Technical University of Munich
TUM School of Engineering and Design

Map-based Long-term Localization and Mapping

Incorporating LiDAR, IMU and Camera Sensors

Miguel Arturo Vega-Torres

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Christoph Holst

Prüfende der Dissertation:

1. Prof. Dr.-Ing. André Borrmann
2. Prof. Dr. Stefan Leutenegger
3. Prof. Dr.-Ing. Borja García de Soto

Die Dissertation wurde am 21.08.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 19.11.2024 angenommen.

Acknowledgements

I am thankful for the continuous support and guidance provided by my esteemed supervisors, Prof. Dr.-Ing. André Borrmann and Dr.-Ing. Alex Braun. Prof. Dr.-Ing. André Borrmann's constant encouragement and insightful discussions throughout my PhD have been crucial in shaping the direction of my research. Dr.-Ing. Alex Braun's assistance and guidance during my journey was also invaluable. This project could not have been completed without their mentorship and feedback.

I am sincerely grateful to the European Horizon 2020 research project, INTREPID, for generously funding most of my doctoral work. Beyond financial support, my involvement in the project afforded me invaluable opportunities to test and refine my algorithms in realistic scenarios. Additionally, I engaged in valuable international collaborations, including interactions with industry professionals. This exposure significantly enhanced the quality of my research, always keeping the end-user in focus. To all dedicated members of the INTREPID consortium, including Gustav Tolt, Nicholas Vertos, and Stephane Cascio, thank you for fostering a collaborative and enriching environment.

Special thanks go to Prof. Dr.-Ing. Borja García de Soto and the New York University Abu Dhabi (NYUAD) for hosting and advising me during my external research stay. Professor Borja and his group welcomed me warmly, and their insights have left a lasting impact, contributing to my academic and personal growth.

A particular mention goes to my mentor, Dr. Denis Wohlfeld. His persistent support, critical and highly analytical thinking, and engaging conversations were a constant source of inspiration, playing a vital role throughout and beyond my doctoral journey.

I extend my heartfelt thanks to my colleagues at the Technical University of Munich, including Mansour Mehranfar, Jonas Urs Schlenger, Divya Singh, Martin Slepicka, Andrea Carrara, Florian Noichl, Fiona Claire Collins, Kasimir Forth, Changyu Du, Yuandong Pan, Mohammad Saeed Mafipour, Mohammad Reza Kolani, Jiabin Wu, Dr. Stavros Nousias, Fabian Pfitzner, Jan Clever, Vijaya Holla, Ina Heise, Sebastian Esser, Ann-Kristin Dugstad, Maikel Brinkhoff, Konstantinos Gkrispanis and Ester Radi. Our collaborative environment fueled creativity and resilience, helping me to persevere through challenging times.

Lastly, I owe immense gratitude to my partner, my family, and my friends. Their constant love, firm belief in my choices, and continuous encouragement have sustained me throughout this journey.

This thesis stands as a symbol of the collaborative energy and combined insights of everyone listed above. My name may be on the cover, but this accomplishment is due to the collaborative efforts of many, and I am profoundly grateful to everyone involved.

Miguel Arturo Vega Torres
August, 2024

Abstract

This dissertation addresses the challenge of achieving long-term localization and mapping in changing environments using Light Detection and Ranging (LiDAR), Inertial Measurement Unit (IMU), and camera sensors integrated with reference 3D building information models (BIM models) or point clouds. The central research question investigates how these reference maps can be used to align and correct sensor data, thereby enhancing accuracy and robustness in mapping.

The developed methodology introduces novel frameworks, such as OGM2PGBM, SLAM2REF, and BIMCaP, that address key challenges in sensor pose refinement and map alignment. OGM2PGBM converts Occupancy Grid Maps (OGMs) into Pose Graph-based Maps (PGBMs), facilitating the transition from Particle Filter (PF) to Graph-based Localization (GBL) algorithms and enhancing pose tracking accuracy, which is crucial for autonomous mobile robots operating in dynamic environments. SLAM2REF integrates new place recognition descriptors and registration algorithms to align drifted session data acquired with Simultaneous Localization and Mapping (SLAM) systems with reference maps, enabling reliable ground truth pose calculations. BIMCaP refines sensor poses by combining semantic landmarks with depth completion methods to integrate LiDAR with camera measurements, effectively registering point clouds, even in cluttered environments, and ensuring accurate alignment with reference maps.

Extensive evaluations with open-access datasets demonstrate significant improvements in SLAM map accuracy, advancing sensor pose precision and semantic landmark extraction for long-term navigation and management.

This research offers practical benefits across various fields. It enables reliable ground truth pose estimation for evaluating SLAM and localization algorithms, minimizing the need for costly sensors. Developers of autonomous vehicles benefit from robust real-time pose-tracking algorithms that enhance localization in changing environments. In Augmented Reality (AR), the methods ensure precise alignment of real-world and virtual data, enhancing applications in architecture and construction. BIMCaP can also assist mobile robotics in retail by aligning data with permanent reference map elements, improving map accuracy and change detection for better inventory management. Additionally, this research advances semantic segmentation for construction sites, contributing to safer and more efficient construction practices through improved digital mapping.

The open-source datasets and algorithms provided promote reproducibility and benchmarking, accelerating the development of more effective and resilient methodologies. The results inspire further research into long-term collaborative robot mapping and robust pose tracking in changing environments.

Zusammenfassung

Diese Dissertation befasst sich mit der Herausforderung, langfristige Lokalisierung und Kartierung in sich verändernden Umgebungen zu erreichen, indem LiDAR, IMU und Kamerasensoren mit Referenz-3D-BIM-Modellen oder Punktwolken integriert werden. Die zentrale Forschungsfrage untersucht, wie diese Referenzkarten verwendet werden können, um Sensordaten auszurichten und zu korrigieren, wodurch die Genauigkeit und Robustheit der Kartierung verbessert wird.

Die entwickelte Methodik führt neuartige Rahmenwerke wie OGM2PGBM, SLAM2REF und BIMCaP ein, die die wichtigsten Herausforderungen bei der Verfeinerung der Sensorposition und der Kartenausrichtung angehen. OGM2PGBM wandelt OGMs in PGBMs um, was den Übergang von PF- zu GBL-Algorithmen erleichtert und die Genauigkeit der Posenverfolgung verbessert, was für autonome mobile Roboter in dynamischen Umgebungen entscheidend ist. SLAM2REF integriert neue Ortserkennungsdeskriptoren und Registrierungsalgorithmen, um mit SLAM-Systemen erfasste Daten von drifted sessions mit Referenzkarten abzugleichen und so zuverlässige Posenberechnungen zu ermöglichen. BIMCaP verfeinert Sensorposen durch die Kombination von semantischen Landmarken mit Tiefenkomplettierungsmethoden, um LiDAR mit Kameramessungen zu integrieren, Punktwolken selbst in unübersichtlichen Umgebungen effektiv zu registrieren und eine genaue Ausrichtung mit Referenzkarten zu gewährleisten.

Umfassende Evaluierungen mit frei zugänglichen Datensätzen zeigen signifikante Verbesserungen in der SLAM-Kartengenauigkeit, die die Präzision der Sensorposition und die semantische Landmarkenextraktion für die langfristige Navigation und Verwaltung verbessern.

Diese Forschung bietet praktische Vorteile in verschiedenen Bereichen. Sie ermöglicht zuverlässige Ground-Truth-Pose-Schätzungen zur Bewertung von SLAM und Lokalisierungsalgorithmen und minimiert den Bedarf an teuren Sensoren. Entwickler autonomer Fahrzeuge profitieren von robusten Echtzeit-Pose-Tracking-Algorithmen, die die Lokalisierung in sich verändernden Umgebungen verbessern. In der AR gewährleisten die Methoden eine präzise Ausrichtung von realen und virtuellen Daten und verbessern Anwendungen in Architektur und Bauwesen. BIMCaP kann auch mobilen Robotern im Einzelhandel helfen, indem es Daten mit permanenten Referenzkarten-Elementen ausrichtet, die Karten-Genauigkeit und Änderungsdetektion für ein besseres Bestandsmanagement verbessert. Darüber hinaus fördert diese Forschung die semantische Segmentierung von Baustellen und trägt zu sichereren und effizienteren Baupraktiken durch verbesserte digitale Modellierung bei.

Die bereitgestellten Open-Source-Datensätze und -Algorithmen fördern die Reproduzierbarkeit und Benchmarking und beschleunigen die Entwicklung effektiverer und widerstandsfähigerer Methoden. Die Ergebnisse inspirieren weitere Forschungen zur langfristigen kollaborativen Roboterkartierung und robusten Pose-Verfolgung in sich verändernden Umgebungen.

