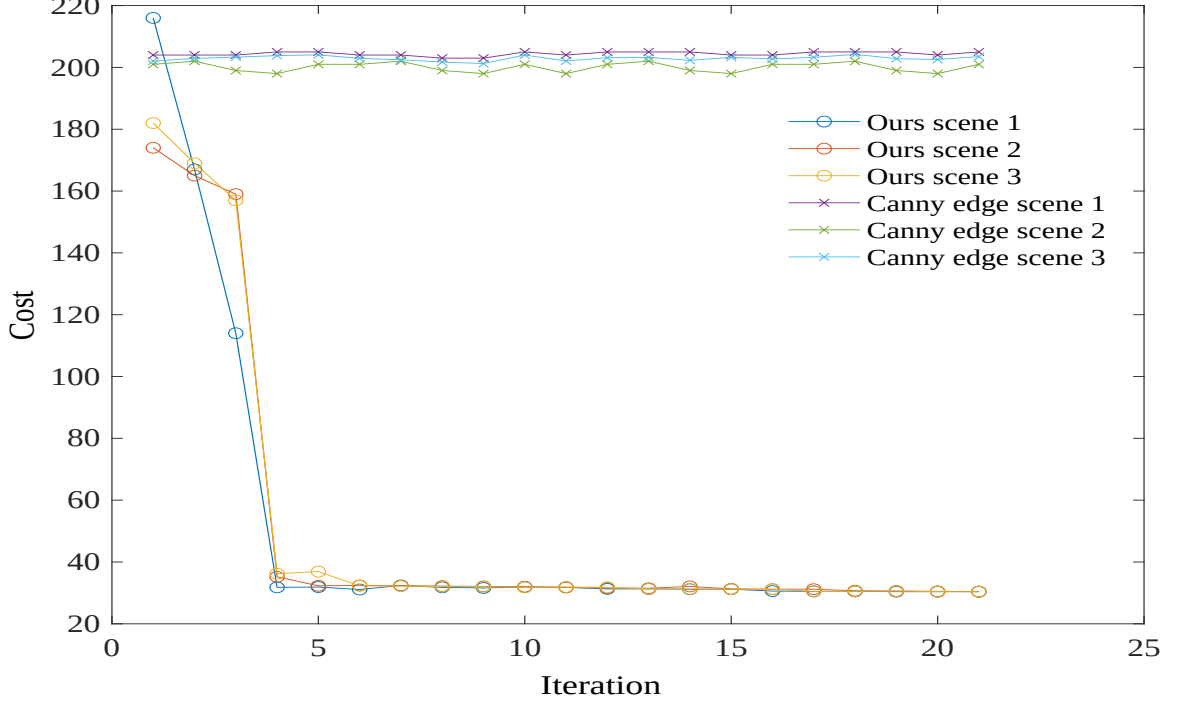


Figure 3.11: Cost comparison using proposed thermal edge detector and Canny edge detector of LiDAR-thermal calibration in different scenes.

Precision and Accuracy

To evaluate the precision of the proposed method, we calculate the calibration error relative to the mean value of extrinsic statistics to test the variance of the proposed method. All results are obtained running batch optimization on 8 frames according to the simulation results that the extrinsic error stability is observed upon surpassing the threshold of 8 frames of data. We compared our method with several others, including an indirect calibration method as a control group, i.e., $T_{ThLiDAR} = T_{ThC} \cdot T_{CLiADR}$, as used by [47]. Although the error in T_{CLiADR} by Fu's [47] method is similar to other methods, the overall error of $T_{ThLiDAR}$ is larger due to the accumulated uncertainty of the indirect process. The boxplot in Fig. 3.12 shows that in the six axes for rotation and translation, our method achieved the smallest variance in five directions out of six axes in LiDAR-RGB calibration, and the smallest variance in four directions out of six axes in LiDAR-thermal calibration. In the yaw direction, our mean error is closer to zero.

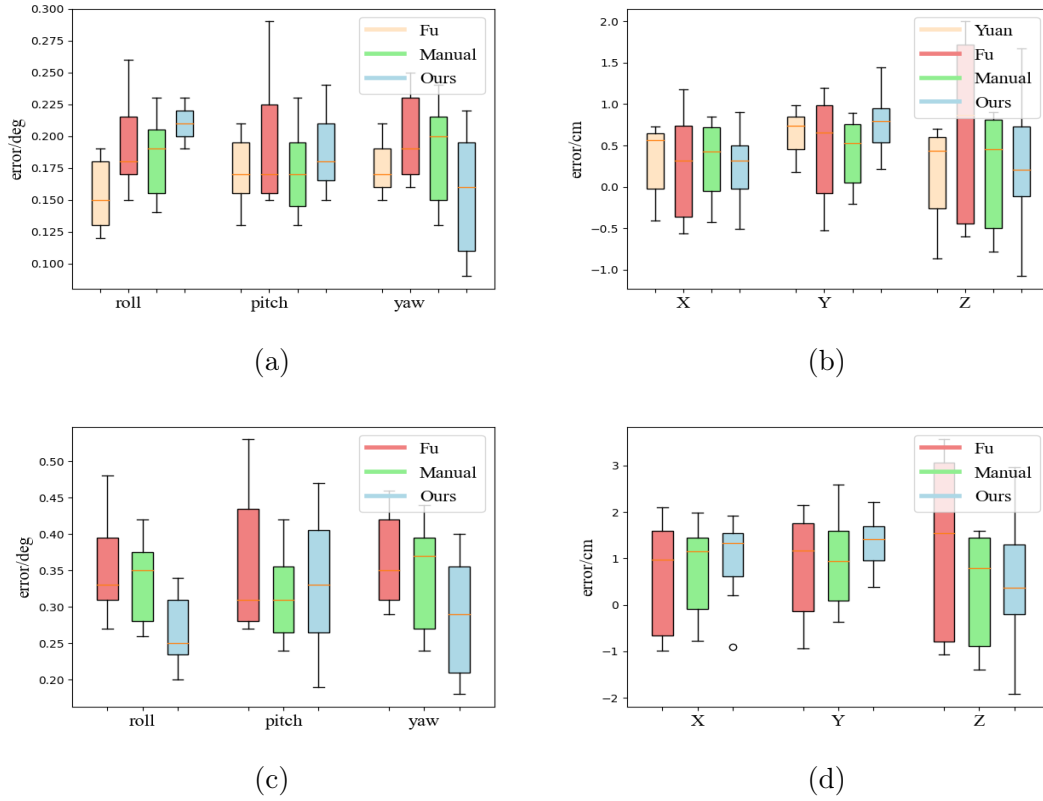


Figure 3.12: Comparison experiment of calibration accuracy. (a) LiDAR-RGB rotation error. (b) LiDAR-RGB translation error. (c) LiDAR-thermal rotation error. (d) LiDAR-thermal translation error.

To further validate the effectiveness of our proposed method, a comparative analysis is conducted with the extrinsic from the CAD model as ground truth. In real experiments, we do not have the absolute ground truth as in simulations. Therefore, we use a CAD model as the ground truth. However, it should be noted that using CAD for extrinsic calibration can introduce large reprojection errors. Table 3.4 presents the comparison results of calibration errors. It is evident that our method achieves superior results in terms of reprojection error among autonomous targetless extrinsic calibration compared to Yuan *et al.* [36] and Fu *et al.* [36] for both LiDAR-RGB and LiDAR-thermal extrinsic calibration. In terms of translation and rotation accuracy, our method achieves results comparable to other methods. Additionally, it performs on par with the manual method. Particularly, our method demonstrates remarkable accuracy in both LiDAR-RGB calibration and LiDAR-thermal calibration, with overall translation errors of 2.77 cm and 3.86 cm, and overall rotation errors of 0.21 degrees and 0.46 degrees, respectively. Moreover, our method enables the direct and autonomous targetless calibration of both the thermal camera and LiDAR. Previously, no existing method offered the capability to directly

calibrate the combination of LiDAR and thermal sensors in such a manner.

Table 3.4: Extrinsic Calibration Comparison

Method	LiDAR-RGB			LiDAR-thermal		
	$e_t(\text{cm})$	$e_r(\text{deg})$	$e_{res}(\text{pix})$	e_t	e_r	e_{res}
Yuan’s [36]	2.75	0.17	1.3	-	-	-
Fu’s [47]	2.81	0.23	1.9	4.39	0.48	2.3
Ours	2.77	0.21	1.2	3.86	0.46	1.3
Manual	2.30	0.19	1.3	4.20	0.37	1.8

Outlier Rejection

Our original calibration algorithm already possesses basic outlier rejection functionality. By iteratively shrinking the edge matching threshold (gradually reducing from 30 pixels to 5 pixels), it effectively enhances robustness against noise. Additionally, the current algorithm still relies on structured environments to achieve stable calibration, resulting in a low number of outliers in the experimental data. For further study of the impact of the wrong matching in the calibration process, RANSAC and different robust kernels were implemented in the system. By comparing the calibration performance under different combinations of RANSAC activation states and kernel functions, we clarify their core values in LiDAR-camera extrinsic parameter calibration: kernel functions serve as the key to improving calibration accuracy, suppressing outlier interference through robust loss functions; RANSAC acts as a supplementary means for filtering explicit outliers, yet exhibits a significant speed bottleneck in low-outlier scenarios.

The computation time is intuitively presented in Fig. 3.13, where the introduction of RANSAC substantially increases the computational overhead — the total time consumption of RANSAC groups (e.g., 918.0 s for RANSAC+Tukey) is much higher than that of NoRANSAC groups (e.g., only 401.5 s for NoRANSAC+Quadric). Fig. 3.14 shows the comparison of extrinsic calibration parameters across different methods with and without RANSAC integration (combined with null, Huber, and Tukey robust loss functions). (a) Rotation angles (Roll, Pitch, Yaw) in degrees following the Z-Y-X Euler angle order, where the Yaw parameter is marked with diamond symbols for clear identification; all methods exhibit highly consistent rotation results despite the loss reduction by RANSAC. (b) Translation components (X, Y, Z) in meters, showing minimal discrep-

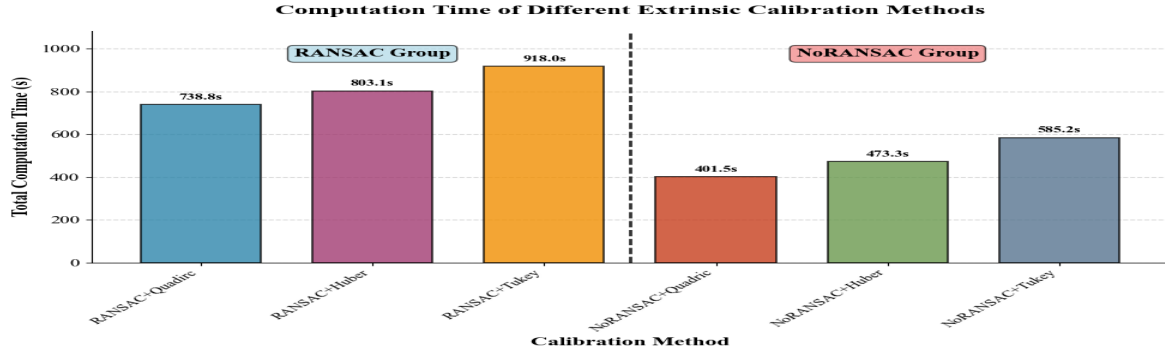


Figure 3.13: Computational Time comparison of different calibration settings

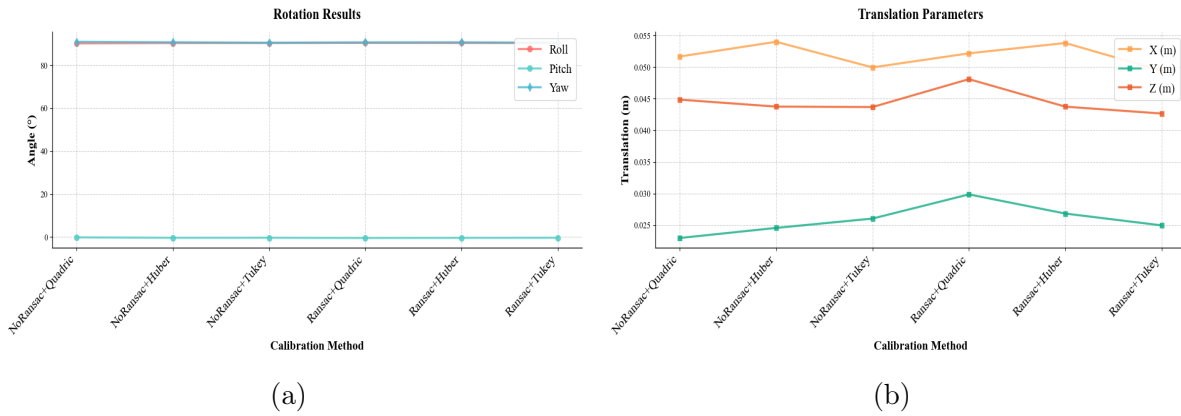


Figure 3.14: Extrinsic calibration parameters of methods with or without RANSAC (and null Huber Tukey robust loss functions). (a) Euler angles (Roll Pitch Yaw deg) with Yaw marked by diamonds; (b) Translation components (X Y Z, m) with minimal discrepancies.

ancies among all calibration schemes, which verifies that RANSAC only enhances outlier rejection without compromising the consistency of final extrinsic parameter outputs.

In practical applications, we recommend prioritizing the "NoRansac + Huber kernel function" scheme to balance accuracy and efficiency; if extreme accuracy is required, the computational cost can be reduced by optimizing RANSAC parameters. Experimental results also demonstrate that in low-outlier scenarios, the residual suppression effect of kernel functions is sufficient to meet engineering requirements, while the gain from RANSAC is limited. Excessively pursuing an "outlier-free" solution will instead lead to a waste of efficiency. The core value of RANSAC lies in screening valid inliers in high-outlier scenarios (outlier ratio $> 10\%$). However, the outlier ratio in this experiment is only 1–5%. The "random sampling–model evaluation–inlier counting" loop of RANSAC generates a large amount of redundant computation, resulting in benefits far lower than the costs.



Figure 3.15: Image fusion result of RGB image and LiDAR point cloud. (a) Raw RGB image, the FoV of the thermal camera is shown in the red box. (b) RGB image with LiDAR point cloud projection.

Fusion Results

Fig. 3.15 shows the fusion result of the LiDAR point cloud and the RGB image in the stereo camera. It can be seen from the fusion result, that the LiDAR point cloud edges are in good alignment with the RGB image edges. LiDAR points near the LiDAR center are represented by red color and points far from the LiDAR are represented by blue color. The trend of color changes in the image is consistent with reality.

Fig. 3.16 shows a qualitative fusion result of the LiDAR point cloud and the thermal image at the same scene as depicted in Fig. 3.15, using the parameters obtained by our calibration method. Since the FOV of the thermal camera is much smaller than the RGBD camera and LiDAR, the thermal information only covers a part of the other two sensors. Still, it demonstrates the feasibility of the proposed method.

3.5 Conclusion

In this work, we have presented an autonomous-targetless calibration method for LiDAR-RGB and LiDAR-thermal calibration, representing the first autonomous-targetless approach capable of directly calibrating a thermal camera and a LiDAR, as supported by the literature review. The method extracts line features from 2D thermal images and 3D point clouds and solves the extrinsic parameters based on the line feature correspondences. We have developed a simulation calibration suite on the Gazebo platform, incorporating a

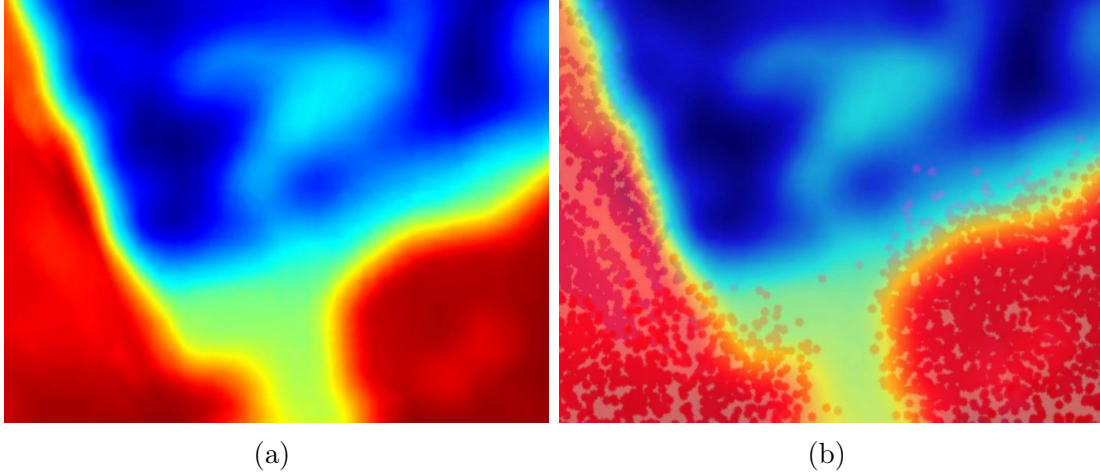


Figure 3.16: Image fusion result of thermal image and LiDAR point cloud. The FoV of the thermal camera is shown in the red box in Fig. 3.15. (a) Thermal image. (b) Thermal image with LiDAR point cloud projection.

thermal camera, an RGB-D camera, a LiDAR and buildings with different temperatures. The performance of the proposed method has been rigorously assessed through systematic estimation using synthetic data and qualitative validation using real-world data. In both LiDAR-RGB calibration and LiDAR-thermal calibration, the method has demonstrated exceptional accuracy, with overall translation errors of 2.77cm and 3.86cm, and overall rotation errors of 0.21 degrees and 0.46 degrees, respectively. These results are comparable to those achieved by traditional target-based manual methods. Different thermal scenes have been analyzed based on the thermography process. The presence of distinguishable and well-aligned edges in different modalities (thermal-RGB-LiDAR) has been identified as the key to obtaining ideal calibration results.

As part of future work, we plan to explore autonomous bad scene rejection mechanisms and enhance the calibration accuracy through the utilization of a more precise thermal edge detection method. Such functionality was previously challenging to implement; however, with the rapid advancement of AI, especially Visual Language Model (VLM) have demonstrated comparable capabilities. A key limitation, nevertheless, is that VLM currently lacks the feasibility for onboard real-time deployment, which could be investigated as a promising research direction in future studies.

Chapter 4

Feedforward-feedback LiDAR Visual Inertial System

4.1 Research Background

SLAM [7] is crucial for robot operation in unknown environments, such as UAVs search and rescue, and building inspection tasks. Common SLAM systems utilize LiDAR, and visual sensors for exteroceptive perception and inertial sensors for interoceptive state estimation. Previous research frameworks primarily relied on a single external perception sensor (like a camera or LiDAR) combined with an IMU. For UAVs, cost-effective configurations often include VIO, SVIO, and LIO.

As the demand for operating intelligent robots in the real world increases, these robots often need to work in challenging environments. Above mentioned systems that rely on a single exteroceptive sensor often cannot provide the accuracy and robustness needed for pose estimation in these challenging conditions. For example, VIO is prone to failure in outdoor environments where the lighting conditions may change rapidly, and the performance of LIO is affected by the environment geometry. To tackle this, several recent works proposed LVIO that fuses the camera and LiDAR measurement. Existing systems usually focus on the fusion camera measurements and LiDAR measurements. Similar to the fusion of a single exteroceptive sensor with an IMU, Kalman filter or factor graph optimization approaches are usually applied to fuse camera and LiDAR measurement.

Contemporary filter-based VIO or LIO systems predominantly employ either a filter-based or an optimization-based framework for the fusion of different data. Both frameworks hinge on the identification of the most suitable prediction that fulfills each individual measurement, such as those derived from visual cameras or an Inertial Measurement

Unit. As the input data escalates, there is a corresponding increase in the demand for computational power. This poses a significant challenge for robotic systems, especially those with constraints on size, weight, and power, such as Unmanned Aerial Vehicles. High-performance pose estimators, in these instances, can consume an excessive amount of computational resources.

Building on our previous work [8], we propose a feedforward-feedback based framework. Unlike other frameworks where gyroscope data is used to predict rotation and accelerometer measurements are used to calculate pose through integration, our framework leverages the complementary nature of gyroscope and accelerometer measurements. Gyroscope data is accurate in the short term but drifts over time (low-frequency noise), while accelerometer data is noisy in the short term but accurate over time (high-frequency noise). We use measurements from both the gyroscope and accelerometer to jointly predict the robot's rotation. We introduce a feedforward-feedback-based complementary filter framework, integrating IMU data and visual LiDAR into the pose estimation pipeline. This structure combines the advantages of the Kalman filter method and includes several decoupled models to partition the process of LiDAR stereo vision inertial fusion. The complementary filter is used to process IMU data, and the feedforward and feedbacks are used to combine measurements from different sensors. With a better initial value, the update step of sensor fusion can converge more quickly and achieve more accurate pose estimation. This approach enhances the precision and robustness of pose estimation, which is crucial for the operation of robots in complex environments.

In summary, the aim of this work is to tackle the issue of fast and robust LiDAR-visual-inertial pose estimation, especially for systems that are restricted by resource limitations. Fig. 4.1 shows the system architecture of FLiVIS, the whole system consists of a VIO subsystem and a LIO subsystem. The VIO subsystem is based on FLVIS [8] and the LIO subsystem is based on Fast-lio2 [9]. The proposed system is evaluated on the KITTI dataset [10] and NTU VIRAL dataset [11] to test its applicability in autonomous driving and UAV inspection tasks. The results demonstrate that FLiVIS is comparable to state-of-the-art systems, and it can achieve real-time processing on various platforms, from laptops to onboard processors.

In this chapter, we present a LVIO based on feedforward and feedbacks. Compared to traditional Kalman filter-based methods or optimization-based methods for sensor fusion, the proposed system achieves sensor fusion through feedforward and feedbacks. This system, named Feedforward-feedback LiDAR Visual Inertial System (FLiVIS) consists of a VIO subsystem and a LIO subsystem, these two subsystems are coupled through complementary filters.

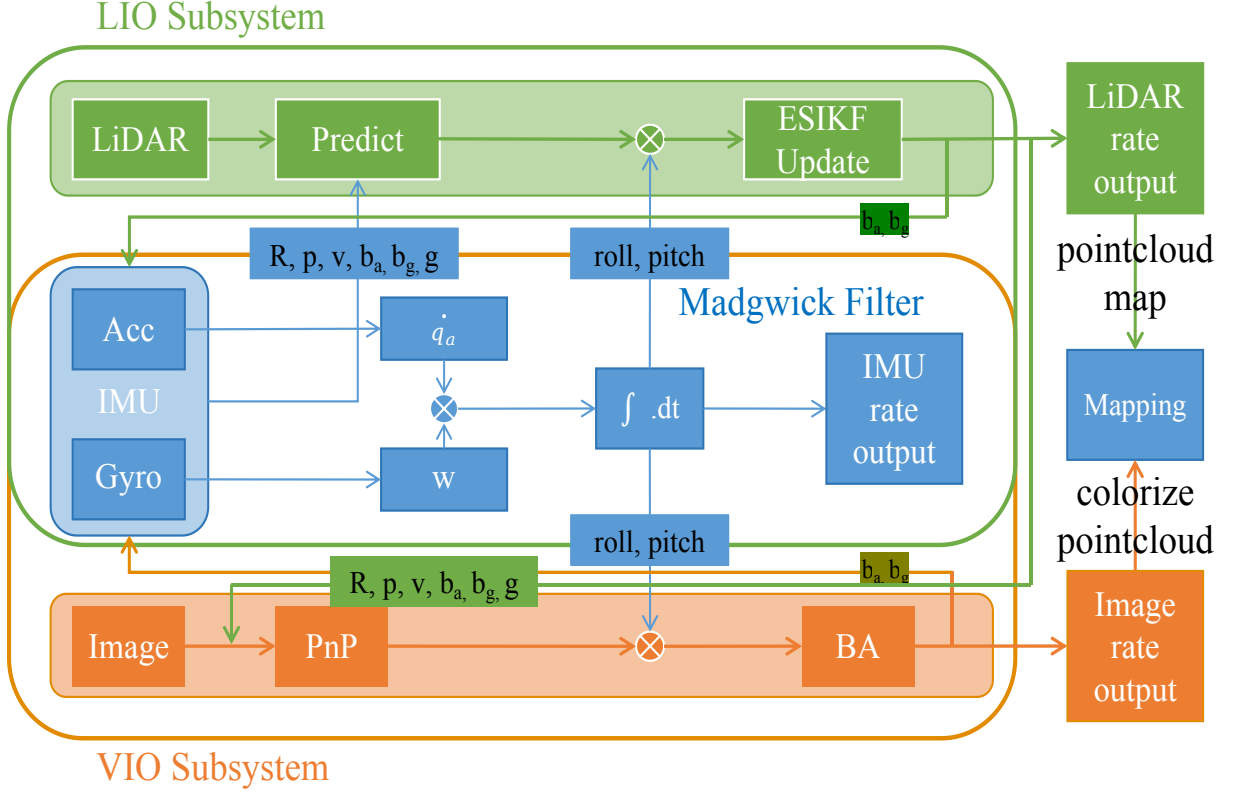


Figure 4.1: System architecture of the proposed pipeline. The inputs are the measurements from the LiDAR, IMU, and camera. The IMU data is processed by the Madgwick filter, and the fusion of IMU estimation and the other two sensors is achieved through feedforward and feedback. The LIO states are maintained by an ESIKF, and VIO mainly depends on feedforward and feedback loops as in FLVIS.

Instead of directly integrating gyroscope data and accelerometer data, our framework leverages the complementary nature of gyroscope and accelerometer measurements. FLiVIS is evaluated on public datasets, it achieves a relative translation error of 0.68% on the KITTI dataset and 0.138 *m* absolute translation error on the NTU-Viral dataset, respectively. The experiment results demonstrate the accuracy and robustness of FLiVIS with respect to other state-of-the-art counterparts. FLiVIS is capable of accommodating both multi-line spinning LiDARs and emerging solid-state LiDARs, which employ distinct scanning patterns. Additionally, it can perform real-time operations on a range of platforms, from laptops to onboard processors.

In summary, the aim of this chapter is to tackle the issue of fast and robust LiDAR-visual-inertial pose estimation, especially for systems that are restricted by resource limitations. Fig. 4.1 shows the system architecture of FLiVIS, the whole system consists of a VIO subsystem and a LIO subsystem. The VIO subsystem is based on FLVIS [8] and the LIO subsystem is based on Fast-lio2 [9]. The proposed system is evaluated on the

KITTI dataset [10] and NTU VIRAL dataset [11] to test its applicability in autonomous driving and UAV inspection tasks. The results demonstrate that FLiVIS is comparable to state-of-the-art systems, and it can achieve real-time processing on various platforms, from laptops to upboard processors.

4.2 Methodology

In Section 4.2.1, we first compare the similarities and differences between the complementary filter, Kalman filter, and optimization-based methods. We believe the main difference lies in the perspective from which sensor fusion is viewed: the complementary filter approaches it from the frequency domain and signal perspective, while the Kalman filter and optimization-based methods model the sensor fusion problem from the perspective of maximum a posterior probability. In Section 4.2.2, we use the one-dimensional complementary filter as an example to derive the different forms of signal fusion from the frequency domain to the time domain. In Section 4.2.3, we demonstrate the Madgwick filter in three-dimensional space based on IMU data and clarify its role in FLiVIS. Finally, we explain our visual and LiDAR measurement model in section 4.2.4.

4.2.1 Complementary Filter

In conventionally Kalman filter-based or optimization-based methods, only gyroscope measurement is used for orientation estimation. The accelerometer data is ignored in the orientation update step. Note that, in the presence of gravity, the accelerometer measurement can be used for orientation estimation under the following two assumptions:

1. During the moving process of the unmanned vehicle, the forces are essentially balanced, i.e., the vehicle is moving at uniform speed or hovering at a fixed point. That is, the accelerometer measurement in the global coordinate is $(0, 0, g)^\top$.
2. The sensor measurement data only involves high-frequency or low-frequency noise, that is, the sensor measurement equation is as follows:

$$\begin{aligned}\hat{a}(t) &= a(t) + n_H(t) \\ \hat{w}(t) &= w(t) + n_L(t),\end{aligned}\tag{4.1}$$

here, the variable with a caret (^) represents the measured value, while the variable without a caret represents the ground truth value. Note that the two assumptions hold for most autonomous vehicle operation periods, we fuse $\hat{a}$ into the orientation estimation pipeline.

Although subject to noise, orientation estimates from accelerometer measurements exhibit sustained accuracy over long periods. Conversely, orientation estimates derived from gyroscope measurements offer precision over short periods but are susceptible to drift over long durations. These distinct characteristics can be exploited and harnessed within the frequency domain [144]. Specifically, orientation estimates predicated on gyroscope measurements demonstrate advantageous properties at high frequencies, thus the gyroscope measurement is filtered by a High Pass Filter (HPF). In contrast, orientation estimates obtained from accelerometer measurements exhibit favorable attributes at low frequencies, thereby the accelerometer measurement is filtered by a Low Pass Filter (LPF). This approach facilitates the effective utilization of the unique properties inherent to each sensor's measurements provided by the IMU, thereby optimizing system performance. The diagram of a complementary filter on IMU measurement is shown in Fig. 4.2. The mathematical derivation of the one-dimensional complementary filter and Madgwick filter are detailed in 4.2.2 and 4.2.3 respectively.

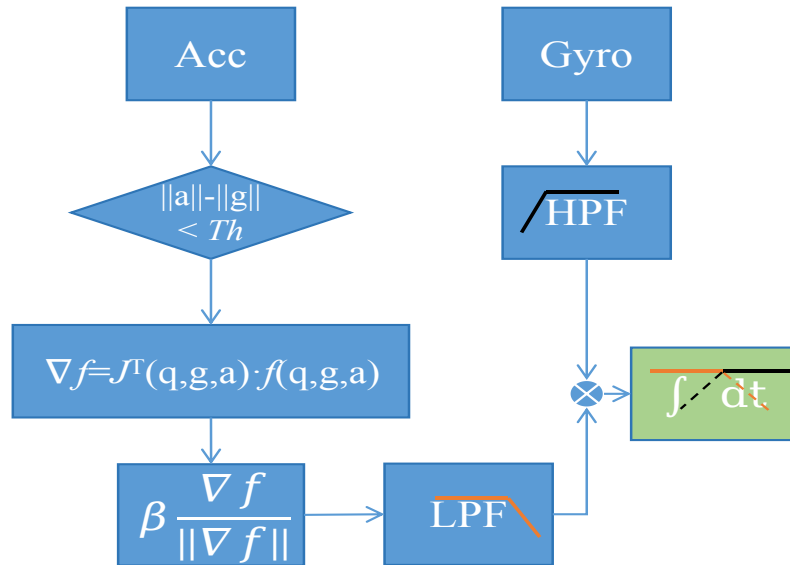


Figure 4.2: Madgwick filter forms the perspective of the complementary filter.

4.2.2 One Dimensional Complementary Filter from Frequency Domain to Time Domain

Consider the one-dimensional case, assuming the IMU rotates along one axis (taking the X-axis as an example here). We denote the estimation of an angle θ from gyroscope measurement by $\hat{\theta}_w$ and the estimation from accelerometer measurement by $\hat{\theta}_a$. $\hat{\theta}_w$ is obtained by integrating gyroscope measurement w_{gyro} and $\hat{\theta}_a$ is obtained by calculating the angle between gravity and the acceleration direction. The Laplace transforms of θ , $\hat{\theta}_a$, $\hat{\theta}_w$, and $\hat{w}$ are denoted by $\Theta(s)$, $\hat{\Theta}_a(s)$, $\hat{\Theta}_w(s)$, and $\hat{W}(s)$, respectively. The complementary filter computes $\Theta(s)$ as:

$$\begin{aligned}\Theta(s) &= LPF(s)\hat{\Theta}_a(s) + HPF(s)\hat{\Theta}_w(s) \\ &= LPF(s)\hat{\Theta}_a(s) + (1 - LPF(s))\frac{1}{s}\hat{W}(s).\end{aligned}\quad (4.2)$$

Where $LPF(s)$ is a low-pass filter and $HPF(s) = 1 - LPF(s)$ is hence a high-pass filter. The insight behind the name complementary filter of (4.2) is that it leverages the complementarity nature of the measurement provided by the gyroscope and the accelerometer.

Taking $LPF(s) = \frac{C(s)}{s+C(s)}$ as a first-order low-pass filter, the output of the complementary filter can be written as:

$$\Theta(s) = \frac{C(s)}{s+C(s)}\hat{\Theta}_a(s) + \frac{s}{s+C(s)}\frac{1}{s}\hat{W}(s).\quad (4.3)$$

Set $C(s)$ in the low-pass filter as $\frac{1}{a}$, (4.3) becomes:

$$\Theta(s) = \frac{1}{as+1}\hat{\Theta}_a(s) + \frac{1}{s}\frac{as}{as+1}\hat{W}(s).\quad (4.4)$$

Substituting $s = \frac{1-z^{-1}}{T}$ into (4.4), we have the Z-transform of the complementary filter:

$$\Theta(z) = \frac{1}{a\frac{1-z^{-1}}{T}+1}\hat{\theta}_a(z) + \frac{a}{a\frac{1-z^{-1}}{T}+1}\hat{W}(z).\quad (4.5)$$

Convert the Z-transform (4.5) to the discrete-time difference form, we have:

$$\begin{aligned}\theta(t) &= \frac{T}{a+T}\hat{\theta}_a(t) + \frac{a}{a+T}(\theta(t-1) + Tw(t)) \\ &= (1-\gamma)\hat{\theta}_a(t) + \gamma(\theta(t-1) + Tw(t)),\end{aligned}\quad (4.6)$$

where γ is $\frac{a}{a+T}$. (4) and (8) are equivalent, except that they represent signal fusion from the perspectives of the frequency domain and time domain, respectively. In the frequency domain, the signal is filtered using a high-pass or low-pass filter to remove noise. In the time domain, it is represented as a weighted complementary filter. In (8), the parameter $\gamma = \frac{a}{a+T}$, represents the weight assigned to the accelerometer, which corresponds to the low-frequency signal. The choice of this weight is related to the cutoff frequency, denoted as a of the low-pass filter. This allows us to derive the complementary filter from the frequency domain to the time domain for one dimension. Having completed the derivation of the complementary filter for one dimension, we now move on to deriving the 3D complementary filter using IMU data as outlined in Section 4.2.3.

4.2.3 Complementary Filter on Quaternions

Having the one-dimensional complementary filter, we can extend it to three-dimensional rotation. In order to achieve a high-precision three-dimensional attitude estimation, we introduce the Madgwick filter here. Madgwick filter is primarily used for attitude estimation, utilizes data from the accelerometer, and applies it to the orientation propagation. It becomes particularly effective when the visual-inertial sensor is nearing a uniform state and demonstrating steady motion as the assumption in 4.2.1. In scenarios where there is an absence of external acceleration (i.e., the sensor is either in a state of uniform motion or remains steady), the gravitational field is expected to align with the acceleration measurement field. This alignment is a critical aspect of maintaining the accuracy of the attitude estimation, and can be described by:

$$a = q_{iw}^{\oplus \top} q_{iw}^+ g = R_{iw}(q)g. \quad (4.7)$$

The R_{iw} can be explicitly rewrite as:

$$\begin{bmatrix} 1 - 2q_y^2 - 2q_z^2 & 2q_xq_y - 2q_zq_w & 2q_xq_z - 2q_yq_w \\ 2q_xq_y + 2q_zq_w & 1 - 2q_x^2 - 2q_z^2 & 2q_yq_z + 2q_xq_w \\ 2q_xq_z - 2q_yq_w & 2q_yq_z + 2q_xq_w & 1 - 2q_x^2 - 2q_y^2 \end{bmatrix} \quad (4.8)$$

By the alignment of the orientations of the gravitational field and the acceleration measurement field, it becomes possible to feed back the acceleration to the orientation estimation. Specifically, when the norm of the incoming acceleration approximates the magnitude of gravity $\|a\| - \|g\| < \pi_{sh}$, the orientation estimation derived from the ac-

celerometer can be calculated by minimizing the error term by solving (4.9) and (4.10).

$$\arg \min_{q_a} f(q_a, g, a). \quad (4.9)$$

$$f(q_a, g, a) = R_{iw}(q_a)g - a. \quad (4.10)$$

Where q_a denotes the orientation solved by accelerometer measurement. Assuming that the direction of gravity is aligned with the vertical direction, or z-axis during the UAV flight task. By substituting $g = (0, 0, 1)$ and the normalized accelerometer measurement $\hat{a}$, we can obtain a simplified version of the objective function and the Jacobian, as defined by (4.11) and (4.2.3).

$$f(q_a, g, a) = \begin{bmatrix} 2q_x q_z - 2q_y q_w - a_x \\ 2q_y q_z + 2q_x q_w - a_y \\ 1 - 2q_x^2 - 2q_y^2 - a_z \end{bmatrix}, \quad (4.11)$$

and the Jacobian of $f(q_a, g, a)$ is defined by:

$$J_{q_a}(q_a, g, a) = \begin{bmatrix} -2q_y & 2q_z & -2q_w & 2q_x \\ 2q_x & 2q_w & 2q_z & 2q_y \\ 0 & -4q_x & -4q_y & 0. \end{bmatrix} \quad (4.12)$$

This optimization problem can be solved by the gradient descent algorithm:

$$q_a(n+1) = q_a(n) - \mu \frac{\nabla f(q_a(n), g, a)}{\|\nabla f(q_a(n), g, a)\|}, n = 1, 2, \dots, n, \quad (4.13)$$

where $\nabla f(q_a(n), g, a) = J_q^\top(q_a(n), g, a)f(q_a(n), g, a)$. An appropriate value of μ will ensure the convergence rate of q is limited to the physical orientation rate as this avoids overshooting due to an unnecessarily large step size. The range of a reasonable μ is given by [145] as in (4.14).

$$\mu = \alpha \|\dot{q}_a\| \Delta t, \quad \alpha > 1 \quad (4.14)$$

Thus the orientation by the accelerometer is solved and can be further fused with the orientation estimate by the gyroscope measurement. Initially, the fuse orientation is given by (4.15) as in (4.2) and (4.6) :

$$q(t) = \gamma q_a + (1 - \gamma)q_w, \quad 0 \leq \gamma \leq 1. \quad (4.15)$$

Following [145], (4.15) is converted to (4.16)

$$q(n) = q(n-1) + \dot{q}\Delta t, \quad (4.16)$$

where $\dot{q}$ is given by (4.17):

$$\dot{q} = \frac{1}{2}q_w \otimes w - \beta \frac{\nabla f(q_a(n), g, a)}{\|\nabla f(q_a(n), g, a)\|}. \quad (4.17)$$

The orientation given by (4.16) is fed into the pipeline as shown in Fig. 4.1 to offer a better orientation initialization for the following steps.

4.2.4 Visual and LiDAR measurement model

FLiVIS is composed of VIO and LIO subsystems. The VIO subsystem builds upon our prior work, FLVIS [8], while the LiDAR measurement model in LIO can be substituted with any LiDAR odometry. In this study, we employ measurement models based on Fast-Lio2 [9]. This modular approach allows our LVIO to adapt to a wide range of environments. It's worth noting that the same LiDAR point registration method may yield significantly different results in varying environments. This study emphasizes the use of a complementary filter for IMU data processing and a framework based on feedforward and feedbacks for sensor fusion. As a result, we decouple the visual and LiDAR measurement models from the feedforward-feedback based LVIO system. The following sections will provide a detailed introduction to both the visual and LiDAR measurement models.

VIO Subsystem

The visual measurement model, shown in Fig. 4.3, is based on the extraction of visual feature points and the minimization of re-projection error. The 3D points are obtained through triangulation. The 3D points are then reprojected onto the image plane to compute the re-projection error with their 2D matches. Denote a 3D point $P_i = [X_i, Y_i, Z_i]^\top$ and its corresponding 2D point $p_i = [u_i, v_i]^\top$. Take the camera intrinsic matrix K_c , we can solve the camera pose by minimizing the re-projection error as follows:

$$T_c = \arg \min_{T_c} \frac{1}{2} \sum_{i=1}^N \|p_i - K T P_i\|^2. \quad (4.18)$$

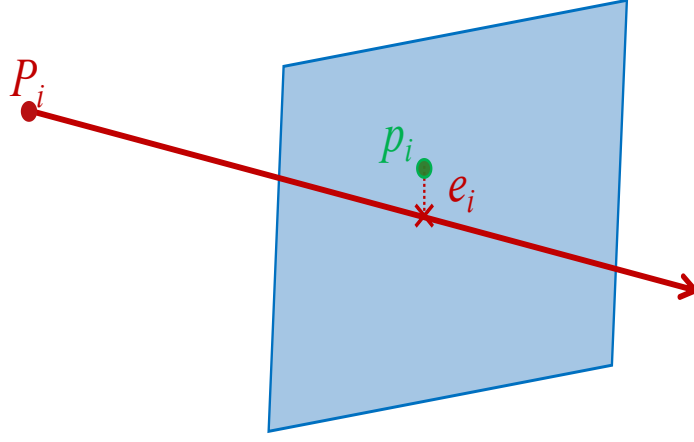


Figure 4.3: Visual measurement model, the error is the 2D distance between the 3D waypoint P_i and the 2D image point p_i on the image plane.

LIO Subsystem

In the LIO subsystem of FLiVIS, the LiDAR measurement model is based on the point-to-plane measurement model of the ikd-tree [9]. The points collected by the LiDAR are first back-propagated through the IMU data and timestamps to obtain undistorted point clouds. As shown in Fig. 4.4, the nearest point is obtained through the K-nearest-neighbor search, Q_j is a point in the K nearest neighbor of the point P_i , and the norm n_i is calculated. Finally, the pose of the LiDAR is obtained by minimizing the following cost function:

$$T_{LiDAR} = \arg \min_{T_{LiDAR}} \frac{1}{2} \sum_{i=1}^N \|n_i(T_{LiDAR}P_i - Q_j)\|^2. \quad (4.19)$$

LIO and VIO Fusion

The entire system relies primarily on LIO (LiDAR-Inertial Odometry), with the state variables maintained by the ESIKF (Error-State Iterated Kalman Filter). During the predict stage, the roll and pitch are estimated using IMU data processed by the Madgwick Filter [8], given that the IMU frame rate is significantly higher than that of the LiDAR. The visual module is mainly utilized for coloring the point cloud during the mapping process. As illustrated in Fig. 4.1, we take the output of LIO as the reference and transmit it to the VIO (Visual-Inertial Odometry), enabling the VIO to perform only frame-to-frame updates. We have also incorporated content on thermal mapping. Fig. 4.5 (a) and (b) display the RGB image and thermal image of the same scene from the VIVID Dataset. It can be observed that the RGB image is relatively clear, whereas the thermal

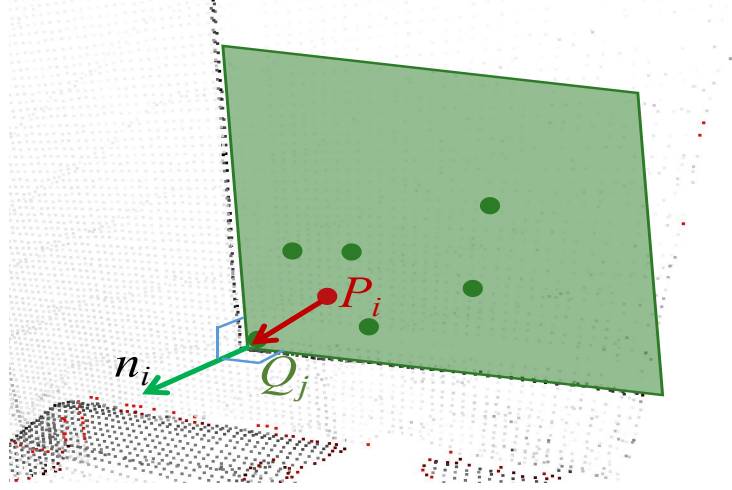


Figure 4.4: LiDAR measurement model, the vector between point P_i (in red) in the current scan and its neighbor point Q_j (in green) is orthogonal to the normal vector of the k -nearest neighbors of P_i .

image contains very few distinct features. In scenarios with scarce image features, the VIO operates in a degenerate state; thus, we rely more heavily on the LIO submodule, with the images solely used for point cloud colorization. This approach ensures that colorized point clouds can still be obtained even under visually degenerate conditions, such as in thermal-LiDAR-inertial mapping. The results of the colorized point clouds are presented in Fig. 4.5.

4.3 Experiments

In this section, we evaluate the proposed system using the public dataset KITTI [10] and NTU-Viral [11] to simulate the running environments of both ground and aerial vehicles. The VIO, LIO subsystems, and the LVIO system are tested. An ablation study is also conducted to demonstrate the effects of the Madgwick filter introduced in 4.2.3.

4.3.1 KITTI Dataset

In this section, we compare the performance of FLiVIS with three state-of-the-art algorithms, namely FLVIS [8], LOAM [63], and line-based LVO [146], utilizing the KITTI dataset [10]. Given that the frequency of the LiDAR, visual, and IMU data is 10HZ, The LIO and VIO subsystems update at the same frequency. Our primary data sources are

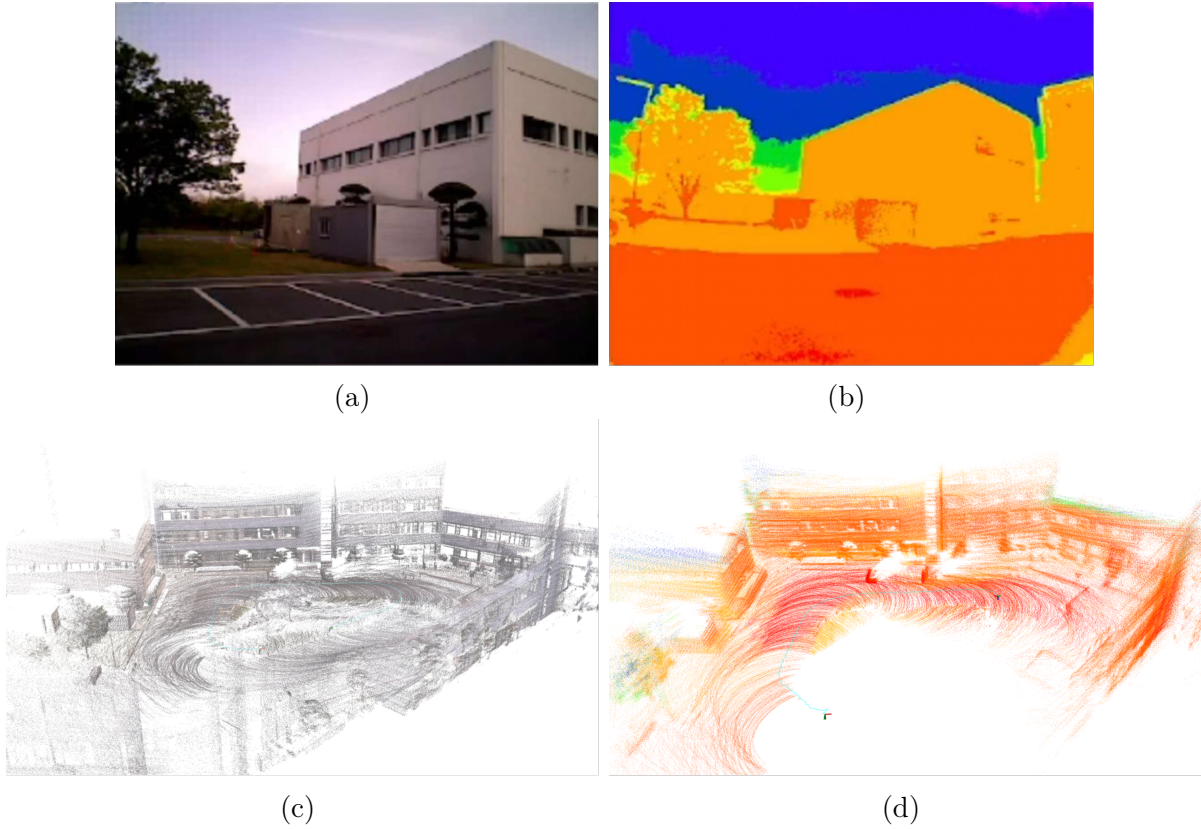


Figure 4.5: LIO and VIO fusion results. (a) RGB images of the building. (b) Thermal image of the building. (c) Point cloud map colorized by RGB images. (d) Point cloud map colorized by thermal images.

LiDAR data and stereo images. For FLVIS, we retained the same parameters as our previous setting in our previous work. Fast-Lio, serving as the LO subsystem, forms FLiVIS in conjunction with FLVIS. For LOAM [63], scan-to-scan LiDAR odometry and scan-to-map optimization are employed. The results of the LVO [146] are consistent with those reported in the original paper, as no open-source code is available. It is worth noting that the original paper did not include information regarding rotation error for the line-based LVO method. Thus, a comparison of rotation error between the studies cannot be made.

The relative translation error is shown in Table 4.1, the LVO-based methods are more accurate than the VO-based and LO-based methods for all sequences. While the line-based LVO [146] achieves the highest accuracy in sequences 04, 06, and 10, the average accuracy of FLiVIS exceeds the line-based LVO [146] by a large margin. It should be noted that the accuracy of LOAM, as presented in Table 4.1, is not as accurate as the results reported on the KITTI odometry leaderboard. This discrepancy can be attributed to the focus on real-time performance. Therefore, the canonical LOAM [63] was chosen as our baseline. The LOAM results showed on the KITTI odometry leaderboard do not operate in real-time. Instead, they were processed at a rate that is 10% of real-time speed, taking approximately one second to process a single scan. For more details, please refer to Section 7.4 of [112].

Table 4.1: Quantitative Results on KITTI Dataset

Sequence	VO	LO	LVO	
	FLVIS	LOAM	line-based LVO	FLiVIS
02	2.30/5.7	1.33/15.7	1.38	0.58/1.5
04	2.77/5.6	1.39/4.7	0.42	0.54/ 0.5
06	2.14/2.4	1.29/5.6	0.61	1.1/ 1.8
08	1.22/5.2	0.64/5.2	1.27	0.32/1.3
10	1.18/2.5	1.97/6.5	0.83	0.84/ 0.8
average	1.92/4.28	1.32/7.54	0.90	0.68/1.02

Note that the performance is represented in 'A/B' format, where 'A' indicates the relative translation error in %, and 'B' indicates the relative rotational error in 10^{-3} degrees per meter.

Outdoor scenes pose significant challenges for visual odometry due to factors such as changes in lighting and a lack of feature points. When there are too few feature points within the camera's field of view to perform feature tracking and matching, FLVIS may fail or be subject to drift, Fig. 4.6 shows a challenging scene for VIO where more than

half of the image is occupied with the sky. Visual feature points are concentrated in the middle and lower parts of the image, with a very focused distribution. This can easily lead to divergence during the optimization process, causing drift in visual odometry, and even failure.

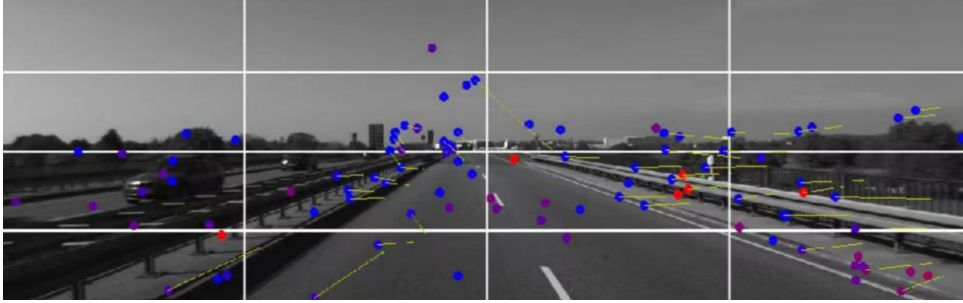


Figure 4.6: Challenging scene for feature-based VIO, where the majority of an image is occupied by the sky, with feature points concentrated in the middle and lower parts of the image. This situation can easily lead to the failure of feature tracking algorithms, and the pose estimation algorithm may become unstable, or even diverge.