Contents

Abbreviations

AI	Artificial Intelligence
AMCL	Adaptive Monte Carlo Localization
APE	Absolute Pose Error
API	Application Programming Interface
AR	Augmented Reality
ATE	Absolute Trajectory Error
BA	Bundle Adjustment
BIM	Building Information Modeling
BIM model	building information model
BIRS	Building Information Robotic System
BlenSor	Blender Sensor Simulation Toolbox
BRIEF	Binary Robust Independent Elementary Features
CAD	Computer-aided Design
CC	CloudCompare
COCO	Common Objects in Context dataset
CVAT	Computer Vision Annotation Tool
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DL	Deep Learning
DLIO	Direct LiDAR Inertial Odometry
DoF	Degrees of Freedom
DT	Digital Twin
EKF	Extended Kalman Filter
FACaP	Floorplan-Aware Camera Poses Refinement
FAST	Features from Accelerated Segment Test
FoV	Field of View
FPFH	Fast Point Feature Histograms
GBL	Graph-based Localization
GICP	Generalized ICP
GMCL	General Monte Carlo Localization
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GT	Ground Truth
ICP	Iterative Closest Point
IFC	Industry Foundation Classes
IMU	Inertial Measurement Unit
IP	Ingress Protection
IPS	Indoor Positioning System
ISC	Indoor Scan Context
ISCD	Indoor Scan Context Descriptor

KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
KLD	Kullback-Leibler distance
KNN	K-nearest neighbors
LiDAR	Light Detection and Ranging
LIO	LiDAR-Inertial Odometry
LIVO	LiDAR-Inertial-Visual Odometry
LOAM	LiDAR Odometry and Mapping
MAP	Maximum A Posteriori
MAV	Micro Aerial Vehicle
MCL	Monte Carlo Localization
MDC	Motion Distortion Correction
MEP	Mechanical, Electrical, and Plumbing
MLE	Maximum Likelihood Estimation
MME	Map Mean Entropy
MPV	Mean Plane Variance
ND	Negative Difference
NeRF-VO	Neural Radiance Fields for Visual Odometry
NIR	Near-infrared
NND	Nearest Neighbor Distance
NWC	Normalized Work Capacity
OBJ	Wavefront .obj file
OGM	Occupancy Grid Map
ORB	Oriented FAST and Rotated BRIEF
P2P	Point-to-Point
PC	Personal Computer
PD	Positive Difference
PF	Particle Filter
PGBM	Pose Graph-based Map
PGM	Portable Gray Map
PNG	Portable Network Graphics
PP	Path Planner
RANSAC	Random Sample Consensus
RE	Rotational Error
RGB	red, green, and blue
RGB-D	Red-Green-Blue-Depth
RMSE	Root Mean Square Error
ROS	Robot Operating System
RTK	Real-time Kinematic
RViz	ROS visualization
SC	Scan Context
Scan-Map deviations	discrepancies between the reference map and the current state of the real-world

SCD	Scan Context Descriptor
SD	Session Data
SDF	Simulation Definition Format
SDK	Software Development Kit
SER	Similar Energy Region
SLAM	Simultaneous Localization and Mapping
SOTA	State-of-the-art
SR	Success Rate
STL	stereolithography
SVG	Scalable Vector Graphics
TE	Translational Error
TEASER	Truncated Least Squares Estimation And SEmidefinite Relaxation
TLS	Terrestrial Laser Scanner
TRIM	Translational and Rotational Invariant Measurement
TUM	Technical University of Munich
UAV	Unmanned Aerial Vehicle
UE	Unaltered Element
UGV	Unmanned Ground Vehicle
URDF	Universal Robot Description Format
UV	Unmanned Vehicle
V-SLAM	Visual-SLAM
VAE	Variational Auto-Encoder
VG	Voxel Grid
VIO	Visual-Inertial Odometry
VL	Vanishing Line
VO	Visual Odometry
VP	Vanishing Point
YAML	YAML Ain't Markup Language

Chapter 1

Introduction

1.1 Background

Currently, mobile mapping systems integrated into robots or handheld devices equipped with advanced sensors facilitate the rapid creation of updated 3D maps. These systems utilize State-of-the-art (SOTA) SLAM algorithms to ensure efficiency. However, these maps are in their local coordinate systems and, therefore, separated from any prior information. Additionally, they might contain potential drift issues, rendering them unsuitable for creating accurate updated map representations, comparative analysis, or change detection.

Moreover, several real-world applications require the capacity to align, compare, and manage 3D data received at various intervals that may be separated by lengthy intervals of time. This process is referred to as long-term map management.

The field of autonomous navigation has experienced significant advancements driven by the need for precise and reliable localization and mapping in dynamic environments. Combining multiple sensor modalities, such as LiDAR, IMUs, and cameras, has proven to be a promising approach for enhancing the robustness and accuracy of mapping systems.

Despite these advancements, challenges remain in achieving long-term localization and mapping, particularly in environments with significant changes and complex structures.

1.2 Problem

The primary challenge addressed in this dissertation is the development of comprehensive methods for long-term localization and mapping that leverage the complementary strengths of LiDAR, IMU, and camera sensors and that utilize reference 3D BIM models or point clouds for alignment and correction of the sensor poses. Current approaches often struggle with the limitations of individual sensors, particularly in changing and cluttered

environments. This thesis aims to overcome these challenges by studying and integrating different sensor technologies and advanced SOTA algorithms to provide robust solutions for long-term localization and mapping with reference maps. In robotics, many systems necessitate real-time performance to adapt to the rapidly changing real-world environment. Therefore, in this dissertation, real-time systems for alignment and localization were explored. However, due to the inherent accuracy limitations of real-time methods, this research will transition from exclusively relying on these methods to emphasizing the acquisition of high-precision maps. This shift is expected to enhance the effectiveness and reliability of map management over extended periods. In essence, the focus will move from immediate but less accurate solutions to more precise and sustainable approaches, thereby improving long-term outcomes.

1.3 Motivation

Long-term map management is crucial since the real world constantly evolves and changes. This applies to both humans who want to utilize the map to comprehend the current situation and its evolution and to autonomous robots for effective and fast navigation.

Moreover, achieving accurate alignment and effective management of extensive datasets represent significant challenges in enabling the creation of Digital Twins (DTs) for cities and buildings (Borrmann et al., 2024; Mylonas et al., 2021). While the definition of a DT is still not standardized in the built environment, in this dissertation, a DT is a dynamic virtual representation of a physical object or system across its life cycle, distinguishing itself from other digital models due to its connection (i.e. real-world data transmission) to the physical object to enable understanding, learning, and reasoning (Feng et al., 2021; Mylonas et al., 2021). As explained by (Botín-Sanabria et al., 2022), in complex implementations, automatic alignment of 3D data becomes imperative to achieve DTs with maturity levels of 3 or higher. Such levels necessitate the augmentation of models with a continuous flow of real-world information.

1.3.1 Concrete Areas of Application

The alignment of sensor measurements with reference maps is a critical technology across various fields, enabling systems to accurately navigate and understand their environment. This section systematically explores the application areas, boundary conditions, and challenges associated with this technology, setting the stage for discussing state-of-the-art techniques and identifying research gaps.

Requiring Real-time Processing

In applications requiring real-time processing, such as emergency situations, augmented reality, and autonomous robot localization and navigation, challenges are particularly pronounced. In emergency scenarios, the fast movement of the scanning sensor, the presence of debris, and significant changes in the environment may result in low correspondences with reference maps, complicating correct alignment. Robust AR applications must deal with different types of clutter, for example, in a residential or office building, furniture that is not present in the reference map or construction sites, materials, and varying levels of geometric correlation with the reference map depending on the construction phase. Autonomous robots face similar issues, where changes in the environment, such as repositioning furniture, disrupt the alignment process.

These dynamic and cluttered environments demand robust, adaptive solutions to ensure accurate real-time alignment of sensor measurements with reference maps. For instance, fast-updated 3D digital maps can help first responders improve situational awareness and make effective, safe decisions to save lives during emergencies before putting themselves in highly risky situations (Alliez et al., 2020; He et al., 2021). Similarly, real-time alignment enables autonomous robots to localize themselves within a reference map (such as a BIM model), facilitating various autonomous robotic activities such as path planning (Dugstad et al., 2022), object inspection, (K. Kim & Peavy, 2022), and maintenance and repair operations (S. Kim et al., 2021; X. Xu et al., 2021). In recent years, significant research has focused on practical applications and methodologies for integrating robotic systems in construction, highlighting the urgency for future development in this field Gopee et al. (2023), B. R. K. Mantha et al. (2020, 2020), Prieto, Giakoumidis, and García de Soto

(2024), Prieto, Xu, and García de Soto (2024), Soto and Skibniewski (2020), X. Xu and Garcia de Soto (2020, 2023), and X. Xu and García de Soto (2022).

Applications requiring rapid processing demand that data resulting from the alignment is immediately usable. However, if the alignment is incorrect or the system fails, the resulting 3D map or calculated pose becomes unusable. Consequently, allowing additional time for refining the acquired map can often be advantageous, ensuring higher accuracy and reliability.

When Offline Processing is Acceptable

Nonetheless, in environments where offline processing is acceptable, such as construction sites, the retail industry, and surveying with prior maps, several challenges also arise.

Construction sites are often cluttered with materials and tools, which can occlude sensor measurements and introduce noise (such as reflections). In the retail industry, the dynamic nature of product placement and shelf arrangements often leads to discrepancies between the reference map and the current state of the environment. As a result, only permanent structural elements, such as columns and walls, can be reliably used for alignment. Similarly, for surveying with prior maps, initial maps may differ due to new elements introduced after the map was created.