The system becomes more robust after incorporating LiDAR odometry. When visual odometry encounters a failure, the system re-initializes visual odometry based on the output of LiDAR odometry. A qualitative comparison between the SVIO FLVIS and proposed LVIO FLiVIS can be seen in Fig. 4.7, it shows the result of the two algorithms on sequence 12 in the KITTI dataset. The error of FLiVIS is smaller than that of FLVIS, and along the whole trajectory, it is aligned with the ground truth.

4.3.2 NTU-Viral Dataset

As stated in [11] Section 1.1, there exists a distinct dichotomy between aerial and terrestrial autonomous vehicles, characterized by the differences in their onboard sensors selection and movement dynamics. In the past, 3D LiDAR was often disregarded for aerial datasets due to payload constraints. The emphasis is typically on high frame-rate camera systems, which provide self-localization capabilities during high-speed and aggressive maneuvers. VIO and V-SLAM systems have been extensively studied to handle increasingly extreme conditions. With the development of LiDAR technology and the challenges of increasingly complex environments, there is now a trend towards equipping UAVs with LiDAR. The NTU-Viral dataset collected data across various outdoor scenarios with a UAV equipped with LiDAR-visual-inertial and UWB sensors. It provides ground truth data for evaluating the accuracy of SLAM algorithms.

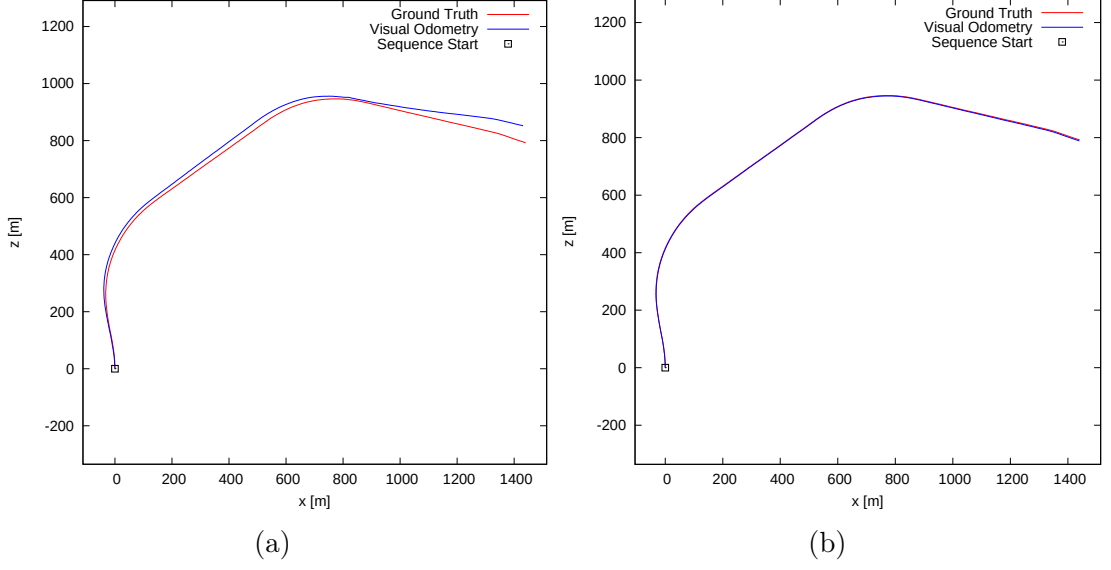


Figure 4.7: Qualitative comparison on KITTI dataset. (a) FLVIS result, and (b) FLiVIS result.

The sensor selection for the NTU-Viral dataset is similar to other autonomous driving datasets, but the installation method differs due to the structure of the drone. On the drone, one LiDAR points downwards and another is horizontal. The small baseline does not affect the performance of SVO in indoor environments, but in outdoor environments with greater depth, it poses additional challenges for stereo triangulation.

From the perspective of data collection, there are significant differences between aerial and ground datasets. Ground vehicles can essentially be considered as moving on a two-dimensional plane, while drones experience large changes in roll and pitch. Drones often perform actions of acceleration and sudden stops. The aerial dataset is smaller than the ground vehicle dataset because the endurance time of ground vehicles far exceeds that of small drones. Shorter paths and more intense motion make drone positioning more challenging than ground vehicle positioning.

Accuracy Comparison

The performance of FLiVIS with state-of-the-art SLAM algorithms using different sensor combinations, i.e. Vins-fusion [75], M-LOAM [147], Fast-livo [76], LVI-SAM [92], and Fst-livo2 [148] are evaluated and compare in this section. As can be seen from Table 4.2, FLiVIS achieved the highest accuracy in all sequences, and it performed better than Fast-LiVO, which is also an LVIO. It can be observed that the localization accuracy progressively increases from VIO to LO and then to LVIO. This is because LO results

Table 4.2: Absolute Trajectory Error in (m) on NTU-Viral Dataset

Seq. num	Vins-fusion	M-LOAM	Fast-livo	LVI-SAM	FLiVIS	Fast-livo2
eee_01	0.608	0.249	0.280	3.901	0.130	0.068
eee_02	0.506	0.166	0.170	0.128	0.125	0.051
eee_03	0.494	0.232	0.230	0.287	0.162	0.068
sbs_01	0.508	0.173	0.290	0.254	0.142	0.062
sbs_02	0.564	0.147	0.220	0.221	0.140	0.061
sbs_03	0.878	0.153	0.220	0.309	0.132	0.060
average	0.593	0.187	0.235	0.239	0.138	0.0616

are more reliable under varying lighting conditions, and during intense movements, IMU data provides additional robustness¹.

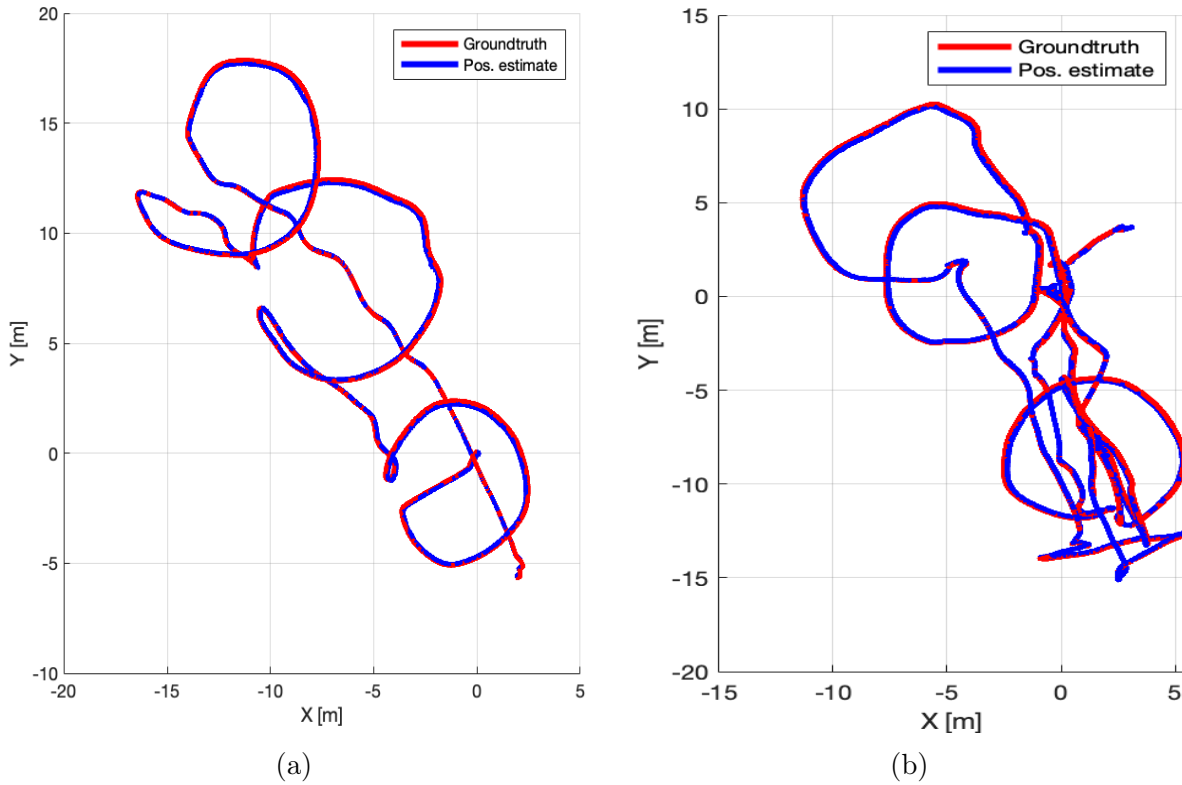


Figure 4.8: Trajectory of FLiVIS on the NTU-Viral dataset. (a) sequence eee_03, and (b) sequence sbs_03.

Fig. 4.8 shows the trajectories of FLiVIS and ground truth in the eee_03 sequence and sbs_03 sequence. Compared with Fig. 4.7, it can be seen that the drone's motion

¹Our method FLiVIS was developed from 2023 to early 2024, which outperforms all the compared state-of-the-art methods published before 2024, while its accuracy is lower than that of Fast-livo2 (2025).

trajectory is more curved and irregular, while the vehicle’s motion trajectory is almost a smooth curve. Moreover, the drone’s motion range is smaller, with the overall range within 100 meters, while in Fig. 4.7, the vehicle’s motion range in the x and y directions is over 1000 meters. Despite this, the trajectory derived from FLiVIS is almost identical to the ground truth, demonstrating the accuracy and robustness of the method we proposed.

Fig. 4.9 shows the LiDAR mapping result and the alignment of LiDAR mapping with Google Maps. It can be seen that the LiDAR map aligns well with Google Maps.

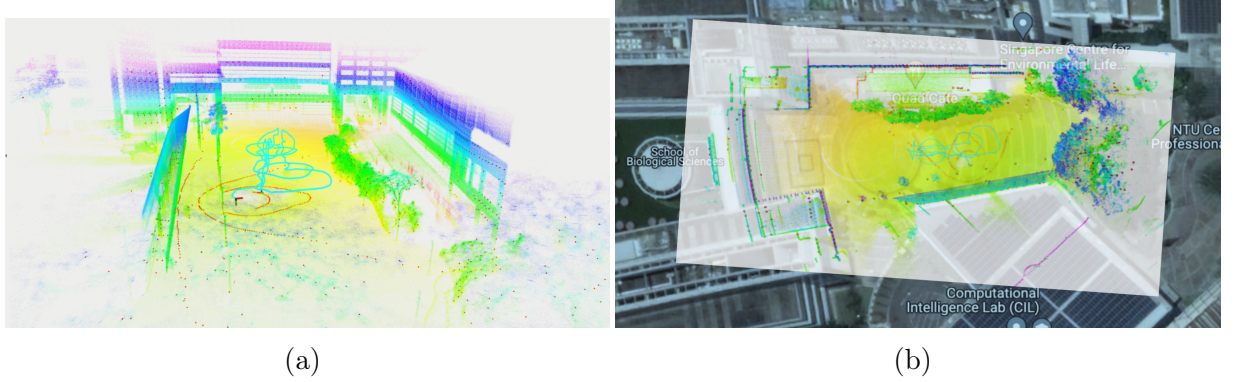


Figure 4.9: Mapping result of School of Biological Sciences, (a) LiDAR mapping result, and (b) top-down view of LiDAR mapping aligns with Google map.

Computational Cost Comparison

The average computational time per LiDAR scan and image frame of our proposed system were tested on a laptop (Lenovo ideapad s530-131wl) equipped with an Intel Core i7-8565U CPU and 8 GB RAM. Our evaluations span the public NTU-VIRAL dataset, covering typical indoor-outdoor hybrid scenes in the sequence. Compared with the high-end desktop platform (Intel i7-10700K, 32GB RAM) used in Fast-LIVO2, the Lenovo ideapad s530-131wl adopts a low-power mobile CPU with weaker computational performance (fewer cores and lower base frequency). As a result, both FLVIS and Fast-LIVO2 show increased processing time on this laptop platform, which conforms to the inherent correlation between hardware performance and algorithm computational efficiency. The detailed computational time of each sequence and the average value are summarized in Table 4.3, where the total processing time of each method is the sum of LiDAR sub-module time and image (stereo image for FLVIS) sub-module time. Our processing time is slightly slower than that of Fast-LIVO2. While Fast-LIVO2 outperforms our method in both accuracy and processing speed, its algorithm imposes extremely high requirements on hardware synchronization. In contrast, our approach adopts a loosely coupled method, enabling us to obtain colorized point clouds at a lower cost.

Table 4.3: Computational Time (ms) on i7-8565U, 8GB RAM evaluated on NTU-Viral Dataset

Seq. num	FLiVIS (LiDAR/stereo image)	Fast-LIVO2 (LiDAR/image)
eee_01	42.74 (19.27 / 23.47)	35.47 (19.65 / 19.16)
eee_02	39.54 (19.05 / 20.48)	36.18 (18.47 / 16.61)
eee_03	38.41 (16.91 / 21.50)	37.36 (17.79 / 17.34)
sbs_01	39.14 (18.42 / 20.72)	35.81 (15.92 / 16.56)
sbs_02	40.18 (18.77 / 21.40)	37.14 (19.02 / 17.00)
sbs_03	44.51 (19.72 / 24.79)	36.18 (20.29 / 16.61)
average	40.75 (18.69 / 22.06)	36.36 (18.69 / 17.21)

4.4 Conclusion

In this work, we introduce a LiDAR-stereo-visual-inertial pose estimation method, which comprises an LIO subsystem and a VIO subsystem. The issue of sensor fusion is approached from a frequency domain perspective, and data from different sensors are integrated using a complementary filter. Our proposed system has been tested on the KITI and NTU-Viral datasets, achieving a relative translation error of 0.68 % and an absolute translation error of 0.138 m respectively. These results are competitive with those of other state-of-the-art algorithms.

Chapter 5

Map Optimization Based on LiDAR Bundle Adjustment

5.1 Research Background

Contemporary robotic automation systems have emerged as a revolutionary force in industrial evolution, fundamentally enhancing workplace safety protocols and redefining workforce efficiency standards. These advancements are particularly evident in mission-critical sectors such as national infrastructure preservation and urban security operations. In many hazardous industrial scenarios, robots can replace human workers, reducing their exposure to danger. Continuous 24-hour patrols can ensure equipment safety and reduce the labor burden on workers [12]. UAVs offer visual assessments of structures, eliminating the need for manual inspections. This approach can reduce inspection costs, minimize public disruption, and provide engineers with a better three-dimensional understanding of the system [13]. Common examples include high-risk infrastructure inspection, such as bridge structural health monitoring [105], or power line patrolling [106].

The current mainstream architecture of autonomous robotic systems has formed a standardized paradigm consisting of four core modules: SLAM (Simultaneous Localization and Mapping), motion control, path planning, and environmental perception. UAVs or aerial robots development also follows this paradigm, with the development of the open-source community significantly accelerating the technology implementation process. SLAM subsystems typically employ LiDAR-inertial-camera fusion pipelines [14]. Motion control layers implement suitable controllers according to different types of robot models, for example, MPC schemes with disturbance observers [15]. Path planning modules integrate sampling-based algorithms (RRT*/A*) [16] to find a path in the presence of ob-

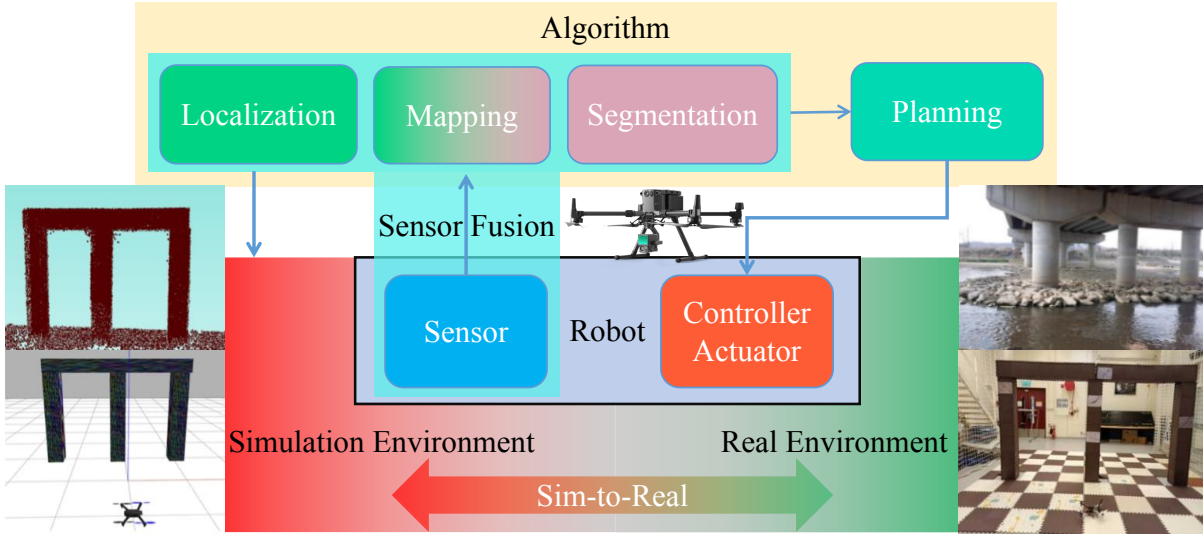


Figure 5.1: Bridge inspection system architecture, from sensor fusion to perception, planning, and control

stacks through collision checking. Environmental perception stacks leverage deep learning architectures for multimodal sensor fusion and target detection [17], segmentation [18], for object recognition tasks.

Despite significant progress, existing systems still have three core deficiencies: Firstly, the accuracy of LiDAR mapping is insufficient, as point cloud mapping is often noisy and fails to meet the precision requirements for bridge management. Secondly, the geometric maps produced by mainstream SLAM systems lack semantic layers, making it difficult for inspection robots to understand the functional attributes of specialized components such as "bridge expansion joints" or higher-level semantic information. Thirdly, a bidirectional closed-loop between simulation and reality has not yet been established. Current sim2real technologies only achieve unidirectional transfer, lacking a reverse optimization mechanism for simulation models using real data. To address the above bottlenecks, this study proposes a hierarchical semantic SLAM framework and a bidirectional sim2real system. Innovatively, it constructs a three-level semantic map architecture: the bottom-level geometric layer (point cloud), the mid-level component layer (associating geometric elements and features), and the top-level functional layer (semantic information collection for bridge management, etc.). The main contribution of this work are concluded as follows:

- **Structural Feature-Enhanced LiDAR Bundle Adjustment:** Incorporates structural primitives (corner points, linear edges, planar surfaces) into LiDAR bundle adjustment. Improve map consistency in feature-deprived environments like bridges.

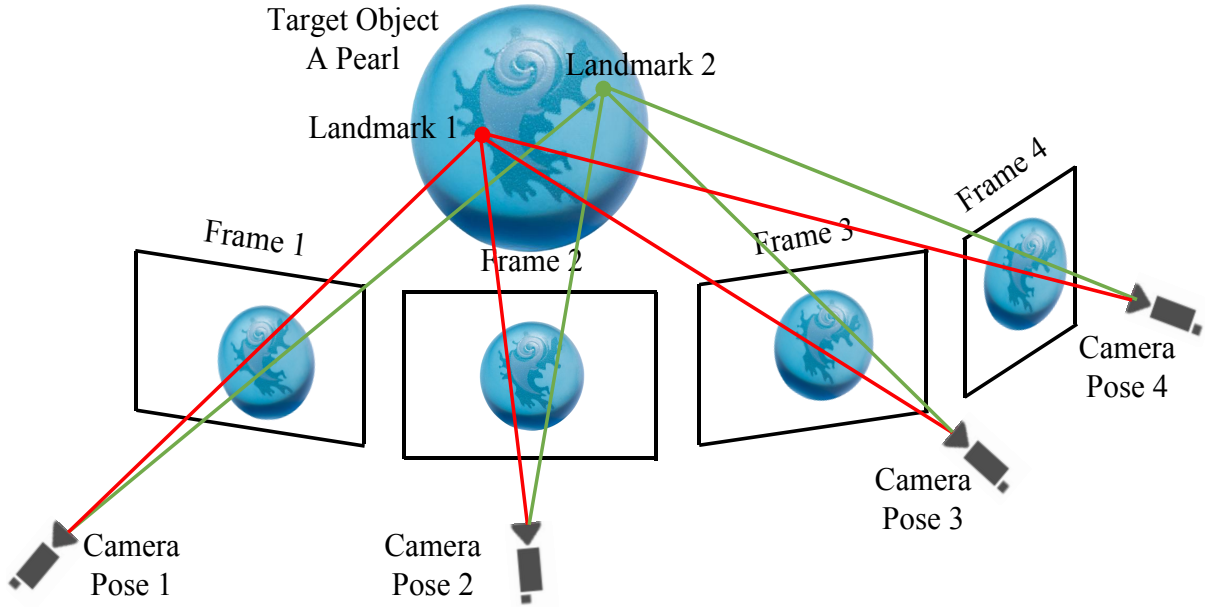


Figure 5.2: Visual bundle adjustment illustration.

- Hierarchical Semantic SLAM Architecture: Geometric Layer: High-fidelity 3D point clouds with topology-aware feature descriptors; Structural Layer: Parametric models of load-bearing components via automatic CAD template matching; Semantic Layer: Semantic attributes fused via 3DSG.
- Bidirectional Sim2Real Framework with Physical Consistency: Dual validation across Gazebo simulation and physical bridge environments, enables full-coverage structural inspection.

5.2 Methodology

The front-end introduced in 4 can be used for online inspection; however, for infrastructure management, the accuracy of front-end mapping is often insufficient. Whether using LiDAR mapping or visual mapping, the results are frequently noisy, and there can be inconsistencies between different inspection tasks. Therefore, back-end processing is necessary for long-term infrastructure management. Typically, BA techniques are used for back-end processing.

While both LiDAR and visual odometry in the front-end yield noisy outputs and cross-task inconsistencies that hinder their application in long-term infrastructure management, back-end optimization via BA techniques addresses this critical limitation. Among these

approaches, visual BA has achieved substantial maturity through decades of extensive research, with its core framework, as shown in Fig. 5.2, well-established in the field. visual BA typically employs feature extractors and descriptors (e.g., SIFT [149], ORB [150]) to align cross-frame visual features, then jointly optimizes the 3D coordinates of landmarks and camera poses. In contrast, LiDAR bundle adjustment remains in its infancy, having received far less attention and lacking a readily visualizable conceptual framework.

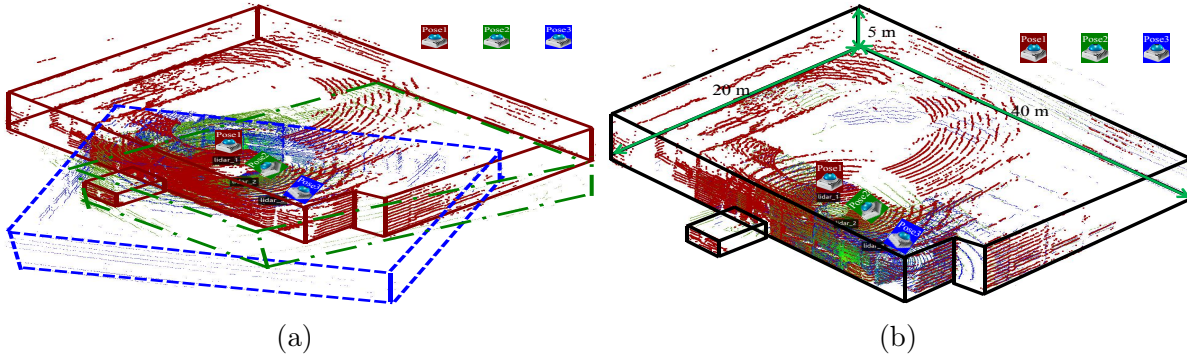


Figure 5.3: LiDAR bundle adjustment illustration. (a) Raw LiDAR point cloud, red for the first frame, green for the second frame, and blue for the third frame. (b) Point cloud map after LBA

LBA has drawn methodological insights from visual BA, specifically adopting the paradigm of refining LiDAR reconstructions to optimize 3D scene structures and LiDAR poses jointly. We depict the LBA process in Fig. 5.3, the LiDAR captures raw data of a gymnasium at different poses, and the raw point cloud is then aligned by the LBA algorithm. Notably, while both visual BA and LBA integrate bundle adjustment with uncertainty modeling, they differ fundamentally in three key aspects:

- Differences in the intrinsic nature of observation sources and data characteristics.
- Differences in the origins and modeling logic of uncertainty.
- Differences in the optimization objectives and constraints of bundle adjustment.

These fundamental discrepancies not only distinguish the two approaches but also underscore the unique innovative value of LBA.

Feature-based visual SLAM requires feature point extraction, resulting in sparse maps, whereas LiDAR mapping can provide dense point cloud mapping. In this section, we introduce LiDAR-BA based on the SLAM front-end FLiVIS. When performing LBA, it is impossible to find accurate point-to-point correspondences as with cameras. Therefore, plane and line features are commonly used. Here, we introduce LBA in the order of point-to-line-to-plane.

5.2.1 LBA without Uncertainties

we derive the deterministic LBA method based on [100].

Point-to-Point LBA Suppose we have M_{frame} point cloud sets. Each frame contains N_{point} points, N_{line} lines, and N_{plane} planes $S_1 = \{p_1^1, \dots, p_1^{N_{point}}\}$ and $S_2 = \{p_2^1, \dots, p_2^{N_{point}}\}$. If the two point clouds are correctly registered, then for the matching point pairs $p_1^i \in S_1$, and $p_2^j \in S_2$, the following equation holds:

$$p_1^i = T_{12}p_2^j. \quad (5.1)$$

The optimization goal becomes:

$$T^*, p^* = \arg \min_{T, p} \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \|T_{wi}p_i^j - q_w^j\|_2^2. \quad (5.2)$$

However, since point-wise matching is only achievable in simulated data and impossible for LiDAR point clouds, the point-to-point LBA is included solely for theoretical completeness and will not be applied in the code implementation.

Point-to-Edge LBA The goal of point-to-edge BA is to minimize the distance from each edge feature point p_i to the corresponding edge line. The edge is represented by the center point q and the direction vector d . The j -th line in the world frame is $l^j = (q^j, d^j)$, and the sum of the squared distances from the point to the line is:

$$(T^*, q^*, d^*) = \arg \min_{T, n, q} \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \left\| (I - d_w^j d_w^{j\top}) (T_{wi}p_i^j - q_w^j) \right\|_2^2, \quad (5.3)$$

where d_w^j is the j -th edge direction unit vector in the world frame

The optimization problem can also be decomposed into:

$$(T^*, q^*, d^*) = \arg \min_T \left(\min_{q, d} \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \left\| (I - d_w^j d_w^{j\top}) (T_{wi}p_i^j - q_w^j) \right\|_2^2 \right). \quad (5.4)$$

According to [100], its analytical solution is determined by the trace and maximum

eigenvalue of the matrix A :

$$\begin{aligned} \min_{n,q} \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \left\| (I - d_w^j d_w^{j\top})(T_{wi} p_i^j - q_w^j) \right\|_2^2 \\ = \text{Tr}(A) - \lambda_1(A) = \lambda_2(A) + \lambda_3(A), \end{aligned} \quad (5.5)$$

where:

- $\text{Tr}(A) = \frac{1}{N} \sum_{i=1}^N \|p_i - \bar{p}\|_2^2$: trace of matrix A
- $\bar{p} = \frac{1}{N} \sum_{i=1}^N p_i$: mean of feature points
- $A = \frac{1}{N} \sum_{i=1}^N (p_i - \bar{p})(p_i - \bar{p})^\top$: covariance matrix
- $d^* = u_2$: eigenvector corresponding to the second largest eigenvalue $\lambda_2(A)$
- $\lambda_3(A)$: minimum eigenvalue of A , corresponding to eigen vector $n^* = u_3$
- $q^* = \bar{p}$: optimal reference point

That is, project all the points of the M frame to the world coordinate system, overlap them, and reduce the next two eigenvalues of the points corresponding to each line.

Point-to-Plane BA The goal of point-to-plane BA is to minimize the sum of the squared distances from each planar feature point p_i (whose position depends on the pose T_{s_i}) to the plane $plane = (q, n)$ q and normal vector n , j-th plane in i-th frame is $plane_i^j = (q_i^j, n_i^j)$, point to line:

$$(T^*, n^*, q^*) = \arg \min_{T,n,q} \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \left(n_w^{j\top} (T_{wi} p_i^j - q_w^j) \right)^2, \quad (5.6)$$

Where:

- n : Plane normal vector (unit vector)
- q : any reference point on the plane

The optimization problem can be decomposed into two steps:

$$\begin{aligned} (T^*, n^*, q^*) = \\ \arg \min_T \left(\min_{n,q} \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \left\| n_w^{j\top} (T_{wi} p_i^j - q_w^j) \right\|_2^2 \right). \end{aligned} \quad (5.7)$$

As with [100], its analytical solution is related to the eigenvalues of the matrix A :

$$\arg \min_T \left(\min_{n,q} \frac{1}{N} \sum_{i=1}^M \sum_{j=1}^N \left\| n_w^j{}^\top (T_{wi} p_i^j - q_w^j) \right\|_2^2 \right) = \lambda_3(A), \quad (5.8)$$

where:

- $\bar{p} = \frac{1}{N} \sum_{i=1}^N p_i$: mean of feature points
- $A = \frac{1}{N} \sum_{i=1}^N (p_i - \bar{p})(p_i - \bar{p})^\top$: covariance matrix
- $\lambda_3(A)$: minimum eigenvalue of A , corresponding to normal vector $n^* = u_3$
- $q^* = \bar{p}$: optimal reference point

That is, project all the points of the M frame to the world coordinate system, overlap them, and reduce the smallest eigenvalue of the points corresponding to each line. The ultimate goal of the BA problem can be unified as minimizing the eigenvalue function:

$$T^* = \arg \min_T \lambda_k(p(T)), \quad (5.9)$$

where $p = [p_1^\top, \dots, p_N^\top]^\top$ represents the set of all feature points, and λ_k depends on the feature type (plane is λ_1 , edge λ_1 and λ_2 ; point λ_1 , λ_2 , and λ_3).

Many methods use voxels to process LiDAR point clouds, balancing computational speed and accuracy, but this approach is effective only in environments with specific structural features. Since point-wise matching is impossible, we introduce uncertainty from conventional LiDAR odometry, shifting from point-to-point ICP to weighted ICP or NDT. This incorporates uncertainty and seeks solutions for point-to-point matching and LiDAR BA from a statistical perspective. Multi-frame point cloud registration, typically completed iteratively using the EM algorithm, is an offline method unsuitable for robotic applications. Therefore, we integrate existing LBA concepts [100] with EM-based multi-frame point cloud registration [97] to propose an LBA with uncertainties method for point cloud refinement.

5.2.2 LBA with Uncertainties

In the above lidar BA process, line and surface features are implicitly expressed, and the optimization is done by minimizing the λ in each voxel or point cluster, without

considering uncertainty. Although there is a line direction vector or plane normal vector when calculating the error at the beginning, it does not appear in the optimization process. We now solve this problem from a probability perspective and derive explicit features as mapping components for infrastructure management.

Point-to-Point LBA with uncertainty Recall point-to-point LBA in (5.2), each point and its variance are represented as $p_i \sim \mathcal{N}(\mu_i, \Sigma_i)$. Assume there is a virtual frame $V \sim \mathcal{N}(\mu_{GT}, \Sigma_{GT})$, where the μ_{GT} is the ground truth mean value of the point cluster and Σ_{GT} is the ground truth covariance of the point cluster. After the transformation T , the probability likelihood function of each point matching the virtual frame is

$$P(\{p_i\}|T, q) = \prod_{i=1}^N \mathcal{N}((T_{wi}u_i^j - u_w^j); 0, \Sigma_{\text{proj}}). \quad (5.10)$$

Taking $-\text{Log}()$ of the probability density function gives

$$\begin{aligned} & -\text{Log}(\arg \max P(\{p_i\}|T, q)) = \\ & \arg \min (T_{wi}u_i^j - u_w^j)^\top \Sigma_w^{j-1} (T_{wi}u_i^j - u_w^j) \end{aligned} \quad (5.11)$$

Then take the negative logarithm of the probability density function

$$\arg \min_{T_{vi}, u_i} \sum_{i=1}^m \sum_{j=1}^n (T_{wi}u_i^j - u_w^j)^\top \Sigma_w^{j-1} (T_{wi}u_i^j - u_w^j). \quad (5.12)$$

Then, the cost function becomes:

$$\begin{aligned} & \arg \max_{T_{wi}, u_i, \Sigma_i} \prod_{i=1}^M \prod_{j=1}^N P(T_{wi}p_i^j; \mu_i^j, \Sigma_i^j) \\ & = \arg \min_{T_{vi}, u, \Sigma_t} \sum_{i=1}^M \sum_{j=1}^N \left((T_{wi}u_i^j - u_w^j)^\top \Sigma_w^{j-1} (T_{wi}u_i^j - u_w^j) \right) \\ & = \arg \min_{T_{vi}, u, \Sigma_t} \sum_{i=1}^M \sum_{j=1}^N \|T_{wi}u_i^j - u_w^j\|_{\Sigma_w^{j-1}}^2. \end{aligned} \quad (5.13)$$

Point-to-Edge LBA with uncertainties For point-to-edge LBA, similarly, we assume that there is a virtual frame with an edge (q_{gt}, d_{gt}) , and then transform the points on this edge of other frames to the virtual frame through the transformation matrix T , and then calculate the point-to-edge error, and weighted error with covariance

The geometric error from point to line is defined as the distance projected to the normal plane:

$$e_i = (I - d_w^j d_w^{j\top})(T p_i^j - q_w^j) \quad (5.14)$$

Introducing the probability model and covariance weighting, assuming that the noise of the measurement point p_i^j obeys an anisotropic Gaussian distribution:

$$\hat{p}_i^j \sim \mathcal{N}(p_i^j, \Sigma_p) \quad (5.15)$$

The covariance matrix of the projection error is:

$$\Sigma_{proj} = (I - d_i^j d_i^{j\top}) \Sigma_p (I - d_i^j d_i^{j\top}) \quad (5.16)$$

The point-to-edge alignment probability density function is given by

$$P(\{p_i\} | T, d, q) = \prod_{i=1}^M \prod_{j=1}^N \mathcal{N}\left((I - d_i^j d_i^{j\top})(T p_i^j - q_w^j); 0, \Sigma_{proj}\right). \quad (5.17)$$

Construct a weighted cost function through Mahalanobis distance:

$$\begin{aligned} \arg \min_{T, n, q} \sum_{i=1}^M \sum_{j=1}^N e_i^\top \Sigma_{proj}^{-1} e_i \\ = \arg \min_{T, n, q} \sum_{i=1}^M \left\| (I - d_w^j d_w^{j\top})(T p_i^j - q_w^j) \right\|_{\Sigma_{proj}^{-1}}^2. \end{aligned} \quad (5.18)$$

The final optimization form is:

$$\begin{aligned} (T^*, d^*, q^*) = \\ \arg \min_{T, n, q} \sum_{i=1}^M \sum_{j=1}^N \underbrace{\left\| (I - d_w^j d_w^{j\top})(T p_i^j - q_w^j) \right\|_{\Sigma_p^{-1}}^2}_{\text{weighted projection error}}. \end{aligned} \quad (5.19)$$

Point-to-Plane LBA with uncertainty For point-to-plane LBA, similarly, we assume that there is a virtual frame with a plane $(q_g t, n_g t)$, and then transform the points on this plane of other frames to the virtual frame through the transformation matrix T , and then calculate the point-to-plane error, and weighted error with covariance.

The geometric error from point to plane is a scalar projection:

$$e_i = n_w^j{}^\top (T_{wi} p_i^j - q_w^j) \quad (5.20)$$

Assume that the noise of the measurement point p_i follows a Gaussian distribution:

$$\hat{p}_i^j \sim \mathcal{N}(p_i^j, \Sigma_p) \quad (5.21)$$

The variance is the projection variance in the normal direction:

$$\Sigma_{proj} = n_w^j{}^\top \Sigma_p n_w^j \quad (5.22)$$

By constructing the objective function through the Mahalanobis distance, we give the derivation of the weighted cost function:

$$\arg \min_{T,n,q} \sum_{i=1}^M \sum_{j=1}^N \left\| n_w^j{}^\top (T_{wi} p_i^j - q_w^j) \right\|_{\Sigma_{proj}}^2 \quad (5.23)$$

Final optimization form

$$(T^*, n^*, q^*) = \arg \min_{T,n,q} \sum_{i=1}^M \sum_{j=1}^N \underbrace{\frac{\left(n_w^j{}^\top (T_{wi} p_i^j - q_w^j) \right)^2}{n^\top \Sigma_p n}}_{\text{weighted scalar error}} \quad (5.24)$$

Through the above formulation, points, lines, and planes in LBA are explicitly modeled, by assuming that there is a virtual frame that can give a reference to all points, lines, and planes in all frames, and then align the point cloud of each frame with these features. Then, we use the gradient descent method to optimize the error of the overlapping point cloud relative to each feature, and thus jointly optimize point cloud distribution and sensor measurement.

5.3 Experiment

Our experimental evaluation is structured into two complementary parts, designed to comprehensively assess the efficacy and robustness of LiDAR bundle adjustment (BA) methods under both controlled and real-world conditions.

First, we conduct a performance comparison of various LiDAR BA approaches us-

ing synthetic data. This controlled setting allows us to precisely characterize algorithmic behavior in the presence of known ground-truth trajectories and point cloud geometry, enabling quantitative analysis of accuracy, convergence properties, computational efficiency, and sensitivity to noise or initialization errors.

Second, we evaluate the same set of LiDAR BA methods on real-world data collected during a full-coverage structural inspection of a bridge. This dataset presents practical challenges such as repetitive geometric structures, dynamic environmental conditions, and sensor noise—factors that are difficult to replicate in simulation but critically impact real-world performance. By comparing results across both synthetic and real scenarios, we aim to provide insights into the generalizability, reliability, and operational suitability of current LiDAR BA techniques for infrastructure inspection tasks.

5.3.1 Lidar Bundle Adjustment on Synthesis Data

The LiDAR BA algorithms constitute the backend of geometric SLAM, and also serve as the intermediate layer within the entire semantic SLAM pipeline. A well-established geometric map provides the foundation for subsequent semantic mapping and inspection. Our experiments follow the methodology of BALM [100] and BALM2 [2] for comparative analysis. Specifically, we first conduct Monte Carlo simulations using synthetic data, followed by bridge mapping with GT models, and finally perform voxel count comparisons without GT models.

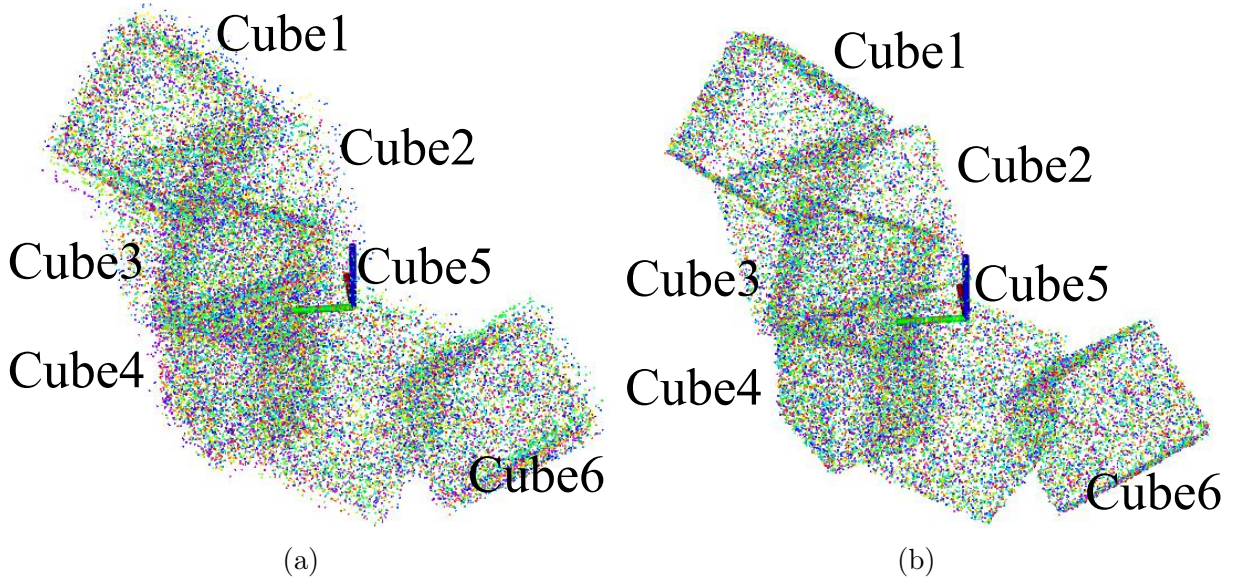


Figure 5.4: Cube feature point cloud before and after LiDAR BA. (a) Before LiDAR BA. (b) After LiDAR BA.

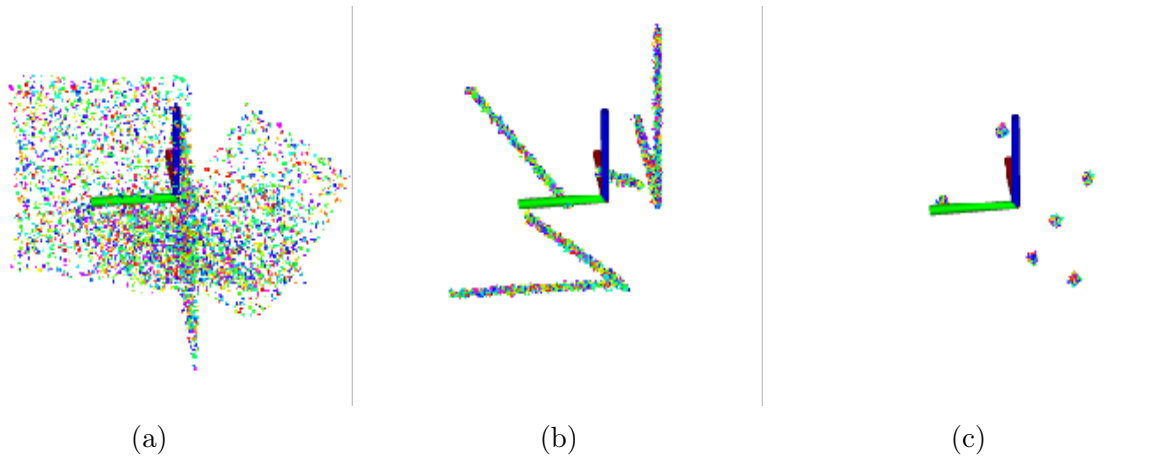


Figure 5.5: Different types of features for LiDAR BA in the simulation. (a) Plane features. (b) Edge features. (c) Corner features.

Following [100] and [151], our synthetic data consists of randomly generated poses along with cube, line, and corner point features. The cube feature point cloud before and after LiDAR BA is shown in Fig. 5.4, and the other types of point cloud features after LiDAR BA are depicted in Fig. 5.5. For each feature, the simulator also generates local observations within individual frames. Feature correspondences across different scans are pre-established. Furthermore, to demonstrate the strengths and weaknesses of different methods, we evaluated their performance under varying parameter configurations. These configurations include the number of features (planes, lines, and corner points), the num-

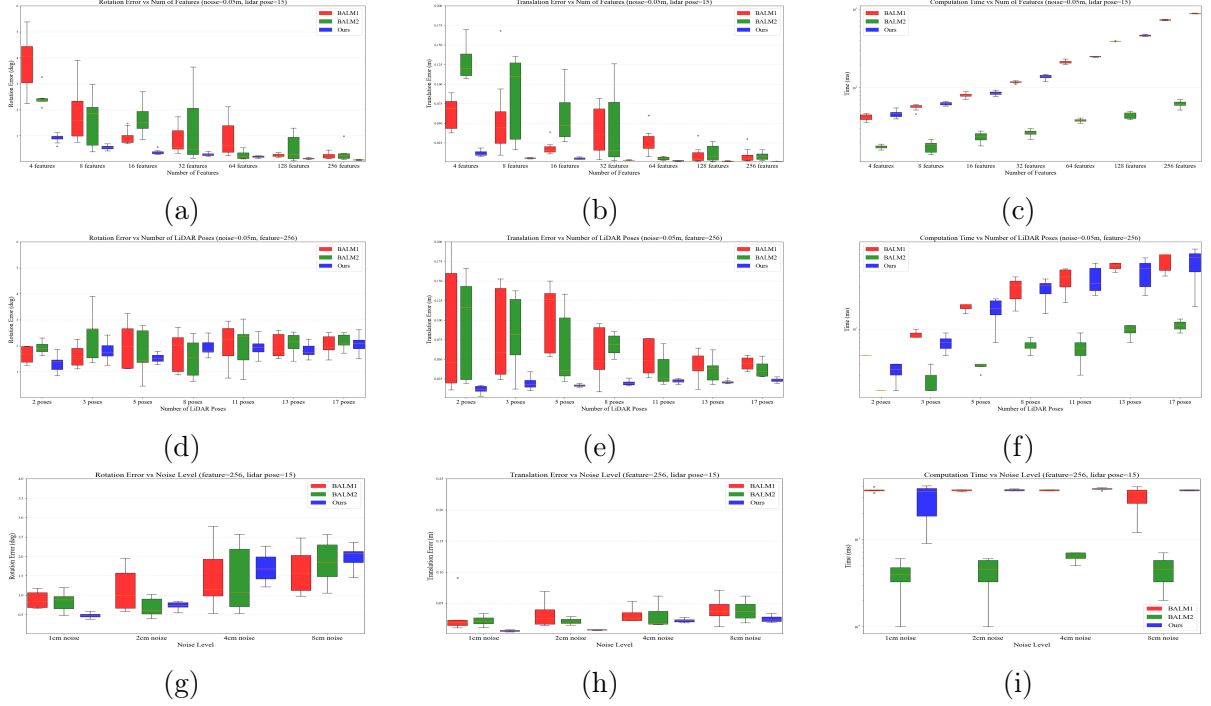


Figure 5.6: Comparison of accuracy and computational efficiency with benchmarks under different experimental configurations. (a)–(c): Fixed point cloud noise (5 cm) and LiDAR pose count (15), feature number 4–256; (a) rotational accuracy, (b) translational accuracy, (c) computational time. (d)–(f): Fixed point cloud noise and feature number (256), LiDAR pose count 2–17; (d) rotational accuracy, (e) translational accuracy, (f) computational time. (g)–(i): Fixed feature number (256) and LiDAR pose count (15), point cloud noise 1–8 cm; (g) rotational accuracy, (h) translational accuracy, (i) computational time.

ber of poses, and observation noise σ . Evaluations were performed using observation sets generated with predefined parameters.

The evaluation results are illustrated in Fig. 5.6, where the 3×3 subfigures are organized to investigate the impact of three key variables (number of features, LiDAR pose count, and point cloud noise) on algorithm performance.

Specifically, Row 1 presents results under the setup where point cloud noise is fixed at 5 cm, LiDAR pose count is set to 15, and the number of features varies from 4 to 256. Under this configuration, our method outperforms all benchmark methods in both rotational and translational accuracy. In contrast, BALM2 [2] achieves the fastest computation speed but exhibits unstable performance when the number of features is small. Row 2 reports experiments with fixed point cloud noise, a fixed feature count of 256, and LiDAR pose count varying from 2 to 17. The results indicate that rotational accuracy is minimally affected by the number of LiDAR poses across all methods, while BALM1 and BALM2 exhibit significantly larger variances in translational accuracy when the number of LiDAR

poses is limited. Row 3 shows performance under the setup where feature count is fixed at 256, LiDAR pose count is set to 15, and point cloud noise ranges from 1 cm to 8 cm. Our method consistently achieves the smallest translational error across all noise levels. For rotational accuracy, it outperforms other methods when noise is low; as noise increases, its rotational error becomes comparable to that of other benchmarks, but it maintains the smallest variance, demonstrating superior robustness.

Overall, our proposed formulation achieves state-of-the-art performance in both accuracy and robustness compared to existing methods. A key reason for this advantage is that most benchmark methods do not explicitly model line and corner points, leading to suboptimal coefficient estimation and thus inaccurate pose results. In contrast, our formulation integrates the states of line and corner points into the optimization framework, which provides additional geometric constraints. These constraints enable the estimation of optimal coefficients given the current pose estimates, effectively minimizing the objective function and enhancing both accuracy and robustness.

5.3.2 Bridge Mapping with GT Model

We verify the effectiveness of the proposed BA in LiDAR mapping experimentally. The LiDAR odometry executes at 10 Hz. Map refinement, triggered upon receipt of every fifth scan, consequently runs at 1 Hz. A sliding window buffer holds the 20 latest scans for processing.

We compare our method with four SOTA methods that focus on the LiDAR BA problem, namely: Eigen-Factor [99], BALM [100], BAREG [151], and BALM2 [2]. We assessed the mapping accuracy by comparing the mapping results against a CAD model. This CAD model was created and updated by the Shenzhen Municipal Road and Bridge Bureau during the bridge’s construction phase. Data were collected at the Entrepreneurship Overpass in Bao’an District, Shenzhen, China. The satellite map and the CAD model of the overpass are shown in Fig. 5.7.



Figure 5.7: The Entrepreneurship Overpass in BaoAn district, Shenzhen, China. (a) Satellite map, and (b) CAD model.

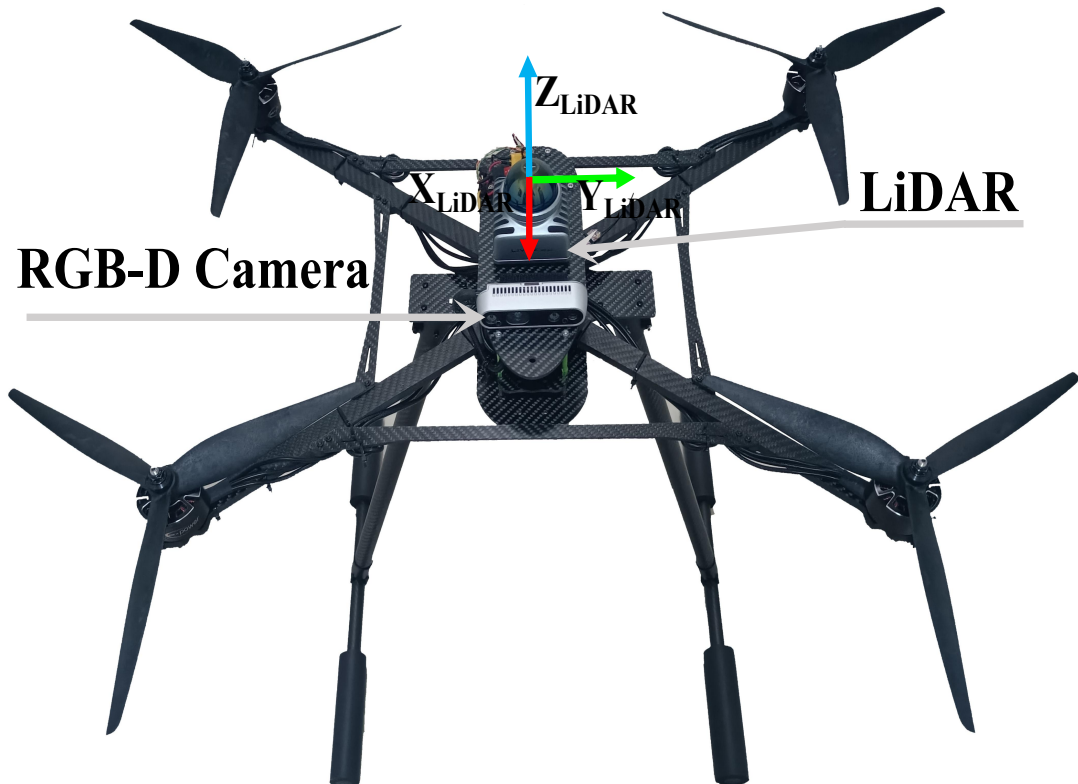


Figure 5.8: Data collection platform, a Livox Mid-360 LiDAR and a Realsense D435 RGB-D camera are mounted on a 15-inch UAV.

The data collection platform, depicted in Fig. 5.8, consists of a Livox MID-360 LiDAR mounted on a 15-inch quadrotor UAV. We acquired four sequences covering one side of the overpass. Data collection was limited to this section because the central span traverses a busy arterial road, while the bridge sections on either side pass over park areas, making UAV flight over the central roadway impractical due to safety regulations and operational constraints. Different from the CAD model, vegetation is also present around the actual bridge structure. The UAV flight path was planned using a pre-established point cloud map to ensure coverage of the vertical surfaces of the bridge piers and deck, while avoiding collisions with both the bridge structure and surrounding trees. The actual flight scenario is depicted in the accompanying Fig. 5.9.