These factors collectively lead to potential inaccuracies in aligning sensor data with reference maps, complicating the task of maintaining accurate and up-to-date environmental representations. Therefore, an automatic map alignment and change detection framework can significantly enhance the integration of mapping devices into existing industry workflows, addressing one of the main barriers to their widespread adoption[1] (NavVis et al., 2022).

For example, an up-to-date 3D digital map can help construction site managers promptly distinguish as-planned and as-built differences, thus reducing the probability of long-schedule delays and high-cost overruns (Braun & Borrmann, 2019; Braun et al., 2020). Another example is the crack detection task (on buildings or bridges), where it is crucial

[1]Compatibility of mapping devices with existing tools is, after the budget, the second most crucial barrier surrounding the usage of mobile mapping devices (NavVis et al., 2022).

not only to identify cracks in an image but also to document their locations relative to a reference model to ensure the information is usable Ko et al. (2021).

Figure 1.1, illustrates several potential applications of the methods proposed in this dissertation.

Figure 1.1: Applications of the research conducted in this dissertation: On the left, applications requiring real-time processing to meet end-user expectations. On the right, applications where offline processing is acceptable.

Besides being useful for autonomous construction site monitoring, an offline accurate alignment of sensor data with a relatively accurate reference map (at least in terms of its permanent structures such as walls and columns) allows the retrieval of the sensor's precise 6-Degrees of Freedom (DoF) Ground Truth (GT) poses in the entire trajectory. 6-DoF refers to the sensor's ability to move freely in three-dimensional space, encompassing three translational movements (forward/backward, up/down, left/right) and three rotational movements (pitch, yaw, roll).

These GT poses serve multiple functions. They enable precise identification of the capture locations of point clouds and images necessary for generating an accurate, updated 3D map. Additionally, they facilitate the assessment of the accuracy of SLAM, odometry, and localization algorithms. This capability is particularly crucial for advancing research and development in this field.

Historically, obtaining GT poses has necessitated costly equipment like Real-time Kinematic (RTK)-corrected Global Navigation Satellite System (GNSS) for outdoor environments or laser trackers and motion capture systems for indoor settings (Y. Liu et al., 2021). However, the expensive costs associated with these methods pose a substantial barrier for

individual researchers. Additionally, acquiring dense GT poses for extended trajectories, especially in indoor scenarios, has been found to be very challenging (L. Zhang et al., 2023).

Recent studies, such as by the authors of ConSLAM (Trzeciak et al., 2023a, 2023b) and Newer College (Ramezani et al., 2020; L. Zhang et al., 2022) datasets, have leveraged Terrestrial Laser Scanner (TLS) point clouds—providing millimeter-precise 3D scans of the environment—to be used as reference GT map and overcome these limitations. Through semi-automatic techniques, researchers have effectively aligned mobile LiDAR measurements with TLS point clouds. This advancement, as well as the methods presented in this dissertation, represent a significant step forward in SLAM research towards automatic, accurate GT pose acquisition methods suitable for both large indoor and outdoor scenarios. Enabling researchers to evaluate SLAM, odometry, and localization frameworks on their own collected sequences.

1.4 State of Practice

For outdoor localization and alignment, Global Positioning System (GPS) is often a viable option due to its widespread availability and effectiveness. However, GPS is impractical for indoor environments because it requires a direct line of sight to at least four satellites—three for determining the 3D position and one for time correction.

To address the limitations of GPS indoors, various Indoor Positioning System (IPS) alternatives use radio signals, such as Wi-Fi or Bluetooth, as well as AprilTags or laser trackers and motion capture systems to achieve accurate pose calculation in indoor settings (Kayhani et al., 2022, 2023; Koide et al., 2022; Lopez-de-Teruel et al., 2017; B. Mantha & Garcia de Soto, 2019; B. R. K. Mantha & Garcia de Soto, 2022). The downside of these systems is that they require additional strategically placed sensors or landmarks, which can increase the cost and effort of implementing such a positioning system. Nevertheless, although not always accessible, 3D prior maps of buildings are increasingly becoming standard in modern construction. These maps, often in the form of BIM models or point clouds, document the state of the building during and after construction or in the design phases.

1.5 State-of-the-Art and Research Gap

Current techniques for aligning sensor measurements with reference maps have achieved significant progress in handling static environments. For example, employing camera sensors (Acharya et al., 2019, 2022; Asadi et al., 2019; Boniardi, Valada, et al., 2019; Haque et al., 2020; Kropp et al., 2018; Sokolova et al., 2022), 2D LiDAR sensors (Boniardi, Caselitz, et al., 2019; Boniardi et al., 2017; Follini et al., 2020; Hendrikx et al., 2021, 2022; Karimi et al., 2020, 2021; K. Kim & Peavy, 2022; S. Kim et al., 2021; Prieto et al., 2020) or 3D LiDAR sensors (Blum et al., 2020; Caballero & Merino, 2021; Ercan et al., 2020; Oelsch et al., 2021, 2022; Shaheer et al., 2022, 2023; Yin et al., 2023).

However, dynamic, changing, and cluttered environments continue to present substantial challenges. The integration of advanced alignment algorithms for accurate map generation and update, improvements in indoor re-localization capabilities, and enhanced system resilience to environmental changes are critical areas for ongoing research. Addressing these research gaps is essential for advancing technology and effectively handling the complex demands of real-world applications.

Concrete use cases demonstrate the value of accurate and up-to-date 3D maps. For instance, construction site managers can use these maps to monitor progress and deviations from plans (which nowadays are almost always present in the form of 4D BIM models), while first responders can leverage them for enhanced situational awareness during emergencies. Mobile robots need to localize themselves within a reference map to benefit from enriched BIM models with up-to-date information, aiding in path planning, object inspection, and maintenance operations.

In conclusion, the diverse applications of sensor alignment with reference maps necessitate specialized approaches to address their specific boundary conditions and challenges. By analyzing these factors and contributing with novel alignment methods, this dissertation aims to contribute to the advancement of localization and mapping technology, particularly in changing and cluttered environments, thereby bridging the existing research gaps.

1.6 Research Objectives and Questions

The main three research questions addressed in this dissertation are:

RQ 1 *How can 3D BIM models be leveraged for real-time 2D LiDAR and image localization systems?*

- *Rationale:* BIM models contains rich 3D spatial and semantic information about built environments, which can significantly enhance localization systems. Current 2D LiDAR and image-based localization systems often face challenges in dynamic and cluttered environments, leading to inaccuracies and failures in real-time applications. By leveraging 3D BIM models, these systems can benefit from pre-existing, detailed environmental data, providing a reference for more precise and robust localization. This research question aims to explore the integration of 3D BIM models with real-time 2D LiDAR or camera data to improve the overall performance and reliability of localization systems in complex environments.

RQ 2 *How can reference 3D BIM models or point clouds be utilized for alignment and correction of session data from 3D LiDAR and IMU measurements?*

- *Rationale:* Accurate alignment and correction of session data are critical for long-term localization and mapping, particularly in environments subject to changes over time. Reference 3D BIM models and point clouds provide a stable and detailed representation of the environment, which can be used as a baseline for aligning and correcting new session data captured by 3D LiDAR and IMU sensors. This research question focuses on developing methods to utilize these reference models to ensure high precision and consistency in the collected data, thereby enhancing the accuracy of the long-term localization and mapping processes.

RQ 3 *How can semantics and LiDAR-camera fusion be utilized to create a robust alignment and correction method of SLAM-acquired real-world 3D data with a BIM model or a semantic 3D map?*

- *Rationale:* Simultaneous Localization and Mapping (SLAM) systems are widely used for real-time mapping and navigation but often suffer from drift and errors, especially in complex environments. Incorporating semantics into the SLAM-acquired data can provide meaningful context while fusing LiDAR and camera data can enhance the robustness of the measurements by leveraging the complementary strengths of these sensors. By aligning and correcting SLAM-acquired 3D data with BIM models or semantic 3D maps, this research aims to develop a more reliable and accurate method for real-world applications. This question seeks to explore advanced techniques that combine semantic understanding and multi-sensor fusion to improve the quality and reliability of SLAM-generated maps, aligning them with a reference 3D map.

In addition to these questions, there are three sub-research questions for each of the presented research questions, which will be elaborated on in the respective chapters.

1.7 Contributions and Implications

This thesis contributes to the field of map-based long-term localization and mapping by addressing key challenges such as deviations between the reference map and the most recently acquired data (Scan-Map deviations) and the limitations of single sensor modalities. The findings have the potential to significantly enhance the robustness and accuracy of mapping and localization systems in changing and dynamic environments.

By integrating LiDAR, IMU, and camera sensors in different ways, this study provides comprehensive approaches that leverage the strengths of each modality, leading to improved performance in real-world applications. This dissertation advances robot localization, navigation, multi-session anchoring, and cross-source point cloud registration by integrating 3D BIM models and point clouds with LiDAR, IMU, and camera measurements. The developed methodologies address long-term localization and mapping challenges in dynamic environments by combining sensor modalities and leveraging 3D reference maps for map alignment and pose correction. These contributions enhance the robustness and accuracy

of mapping systems, facilitating applications in construction site management, emergency response, and autonomous navigation.