Figure 5.9: The UAV is flying around the bridge piers and deck for coverage inspection task.

All sequences underwent two preprocessing steps: motion undistortion and scan down-sampling. To compensate for point cloud distortion caused by continuous LiDAR motion during scanning, we utilized the front-end FLiVIS algorithm. FLiVIS provides an initial pose estimate for each frame while simultaneously estimating IMU biases (along with other state variables) and performing real-time motion undistortion of the points. We retained all points in the scan that had been undistorted by FLiVIS but discarded its odometry output. Subsequently, the processed data from all sequences were downsampled from the original 10 Hz to 1 Hz. This subsampling is necessary because the BA method processes all scans simultaneously, and the 10 Hz scan rate imposes excessive computational load on any BA approach. This downsampling step is analogous to keyframe selection commonly

employed in visual SLAM frameworks. Finally, the undistorted point clouds and initial poses were fed into the BA optimizer to solve for an improved mapping result.

Similar to BALM2 [2], in addition to multi-view registration approaches, we also compared against classical pairwise registration methods, including ICP, GICP, and NDT implementations from the PCL library, as well as incremental mapping results from LIO systems (specifically Fast-LIO2 [83] and FLiVIS [125]). The pairwise registration methods were operated incrementally: each new scan was incrementally registered and merged with previous scans. To limit computational time, only the last 10 point cloud frames were used during the registration of each new scan. Pose estimates from LIO outputs served as the initial trajectory for both feature association and optimization in all BA methods evaluated, including EF, BALM, BAREG, BALM2, and our proposed approach.

Table 5.1: Voxel Count Comparison Experiment

Sequence	G-ICP	NDT	Fast-Lio	EF	BALM	BAREG	BALM2	Ours
Seq. 1	+8874	+6083	+1874	+723	+482	+375	+188	476533
Seq. 2	+8363	+5786	+1851	+740	+500	+396	+196	443825
Seq. 3	+10107	+7109	+2303	+945	+643	+501	+227	664357
Seq. 4	+15104	+10406	+3259	+1270	+857	+668	+258	887590
Seq. 5	+12701	+9204	+3302	+1458	+1024	+822	+332	986545

Note the rightmost column reports the absolute voxel counts produced by our method. The values for all other methods are presented as positive offsets (indicated with a +), representing how many more voxels other methods generate compared to our approach.

The bridge components within the CAD model remain accurate and undergo continuous monitoring, validating their suitability as GT for point cloud map comparison. However, the central overpass area contains uncontrolled vegetation (grass and trees), whose growth patterns are unpredictable. Such vegetation is prevalent, particularly in bridges located away from urban centers, which often exist surrounded by tall trees. Since botanical models are unavailable for these areas, we adopt a quality assessment method proposed by Filatov [152] for environments lacking ground truth maps. This approach partitions space into uniform voxels and counts the number of voxels occupied by LiDAR points. Higher map quality corresponds to fewer occupied voxels, as accurate multi-scan registration maximizes point consistency, minimizing spatial dispersion.

Using 0.1 m voxels, Table 5.1 presents the number of occupied voxels. Notably, a

lower voxel count indicates a more concentrated point cloud and thus a more consistent surface, whereas a higher count signals greater point cloud noise and poorer mapping performance. To highlight comparative performance, we subtract our method’s voxel count from that of each other method for each sequence. Results include both our absolute counts and the relative differences of other methods. Our approach (incorporating corner and edge features) consistently achieves the lowest occupancy across all sequences, followed by BALM2, BAREG, and BALM. Incremental methods exhibit notably poorer performance. This occupancy-based ranking strongly corroborates the structural accuracy trends observed in bridge model comparisons.

Fig. 5.10 visualizes the alignment between our reconstructed point cloud—obtained by registering the point cloud to the global map using our estimated poses—and the ground truth CAD model. Despite operating in unstructured environments, our method performs well, with the reconstructed point cloud aligning closely with the GT model. To quantitatively evaluate mapping accuracy, we compute the distance from each reconstructed point to its nearest neighbor in the GT geometry and compare our results with those of BALM2 [2]. As shown in Fig. 5.11, our method achieves mapping errors comparable to or lower than those of existing approaches. Structural errors primarily stem from pose estimation inaccuracies and imperfections in point cloud structure optimization.

5.4 Conclusion

Our LiDAR BA approach proposes a novel structural representation using planes, edges, and corner points optimized for bridge environments. Algorithms based on this architecture demonstrate exceptional precision in both simulated environments and real-world experiments. Comparative evaluations against state-of-the-art (SOTA) methods across both simulated and real-world datasets show our approach surpasses existing benchmarks in mapping accuracy. Critically, the plane, edge, and corner point features serve as an intermediate layer connecting semantic and geometric hierarchies. Under full-coverage path planning, our algorithm successfully detects and localizes all cracks present on the bridge structure.

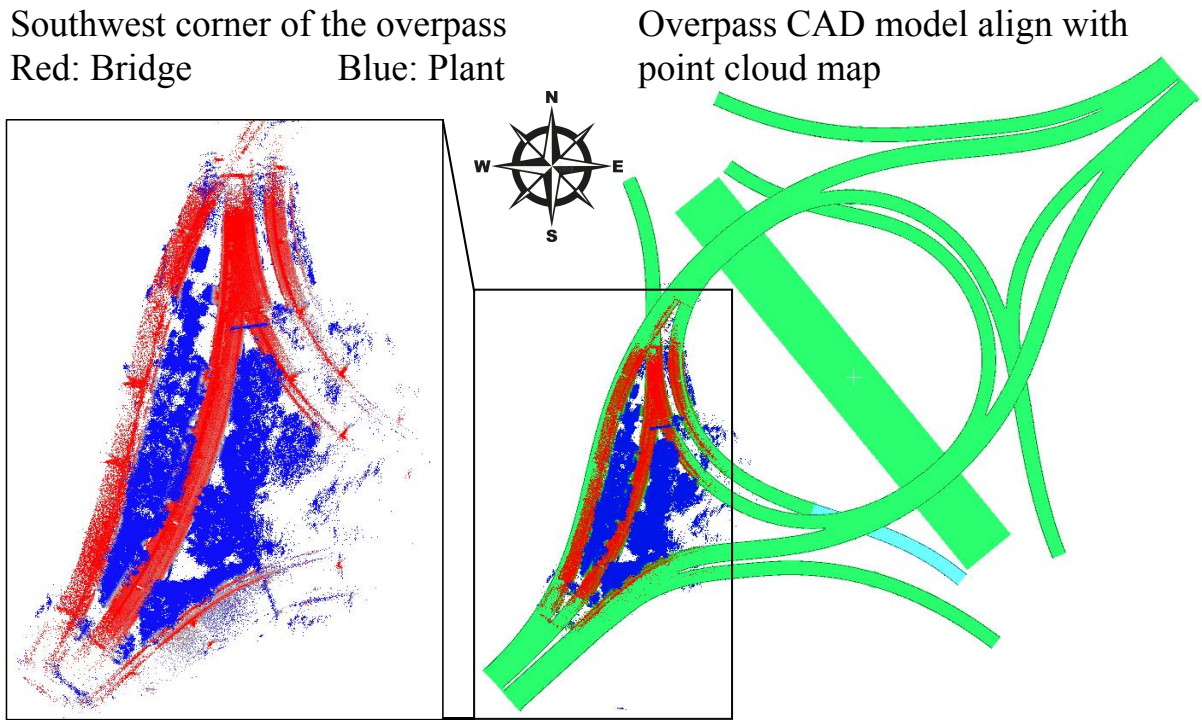


Figure 5.10: LiDAR point cloud mapping visualization results of the southwest corner of the Overpass.

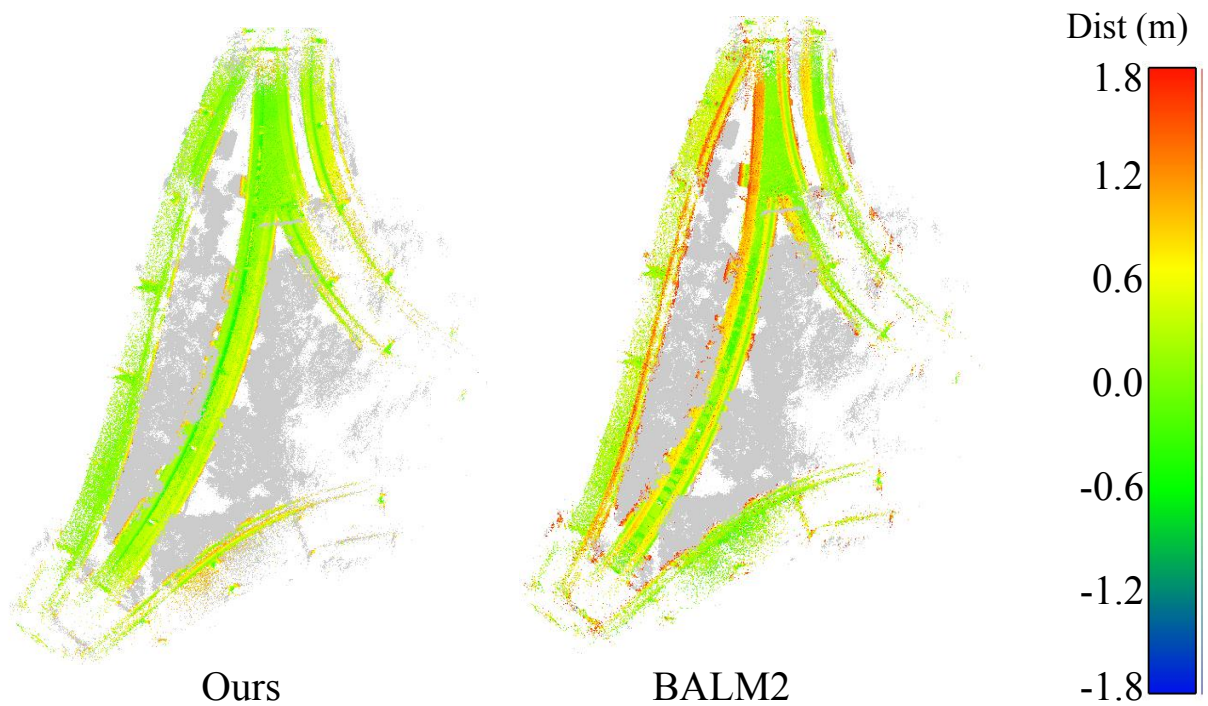


Figure 5.11: Comparative analysis of point cloud-to-CAD structural deviations between our method and BALM2 [2].

Chapter 6

Hierarchical 3DSG-based Semantic-Metric Mapping

6.1 Research Background

Under the dual pressures of current population growth and climate change, food security has become an especially critical challenge for high-density areas like Hong Kong and Singapore. Smart agriculture, empowered by a multimodal technology cluster integrating LLM, robotics, and the IoT, is fundamentally reshaping agricultural production systems. Through the four-dimensional synergy of perception, analysis, decision-making, and execution, a closed-loop system is established, encompassing data collection, intelligent decision-making, precise execution, and feedback evaluation. This innovation not only overcomes the resource efficiency bottlenecks inherent in traditional agriculture but also establishes a scientific paradigm that supports sustainable development. Driven by this new ecosystem of scientific instrumentation, this transformation aligns agricultural and food industries with the United Nations Sustainable Development Goals [19], offering a technological pathway to resolve food security dilemmas in land and resource-constrained regions.

To address these imperatives, we have initiated research on autonomous intelligent vertical farming. Vertical farming addresses critical challenges in densely populated regions (e.g., Hong Kong, Singapore), where arable land is scarce but high-rise buildings and vacant industrial facilities are abundant. Researchers have developed an integrated autonomous intelligent vertical farm system combining UAVs with vertical farming infrastructure [20]. These systems autonomously regulate light environment parameters and nutrient delivery strategies through perceptual feedback data, thereby forming an opti-

mized closed-loop for plant growth. Our work aims to overcome the inherent limitations of traditional agriculture in terms of resource efficiency and environmental adaptability, while simultaneously constructing a closed-loop scientific discovery framework for next-generation smart agriculture.

In recent years, UAV technology has profoundly penetrated various sectors, including surveillance patrols, logistics transportation [153], and geospatial mapping [154]. At the intersection of IoT and smart agriculture, UAVs are evolving from conventional operational tools into intelligent agricultural hubs. Current agricultural scenarios often suffer from widespread infrastructure deficiencies [155], where insufficient Wi-Fi router and base station coverage hinders real-time data transmission. As dynamic aerial nodes, UAVs can establish a "device-edge-cloud" three-tier data transmission architecture [21], effectively bridging the connectivity gap between farmland and digital systems [22]. Drone swarms can execute diversified tasks such as seeding, plant protection, and 3D mapping, acquiring field data through multi-modal sensors. After edge computing preprocessing, agricultural knowledge graphs are constructed to provide structured data support for AI decision-making systems [23]. Their operational paradigm has transcended passive monitoring and evolved into an active agricultural intervention system that integrates precision pesticide application and intelligent harvesting.

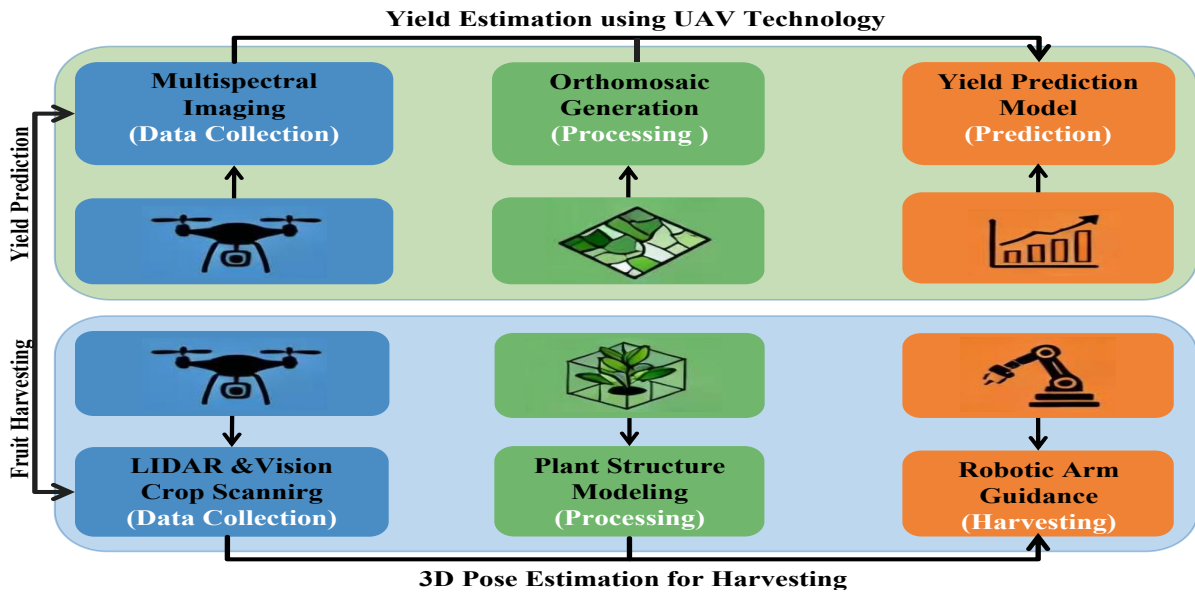


Figure 6.1: Schematic diagram of the dual-subsystem architecture for agricultural inspection and harvesting.

Ideally, a system should both accurately estimate yield and precisely localize each fruit. On the other hand, research on smart agricultural robotics is still relatively nascent,

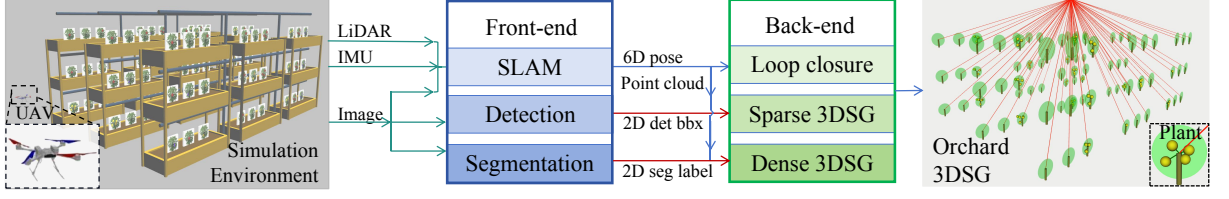


Figure 6.2: System architecture of the proposed framework. Functional blocks are grouped into front-end (low-level perception) and back-end (high-level perception) layers, organized by increasing computational complexity and abstraction level. Data flows are indicated by: **green** for raw sensor data, red for semantic information, and **blue** for system-level communication.

with many pressing challenges yet to be resolved. The community currently finds it difficult to deliver such an ideal solution. In the early stages, it makes sense to divide the agricultural inspection system into two subsystems: an offline statistical subsystem for yield estimation, and an online precise 3D fruit detection subsystem for fruit picking. This dual-subsystem architecture is depicted in Fig. 6.1. This division is motivated by the fact that yield statistics and production planning do not require real-time operation or 3D positional data of every fruit, whereas a harvesting robot needs real-time, high-precision localization of the target fruit but does not depend on knowing the total fruit count across the orchard or field. In summary, functionalities such as full coverage and task allocation—which can be managed by the statistical subsystem—are decoupled from the harvesting operation, which is carried out by robots equipped with the precise fruit mapping subsystem.

Intelligent environmental perception forms the technological foundation for autonomous drone decision-making [13]. However, current mainstream perception solutions exhibit significant limitations: First, SLAM technology generates only semantically deficient point cloud maps (including Visual-SLAM and LiDAR-SLAM), which are inadequate for agricultural knowledge representation. Second, object detection and image segmentation algorithms, predominantly developed on idealized 2D datasets, fail to construct ecologically correlated 3D growth models. Third, a deeper issue lies in existing technologies deconstructing plant organs (leaves, branches, flowers/fruits) into discrete spatial elements [24], neglecting their dynamic coupling with biological attributes and light conditions. Meanwhile, orchard environments exhibit complex textures, microscale targets, and chaotic spatial distributions, causing mainstream 3D object estimation methods (e.g., bounding box regression from autonomous driving [25]) to maintain high error rates in fruit counting and pose estimation tasks. These challenges highlight the necessity of developing agricultural-specific perception frameworks.

To address these challenges, this paper proposes a metric-semantic SLAM framework as shown in Fig. 6.2 for 3DSG-based plant inspection¹. By leveraging the hierarchical structure of scene graphs, our approach significantly enhances fruit counting accuracy through error propagation control in object pose estimation. The systematic analysis of multi-sensor error sources and hierarchical abstraction mechanisms represents a critical advancement in metric-semantic SLAM. The principal contributions are summarized as follows:

- A 3DSG is introduced to construct a multidimensional spatiotemporal representation of plant development, facilitating interpretable growth analysis and traceable, time-series monitoring of agricultural processes.
- A systematic analysis of object estimation uncertainties is conducted based on different sensor models, sensor combinations (including stereo camera, RGB-D camera, and camera-LiDAR combinations), and error propagation modeling, highlighting the critical role of hierarchical abstraction layers in enabling precision agriculture. The effectiveness of the proposed system is validated through both simulated and real-world experiments.
- IoT-based smart vertical farming framework that combines mobile robotic perception systems and automated environmental control mechanisms to enable real-time data collection and closed-loop feedback regulation of crop growth metrics is proposed.

The synergistic development of IoT, robotics, and AI is reshaping the technological paradigms of interdisciplinary laboratories and industrial ecosystems. IoT-enabled vertical farming systems demonstrate significant advantages, achieving yield enhancement while reducing carbon emissions compared to traditional agriculture, thereby providing innovative solutions for sustainable food production. The advancement of robotic technologies further expands the application dimensions of mobile intelligent sensors in vertical farm IoT networks. Based on an autonomous farming system that integrates UAV, sensors, and modular vertical farming units, this study proposes a 3DSG-based hierarchical mapping method for the dynamic monitoring of plant and fruit growth. Through feedback mechanisms, the system optimizes growth conditions by adjusting lighting and nutrient delivery, while the hierarchical mapping architecture reduces detection errors and enables comprehensive 3D visualization. In this chapter, we introduce a intelligent

¹Simulation environment adapted from ICUAS'2024 competition, June 4-7, 2024 in Chania, Greece. Where the authors' team achieved second place

plant inspection system with the following contributions: 1) Pioneering application of 3DSG technology to establish a multi-dimensional spatiotemporal representation model for plant growth processes, supporting interpretable analysis and traceable monitoring; 2) Establishing an uncertainty model through error propagation by systematically analyzing sensor models (covering various common sensor combinations) and integrating these models into 3D object pose estimation algorithms. This highlights the necessity of hierarchical abstraction levels. The system is validated through simulations and real-world experiments, providing a quantitative evaluation of object pose estimation; and 3) An IoT-driven intelligent vertical farming architecture that integrating mobile robotic perception networks and environmental regulation devices, enabling dynamic acquisition and closed-loop control of plant growth parameters.

6.2 Methodology

This section starting from systematically investigates error sources throughout the fruit perception pipeline, spanning from 2D detection to 3D semantic mapping in 6.2.1. Then, we present error analysis of two prevalent depth sensing configurations: i) stereo vision systems in 6.2.2 and ii) LiDAR-camera fusion architectures in 6.2.3, with particular emphasis on systematic errors introduced during depth estimation processes. Then, we analyze absolute/relative errors and abstraction-level implications to justify our hierarchical semantic mapping approach in Section 6.2.4, and at last we formalize our 3DSG framework in Section 6.2.5.

6.2.1 Projection Model and Noise Propagation

In this part, we detail the process of recovering 3D coordinates from 2D image detections, propagating errors in the camera coordinate system, and transforming these errors into the world coordinate system. We use $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ to denote 3D rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ to denote 3D translation vector and $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$ to denote 3D transformation matrix.

The camera intrinsic matrix $\mathbf{K}$ is given by:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (6.1)$$

Rcover 3D Object from Image Plane

Given a 2D detection (u, v) and depth d , the coordinates in the world frame (x_w, y_w, z_w) can be recovered using the camera intrinsic matrix $\mathbf{K}$, and the transformation matrix $\mathbf{T}_{wc}$:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \mathbf{T}_{wc} * \mathbf{K}^{-1} \begin{bmatrix} (u + n_u) \cdot (d + n_d) \\ (v + n_v) \cdot (d + n_d) \\ d + n_d \end{bmatrix} + \mathbf{n}_{pose}. \quad (6.2)$$

where n_u, n_v represent the 2D detection noise, n_d is the depth noise, and $\mathbf{n}_{pose}$ is the camera pose error.

The errors in the recovered 3D coordinates can be calculated using the following formulas:

$$\begin{aligned} \sigma_{x_{cam}}^2 &= \left(\frac{d}{f_x}\right)^2 \sigma_u^2 + \left(\frac{u}{f_x}\right)^2 \sigma_d^2, & \sigma_{z_{cam}}^2 &= \sigma_d^2, \\ \sigma_{y_{cam}}^2 &= \left(\frac{d}{f_y}\right)^2 \sigma_v^2 + \left(\frac{v}{f_y}\right)^2 \sigma_d^2. \end{aligned} \quad (6.3)$$

The covariance matrix of the object position in the world coordinates can be calculated as:

$$\Sigma_{world} = \mathbf{R} \Sigma_{cam} \mathbf{R}^T + \Sigma_{T_{wc}}, \quad (6.4)$$

where Σ_{cam} is the covariance matrix of the object position in the camera coordinates, and $\Sigma_{T_{wc}}$ is the covariance matrix of the transformation T_{wc} .

2D Detection Accuracy

Common detection or segmentation algorithms require objects to occupy similar pixel areas in the image; otherwise, errors are likely to occur. Taking YOLO [156] as an example, the difficulties in detecting small objects at a distance are as follows: i) Resolution Limitation: YOLO divides the input image into a grid, with each grid cell responsible for detecting one object. If the object is too small, it may fall within a single grid cell, making it difficult for the detector to accurately locate and classify it. ii) Feature Extraction: YOLO uses Convolutional Neural Network (CNN) to extract image features. The features

of small objects may be lost or become less prominent during downsampling, leading to decreased detection accuracy. Therefore, when objects are at a greater distance, larger objects are detected with higher accuracy, while smaller objects are more prone to errors.

According to the law of large numbers, the average of a large number of independent random samples converges to the true expected value (mean) if it exists. In the context of depth estimation: Consider a bounding box containing pixels from the same object with similar depth. Each pixel's depth estimate can be viewed as an independent random sample. If we average the depth estimates from all these pixels, we can think of it as an approximation of the true depth. As the number of pixels (samples) increases, the average depth estimate tends to converge to the actual depth (if other factors are controlled). While for small objects, fewer pixels mean less reliable statistical sampling. Individual measurement errors have a larger impact on the overall depth estimate. The law of large numbers still applies, but with fewer samples, the convergence may be slower or less reliable.

As the number of pixels (samples) increases, the average depth estimate tends to converge to the actual depth. For small objects with fewer pixels, the individual measurement errors have a larger impact on the overall depth estimate. Assume a bounding box contains n pixels, where the depth estimate for each pixel i is Z_i , and the standard deviation of the depth estimate is σ_{Z_i} . The average depth estimate over the n pixels is: $\bar{Z} = \frac{1}{n} \sum_{i=1}^n Z_i$. According to the law of large numbers, as n increases, $\bar{Z}$ converges to the true depth Z . This is because the standard deviation of the average depth estimate decreases as $\frac{1}{\sqrt{n}}$, we have:

$$\sigma_{\bar{Z}} = \frac{1}{\sqrt{n}} \sqrt{\frac{1}{n-1} \sum_{i=1}^n \sigma_{Z_i}^2}. \quad (6.5)$$

For the case of small objects with fewer pixels, we have a smaller n . Moreover, since Z is larger for distant objects, the depth standard deviation σ_Z will be higher. With fewer pixels (n smaller), the average depth estimate $\bar{Z}$ will have a larger variance $\sigma_{\bar{Z}}$, and thus the individual measurement errors have a more significant impact on the overall depth estimate.

6.2.2 Depth Estimation Accuracy for Stereo Camera

There are many factors that affect depth error, including object distance, surface/medium characteristics, sensor technology [157], and other parameters. When the sensor combina-

tion is fixed, and both the detected object and the working environment are determined, the main influencing factor is the distance between the camera and the target. Under such situations, errors are more likely to occur when the object is either too far from the camera or too small in size. For distant targets or small objects with a small number of pixels, the main reasons for inaccurate depth estimation include detail loss, matching difficulty, and the impact of noise.

- **Detail Loss:** Fewer pixels mean lower image resolution, resulting in the loss of detailed information in the image. This reduction makes it difficult for algorithms to capture the edges and textures of objects, thereby affecting the accuracy of depth estimation.
- **Matching Difficulty:** Depth estimation usually relies on feature point matching in the image. Fewer pixels lead to a reduced number of feature points, thereby increasing the difficulty of matching and subsequently affecting the accuracy of depth estimation.
- **Noise Impact:** As pixel-per-meter decreases, meaning each pixel corresponds to a larger actual spatial range, the noise accumulated within that space also increases, resulting in a lower signal-to-noise ratio. Low-resolution images are more susceptible to noise, which can interfere with the normal operation of depth estimation algorithms, leading to inaccurate estimation results.

Taking the stereo camera as an example, based on the disparity-depth model, Mallick *et. al* [157] calculated the depth from disparity and determined the accuracy and resolution of depth. Using similar triangles, the depth $Z(\rho)$ of an object can be described by:

$$Z(\rho) = \left(\frac{m}{fb} \rho + \left(\frac{1}{Z_r} + \frac{n}{fb} \right) \right)^{-1}, \quad (6.6)$$

where m and n are the parameters of linear normalization, f is camera focal length, b is the stereo baseline length, and ρ is the normalized disparity [157]. To determine the standard deviation σ_Z , the derivative of $Z(\rho)$ is initially computed with respect to ρ :

$$\begin{aligned} \frac{dZ}{d\rho} &= \frac{d}{d\rho} \left(\left(\frac{m}{fb} \right) \rho + \left(\frac{1}{Z_r} + \frac{n}{fb} \right) \right)^{-1} \\ &= - \left(\frac{m}{fb} \right) \left(\left(\frac{m}{fb} \right) \rho + \left(\frac{1}{Z_r} + \frac{n}{fb} \right) \right)^{-2}. \end{aligned} \quad (6.7)$$

Upon substituting $Z(\rho)$ into (6.7), we obtain $\frac{dZ}{d\rho} = - \left(\frac{m}{fb} \right) Z^{-2}$. The standard deviation

σ_Z is related to the standard deviation of the normalized disparity σ_ρ can be given by:

$$\sigma_Z = \left| \frac{dZ}{d\rho} \right| \sigma_\rho = \left| -\frac{\left(\frac{m}{fb}\right)}{Z^2} \right| \sigma_\rho = \left(\frac{m}{fb}\right) \frac{\sigma_\rho}{Z^2}. \quad (6.8)$$

From the above expression, the error of depth measurement is proportional to the square of the depth. This has been experimentally verified in [157].

Assuming the errors in the estimation of X and Y coordinates are independent, one has:

$$\sigma_X = \left(\frac{mx}{f^2b}\right) Z^2 \sigma_\rho \quad \text{and} \quad \sigma_Y = \left(\frac{my}{f^2b}\right) Z^2 \sigma_\rho, \quad (6.9)$$

where x and y are the coordinates of the pixel in the depth image, and σ_X and σ_Y are the errors in X and Y , respectively. Both the depth accuracy and resolution decrease quadratically with increasing depth from the sensor.

6.2.3 Depth Estimation Accuracy for LiDAR Camera Fusion

The LiDAR measures a point in 3D space using a bearing direction $\boldsymbol{\omega}_i \in \mathbb{S}^2$ and depth d_i . The bearing direction in terms of azimuth θ and elevation ϕ angles, the bearing direction $\boldsymbol{\omega}_i$ is parameterized as:

$$\boldsymbol{\omega}_i = \begin{bmatrix} \cos \theta \cos \phi & \sin \theta \cos \phi & \sin \phi \end{bmatrix}. \quad (6.10)$$

The measured point in the LiDAR frame is denoted as $\mathbf{P}_i = d_i \boldsymbol{\omega}_i$. The uncertainty in the LiDAR measurement arises from the following two sources:

- **Bearing direction noise:** $\boldsymbol{\delta}_{\boldsymbol{\omega}_i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}_i})$ in the tangent plane of $\boldsymbol{\omega}_i$
- **Depth noise:** $\delta_{d_i} \sim \mathcal{N}(0, \Sigma_{d_i})$ in the depth measurement

The true point $\mathbf{P}_i^{\text{gt}}$ relates to the measured point as:

$$\mathbf{P}_i^{\text{gt}} = (d_i + \delta_{d_i})(\boldsymbol{\omega}_i \oplus_{\mathbb{S}^2} \boldsymbol{\delta}_{\boldsymbol{\omega}_i}) \approx d_i \boldsymbol{\omega}_i + \boldsymbol{\omega}_i \delta_{d_i} - d_i \boldsymbol{\omega}_i^\wedge \mathbf{N}(\boldsymbol{\omega}_i) \boldsymbol{\delta}_{\boldsymbol{\omega}_i}, \quad (6.11)$$

where $\mathbf{N}(\boldsymbol{\omega}_i)$ is an orthonormal basis for the tangent plane, and $\boldsymbol{\omega}_i^\wedge$ is the skew-symmetric matrix of $\boldsymbol{\omega}_i$.

The total noise in $\mathbf{P}_i$ is calculated as:

$$\mathbf{w}_i = \boldsymbol{\omega}_i \delta_{d_i} - d_i \boldsymbol{\omega}_i^\wedge \mathbf{N}(\boldsymbol{\omega}_i) \boldsymbol{\delta}_{\boldsymbol{\omega}_i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{P}_i}), \quad (6.12)$$

with covariance:

$$\boldsymbol{\Sigma}_{\mathbf{P}_i} = \boldsymbol{\omega}_i \boldsymbol{\Sigma}_{d_i} \boldsymbol{\omega}_i^T + d_i^2 \boldsymbol{\omega}_i^\wedge \mathbf{N}(\boldsymbol{\omega}_i) \boldsymbol{\Sigma}_{\boldsymbol{\omega}_i} \mathbf{N}(\boldsymbol{\omega}_i)^T \boldsymbol{\omega}_i^{\wedge\top}. \quad (6.13)$$

The LiDAR point $\mathbf{P}_i$ is transformed to the camera frame using extrinsic calibration $\mathbf{T} = (\mathbf{R}, \mathbf{t})$:

$$\mathbf{P}_i^c = \mathbf{R} \mathbf{P}_i + \mathbf{t}. \quad (6.14)$$

The uncertainty in $\mathbf{P}_i^c$ is calculated by:

$$\boldsymbol{\Sigma}_{\mathbf{P}_i^c} = \mathbf{R} \boldsymbol{\Sigma}_{\mathbf{P}_i} \mathbf{R}^T. \quad (6.15)$$

Then, the point $\mathbf{P}_i^c = (X_i^c, Y_i^c, Z_i^c)$ in the camera frame is projected onto the image plane via the following formula:

$$\mathbf{p}_i = \left(f_x \frac{X_i^c}{Z_i^c} + c_x, f_y \frac{Y_i^c}{Z_i^c} + c_y \right), \quad (6.16)$$

where f_x , and f_y are focal lengths and (c_x, c_y) is the principal point.

The Jacobian of Z_i^c with respect to $\mathbf{P}_i$ is computed by:

$$\frac{\partial Z_i^c}{\partial \mathbf{P}_i} = \mathbf{R}_3, \quad (6.17)$$

where $\mathbf{R}_3$ is the third row of $\mathbf{R}$. The variance of Z_i^c is:

$$\sigma_{Z_i^c}^2 = \mathbf{R}_3 \boldsymbol{\Sigma}_{\mathbf{P}_i} \mathbf{R}_3^T. \quad (6.18)$$

Assuming isotropic bearing noise ($\boldsymbol{\Sigma}_{\boldsymbol{\omega}_i} = \sigma_{\boldsymbol{\omega}}^2 \mathbf{I}_{2 \times 2}$):

$$\boldsymbol{\Sigma}_{\mathbf{P}_i} = \sigma_{d_i}^2 \boldsymbol{\omega}_i \boldsymbol{\omega}_i^T + d_i^2 \sigma_{\boldsymbol{\omega}}^2 (\mathbf{I}_{3 \times 3} - \boldsymbol{\omega}_i \boldsymbol{\omega}_i^T). \quad (6.19)$$

The depth uncertainty becomes:

$$\sigma_{Z_i^c}^2 = \sigma_{d_i}^2 (\mathbf{R}_3 \boldsymbol{\omega}_i)^2 + d_i^2 \sigma_{\boldsymbol{\omega}}^2 (\|\mathbf{R}_3\|^2 - (\mathbf{R}_3 \boldsymbol{\omega}_i)^2). \quad (6.20)$$

Note the term $\sigma_{d_i}^2 (\mathbf{R}_3 \boldsymbol{\omega}_i)^2$ is generally negligible in magnitude, while the dominant contribution comes from $d_i^2 \sigma_{\boldsymbol{\omega}}^2 (\|\mathbf{R}_3\|^2 - (\mathbf{R}_3 \boldsymbol{\omega}_i)^2)$. This predominant term exhibits quadratic

growth trend with respect to the distance d_i , which makes the second term in (6.20) becoming increasingly significant as d_i increases. This observation is well aligned with the depth uncertainty analysis results in 6.2.2, and we will introduce hierarchical scene graph to alleviate the uncertainty introduced by small or distant objects in Section 6.2.5.

6.2.4 Absolute and Relative Errors in Hierarchical Structures

In semantic SLAM, objects are typically simplified to only retain their category, size, and pose information. Thus, each object carries information in the form of $[class, size, T \in \mathbb{R}^{4 \times 4}]$, which is the information we need to extract in metric-SLAM. However, the number of pixels generated by each object in the image is different, as shown in Fig. 6.3, meaning that the amount of information in the image (the source of information) will also be different. We are mainly concerned with the uncertainty of object position and size.

According to (6.3), the accuracy of the position is significantly affected by the uv and $depth$ errors. The uncertainty of uv and $depth$ accuracy is influenced by the size of the object in the image, which is affected by the size and distance of the object. This relationship can be expressed by the following formula:

$$obj_image_size = \frac{obj_size \cdot f}{distance}, \quad (6.21)$$

where f is the camera focal length, which can be obtained from the camera intrinsic matrix $\mathbf{K}$. Regarding uv and $depth$ errors, they are affected by the size of the object in the image. In particular, it is necessary to ensure that the object image size reaches a suitable size. In the following, we analyze the relationship between 2D detection errors, depth errors, and object image size.

The localization error of an object can be divided into absolute error and relative error. The absolute error xyz_error is measured in a unified length unit (m). The relative error is given by $\frac{xyz_error}{size_{obj}}$. A natural assumption in object association is that two objects of the same class cannot overlap in 3D space, so the distance threshold between objects is usually set to the size of the object itself, and the size values are determined through statistical evaluation of object distributions. Large objects have larger distance thresholds, while small objects have smaller distance thresholds. Therefore, it is more reasonable and accurate to first associate the larger objects and then the smaller objects within them.

From another perspective, in the real world, larger objects logically contain (or corre-

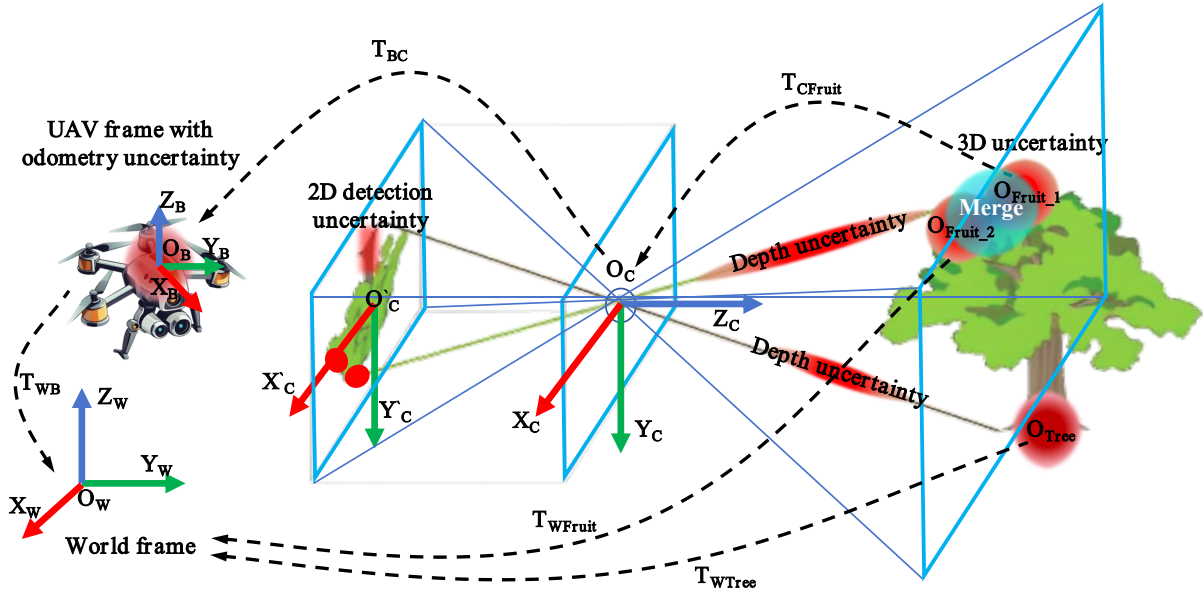


Figure 6.3: Projection model and uncertainty propagation.

spond to) multiple smaller objects (for example, a tree has multiple fruits, and a building has multiple rooms). Smaller objects are less likely to correspond to multiple larger objects (for example, a fruit is not on multiple trees, or a room does not belong to multiple buildings). Therefore, a hierarchical scene graph is a natural organizational structure, allowing the metric-semantic SLAM framework to organize semantic elements at different levels of abstraction. This approach also reduces the memory required for large-scale mapping. More importantly, larger objects are associated with more depth pixels, making the depth values less susceptible to outliers and more robust.

6.2.5 Hierarchical Scene Graph for Fruit Inspection

We use a 3DSG to organize objects at different levels of abstraction. A 3DSG is a layered graph with nodes representing spatial concepts at multiple levels of abstraction (from low-level geometric point clouds to objects, places, and rooms for indoor service robots).

Scene Graph Definition for Plant Inspection

In the context of orchard management, the abstraction level ascends from pure geometric points (devoid of semantic information) to entities such as fruits, with the orchard as the topmost level. While plant stems and leaves should be incorporated into this hierarchical structure, this study simplifies the problem by focusing solely on point clouds,

fruits, plants, and orchards. The relationships between nodes include subordination and adjacency. The formal definition of our 3DSG is presented as follows:

Definition 1. Hierarchical Graph. A graph $G = (V, E)$ is said to be hierarchical if the set of nodes V can be partitioned into l layers, i.e., $V = \bigcup_{i=1}^l V_i$, such that:

1. **Single parent:** Each node $v \in V_i$ at layer i shares an edge with at most one node in the layer V_{i+1} .
2. **Inherent Position Attributes:** Each node $v \in V_i$ at layer i pertains the position attributes from its parent node in the layer V_{i+1} .
3. **Locality:** Each node $v \in V_i$ can only share edges with nodes in the same or adjacent layers, i.e., V_{i-1} , V_i , or V_{i+1} .
4. **Disjoint children:** For any $u, v \in V_i$, the children of u and v , namely $C(u)$ and $C(v)$, are disjoint (i.e., they share no nodes or edges), where

$$C(u) = \{w \in V_{i-1} \mid (w, u) \in E\}, \quad (6.22)$$

denotes the children of u , i.e., the set of nodes in layer V_{i-1} sharing an edge with u . We refer to G as an L -layered hierarchical graph and denote with V_l the set of nodes in layer l . Moreover, we conventionally order the layers from the bottom to the top, i.e., the lowest layer is V_1 , while the top layer is V_L .

For instance, in fruit inspection scenarios, the hierarchical scene graph adheres to the simplified structure defined earlier, with the orchard as the topmost semantic layer that encompasses the entire inspection scope. The layer immediately below the orchard consists of individual plants, which serve as the intermediate carriers connecting the orchard and fruits. The third layer is composed of fruits, the key inspection objects in this context, while the lowest layer corresponds to pure geometric landmarks—specifically the point clouds mentioned previously—without any semantic attributes. In line with the subordination relationship between nodes, each fruit is uniquely subordinate to a single plant (its parent node in the upper layer), and each plant, in turn, belongs to the orchard (the top-level node). This hierarchical structure not only conforms to the actual growth pattern of fruit trees in orchards but also effectively maps the geometric and semantic information of the inspection scene.

Scene Graph Node Generation, Association, and Hierarchical Update for Orchard Object Pose Estimation

Algorithm 2 Add Child to SceneGraph

```
1: if self.children is empty then
2:   add_child()
3: else
4:   for each old_child in self.children do
5:     node_dist = calc_node_dist(old_child, child)
6:     if node_dist < dist_threshold and size_diff < size_threshold then
7:       merge_existing_child_with_child()
8:       return existing_child
9:     else
10:      pass
11:   end if
12: end for
13: add_child()
14: end if
15: update_node_attributes()
16: return child
```

To achieve coherent spatiotemporal tracking of orchard objects, this study proposes an integrated strategy encompassing node generation, association, hierarchical pose estimation, and parent-child coordinated update, coupled with a layer-wise data association mechanism to ensure consistent object representation across consecutive frames. Specifically, the 2D detection process for low-abstraction-level objects is constrained within the bounding boxes of high-abstraction-level entities—for example, fruit detection is limited to the bounding box of its host plant, and the fruit node is then attached to the corresponding plant node via Algorithm 2 (i.e., *add_node(fruit)*) as part of the node generation process. Benefiting from more depth pixels associated with larger objects (high-abstraction-level nodes in the scene graph), their depth estimation exhibits higher robustness against outliers. Thus, the algorithm prioritizes querying high-abstraction-level objects, followed by sequential queries and association of low-abstraction-level ones. Notably, the position update of a parent node triggers synchronous adjustment of its child nodes’ poses, which effectively mitigates matching errors during loop closure.

Algorithm 2 is enhanced with adaptive dual thresholds (3D center distance and size difference) to replace fixed distance thresholds, where the threshold values are dynamically determined by three key factors: (1) the intrinsic physical scale of the object (e.g., canopy width for plants and diameter for fruits), (2) measurement uncertainty induced by sensor noise or detection confidence, and (3) the spatial density of homogeneous objects in the local scene. For illustration, plants are usually sparsely distributed with low pose uncertainty, allowing a relatively loose distance threshold; in contrast, fruits tend to grow in

dense clusters with higher localization ambiguity, demanding stricter thresholds for both position and size. A new detection is only eligible for merging if it meets both threshold criteria and achieves the required confidence—detections with insufficient confidence are directly discarded. If no existing node of the same type satisfies these conditions, a new node is created and inserted into the scene graph at the corresponding hierarchical level.

Once the correspondence between the new detection and an existing node is confirmed (i.e., distance and size differences are smaller than thresholds and confidence requirements are satisfied), the two nodes are merged. The pose parameters (including 3D position and size) are then fused using an Infinite impulse response (IIR) filter to enhance temporal smoothness and suppress sensor noise. This filter assigns higher weights to observations from frames with higher sensor confidence, such as those with denser LiDAR point clouds. To maintain the structural consistency of the scene graph, the pose update of a parent node (e.g., a plant) via merging will drive the coherent adjustment of all its child nodes (e.g., fruits) in the local coordinate frame. This hierarchical, geometry-aware association and update mechanism enables robust tracking of individual fruits and plants across varying viewpoints and time, even under partial occlusion or sensor noise interference.

6.3 Experiment

As analyzed in Section 6.2.1, we conducted experiments using two different sensor configurations: RGB-D and camera-LiDAR fusion. Four different datasets collected in different experimental scenarios were tested. The detailed configurations and parameters for each scenario are presented in Table 6.1. The quantitative evaluation in this Section compares algorithmic fruit counts against GT across multiple fruit varieties. To provide an intuitive measure of counting accuracy, we include bracketed percentages indicating each algorithm’s closeness to GT, where 100% represents perfect alignment with ground truth. Deviations above or below this value reflect performance degradation, with greater differences indicating lower accuracy.

Table 6.1: Experiment Datasets Configuration

Dataset	Sensor	Platform	Scenario	Object	Obj Color	BG Color	Quantity
Gazebo	RGB-D	UAV	Vertical	Fruit	R, Y, P	Oak	50-100
TUM-RGBD	RGB-D	Hand	Indoor	Desktop	Multi color	Gray	50-60
ICUAS’24	Cam-LiDAR	UAV	Orchard	Fake fruits	R, Y	Green	20-30
Hydroponic	Cam-LiDAR	UAV	Lab	Strawberry	Green, red	Green	10-50

For RGB-D datasets, the state-of-art metri-semantic SLAM, OA-SLAM [133], Quadric-SLAM [132], VOOM [158], and 3DSG SLAM, Kimera [24], and Hydra [113] are selected as the main baseline. OA-SLAM, Quadric-SLAM and VOOM use dual quadric as landmarks in SLAM with constraints of detected bounding boxes or instance segmentations, while all three above methods apply flat metric-semantic maps for point cloud landmarks and objects (represented by quadrics) storage, i.e., they are not concerned with estimating higher-level semantics and store all mapping elements at the same level of abstraction. Kimera and Hydra applied 3DSG to organize semantic elements at different levels of abstraction. Their studies, however, mainly considered an ideal simulation environment with accurate semantic segmentation results. It is unrealistic to obtain completely correct and comprehensive semantic segmentation in the real world. Therefore, it remains crucial to improve object mapping accuracy under the influence of noise.

For LiDAR-camera fusion datasets, Kang [159] and Liu's [160] methods are selected as the main benchmarks. Among them, Kang's method is only camera-LiDAR fusion without odometry information, so it can only estimate the fruit position statically in the local frame (the vehicle cannot move, only the manipulator can move to grasp fruits), and cannot calculate the fruit position in the global frame. Liu's method extends Kang's method, but with worse positioning accuracy. Both methods project the LiDAR point cloud onto the image plane to obtain the depth of pixels in the image, and then estimate the 3D position of the fruit based on the depth. In contrast, our sensor fusion method adopts the 3D object detection paradigm commonly used in autonomous driving. We project the object detection bounding box from the image into 3D space, and subsequently filter the point clusters in the 3D space to avoid the influence of noisy point clouds. To address potential calibration challenges from variable extrinsic settings, we employ a map-based fusion strategy where we first pre-build a reference point cloud map using LiDAR odometry, then perform camera-LiDAR fusion exclusively on this pre-built map. This approach enables higher positioning accuracy and more robust performance.

6.3.1 Gazebo Simulation for Fruit Counting

In the Gazebo environment as depicted in Fig. 6.2, various fruits are placed on the vertical plant bed in the garden. Each bed contains one of the three species of plants. Each type of plant has different colors and sizes, namely purple eggplants, red tomatoes, and yellow peppers. A UAV flies in the greenhouse to conduct fruit counting, a depth camera, an IMU are installed on the simulated UAV. The ground truth odometry information is used

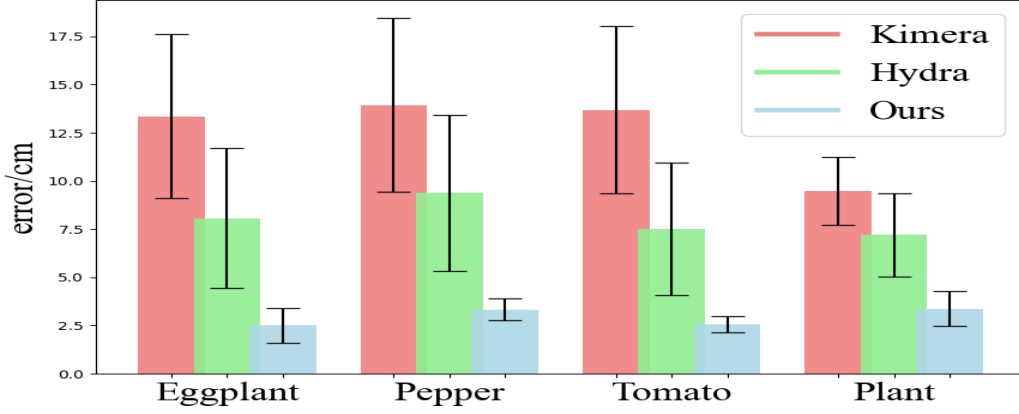


Figure 6.4: Plant and fruit reconstruction position error comparison in centimeters.

to provide the pose information of the UAV.

We conducted a comparative analysis of fruit counting accuracy with Kimera and Hydra frameworks, given their shared experimental configuration utilizing RGB-D camera inputs and ground-truth semantic segmentation annotations. Within an idealized simulation framework, precise semantic segmentation labels can be systematically generated through HSV-based thresholding techniques, effectively bypassing sensor noise artifacts. As shown in Table 6.2, our Gazebo simulation results demonstrate exceptional counting precision: the proposed methodology achieves exact 100% alignment with GT in most test cases, with a single instance at 102% - still well within near-perfect range. This significant performance improvement over comparative methods validates our approach’s effectiveness in fruit counting applications.

Table 6.2: Fruit Counting Accuracy in Gazebo Simulation Setup
Prived by ICUAS’24 Simulation Phase

	Kimera	Hydra	Ours	GT
Eggplant	27 (122%)	24 (109%)	22 (100%)	22
Pepper	49 (114%)	38 (93%)	42 (102%)	41
Tomato	39 (114%)	32 (95%)	34 (100%)	34
Plant	21 (110%)	19 (100%)	19 (100%)	19

It is notable that even with the GT-Trajectory configuration, achieving 100% accuracy for $\%Found$ and $\%Correct$ in the simulation scenes using ground-truth semantics is not realized. There are centimeter-level fruit and plant localization errors in all three methods

as shown in Fig. 6.4. Two primary factors contribute to the imperfect performance of the GT-Trajectory setup. First, Hydra’s object extraction method tends to under-segment nearby objects of the same class, resulting in the predicted object centroid deviating further from the individual ground truth object instances. Second, the underlying metric-semantic reconstruction within the active window (such as Kimera and Voxelblox) can occasionally cause object labels to be projected onto surfaces behind objects, leading to the generation of hallucinated objects. To address these issues, the algorithm design has been refined to be more robust to noise and uncertainty, following the analysis outlined in Section 6.2.1.