Key contributions include a novel open-source method for transforming 2D OGMs into PGBMs and a comprehensive evaluation of SOTA 2D LiDAR localization algorithms. Additionally, a novel open-source approach, called SLAM2REF, was developed for correcting sensor poses using 3D BIM models. The approach includes methods for generating accurate OGMs and 3D session data from BIM models and point clouds. Techniques such as Indoor Scan Context (ISC) and YawICP were introduced for fast place recognition and point cloud registration, which are combined within SLAM2REF in a holistic multi-session anchoring system for aligning and correcting drifted sessions. Moreover, methods were proposed for analyzing and detecting changes in the aligned 3D data.

A method was also proposed that uses the principal normal direction count and feature matching for cross-source global point cloud registration using semantic landmarks. The open-source BIMCaP framework for aligning and correcting sensor measurements with BIM models was introduced, demonstrating improvements over existing methods. These advancements support automated map management and improve compatibility with industry workflows, facilitating the creation and maintenance of DTs for buildings and cities. Overall, this work significantly enhances the capabilities of autonomous robotic systems and supports various practical applications, providing robust solutions for long-term localization, mapping, and DT creation.

A detailed list of open-source contributions can be found in Sections 1.12 and 1.13

1.8 Scope and Limitations

This research explores the use of LiDAR, IMU, and camera sensors for long-term localization and mapping in dynamic environments. While some proposed methodologies integrate these sensors in pairs, a single comprehensive method combining all three for sensor pose refinement was not developed.

The primary focus of this study is the precise alignment of sequential sensor data with a reference map. Nonetheless, this research also provides valuable insights into related areas such as real-time localization (see Section 4.3), map updates, positive and negative change

detection (see Section 5.3.3), depth completion (see Section 6.4), and image semantic segmentation(see Section 6.4), these topics are secondary to the core contributions of this work. For example, in Section 4.3, various methods to enhance real-time localization are introduced and analyzed. However, the main objective remains the creation of an accurate, well-registered, updated map aligned with a reference map, even if this process occurs offline and not in real-time.

Central to this study are indoor structured environments where BIM models or point clouds are available; if none of these reference maps exist, the core of the proposed methods can not be leveraged.

A significant challenge addressed in this study is the alignment or registration of sequential measurements in environments with discrepancies between the reference map and the current state of the real-world (Scan-Map deviations), which are categorized into three types:

1. Deviations caused by clutter or furniture not present in the reference map;
2. Deviations from dynamic, moving elements in the environment during scanning and
3. Alterations to permanent structures like walls and columns.

This research focuses primarily on the first two types of deviations. However, minor discrepancies in permanent features, such as small holes or slight shifts (in the range of ± 3 cm) in individual columns or walls, are not expected to impede the successful application of the framework. It is generally assumed that while Scan-Map deviations are present, the reference map remains a reliable source for localization, with the BIM model or the point cloud maintaining enough geometric precision to reflect the current state of the environment. Permanent elements, including walls, columns, floors, and ceilings, are expected to be accurate within ± 3 centimeters to be considered in the registration process. Considering the range precision of most current LiDAR sensors (which is ± 3 centimeters), even if the map is more accurate than that, a more precise localization is not possible. On the other hand, isolated elements that deviate significantly from their expected positions (e.g., by 1 meter) will not contribute to the alignment process and will most likely be disregarded, assuming that the remaining elements guide the alignment

toward the correct pose. However, if all reference map elements are incorrectly positioned or fail to correspond to real-world measurements, alignment becomes impossible.

Although the proposed methods have potential applications in assisting first responders during emergencies by creating updated and aligned 3D maps, it is crucial to recognize their limitations. For example, in situations like fires where dense smoke fills the interior of buildings, the sensors used in this study may be ineffective in penetrating smoke-filled areas. In such critical scenarios, alternative sensors and techniques, such as thermal cameras, may be more appropriate.

1.9 Structure of the Dissertation

This dissertation is structured as follows:

- **Chapter 1: Introduction** - Provides the background, problem statement, motivation, research objectives, scope, and structure of the dissertation.
- **Chapter 2: Fundamentals** - This chapter presents a concise overview of the foundational techniques that form the basis of the core methods introduced in this dissertation, with particular focus on SLAM and the multi-session anchoring method presented in Chapter 5.
- **Chapter 3: Related Work** - Reviews existing research on visual and LiDAR-based pose estimation as well as on localization and mapping, and the alignment of LiDAR, IMU, and camera sensors with reference maps.
- **Chapter 4: Real-time LiDAR and Image Localization** - This chapter introduces two innovative methodologies designed to enhance real-time localization using BIM models or reference 3D/2D maps. The first method involves a comparative analysis of existing 2D LiDAR real-time localization algorithms and presents the OGM2PGBM framework for transitioning from OGMs to PGBMs. This transition facilitates the use of PFs for rapid global localization, followed by algorithms for continuous pose tracking. The second method focuses on refining camera poses by extracting Vanishing Points (VPs) and Vanishing Lines (VLs) from both camera images and synthetically generated images from a BIM model, thus improving the

accuracy of fast image localization and pose refinement. However, real-time algorithms are intrinsically limited by the computational resources available within the stringent time constraints, leading to inherent trade-offs that often restrict their achievable accuracy. These limitations are addressed in the following two chapters, with a particular emphasis on accurate, non-real-time approaches.

- **Chapter 5: Aligning Integrated Mobile 3D LiDAR-inertial Session Data with a Reference Map** - Building on the OGM2PGBM technique proposed in the previous chapter, this chapter presents the SLAM2REF framework, which addresses key limitations of the methods presented in the previous chapter (Chapter 4) by focusing on post-processing alignment rather than real-time alignment. This framework leverages innovative feature descriptors based on the Scan Context Descriptor (SCD) for place recognition and introduces the novel YawGICP registration algorithm. It incorporates IMU sensor measurements for Motion Distortion Correction (MDC) and integrates these components into a multi-session anchoring framework to align and correct drifted SLAM session data with a reference 3D BIM model or point cloud. This comprehensive approach enhances the accuracy and reliability of localization and mapping, especially in dynamic and challenging environments. Moreover, positive and negative change detection methods in the aligned data are also introduced. However, SLAM2REF still has some limitations, such as a small tolerance for Scan-Map deviations during initial alignment and the requirement for 3D LiDAR measurements with a horizontal Field of View (FoV) of 360 degrees. In the subsequent chapter, these limitations are addressed by incorporating camera data, extracting real-world semantics, and using a Bundle Adjustment (BA) algorithm to enhance sensor poses from restricted FoV measurements.

- **Chapter 6: Artificial Intelligence (AI)-supported Integration of LiDAR and Camera Data with BIM Models and Reference Maps** - This chapter explores AI-driven approaches to improve the alignment of LiDAR data fused with camera data using BIM models or semantic reference maps, with the aim of tackling the main limitations of the SLAM2REF method discussed in the previous chapter (Chapter 5). The first approach addresses the global registration of a SLAM-based reconstructed point cloud with a BIM model, assuming semantically enriched data with minimal drift. The second approach, called BIMCaP, refines sensor poses by

integrating LiDAR and camera data, aligning real-world measurements of structural elements with their corresponding components in a 3D BIM model. Employing only reduced FoV measurements, advanced neural depth completion, and image semantic segmentation algorithms, this chapter aims to achieve precise offline alignment, enhancing the integration of multi-sensor data with BIM models for accurate and reliable mapping.

- **Chapter 7: Conclusion and Further Development** - This chapter concludes the dissertation, summarizing the contributions of each chapter to both the research community and industry. It situates this dissertation within the broader context of the future of map-based long-term localization and mapping and highlights areas, both related and tangential, that deserve further exploration of approaches similar to those proposed.

Figure 1.2 provides a graphical summary of the dissertation's content and illustrates the interrelationships between the proposed methods.

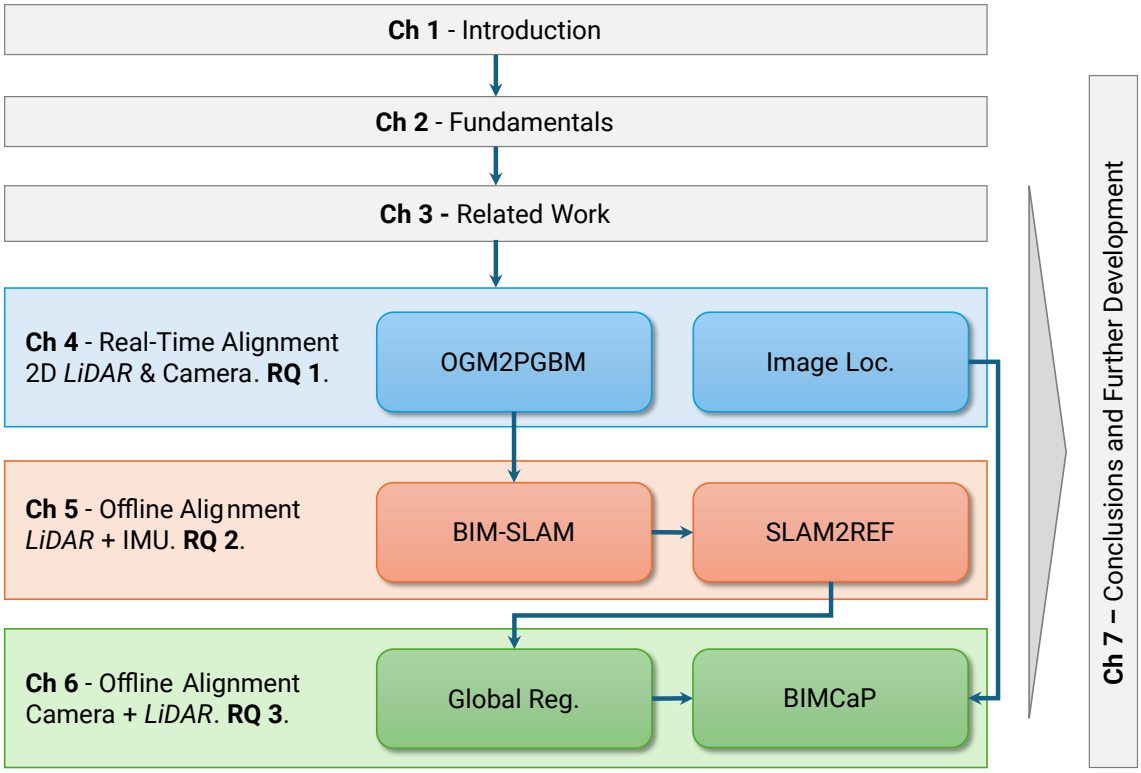


Figure 1.2: Visual overview of the thesis content and structure.

The table below (Table 1.1 provides a summary of the objectives and contributions discussed in this dissertation, along with their corresponding chapters and the sensors utilized for each approach.

Table 1.1: Summary of all contributions and their relation with the objectives, research questions, sensor modalities, and chapters of this dissertation.

Objective & RQ	**Sensors**	**Chapter**	**Contributions**
1. Real-time localization and alignment with a reference map. **RQ 1**.	2D LiDAR and Camera.	4	OGM2PGBM & Image Localization.
2. Highly accurate alignment and correction of SLAM-drifted maps with a reference map. **RQ 2**.	3D LiDAR and IMU.	5	Map to Session Data, SLAM2REF & Map Update.
3. Align low-drift point clouds and refine the sensor poses with a reference map leveraging semantics. **RQ 3**.	3D LiDAR and Camera.	6	Cross-source Global Registration & BIMCaP.

1.10 Publications

Parts of this dissertation have been published in the following peer-reviewed journals and conference papers.

- Vega-Torres, M. A., Braun, A., & Borrmann, A. (2022, September). Occupancy Grid Map to Pose Graph-based Map: Robust BIM-based 2D- LiDAR Localization for Lifelong Indoor Navigation in Changing and Dynamic Environments. In *Proc. of European Conference on Product and Process Modeling 2022*.
  DOI: https://doi.org/10.1201/9781003354222-72
  Discussed in Chapter 4.

- Dantas, R., Peter, S., Wang, X., Vega-Torres, M. A., & Dugstad, A. (2022). Towards Real-time Image Localization with BIM Models. In *Proceedings of 33. Forum Bauinformatik*.
  Discussed in Chapter 4.

- Vega-Torres, M. A., Braun, A., & Borrmann, A. (2023, July). BIM-SLAM: Integrating BIM Models in Multi-session SLAM for Lifelong Mapping using 3D LiDAR. In *Proc. of the 40th International Symposium on Automation and Robotics in Construction (ISARC 2023)*.
  DOI: https://doi.org/10.22260/ISARC2023/0070
  Discussed in the extended version (SLAM2REF) in Chapter 5.

- Vega-Torres, M. A., Braun, A., & Borrmann, A. (2024, July). SLAM2REF: Advancing long-term mapping with 3D LiDAR and reference map integration for precise 6-DoF trajectory estimation and map extension. *Construction Robotics*, 8(2), 13.

DOI: https://doi.org/10.1007/s41693-024-00126-w
Discussed in Chapter 5.

- Vega-Torres, M. A., Braun, A., & Borrmann, A. (2024, July). BIMCaP: BIM-based AI-supported LiDAR-Camera Pose Refinement. In *Proc. of the 31th Int. Conference on Intelligent Computing in Engineering (EG-ICE).*
URL: https://github.com/MigVega/BIMCaP
Discussed in Chapter 6.

- Vega-Torres, M. A., & Pfitzner, F. (2023, September). Investigating Robot Dogs for Construction Monitoring: A Comparative Analysis of Specifications and On-site Requirements. In *Proceedings of the 34th Forum Bauinformatik 2023.*
DOI: https://doi.org/10.13154/294-10094
Discussed in Appendix B.

1.11 Additional Scientific Contributions

During my doctorate, I participated as a collaborator in the following peer-reviewed journal and conference publications that are not covered in this dissertation.

- Vega-Torres, M. A., Braun, A., Bauer, H., Noichl, F., & Borrmann, A. (2021). Efficient vertical object detection in large-high-quality point clouds of construction sites. In *Proc. of the 2021 European Conference on Computing in Construction.*
DOI: https://doi.org/10.35490/EC3.2021.156

- Vega-Torres, M. A., Braun, A., Noichl, F., Borrmann, A., Bauer, H., & Wohlfeld, D. (2022). Recognition of temporary vertical objects in large point clouds of construction sites. *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction*, 174(4), 134-149.
DOI: https://doi.org/10.1680/jsmic.21.00033

- Collins, F., Mafipour, M. S., Noichl, F., Pan, Y., & Vega-Torres, M. A. (2021). Towards applicable scan-to-BIM and scan-to-floorplan: An end-to-end experiment. In *Proc. of the 32nd Forum Bauinformatik.*
DOI: https://doi.org/10.26083/tuprints-00019496

- Du, C., Vega-Torres, M. A., Pan, Y., & Borrmann, A. (2022, September). MV-KPConv: Multi-view KPConv for enhanced 3D point cloud semantic segmentation using multi-modal fusion with 2D image. In *European Conference on Product and Process Modeling 2022.*
DOI: https://doi.org/10.1201/9781003354222-67

- Hassaan, M., Ott, P. A., Dugstad, A.-K., Vega-Torres, M. A., & Borrmann, A. (2023). Emergency floor plan digitization using machine learning. *Sensors*, 23(19).
DOI: https://doi.org/10.3390/s23198344

- Mehranfar, M., Vega-Torres, M. A., Braun, A., & Borrmann, A. (2024). Automated data-driven method for creating digital building models from dense point clouds and images through semantic segmentation and parametric model fitting. *Advanced Engineering Informatics*, 62 (Part A), 10264.
  DOI: https://doi.org/10.1016/j.aei.2024.102643

1.12 Open Source Packages

This dissertation includes several open-source repositories that were developed as part of the research process. These repositories are publicly available on GitHub and contribute to the reproducibility and further development of the work presented. Here are some notable contributions:

1.12.1 OGM2PGBM

OGM2PGBM is a Robot Operating System (ROS) 1/2 package developed to perform transformation from 2D OGMs to PGBMs. It enables the usage of GBL algorithms for pose tracking using a reference map (e.g., with only permanent elements of the environment) for long-term accurate pose-tracking and is a key component in the research presented in Chapter 4, Section 4.3.

Repository Link: https://github.com/MigVega/OGM2PGBM
DOI: https://doi.org/10.5281/zenodo.7330270

1.12.2 SLAM2REF

SLAM2REF is the extended version of BIM-SLAM and is a holistic system that was developed to achieve automatic alignment and correction of LiDAR-based session data with a reference map, which can be either a point cloud or a BIM model. This tool is crucial to calculate precise 6-DoF poses and achieve coherent map extension and is detailed in Chapter 5.

Repository Link 1: https://github.com/MigVega/SLAM2REF

In addition to this contribution, two key packages have been developed. The first enables the generation of session data from a reference 3D point cloud or BIM model; in this repository is also the code that creates accurate 2D OGMs from any of these reference maps (see Section 5.3.1 for further details). The second package facilitates the saving of key information from any LiDAR-based SLAM or odometry system (refer to Section 5.3.2 for more information). The respective repositories are provided below.

Repository Link 2: https://github.com/MigVega/Key-Info-Saver-SLAM
Repository Link 3: https://github.com/MigVega/Map2SessionData

1.12.3 BIMCaP

BIMCaP provides the code to achieve LiDAR-Camera depth completion and pose refinement, filtering only reliable semantic landmarks (such as walls and columns) from the source and target maps, aiding in the tasks described in Chapter 6, Section 6.4.

Repository Link 1: https://github.com/MigVega/BIMCaP

Alongside this contribution, two other essential packages have been released. BIM2SemanticPC contains the code to convert BIM models into semantically enriched point clouds with desired density (used in Sections 6.3 and 6.4). PC2VectorizedFloorPlan comprehends the algorithms needed to convert a semantically enriched point cloud into a vectorized-semantic floor plan (used in Section 6.4).