6.3.2 TUM RGB-D Dataset

Table 6.3: Object Counting Accuracy in TUM RGB-D Dataset and Diomond Dataset

	OA	QISO	VOOM	Ours	GT
Fr2_desk	119 (270%)	64 (145%)	49 (111%)	46 (105%)	44
Fr2_person	126 (280%)	69 (153%)	56 (124%)	49 (109%)	45
Fr3_office	-	-	66 (108%)	58 (95%)	61
Diamond_1	73 (146%)	59 (118%)	53 (106%)	48 (96%)	50
Diamond_2	74 (172%)	78 (181%)	50 (116%)	47 (109%)	43

Since benchmark methods are designed for service robotic applications rather than being specifically targeted towards fruit mapping and counting, we validate the generalizability of our proposed approach beyond agricultural scenarios through testing on the TUM RGB-D dataset, which primarily contains indoor environments. As discussed in OA-SLAM [133], the mere incorporation of bounding boxes or quadric surface models as map landmarks may not substantially enhance SLAM’s localization accuracy. Consequently, rather than prioritizing localization precision comparisons, our evaluation framework emphasizes assessing and comparing the quantitative reconstruction completeness of objects through semantic-aware metrics in SLAM implementations.

This evaluation by counting object numbers reflects the accuracy of object-level mapping. The closer the number of reconstructions is to the ground truth, the better the performance of data association. Since the TUM RGB-D dataset does not provide ground truth 3D reconstruction or the ground truth counts and labels of objects, we obtain the

ground truth object counts by manual labeling and visual inspection.

Table 6.3 summarizes the results. We compare our algorithm with four object data association methods: OA-SLAM, QISO-SLAM, and VOOM. Among them, OA-SLAM and QISO-SLAM, IoU serves as the primary metric for object association. OA-SLAM restricts associations to objects of the same category, whereas QISO-SLAM associates objects based solely on IoU values, without considering their categories, and VOOM applies Wasserstein distance [158] for object association.

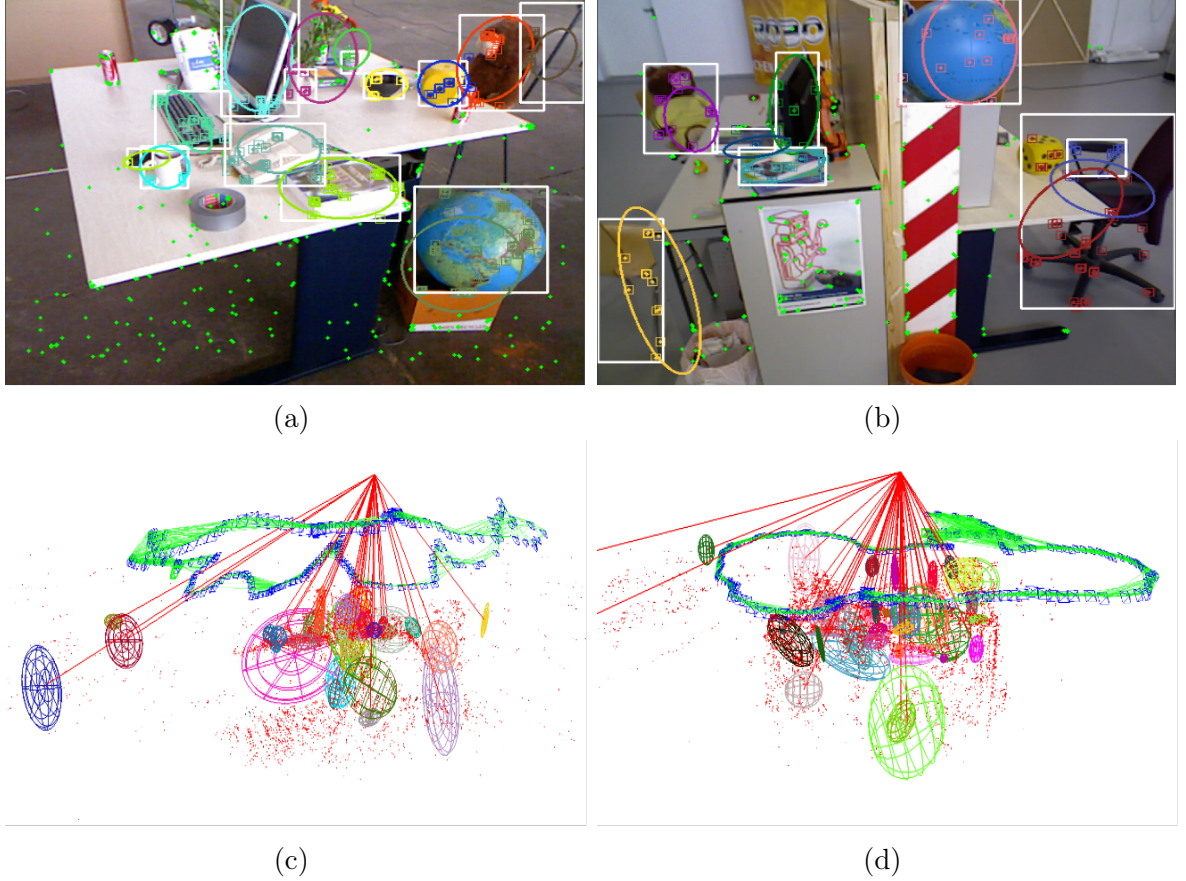


Figure 6.5: TUM RGB-D dataset test results. (a) 2D object detection in fr2_office. (b) 2D object detection in fr3_office. (c) 3DSG in fr2_office. (d) 3DSG in fr3_office.

Comparatively, our method outperforms other algorithms on all sequences by scrutinizing the object’s class, size, and 3D position in the scene graph, allowing for the updating of object poses. Therefore, our method performs well even when dealing with sequences like Fr2_person, where moving persons are present, while VOOM tends to yield a higher number of incorrect counts. The sparse 3DSG reconstruction result of the TUM RGB-D dataset is shown in Fig. 6.5.

The results for OA-SLAM and QISO-SLAM are consistent with those reported in



Figure 6.6: ICUAS'24 dataset collection scenario. (a) The dataset collection orchard environment. (b) The UAV configuration.

Table 6.4: Object Counting Accuracy in ICUAS'24 Dataset

	Liu's(Y)	Ours(Y)	Liu's(R)	Ours(R)	GT(Y R)
Bag 1	21 (117%)	17 (94%)	12 (150%)	8 (100%)	18Y 8R
Bag 2	26 (144%)	21 (117%)	11 (138%)	10 (125%)	18Y 8R
Bag 3	27 (150%)	18 (100%)	9 (113%)	7 (88%)	18Y 8R
Bag 4	29 (126%)	22 (96%)	19 (119%)	14 (88%)	23Y 16R

[158], as the source code for QISO-SLAM remains unavailable. Other results represent the best performance achieved. It can be observed that OA-SLAM exhibits the highest number of redundant objects among all the methods. This outcome is attributed to its utilization of 2D IoU alone for object association, which leads to erroneous object detection, misclassification, and consequently, erroneous generation of 3D objects. And VOOM outperforms QISO-SLAM by incorporating additional criteria for object data association.

6.3.3 ICUAS'24 Competition Dataset

The ICUAS'24 competition is organized by LARICS from the University of Zagreb. Inspired by the scenario of robots in agriculture, the final stage of the competition takes place in an orchard shown in Fig. 6.6 (a). There are two different types of plants in the orchard, with differently colored fruits (yellow and red). A UAV is deployed and

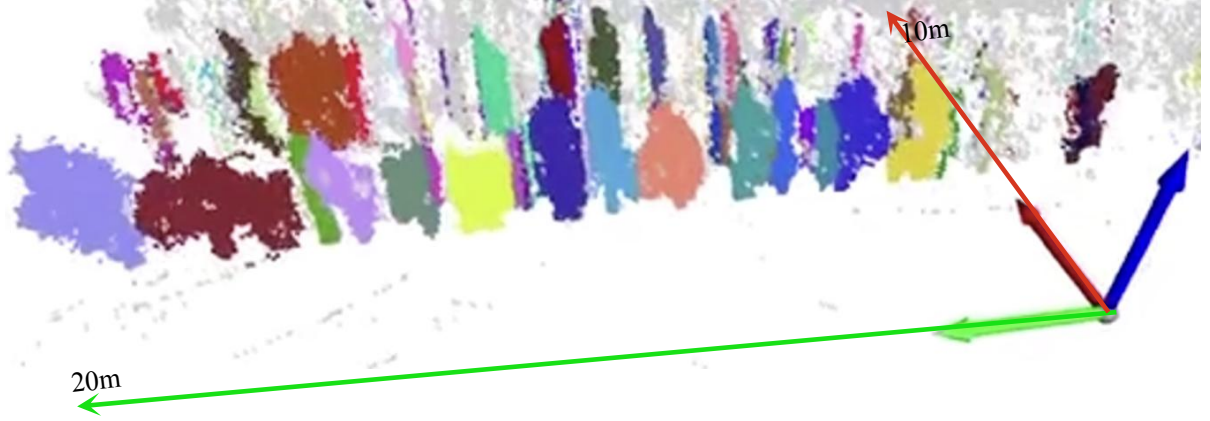


Figure 6.7: Tree Segmentation in the Orchard.

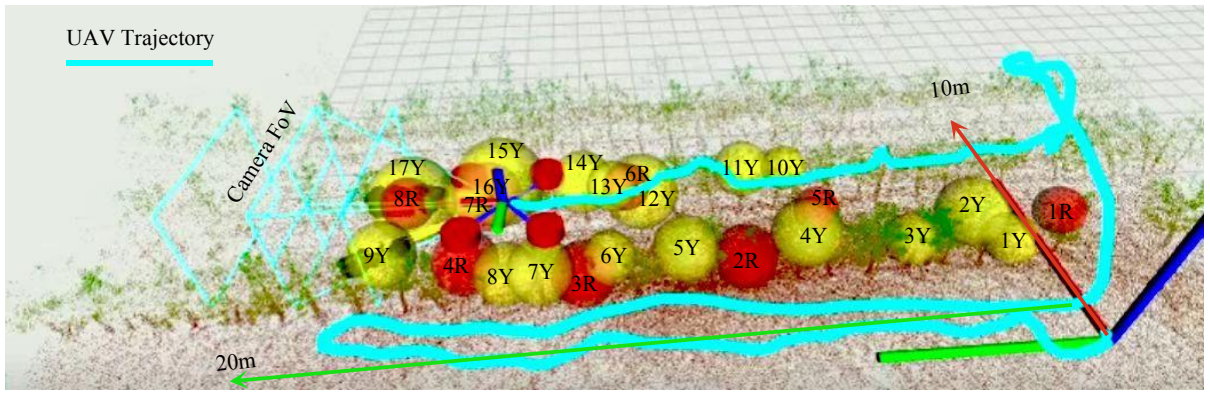


Figure 6.8: The ICUAS'24 orchard mapping results with fruit landmarks and point cloud mapping.

flown manually, both in position and attitude modes, to collect data (including LiDAR point cloud, camera image, GPS signal, and IMU data) for fruit counting. The UAV is equipped with a front-facing camera, slightly pitched-down 3D lidar, GPS receiver, IMU, and Pixhawk controller as shown in Fig. 6.6 (b).

Kang's method and Liu's method primarily focus on fruit localization. However, the ICUAS'24 competition did not provide ground truth 3D fruit locations (in fact, obtaining GT object localization in outdoor, unstructured, and cluttered environments like orchards is very challenging). Therefore, we only compared our method with Liu's method for fruit counting accuracy. Kang's method uses a Livox LiDAR (a solid-state LiDAR with a non-repetitive scan pattern) to accumulate point clouds from one static pose and project them onto the image plane to estimate pixel depth. This method is only suitable for static platforms or robots that need to stay in the same position for a long time, which significantly reduces the robot's efficiency (both in detection and fruit harvesting). Consequently, it is not suitable for mobile robots, especially UAVs. Liu's method builds on

Kang’s method by adopting a sliding window to accumulate point clouds from the LiDAR and then projecting the LiDAR point clouds onto the image plane.

However, due to parallax and occlusion, in an actual orchard, the background depth may be projected onto the same pixels as the foreground depth, which introduces significant errors. Our method directly selects candidate point cloud clusters in 3D space and then probabilistically selects the fruit depth. By using IIR filters and hierarchical scene graphs to select and update the fruit candidate poses, our approach effectively avoids a large number of erroneous estimates. The tree segmentation in the point cloud map is carried out by using the DB-Scan algorithm, as shown in Fig. 6.7.

Since Liu’s algorithm is not open-source, we re-implemented their methods based on the descriptions provided in their paper and tested them on the ICUAS’24 dataset. In this dataset, the UAV flies over two rows of orchards, and the fruit mapping results can be seen in Fig. 6.8. In our method, all fruits are well aligned with the trees, while in Liu’s method, due to background depth noise, many fruits erroneously fall into the empty area between the two rows of orchards. Quantitative results are shown in Table 6.4, where it can be seen that our method achieves higher accuracy compared to Liu’s method. However, some errors persist, primarily due to the cluttered environment and the inherent challenges associated with fruit counting in complex agricultural scenes.

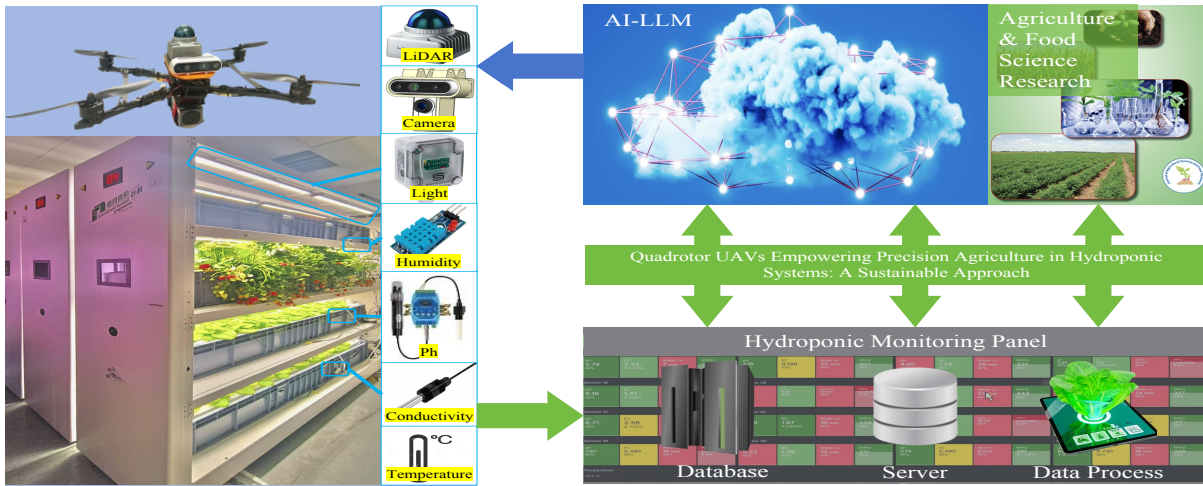


Figure 6.9: System architecture of the intelligent hydroponic laboratory with embedded sensors and UAV.

6.3.4 Hydroponic Laboratory Test for Strawberry Mapping

We also conducted indoor strawberry hydroponic cultivation at the Hong Kong Polytechnic University and performed UAV flight tests in the vertical hydroponics farm lab to collect data and test our fruit counting system. As shown in Fig. 6.9, our hydroponic laboratory is an indoor controlled environment, capable of adjusting growth conditions through various equipment (including temperature, humidity, nutrient supply, CO_2 concentration, and lighting), thereby providing a tailored and stable growth environment for crops. On one hand, indoor cultivation is not limited by climate, allowing crops that are typically grown in winter to be cultivated in summer, thus ensuring continuous and stable year-round production. On the other hand, due to the scarcity of arable land in Hong Kong, indoor controlled hydroponics offers more flexible site selection compared to traditional outdoor greenhouse hydroponics, since they can be set up in factory premises, containers, and other locations, thereby minimizing environmental impact.



Figure 6.10: Ripe and unripe strawberry detection by YOLOv8.

Our indoor controlled hydroponic platform (maintained at a temperature of $22 \pm 1^\circ\text{C}$ and relative humidity of 50-60%) currently has 32 intelligent hydroponic units that can control light intensity, spectrum, and water circulation for research purposes. The hydroponic platform already has many sensors to monitor the plant growth environment, but most of them are for monitoring temperature, humidity, and nutrient composition, lacking sensors for monitoring plant growth status. This limitation is partly due to the structural complexity of the hydroponic platform, which makes the installation of cameras and other sensors challenging, and partly because densely installing cameras on a large-scale hydroponic platform would significantly increase costs. By equipping a UAV with an RGB-D camera and a Livox MID-360 LiDAR, a mobile data collection platform for

crop monitoring is established, substantially reducing sensor installation costs. The sensor setup of the UAV could be found at [14]. In the future, manipulators can be installed on UAVs for active operations such as pollination, pruning, and harvesting. First, we

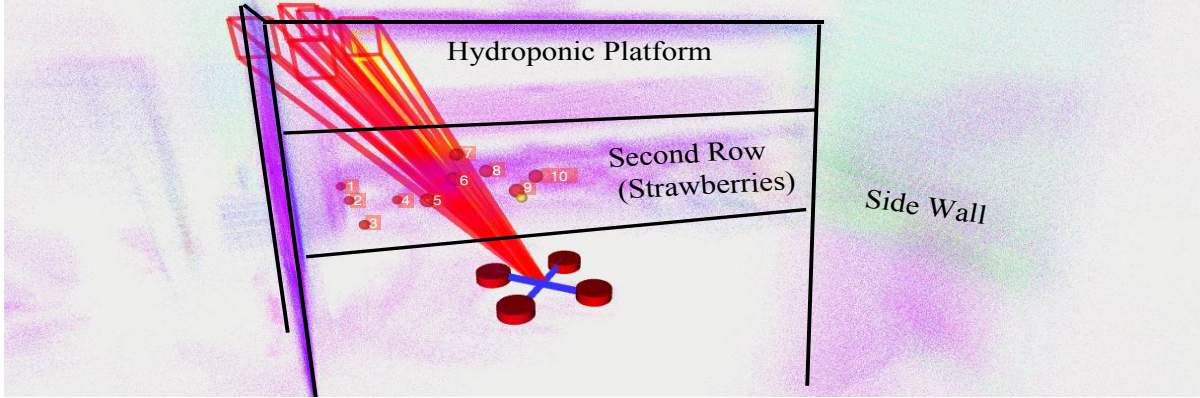


Figure 6.11: 3D strawberry mapping in the hydroponic platform .

collected over 1200 images for strawberry annotation and classification, only annotating strawberries with fully fallen petals. Based on color (green and red), the strawberries were classified into two categories: ripe and unripe. Unripe strawberries are difficult to be accurately identified, and the training process may incorrectly identify ripe strawberries as unripe. In contrast, fully ripe strawberries are relatively easier to be identified correctly. Additionally, manual selections may result in different classifications due to varying levels of experience, posing further challenges for data labeling. In this study, strawberries exhibiting more than 60% red surface area were classified as ripe, while those with less than 60% red area were classified as unripe. The annotated dataset was subsequently used to train a YOLOv8 network for real-time strawberry detection. The detection results are shown in Fig. 6.10. The detected bounding boxes are filtered and then input into the hierarchical scene graph for 3D semantic mapping.

Table 6.5: Strawberry Counting Accuracy in hydroponic Platform

	Liu's [160]	Kang's [159]	Ours	GT
Sequence 1	15 (150%)	9 (90%)	10 (100%)	10
Sequence 2	29 (132%)	17 (77%)	26 (118%)	22
Sequence 3	52 (141%)	29 (78%)	45 (122%)	37

The final mapping of strawberries in the indoor hydroponic platform is presented in Fig. 6.11. It can be seen in the figure that most ripe strawberries were detected and

correctly mapped in 3D space. Since unripe strawberry detection is more challenging, only ripe strawberries are counted for the final comparison. The strawberry counting results are shown in Table 6.5. Accurately detecting unripe strawberries will be our future work direction. For quantitative assessment, ground-truth positions of strawberries were manually measured, with the hydroponic shelf designated as the coordinate origin. In dynamic scenarios, our method achieves 2 cm localization error, which is better than Liu’s method as shown in Fig. 6.12. Based on the above analysis, the proposed algorithm helps to achieve superior accuracy, clearly demonstrating the effectiveness of our approach.

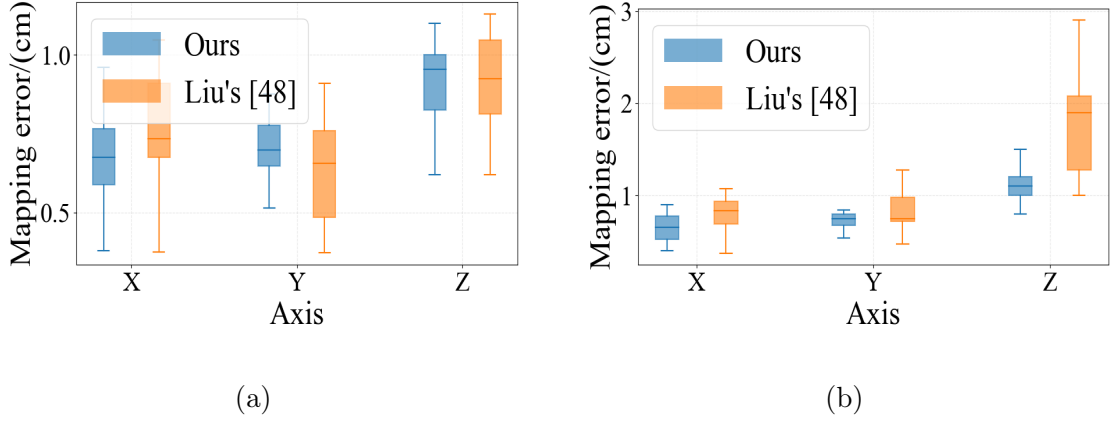


Figure 6.12: Hydroponic platform strawberry 3D mapping accuracy comparison, (a) static mapping accuracy, (b) dynamic mapping accuracy.

6.4 Conclusion

This work, grounded in fruit counting, introduces a hierarchical metric-semantic SLAM system based on 3DSG for object mapping and counting. By leveraging a hierarchical scene-graph structure, the system mitigates mapping errors and significantly enhances fruit counting accuracy. Our work focuses on two key aspects: i), reducing errors in each stage (2D detection, depth estimation, robot odometry), and ii), leveraging a hierarchical structure in the real world to minimize object mapping errors through a top-down data association paradigm. Experimental results across four different scenarios demonstrate both the mapping accuracy of 3D objects and the generality of the proposed algorithm. Experimental evaluations reveal context-dependent performance: In outdoor orchards (Section 6.3.3): Our method reduced fruit localization errors by 40% compared to Liu’s [160] method through 3D clustering, but remained sensitive to extreme foliage density.

In indoor vertical farms (Section 6.3.4): Our method achieved 92% strawberry counting accuracy (dynamic flights) by leveraging hierarchical constraints to overcome occlusion.

Chapter 7

Conclusion and Future Work

In this study, we have introduced an autonomous infrastructure inspection system leveraging UAVs. The emergence of UAV technology has revolutionized infrastructure inspection, offering effective and innovative solutions. Our system combines LiDAR scanning for 3D model construction with cameras for defect detection, addressing critical challenges in the field.

System Overview: Our autonomous infrastructure inspection system encompasses various modules, including control, planning, perception, localization, and mapping. These components work seamlessly together to create a comprehensive solution. The core focus of my work lies in integrating these modules to develop a complete autonomous inspection system suitable for real-world scenarios. By utilizing UAVs, we achieve flexibility, scalability, and cost-effectiveness in infrastructure inspection.

Beyond system synthesis, we have also developed four critical sub-modules:

- Autonomous targetless calibration kit.
- FLiVIS.
- LiDAR BA with uncertainty.
- 3DSG.

The autonomous Targetless Calibration Kit tackles the extrinsic calibration of LiDAR-RGB and LiDAR-thermal sensors. Accurate calibration ensures reliable data fusion and accurate 3D reconstruction. The autonomous nature of this calibration process eliminates the need for manual target placement, streamlining the overall inspection workflow. FLiVIS is a critical component of our system, combining LiDAR, visual, and inertial information for precise localization and mapping. it comprises two subsystems: VIO

Subsystem and LIO Subsystem: Leverages LiDAR measurements for odometry. These subsystems are interconnected through complementary filters, enhancing robustness and accuracy. FLiVIS not only serves as an odometry system for UAVs but also contributes to environmental mapping, bridging the gap between simulation and real-world experiments. LiDAR BA and 3DSG are introduced to improve inspection tasks. LiDAR BA produces a more accurate point cloud map, while 3DSG builds upon this foundational level by providing an organizational framework for semantic information, thereby supplying well-structured semantic data for downstream tasks such as task planning.

Through extensive testing, our proposed system has successfully demonstrated the detection and localization of various types of cracks. As we transition from the controlled laboratory environment to real-world outdoor inspections, we aim to expand our scope beyond bridges to include diverse targets such as buildings, bridges, and fruits.

Currently, Chapters 5 and 6 of our paper focus on the scenarios of bridge inspection and plant inspection, respectively. However, the algorithms in these two sections are not mutually adaptable. This is because LiDAR BA yields suboptimal results in unstructured environments, while the construction logic of the semantic scene graph in bridge scenarios is more complex and less straightforward to establish compared to the relatively simple structure of trees and fruits. In the future, we aim to further strengthen the connection between Chapters 5 and 6, and develop a unified framework for geometric and semantic mapping that can be applied across different environments to support task-level robotic operations.