Repository Link 2: https://github.com/MigVega/BIM2SemanticPC
Repository Link 3: https://github.com/MigVega/PC2VectorizedFloorPlan

1.13 Open Access Datasets

Certain datasets and developments from this dissertation have been disseminated through the following publications:

1.13.1 OGM2PGBM Dataset

The OGM2PGBM dataset contains the sequencers of simulated 3D LiDAR scans along with their corresponding 3D and 2D reference maps used for the real-time localization experiments in Chapter 4, Section 4.3.

Repository Link: https://mediatum.ub.tum.de/1749236
DOI: https://doi.org/10.14459/2024mp1749236

1.13.2 ConSLAM BIM and GT Poses Dataset

This dataset contains the ConSLAM [2] BIM model and GT 6-DoF poses used for the alignment and correction experiments in Chapter 5.

Repository Link: https://mediatum.ub.tum.de/1743877
DOI: https://doi.org/10.14459/2024MP1743877

1.13.3 CMS Sensor Mounting System

This repository includes 3D-digital-modeled parts designed for mounting a LiDAR-Camera mapping system onto a robot. The modular components, many of which are 3D-printable, allow for the integration of multiple sensors. While specifically developed for the Go1-legged robot, the system is adaptable for use on any robot with a flat surface or as a handheld device by detaching the sensors from the components that house the Personal Computer (PC) and batteries. Further details about the system can be found in Appendix B.

Repository Link: https://mediatum.ub.tum.de/1750434
DOI: https://doi.org/10.14459/2024mp1750434

[2] ConSLAM is also an open-access dataset accessible here: https://github.com/mac137/ConSLAM

1.13.4 Layout Prediction Dataset

The Layout Prediction dataset contains labels for over 200 images of real-world construction sites, which are part of Sequence 2 of the ConSLAM dataset. This dataset adheres to the conventions defined in SRW-Net. The dataset includes the original images accompanied by layout annotations. These annotations consist of lines representing various architectural elements, such as walls, ceilings, and doors, with each line specified by coordinates and categorized in a dictionary according to its type.

Repository Link: https://mediatum.ub.tum.de/1751462

DOI:https://doi.org/10.14459/2024mp1751462

1.14 Summary of Open Contributions

Table 1.2 summarizes and provides hyperlinks to all open contributions and relates them to the corresponding parts of this dissertation.

Table 1.2: Hyperlinks to the open-access data and open-source code created in the frame of this dissertation.

Part	**Code - GitHub Links**	**Data - Hyperlinks**
Sec. 4.3	OGM2PGBM	OGM2PGBM Dataset
		Layout Prediction Dataset
Ch. 5	SLAM2REF	ConSLAM BIM
	Key-Info-Saver	and GT Poses
	Map2SessionData	
Sec. 6.4	BIMCaP	Pre-trained RTMDet model
	BIM2SemanticPC	254 labeled images
	PC2VectorizedFloorPlan	All pseudo-labels of ConSLAM Seq. 2
		88 labels 19 classes
Apdx. B	Mapping System	CMS Sensor Mounting System

Chapter 2

Fundamentals

Before presenting the current SOTA methodologies, an introduction to the theoretical concepts behind localization and mapping algorithms, as well as the multi-session anchoring process employed in Chapter 5, is presented. For better understanding, a table with all mathematical variables and the corresponding description can be found in the appendix A.

2.1 Foundational Concepts

Prior to explaining the basics of multi-session anchoring and factor graph, it is important to differentiate between four important concepts: Odometry, Localization, SLAM and BA.

- Odometry refers to the process of estimating an agent's change in position and orientation over time by integrating sequential data from motion sensors, such as wheel encoders, IMUs, cameras, or LiDAR sensors. This process is usually performed by extracting ego-motion parameters from correspondences between sequential frames. While it provides continuous updates on the agent's movement, odometry is prone to accumulating errors over time, leading to *drift* (Agostinho et al., 2022).

- Localization involves determining an agent's precise position and orientation within a known environment. Unlike odometry, which tracks relative movement, localization aims to pinpoint the agent's location on a pre-existing map, correcting any errors from odometry by referencing fixed landmarks or features in the environment (Kumar & Muhammad, 2023).

- SLAM is a more complex process that combines the tasks of self-localization and mapping. In SLAM, a robot simultaneously builds a map of an unknown environment and determines its location within that map. (Gutmann & Schlegel, 1996). SLAM extends the capabilities of traditional odometry-based systems by enabling

long-term data associations, commonly referred to as loop closures when the agent revisits previously explored locations (G. Kim et al., 2022). These additional data associations facilitate the optimization of the entire map through pose-graph optimization techniques (Dellaert, Kaess, et al., 2017). Unlike localization systems, which rely on a pre-existing map for pose estimation, SLAM concurrently constructs a map in a local coordinate system that is independent of prior information. As a result, SLAM can still be susceptible to drift and may require significant storage capacity, substantial computational resources, and rapid data transmission capabilities (Kumar & Muhammad, 2023).

- Bundle Adjustment is an optimization technique for refining the 3D structure of a scene and the camera poses that captured it. It involves jointly optimizing the positions and orientations of the cameras (or robot poses) and the 3D coordinates of observed landmarks to minimize the overall reprojection error across all images or observations. Unlike SLAM, which processes data sequentially as it becomes available and considers the order of the measurements to maintain temporal consistency, BA treats all observations as a whole without necessarily considering the sequence in which the measurements were captured. This allows for a more global optimization; however, it is typically used as a post-processing step rather than in real-time. The key difference is that while SLAM focuses on real-time operations, relying on sequential measurements and the fusion of multiple sensor inputs, BA does not assume a sequential structure. Instead, it is a global optimization technique used to refine the accuracy of the map and sensor poses after an initial estimation.

2.2 SLAM and Multi-Session Anchoring

In multi-session anchoring, similar to SLAM or a tracking scenario, the objective is to optimize the posterior probability of the poses in a trajectory based on collected measurements. In other words, the goal is to find the poses for which the provided measurements have the highest probability.

However, in multi-session anchoring, the goal is also to find the best alignment between sessions. Each session consists of successive sensor data collected from a specific location at varying time intervals.

2.2.1 Factor Graph Problem

These types of problems can be formulated as a Maximum A Posteriori (MAP) estimate that maximizes the posterior density $p(X|z)$ of the states X given the measurements Z. Instead of using Bayes Net, the problem can be considered as a factor graph factorization in which each factor is proportional to a conditional probability density.

While Bayesian nets provide a practical modeling framework, factor graphs facilitate rapid inference. Like Bayesian networks, factor graphs enable the representation of a joint density as a product of factors (Dellaert, Kaess, et al., 2017).

In robotics, various challenges, including pose estimation, planning, and optimal control, often involve solving optimization problems. These problems typically center around maximizing or minimizing objectives composed of numerous local factors or terms specific to small subsets of variables. Factor graphs allow the encapsulation of this local structure, with factors representing functions related to subsets of variables (Dellaert, 2021).

A factor graph $F = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ comprises nodes connected by edges $e_{ij} \in \mathcal{E}$. The nodes can be of two types: factors $\phi_i \in \mathcal{U}$ and variables $x_i \in \mathcal{V}$. The factor graph represents the factorization of a global function, where each factor is a function of the variables in its adjacency set. Given that X_i is the group of variables x_i connected to a factor ϕ_i, a factor graph specifies the factorization of a global function $\phi(X)$ as

$$\phi(X) = \prod_i \phi_i(X_i).$$

Stated differently, each factor ϕ_i relies solely on the adjacent variables X_i and is connected to other factors via the edges e_{ij}.

An elegant representation of a SLAM problem is called *pose SLAM*, which eliminates the need to directly include landmarks in the optimization process. The focus of pose SLAM is to predict the robot's trajectory based on constraints from odometry and loop closures between the different poses in a trajectory (Jurić et al., 2021). These odometry constraints, describing the relative poses, can be derived from various sources (e.g., camera or wheel encoders).

In general, MAP inference inference is a fundamental probabilistic estimation technique that seeks to find the most likely configuration of a set of variables given observed data. This estimation is based on Bayes' theorem, which expresses how prior knowledge about the variables can be updated with new measurements:

$$p(X|Z) \propto p(Z|X)p(X)$$

Here, $p(Z|X)$ represents the likelihood of the observations given the state, and $p(X)$ is the prior distribution, which encodes any prior knowledge about X. MAP estimation then seeks to maximize this posterior distribution to obtain the most probable state estimate.

In the context of factor graphs, MAP inference corresponds to maximizing the product of all factor potentials (Dellaert, Kaess, et al., 2017):

$$X^{\text{MAP}} = \underset{x}{\text{argmax}} \prod_i \phi_i(X_i) \tag{2.1}$$

Assuming that all factors can be modeled by a measurement function h_i, with normally distributed priors and factors from measurements z_i with zero-mean Gaussian noise models Σ_i, the conditional density $p(z_i|x_i, l_i)$ on the measurement z_i is given by:

$$p(z_i|x_i, l_i) = \mathcal{N}(z_i; h_i(x_i, l_i), \Sigma_i) = \frac{1}{\sqrt{|2\pi\Sigma_i|}} \exp\left\{-\frac{1}{2} \|h_i(x_i, l_i) - z_i\|^2_{\Sigma_i}\right\}$$

Thus, we face factors that are proportional to:

$$\phi_i(X_i) \propto \exp\left\{-\frac{1}{2} \|h_i(X_i) - z_i\|^2_{\Sigma_i}\right\} \tag{2.2}$$

Now, considering the prior distribution on X, we assume a Gaussian prior with mean μ_X and covariance Λ:

$$p(X) = \mathcal{N}(X; \mu_X, \Lambda) \propto \exp\left(-\frac{1}{2}\|X - \mu_X\|^2_\Lambda\right)$$

Taking the negative log of Eq. (2.1) and incorporating the prior term, we obtain the following minimization problem:

$$
\begin{aligned}
X^{\mathrm{MAP}} &= \underset{x}{\operatorname{argmin}} - \log \prod_i \phi_i(X_i) p(X) \\
&= \underset{x}{\operatorname{argmin}} \sum_i \|h_i(X_i) - z_i\|^2_{\Sigma_i} + \|X - \mu_X\|^2_{\Lambda}
\end{aligned}
\tag{2.3}
$$

This contrasts with Maximum Likelihood Estimation (MLE), which does not include the prior term and instead only maximizes the likelihood $p(Z|X)$, leading to:

$$
X^{\mathrm{MLE}} = \underset{x}{\operatorname{argmin}} \sum_i \|h_i(X_i) - z_i\|^2_{\Sigma_i}
$$

Thus, while MAP incorporates prior knowledge into the estimation process, MLE relies solely on the observed data.

2.2.2 Encounters or Loop Closures

In the context of multi-session anchoring, inter-session, or between sessions, loop closure detections, also called *encounters* $\mathbf{c}$ (which are also poses in the special Euclidean group SE(3)), can be added to the non-linear least squares formulation in Eq. (2.3) with the following Gaussian measurement equation:

$$
\mathbf{c} = h\left(\mathbf{x}_R, \mathbf{x}_Q\right) + \eta,
$$

where $h(.)$ is a relative measurement prediction function, and η is a normally distributed zero-mean measurement noise with covariance Σ_c. Furthermore, $\mathbf{x}_R$ and $\mathbf{x}_Q$ are the sensor poses in the two sessions $\mathcal{S}_{\mathcal{R}}$ and $\mathcal{S}_{\mathcal{Q}}$, respectively. This yields the following conditional density $p(\mathbf{c}|\mathbf{x}_R, \mathbf{x}_Q)$ on the measurement $\mathbf{c}$

$$
p(\mathbf{c}|\mathbf{x}_R, \mathbf{x}_Q) = \frac{1}{\sqrt{|2\pi\Sigma_c|}} \exp\left\{-\frac{1}{2}\left|\left|h(\mathbf{x}_R, \mathbf{x}_Q) - \mathbf{c}\right|\right|^2_{\Sigma_c}\right\}.
$$

Similarly, an odometry model $f(.)$, which usually incorporates a scan-matching process, among other techniques, produces constraints $\mathbf{u}_i^s$ between consecutive poses: $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$.

Unifying the encounter measurement model $h(.)$ together with the odometry model $f(.)$ in Eq. (2.3), we obtain the following equation (omitting intra-session loop closures for simplicity).

$$X^{\text{MAP}} = \underset{x}{\operatorname{argmin}} \left\{ \sum_{\mathcal{S}} \left(\|\mathbf{p}_s - \mathbf{x}_{s,0}\|^2_{\Sigma_P} + \sum_{i \in M_s} \|f_i\left(\mathbf{x}_{s,i}, \mathbf{u}_{s,i}\right) - \mathbf{x}_{s,i+1}\|^2_{\Sigma_O} \right) + \sum_{j \in N_e} \|h_j\left(\mathbf{x}_{R,j}, \mathbf{x}_{Q,j}\right) - \mathbf{c}_j\|^2_{\Sigma_c} \right\} \tag{2.4}$$

Where $\mathcal{S} \in \{\mathcal{S}_Q, \mathcal{S}_R\}$, M_s is the number of poses in the session $\mathcal{S}$, and N_e is the number of encounters between sessions.

Here, the initial pose of each session is directly incorporated as a prior factor $\mathbf{p}_s$. This fixes the initial pose to the origin, effectively eliminating that gauge of freedom, i.e., assigning a local reference coordinate system to each session.

2.2.3 Anchor Nodes

As in a multi-robot mapping problem, having two sessions or more requires a strategy to handle the fact that the sessions can have different initial poses and, therefore, other initialization prior (Lajoie & Beltrame, 2024).

Anchor nodes are employed to address this problem and facilitate the integration of inter-session constraints.

The anchor Δ_Q is a SE(3) pose for the session $\mathcal{S}_Q$ that determines how the entire trajectory is positioned concerning a global coordinate frame.

Essentially, the individual pose graphs of each session are maintained in their respective local frames and are bound with anchor factors to the global frame. For each session, an anchor node is added to the pose graph problem as the first pose of the session; this pose can be selected arbitrarily (usually set to the origin).

During the initial encounter, no modifications are made to the pose graphs of the respective sessions; only the anchor nodes change, bringing both graphs to a global coordinate system where they can be compared. In subsequent encounters, information can propagate between the two pose graphs, similar to the scenario of loop closures in a single session. The incorporation of anchor nodes makes efficient updates and quick optimization feasible.

As described by B. Kim et al. (2010), the anchor nodes allow us to estimate the offset between sessions. Moreover, they provide faster convergence to least-squares solvers and

allow each session to optimize their poses before considering global constraints, such as from inter-session loop closures (Ozog et al., 2016).

This feature is advantageous for *long-term mapping* since it enables the production of the first consistent map of the environment when the data is gathered. Whenever a map containing a new session is constructed in a posterior period, and at least one encounter is detected, the anchor nodes allow the computation of the transformation that aligns this recent session with the previously acquired session. Subsequent inter-session loop closure detections will allow correction and improvement of both sessions.

After concluding the theoretical introduction to the method for aligning multiple sessions that will be presented in Chapter 5, the subsequent section will delve into the latest State-of-the-art (SOTA) techniques for achieving alignment with a reference map, with a particular focus on BIM models. Prior to this, the section will provide a concise literature review of visual and LiDAR-based odometry and SLAM systems.

Chapter 3

Related Work

In this chapter, the SOTA in Visual-SLAM (V-SLAM) and LiDAR-based localization and SLAM are discussed. Subsequently, methods that use maps as a reference to enhance localization and mapping systems are examined.

3.1 Visual Pose Estimation

V-SLAM is a critical component in robotics and computer vision, enabling systems to simultaneously localize themselves and map their surroundings primarily using camera measurements. In this section, these systems are broadly categorized into three main groups. The first group consists of camera-only V-SLAM or Visual Odometry (VO) systems, which rely solely on measurements from monocular, stereo, Red-Green-Blue-Depth (RGB-D), or event cameras. The second group includes Visual-Inertial Odometry (VIO) systems, which integrate additional IMU measurements with camera data. This integration has demonstrated significant potential in various application scenarios, enhancing the robustness and accuracy of V-SLAM systems. The third group encompasses alternative approaches, such as those integrating GPS measurements for enhanced localization and others focusing on 3D semantic understanding.

3.1.1 Visual-only Approaches

Perhaps the first way of pose estimation with camera sensors was with monocular cameras. In this regard, Zienkiewicz et al. (2016) presented a method for real-time surface reconstruction using monocular cameras, which adapts to varying levels of detail by dynamically tessellating a triangular mesh. Platinsky et al. (2017) examined the performance differences between sparse joint optimization and dense alternation in monocular VO. They proposed a method for comparing the accuracy of SLAM frontends, demonstrating relative parity between the approaches under current computational capabilities. More recently,

Lukierski et al. (2022) explored the use of monocular multi-directional cameras to estimate the dimensions of enclosed spaces, enhancing the capability to quickly and accurately map indoor environments.

The V-SLAM can also be classified as feature-based or direct SLAM depending on how the measurements are used. The feature-based SLAM repeatedly detects features in images and utilizes descriptive features for tracking and depth estimation (Azzam et al., 2020). Some fundamental frameworks for this feature-based system include MonoSLAM (Davison et al., 2007), ORB-SLAM versions one, two and three (Campos et al., 2021; Mur-Artal et al., 2015; Mur-Artal & Tardós, 2017), and SOFT2 (Cvišić et al., 2022). Instead of using any feature detectors and descriptors, the direct SLAM method uses the whole image. Examples of direct SLAM include LSD-SLAM (D. Caruso et al., 2015; Engel et al., 2014), and SVO (Forster et al., 2017).