Bibliography

- [1] L. Carlone, A. Kim, T. Barfoot, D. Cremers, and F. Dellaert, “Slam handbook: From localization and mapping to spatial intelligence,” 2025.
- [2] X. L. Liu, Zheng and F. Zhang, “Efficient and consistent bundle adjustment on lidar point clouds,” *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4366–4386, 2023.
- [3] O. UAS, “Oregon uas innovation challenge,” <https://oregonuas.org/pages/oregon-uas-innovation-challenge>, 2024, accessed: Dec. 10, 2024.
- [4] D. N. Enterprise, “Drone technology has changed the world as we know it: Are you ready to embrace it?” <https://enterprise.dronenerds.com/blog/drone-knowledge/drone-technology-has-changed-the-world-as-we-know-it-are-you-ready-to-embrace-it/>, 2024, accessed: Dec. 10, 2024.
- [5] T. U. Expo, “Drone powerline inspection: What to consider and how,” <https://www.theutilityexpo.com/news/drone-powerline-inspection-what-to-consider-and-ho>, 2024, accessed: Dec. 10, 2024.
- [6] T. B. R. Company, “Inspection drone global market report,” <https://www.thebusinessresearchcompany.com/report/inspection-drone-global-market-report>, 2024, accessed: Dec. 10, 2024.
- [7] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Deniston, S.-P. Deschênes, K. Harlow, S. Khattak *et al.*, “Present and future of slam in extreme environments: The darpa subT challenge,” *IEEE Trans. Robot.*, 2023.
- [8] S. Chen, Y. Feng, C.-Y. Wen, Y. Zou, and W. Chen, “Stereo visual inertial pose estimation based on feedforward and feedbacks,” *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 6, pp. 3562–3572, 2023.
- [9] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-lio2: Fast direct lidar-inertial odometry,” *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, 2022.

- [10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [11] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint," *Int. J. Robot. Res.*, vol. 41, no. 3, pp. 270–280, 2022.
- [12] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield *et al.*, "The grand challenges of science robotics," *Sci. Robot.*, vol. 3, no. 14, p. eaar7650, 2018.
- [13] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [14] W. Yang, H. Luo, K.-W. Tse, H. Hu, K. Liu, B. Li, and C.-Y. Wen, "Autonomous targetless extrinsic calibration of thermal, rgb, and lidar sensors," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024.
- [15] K. Liu, P. Yang, R. Wang, L. Jiao, T. Li, and J. Zhang, "Observer-based adaptive fuzzy finite-time attitude control for quadrotor uavs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 6, pp. 8637–8654, 2023.
- [16] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 478–485, 2021.
- [17] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, 2022.
- [18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," in *2023 IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3992–4003.
- [19] UN Secretary-General, "Progress towards the sustainable development goals: Report of the secretary-general," New York, NY, USA, 2024. [Online]. Available: <https://un.org/sdgs/reports>
- [20] H. Ibraiwish, M. W. Eltokhey, and M.-S. Alouini, "Uav-assisted vlc using led-based grow lights in precision agriculture systems," *IEEE Internet of Things Mag.*, vol. 7, no. 3, pp. 100–105, 2024.

-
- [21] K. Yu, Z. Feng, J. Yu, T. Chen, J. Peng, and D. Li, "Secure ultra-reliable and low latency communication in uav-enabled noma wireless networks," *IEEE Trans. Veh. Technol.*, 2024.
- [22] F. Pan, Z. Pang, M. Luvisotto, M. Xiao, and H. Wen, "Physical-layer security for industrial wireless control systems: Basics and future directions," *IEEE Ind. Electron. Mag.*, vol. 12, no. 4, pp. 18–27, 2018.
- [23] R. Lu, R. Bai, R. Li, L. Zhu, M. Sun, F. Xiao, D. Wang, H. Wu, and Y. Ding, "A novel sequence-to-sequence-based deep learning model for multistep load forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 638–652, 2024.
- [24] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *Int. J. Robot. Res.*, vol. 40, no. 12-14, pp. 1510–1546, 2021.
- [25] K. Wang, T. Zhou, X. Li, and F. Ren, "Performance and challenges of 3d object detection methods in complex scenes for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1699–1716, 2022.
- [26] A. Khurana and K. Nagla, "Extrinsic calibration methods for laser range finder and camera: A systematic review," *Mapan*, vol. 36, no. 3, pp. 669–690, 2021.
- [27] X. Li, Y. Xiao, B. Wang, H. Ren, Y. Zhang, and J. Ji, "Automatic targetless lidar-camera calibration: A survey," *Artif. Intell. Rev.*, pp. 1–39, 2022.
- [28] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [29] J.-Y. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/, 2004.
- [30] M. Rufli, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images," in *2008 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2008, pp. 3121–3126.
- [31] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, 2020.

- [32] J. Maye, P. Furgale, and R. Siegwart, “Self-supervised calibration for robotic systems,” in *Proc. IEEE Intell. Veh. Symp.*, 2013, pp. 473–480.
- [33] R. Unnikrishnan and M. Hebert, “Fast extrinsic calibration of a laser rangefinder to a camera,” *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*, 2005.
- [34] J. Beltrán, C. Guindel, F. García *et al.*, “Automatic extrinsic calibration method for lidar and camera sensor setups,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17 677–17 689, 2022.
- [35] Z. Pusztai and L. Hajder, “Accurate calibration of lidar-camera systems using ordinary boxes,” in *Proceedings of the IEEE Int. Conf. on Computer Vision Workshops*, 2017, pp. 394–402.
- [36] C. Yuan, X. Liu, X. Hong, and F. Zhang, “Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7517–7524, 2021.
- [37] Y. Lin, V. Larsson, M. Geppert, Z. Kukelova, M. Pollefeys, and T. Sattler, “Infrastructure-based multi-camera calibration using radial projections,” in *Eur. Conf. Comput. Vis.*, 2020, pp. 327–344.
- [38] L. Zhou, Z. Li, and M. Kaess, “Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences,” in *2018 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2018, pp. 5562–5569.
- [39] T. Tóth, Z. Pusztai, and L. Hajder, “Automatic lidar-camera calibration of extrinsic parameters using a spherical target,” in *2020 IEEE Int. Conf. Robot. Autom.* IEEE, 2020, pp. 8580–8586.
- [40] J. Levinson and S. Thrun, “Automatic online calibration of cameras and lasers.” in *Robotics: Science and Systems*, vol. 2. Citeseer, 2013, p. 7.
- [41] G. Pandey, J. McBride, S. Savarese, and R. Eustice, “Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 26, no. 1, 2012, pp. 2053–2059.
- [42] M. R. U. Saputra, C. X. Lu, P. P. B. de Gusmao, B. Wang, A. Markham, and N. Trigoni, “Graph-based thermal-inertial slam with probabilistic neural networks,” *IEEE Trans. Robot.*, 2021.

-
- [43] S. Prakash, P. Y. Lee, T. Caelli, and T. Raupach, “Robust thermal camera calibration and 3d mapping of object surface temperatures,” in *Thermosense XXVIII*, vol. 6205. SPIE, 2006, pp. 182–189.
 - [44] S. Vidas, P. Moghadam, and M. Bosse, “3d thermal mapping of building interiors using an rgb-d and thermal camera,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2311–2318.
 - [45] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark, “A mask-based approach for the geometric calibration of thermal-infrared cameras,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 6, pp. 1625–1635, 2012.
 - [46] J. T. Lussier and S. Thrun, “Automatic calibration of rgb-d and thermal cameras,” in *2014 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2014, pp. 451–458.
 - [47] T. Fu, H. Yu, W. Yang, Y. Hu, and S. Scherer, “Targetless extrinsic calibration of stereo, thermal, and laser sensors in structured environments,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.
 - [48] J. Zhang, P. Siritanawan, Y. Yue, C. Yang, M. Wen, and D. Wang, “A two-step method for extrinsic calibration between a sparse 3d lidar and a thermal camera,” in *2018 15th IEEE Int. Conf. Control, Autom., Robot. Vis.* IEEE, 2018, pp. 1039–1044.
 - [49] J. Zhang, Y. Liu, M. Wen, Y. Yue, H. Zhang, and D. Wang, “L2v2t2 calib: Automatic and unified extrinsic calibration toolbox for different 3d lidar, visual camera and thermal camera,” in *Proc. IEEE Intell. Veh. Symp.*, 2023, pp. 1–7.
 - [50] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
 - [51] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. J. Berles, “S-ptam: Stereo parallel tracking and mapping,” *Robotics and Autonomous Systems*, vol. 93, pp. 27–42, 2017.
 - [52] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam,” *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

- [53] R. Wang, M. Schworer, and D. Cremers, “Stereo dso: Large-scale direct sparse visual odometry with stereo cameras,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3903–3911.
- [54] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *2014 IEEE Int. Conf. Robot. Autom.*. IEEE, 2014, pp. 15–22.
- [55] R. Mur-Artal and J. D. Tardós, “Probabilistic semi-dense mapping from highly accurate feature-based monocular slam.” in *Robotics: Science and Systems*, vol. 2015. Rome, 2015.
- [56] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *Eur. Conf. Comput. Vis.*. Springer, 2014, pp. 834–849.
- [57] C. Fu, A. Carrio, and P. Campoy, “Efficient visual odometry and mapping for unmanned aerial vehicle using arm-based stereo vision pre-processing system,” in *2015 Int. Conf. Unmanned Aircraft Syst.*. IEEE, 2015, pp. 957–962.
- [58] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [59] P. Geneva, K. Eickenhoff, W. Lee, Y. Yang, and G. Huang, “Openvins: A research platform for visual-inertial estimation,” in *Proc. IEEE Int. Conf. Robot. Autom.*. IEEE, 2020, pp. 4666–4672.
- [60] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *Proceedings 2007 IEEE Int. Conf. Robot. Autom.*. IEEE, 2007, pp. 3565–3572.
- [61] Z. Liu, F. Zhang, and X. Hong, “Low-cost retina-like robotic lidars based on incommensurable scanning,” *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 1, pp. 58–68, 2021.
- [62] R. Milijas, L. Markovic, A. Ivanovic, F. Petric, and S. Bogdan, “A comparison of lidar-based slam systems for control of unmanned aerial vehicles,” in *2021 Int. Conf. Unmanned Aircraft Syst.*. IEEE, 2021, pp. 1148–1154.
- [63] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time.” in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [64] A. Censi, “An icp variant using a point-to-line metric,” in *2008 IEEE Int. Conf. Robot. Autom.*. Ieee, 2008, pp. 19–25.

-
- [65] J. W. Kim and B. H. Lee, “Robust and fast 3-d scan registration using normal distributions transform with supervoxel segmentation,” *Robotica*, vol. 34, no. 7, pp. 1630–1658, 2016.
 - [66] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, “Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, 2022.
 - [67] H. Guo, J. Zhu, and Y. Chen, “E-loam: Lidar odometry and mapping with expanded local structural information,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1911–1921, 2022.
 - [68] J. Lin and F. Zhang, “Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov,” in *2020 IEEE Int. Conf. Robot. Autom.* IEEE, 2020, pp. 3126–3131.
 - [69] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *2020 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2020, pp. 5135–5142.
 - [70] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, “Lins: A lidar-inertial state estimator for robust and efficient navigation,” in *2020 IEEE Int. Conf. Robot. Autom.* IEEE, 2020, pp. 8899–8906.
 - [71] H. Ye, Y. Chen, and M. Liu, “Tightly coupled 3d lidar inertial odometry and mapping,” in *2019 IEEE Int. Conf. Robot. Autom.* IEEE, 2019, pp. 3144–3150.
 - [72] J. Lin, C. Zheng, W. Xu, and F. Zhang, “R2live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7469–7476, 2021.
 - [73] J. Graeter, A. Wilczynski, and M. Lauer, “Limo: Lidar-monocular visual odometry,” in *2018 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2018, pp. 7872–7879.
 - [74] T. Shan, B. Englot, C. Ratti, and D. Rus, “Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping,” in *2021 IEEE Int. Conf. Robot. Autom.* IEEE, 2021, pp. 5692–5698.
 - [75] T. Qin, J. Pan, S. Cao, and S. Shen, “A general optimization-based framework for local odometry estimation with multiple sensors,” *arXiv preprint arXiv:1901.03638*, 2019.

- [76] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, “Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry,” in *2022 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2022, pp. 4003–4009.
- [77] M. Kok, J. D. Hol, and T. B. Schön, “Using inertial sensors for position and orientation estimation,” *arXiv preprint arXiv:1704.06053*, 2017.
- [78] I.-H. Kim, S. Yoon, J. H. Lee, S. Jung, S. Cho, and H.-J. Jung, “A comparative study of bridge inspection and condition assessment between manpower and a uas,” *Drones*, vol. 6, no. 11, p. 355, 2022.
- [79] S. Jung, S. Song, P. Youn, and H. Myung, “Multi-layer coverage path planner for autonomous structural inspection of high-rise structures,” in *2018 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2018, pp. 1–9.
- [80] S. Jung, D. Choi, S. Song, and H. Myung, “Bridge inspection using unmanned aerial vehicle based on hg-slam: Hierarchical graph-based slam,” *Remote Sensing*, vol. 12, no. 18, p. 3022, 2020.
- [81] Q. Li, G. Yang, C. Gao, Y. Huang, J. Zhang, D. Huang, B. Zhao, X. Chen, and B. M. Chen, “Single drone-based 3d reconstruction approach to improve public engagement in conservation of heritage buildings: A case of hakka tulou,” *Journal of Building Engineering*, p. 108954, 2024.
- [82] M. Oelsch, M. Karimi, and E. Steinbach, “R-loam: Improving lidar odometry and mapping with point-to-mesh features of a known 3d reference object,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2068–2075, 2021.
- [83] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-liv2: Fast direct lidar-inertial odometry,” *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [84] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, others, and L. Carlone, “Present and future of slam in extreme environments: The darpa sub challenge,” *IEEE Trans. Robot.*, vol. 40, pp. 936–959, 2023.
- [85] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, “Observability-based rules for designing consistent ekf slam estimators,” *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 502–528, 2010.
- [86] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “isam2: Incremental smoothing and mapping using the bayes tree,” *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.

-
- [87] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam,” *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [88] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *Eur. Conf. Comput. Vis.* Cham: Springer International Publishing, 2014.
- [89] M. Labbé and F. Michaud, “Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, 2019.
- [90] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2020, pp. 5135–5142.
- [91] J. Lin and F. Zhang, “R3live++: A robust, real-time, radiance reconstruction package with a tightly-coupled lidar-inertial-visual state estimator,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024.
- [92] T. Shan, B. Englot, C. Ratti, and D. Rus, “Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping,” in *2021 IEEE Int. Conf. Robot. Autom.*, 2021, pp. 5692–5698.
- [93] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment—a modern synthesis,” in *Vision Algorithms: Theory and Practice: Int. Workshop on Vision Algorithms, Corfu, Greece, September 21–22, 1999 Proceedings.* Springer, 2000, pp. 298–372.
- [94] V. Usenko, J. Engel, J. Stückler, and D. Cremers, “Direct visual-inertial odometry with stereo cameras,” in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2016, pp. 1885–1892.
- [95] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.
- [96] X. L. Liu, Zheng and F. Zhang, “Efficient and consistent bundle adjustment on lidar point clouds,” *IEEE Trans. Robot.*, 2024.
- [97] G. D. Evangelidis and R. Horaud, “Joint alignment of multiple point sets with batch and incremental expectation-maximization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1397–1410, Jun. 2017.

- [98] J. Zhu, R. Guo, Z. Li, J. Zhang, and S. Pang, "Registration of multi-view point sets under the perspective of expectation-maximization," *IEEE Trans. Image Process.*, vol. 29, pp. 9176–9189, 2020.
- [99] G. Ferrer, "Eigen-factors: Plane estimation for multi-frame and time-continuous point cloud alignment," in *2019 IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2019.
- [100] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3184–3191, 2021.
- [101] J. Li, T.-M. Nguyen, S. Yuan, and L. Xie, "PSS-BA: LiDAR bundle adjustment with progressive spatial smoothing," in *2024 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 1124–1129.
- [102] X. Ji, S. Yuan, J. Li, P. Yin, H. Cao, and L. Xie, "SGBA: Semantic gaussian mixture model-based LiDAR bundle adjustment," *IEEE Robot. Autom. Lett.*, vol. 9, no. 12, pp. 10 922–10 929, 2024.
- [103] C. Gao, X. Wang, R. Wang, Z. Zhao, Y. Zhai, X. Chen, and B. M. Chen, "A uav-based explore-then-exploit system for autonomous indoor facility inspection and scene reconstruction," *Automation in Construction*, vol. 148, p. 104753, 2023.
- [104] A. Ollero, A. Suarez, C. Papaioannidis, I. Pitas, J. M. Marredo, V. Duong, E. Ebeid, V. Kratky, M. Saska, C. Hanoune *et al.*, "Aerial-core: Ai-powered aerial robots for inspection and maintenance of electrical power infrastructures," *arXiv preprint arXiv:2401.02343*, 2024.
- [105] K.-W. Tse, R. Pi, W. Yang, X. Yu, and C.-Y. Wen, "Advancing uav-based inspection system: The ussa-net segmentation approach to crack quantification," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024.
- [106] L. Yang, J. Fan, Y. Liu, E. Li, J. Peng, and Z. Liang, "A review on state-of-the-art power line inspection techniques," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9350–9365, 2020.
- [107] Y. Tan, S. Li, H. Liu, P. Chen, and Z. Zhou, "Automatic inspection data collection of building surface based on bim and uav," *Autom. Constr.*, vol. 131, p. 103881, 2021.

-
- [108] Y. D. V. Yasuda, F. A. M. Cappabianco, L. E. G. Martins, and J. A. B. Gripp, "Aircraft visual inspection: A systematic literature review," *Comput. Ind.*, vol. 141, p. 103695, 2022.
- [109] H. W. Tong, B. Li, H. Huang, and C. y. Wen, "Coverage path planning for autonomous aircraft inspection using uavs," in *AIAA SCITECH 2025 Forum*, 2025, p. 0216.
- [110] H. Zhang, L. Du, S. Bao, J. Yuan, and S. Ma, "Lvio-fusion: Tightly-coupled lidar-visual-inertial odometry and mapping in degenerate environments," *IEEE Robot. Autom. Lett.*, 2024.
- [111] M. Oelsch, M. Karimi, and E. Steinbach, "Ro-loam: 3d reference object-based trajectory and map optimization in lidar odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6806–6813, 2022.
- [112] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, pp. 401–416, 2017.
- [113] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *Int. J. Robot. Res.*, p. 02783649241229725, 2024.
- [114] J. Wang, B. Li, and M. Q.-H. Meng, "Kinematic constrained bi-directional rrt with efficient branch pruning for robot path planning," *Expert Syst. with Appl.*, vol. 170, p. 114541, 2021.
- [115] T. Li, Q. Feng, Q. Qiu, F. Xie, and C. Zhao, "Occluded apple fruit detection and localization with a frustum-based point-cloud-processing approach for robotic harvesting," *Remote Sens.*, vol. 14, no. 3, p. 482, 2022.
- [116] Y. Tang, J. Qiu, Y. Zhang, D. Wu, Y. Cao, K. Zhao, and L. Zhu, "Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: A review," *Precis. Agric.*, vol. 24, no. 4, pp. 1183–1219, 2023.
- [117] M. Suresh Kumar and S. Mohan, "Selective fruit harvesting: Research, trends and developments towards fruit detection and localization—a review," *Proc. Inst. Mech. Eng. C*, vol. 237, no. 6, pp. 1405–1444, 2023.

- [118] Y. Lu and S. Young, “A survey of public datasets for computer vision tasks in precision agriculture,” *Comput. Electron. Agric.*, vol. 178, p. 105760, 2020.
- [119] C. Smitt, M. Halstead, P. Zimmer, T. Läbe, E. Guclu, C. Stachniss, and C. McCool, “Pag-nerf: Towards fast and efficient end-to-end panoptic 3d representations for agricultural robotics,” *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 907–914, 2023.
- [120] C. Smitt, M. Halstead, T. Zaenker, M. Bennewitz, and C. McCool, “Pathobot: A robot for glasshouse crop phenotyping and intervention,” in *IEEE Int. Conf. Robot. Autom.*. IEEE, 2021, pp. 2324–2330.
- [121] G. Roggiolani, M. Sodano, T. Guadagnino, F. Magistri, J. Behley, and C. Stachniss, “Hierarchical approach for joint semantic, plant instance, and leaf instance segmentation in the agricultural domain,” in *IEEE Int. Conf. Robot. Autom.*. IEEE, 2023, pp. 9601–9607.
- [122] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [123] S. Thrun, “Probabilistic robotics,” *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.
- [124] Y. Zhang, W. Liang, M. Yuan, H. He, J. Tan, and Z. Pang, “Monocular visual-inertial and robotic-arm calibration in a unifying framework,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 146–159, 2021.
- [125] W. Yang, H. Hu, K.-w. Tse, S. Chen, W. Wen, and C.-y. Wen, “Lidar stereo visual inertial pose estimation based on feedforward and feedbacks,” in *Int. Conf. Unmanned Aircraft Syst.*. IEEE, 2024, pp. 1042–1049.
- [126] S. Chen, Y. Feng, C.-Y. Wen, Y. Zou, and W. Chen, “Stereo visual inertial pose estimation based on feedforward and feedbacks,” *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 6, pp. 3562–3572, 2023.
- [127] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, “Observability-based rules for designing consistent ekf slam estimators,” *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 502–528, 2010.
- [128] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam,” *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

-
- [129] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, “Pl-slam: Real-time monocular visual slam with points and lines,” in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2017, pp. 4503–4508.
 - [130] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.
 - [131] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.
 - [132] L. Nicholson, M. Milford, and N. Sünderhauf, “Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 1–8, 2018.
 - [133] M. Zins, G. Simon, and M.-O. Berger, “Oa-slam: Leveraging objects for camera relocalization in visual slam,” in *Proc. IEEE Int. Symp. Mixed Augment. Reality.* IEEE, 2022, pp. 720–728.
 - [134] K.-W. Tse, R. Pi, Y. Sun, C.-Y. Wen, and Y. Feng, “A novel real-time autonomous crack inspection system based on unmanned aerial vehicles,” *Sensors*, vol. 23, no. 7, p. 3418, 2023.
 - [135] Y. Wang, B. Xu, W. Fan, and C. Xiang, “Qiso-slam: Object-oriented slam using dual quadrics as landmarks based on instance segmentation,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 4, pp. 2253–2260, 2023.
 - [136] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, “3d scene graph: A structure for unified semantics, 3d space, and camera,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5664–5673.
 - [137] J. Zhang, R. Zhang, Y. Yue, C. Yang, M. Wen, and D. Wang, “Slat-calib: Extrinsic calibration between a sparse 3d lidar and a limited-fov low-resolution thermal camera,” in *2019 IEEE Int. Conf. Robot. Biomimet.* IEEE, 2019, pp. 648–653.
 - [138] J. Sola, J. Deray, and D. Atchuthan, “A micro lie theory for state estimation in robotics,” *arXiv preprint arXiv:1812.01537*, 2018.
 - [139] W. Xu and F. Zhang, “Fast-lío: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, 2021.

- [140] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 679–698, 1986.
- [141] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [142] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.
- [143] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-lio2: Fast direct lidar-inertial odometry,” pp. 2053–2073, 2022.
- [144] W. T. Higgins, “A comparison of complementary and kalman filtering,” *IEEE Trans. Aerosp. Electron. Syst.*, no. 3, pp. 321–325, 1975.
- [145] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, “Estimation of imu and marg orientation using a gradient descent algorithm,” in *2011 IEEE Int. Conf. Rehabil. Robot.*. IEEE, 2011, pp. 1–7.
- [146] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, “Lidar-monocular visual odometry using point and line features,” in *2020 IEEE Int. Conf. Robot. Autom.*. IEEE, 2020, pp. 1091–1097.
- [147] J. Jiao, H. Ye, Y. Zhu, and M. Liu, “Robust odometry and mapping for multi-lidar systems with online extrinsic calibration,” *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 351–371, 2021.
- [148] C. Zheng, W. Xu, Z. Zou, T. Hua, C. Yuan, D. He, B. Zhou, Z. Liu, J. Lin, F. Zhu *et al.*, “Fast-livo2: Fast, direct lidar-inertial-visual odometry,” *IEEE Trans. Robot.*, 2024.
- [149] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [150] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *Proc. IEEE Int. Conf. Comput. Vis.*. IEEE, 2011, pp. 2564–2571.
- [151] H. Huang, Y. Sun, J. Wu, J. Jiao, X. Hu, L. Zheng, L. Wang, and M. Liu, “On bundle adjustment for multiview point cloud registration,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8269–8276, 2021.

-
- [152] A. Filatov, A. Filatov, K. Krinkin, B. Chen, and D. Molodan, “2d slam quality evaluation methods,” in *2017 21st Conf. Open Innovations Assoc.* IEEE, 2017, pp. 120–126.
- [153] W. Shang, J. Fu, J. Ma, J. Lin, J. Tian, and M. Yu, “Internet of things-based low-carbon distribution route optimization for logistics,” *IEEE Internet Things J.*, 2024.
- [154] C. Friess, V. Niculescu, T. Polonelli, M. Magno, and L. Benini, “Fully onboard slam for distributed mapping with a swarm of nano-drones,” *IEEE Internet Things J.*, 2024.
- [155] M. Raj, S. Gupta, V. Chamola, A. Elhence, T. Garg, M. Atiquzzaman, and D. Niyato, “A survey on the role of internet of things for adopting and promoting agriculture 4.0,” *J. Netw. Comput. Appl.*, vol. 187, p. 103107, 2021.
- [156] M. P. Mathew and T. Y. Mahesh, “Leaf-based disease detection in bell pepper plant using yolo v5,” *Signal, Image and Video Processing*, pp. 1–7, 2022.
- [157] T. Mallick, P. P. Das, and A. K. Majumdar, “Characterizations of noise in kinect depth images: A review,” *IEEE Sens. J.*, vol. 14, no. 6, pp. 1731–1740, 2014.
- [158] Y. Wang, C. Jiang, and X. Chen, “VOOM: Robust Visual Object Odometry and Mapping using Hierarchical Landmarks,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2024.
- [159] H. Kang, X. Wang, and C. Chen, “Accurate fruit localisation using high resolution lidar-camera fusion and instance segmentation,” *Comput. Electron. Agric.*, vol. 203, p. 107450, 2022.
- [160] T. Liu, H. Kang, and C. Chen, “Orb-livox: A real-time dynamic system for fruit detection and localization,” *Comput. Electron. Agric.*, vol. 209, p. 107834, 2023.