In the realm of dynamic environments, B. Xu et al. (2019) proposed MID-Fusion, an octree-based object-level multi-instance dynamic RGB-D SLAM system. This system provides robust camera tracking and continuously estimates geometric, semantic, and motion properties for objects in the scene. In parallel, Vespa et al. (2019) introduced an adaptive-resolution octree-based volumetric SLAM pipeline that dynamically selects the appropriate integration scale based on sensor resolution and distance. Their approach improves reconstruction quality and efficiency. Henning et al. (2022) introduced BodySLAM, a monocular SLAM system that jointly estimates human body parameters and camera poses. Their novel human motion model improves the accuracy of both human body and camera pose estimates.

3.1.2 Visual-Inertial Approaches

Combining visual and inertial measurements has become popular in mobile robotics since the two sensing modalities offer complementary characteristics that make them the ideal choice for accurate VIO or visual-inertial SLAM.

VIO research has developed this into two primary methodologies: tightly coupled and loosely coupled systems. Loosely coupled VIO systems handle visual and IMU data separately before selectively fusing or optimizing the results, such as in the works by Konolige et al. (2011) and Tardif et al. (2010). The loosely coupled approach reduces computa-

tional complexity and enhances system flexibility, though with potential compromises in accuracy. In contrast, tightly coupled VIO systems, exemplified by the MSCKF method proposed by Mourikis and Roumeliotis (2007), integrate camera image data and IMU measurements within a unified optimization framework, ensuring high accuracy and robustness at the expense of increased computational complexity. Notable examples of similar methods include OKVIS (Leutenegger, 2020), OKVIS2 (Leutenegger, 2022), VINS-MONO (Qin et al., 2018), and ICE-BA (H. Liu et al., 2018).

DM-VIO (von Stumberg & Cremers, 2022) is a monocular VIO system that introduces delayed marginalization and poses graph BA to improve accuracy. Delayed marginalization allows for updated linearization points and better integration of IMU data, enhancing photometric uncertainty capture and scale estimation. DM-VIO outperformed even stereo-inertial systems, using just a single camera and an IMU.

More recently, Laina et al. (2024) focus on scalable autonomous drone flight in forest environments using visual-inertial sensors. Their system leverages visual-inertial SLAM for accurate state estimation and introduces a sub-mapping framework to manage drift and corrections. This approach ensures safe and efficient navigation in dense, unstructured environments without relying on LiDAR.

Other approaches have focused on visual-inertial dense reconstruction; for example, Laidlow et al. (2017) proposed a RGB-D-inertial SLAM system that jointly optimizes camera pose, velocity, IMU biases, and gravity direction while maintaining a globally consistent, fully dense surfel-based 3D reconstruction. In the same direction, Xin et al. (2023) proposed SimpleMapping, a method that uses sparse depth from VIO and a multi-view stereo neural network to achieve high-quality 3D reconstruction.

3.1.3 Other Advancements in Visual Pose Estimation

To mitigate long-term drift and ensure scale observability, integrating VIO systems with GNSS measurements has been investigated, as GNSS provides absolute measurements in the global frame (Cioffi & Scaramuzza, 2020). Early research involved aligning local VIO estimations to a global coordinate system by loosely coupling GNSS observations within optimization frameworks or using an Extended Kalman Filter (EKF) framework (Leutenegger, Lynen, et al., 2014; Leutenegger, Melzer, et al., 2014; Leutenegger & Siegwart, 2012;

Yu et al., 2019). More recent approaches tightly fuse visual and inertial measurements with raw GNSS data, such as Doppler shifts and pseudo ranges (J. Liu et al., 2020). For example, Cao et al. (2021) propose a non-linear optimization-based GNSS-Visual-Inertial-Odometry system for real-time, drift-free state estimation. Another method, proposed by Boche et al. (2022), addresses the issues related to global reference frame initialization and compensates for GPS signal outages, enhancing trajectory accuracy and robustness.

Expanding on semantic understanding, Landgraf et al. (2021) developed SIMstack, which leverages a depth-conditioned Variational Auto-Encoder (VAE) to predict 3D shapes and instance segmentation from single depth views, proving useful for applications like precise object grasping in robotics. In object pose estimation, Merrill et al. (2022) presented a keypoint-based object-level SLAM framework that provides globally consistent 6-DoF pose estimates. Their system uses camera pose information from SLAM to track key points on symmetric objects and predict Gaussian covariance for key points. In another work, Papatheodorou et al. (2023) introduced a framework for object-centric exploration using Micro Aerial Vehicle (MAV). This approach not only maps the environment but also identifies and reconstructs specific objects with high detail, thereby enabling more context-aware exploration.

DirectTracker (Gladkova et al., 2022) combines direct image alignment for short-term tracking with sliding-window photometric bundle adjustment for 3D object detection. It refines object proposals using an optimization-based cost function that integrates 3D and 2D cues, and evaluates performance using the higher-order tracking accuracy metric. (Muhle et al., 2023) propose a method that introduces a differentiable nonlinear least squares framework to address uncertainty in relative pose estimation from feature correspondences. It features a symmetric probabilistic normal epipolar constraint and a technique for estimating feature position covariance.

Event cameras, as discussed by Gallego et al. (2022), offer high temporal resolution, dynamic range, and low power consumption. Their review highlights the potential of event-based vision in SLAM applications, particularly in challenging fast-movement scenarios for traditional cameras.

Further enhancing VO, Naumann et al. (2024) introduced Neural Radiance Fields for Visual Odometry (NeRF-VO), a system that integrates learning-based sparse VO with

neural radiance fields to achieve superior performance in camera tracking and dense reconstruction, outperforming SOTA methods across various datasets.

Abaspur Kazerouni et al. (2022), D. Cai et al. (2024), Jia et al. (2022), Lai (2022), and Macario Barros et al. (2022) provide a comprehensive review of V-SLAM systems.

These advancements illustrate the continuous evolution of visual SLAM technologies, addressing various challenges and enhancing their accuracy, robustness, and efficiency.

3.2 LiDAR-Based Pose Estimation

Recent advancements in hardware have enabled SLAM and pose estimation research to achieve more accurate 3D representations of the environment. In comparison with cameras, LiDAR sensors offer more precise distance measurements to objects within the FoV of the sensor, significantly enhancing the accuracy of localization and mapping systems. In the following subsections, the SOTA in LiDAR-based odometry and SLAM systems are discussed.

3.2.1 LiDAR-only Approaches

Some of the earliest LiDAR-based SLAM approaches were founded on PF algorithms, such as FastSLAM (Montemerlo, 2003) and Gmapping (Grisetti et al., 2007), the latter of which remains widely used today.

Subsequently, the algorithms evolved to utilize pose-graph and optimization back-ends (Grisetti et al., 2010; Kaess et al., 2008, 2012; Kümmerle et al., 2011), ensuring better coherence of acquired measurements and reducing long-term drift. For example, Cartographer (Hess et al., 2016) introduced real-time loop closure detection in SLAM, using a front-end with a Ceres-based scan-matcher for building trajectories and a back-end with Sparse Pose Adjustment to reduce errors upon revisiting locations. SLAM Toolbox (Macenski & Jambrecic, 2021), based on Karto SLAM (Konolige et al., 2010), similarly employs local and global optimization, offering various mapping modes and better handling of environmental changes. It filters out incorrect constraints strictly, whereas Cartographer uses the Huber loss function for loop closure, which, as will be discussed in Section 4.3, can affect localization in changing environments.

The previously mentioned algorithms were developed to work mainly with 2D LiDAR sensors; the development of 3D LiDARs has allowed us to acquire more information from the space in the surroundings, and new methods have also emerged.

(J. Zhang & Singh, 2014) introduced LiDAR Odometry and Mapping (LOAM), perhaps one of the most influential 3D LiDAR pose estimation algorithms. LOAM estimates the robot's odometry by registering sequential scans using Iterative Closest Point (ICP), leveraging planar and edge features to construct a sparse feature map. This approach has inspired several subsequent methods, such as LeGO-LOAM (Shan & Englot, 2018) and F-LOAM (H. Wang et al., 2021). However, these methods demand parameter tuning for feature extraction, which is heavily influenced by the sensor's resolution and the structure of the environment.

Behley and Stachniss (2018) proposed the surfel-based SuMa for LiDAR odometry and mapping, later extended to include semantics (X. Chen et al., 2019) and handle dynamic objects (X. Chen et al., 2021). Y. Wang et al. (2021) propose an efficient 3D LiDAR reconstruction framework designed for large-scale exploration tasks. Their system integrates long-range LiDAR scans at high frequency and supports dynamic correction of 3D reconstructions.

Some approaches address the odometry estimation, focusing on real-time performance and accuracy. For example, Pan et al. (2021) introduced a multi-metric approach (MULLS) with robust results across various scenarios but requiring extensive parameter tuning. Dellenbach et al. (2022) proposed CT-ICP, incorporating IMU-free motion distortion correction into registration, yielding excellent results but with added complexity and the need for prior knowledge of the robot's motion profile. Vizzo et al. (2023a) advocating for a constant velocity model, propose KISS-ICP a LiDAR-only odometry method that relies on few parameters and does not require prior motion profile knowledge. By minimizing a simpler point-to-point metric, KISS-ICP achieves comparable or superior odometry performance, with a voxelized, downsampled point cloud map representation simplifying implementation.

However, while the constant velocity assumption may be valid for data collected using LiDAR mounted on autonomous vehicles with straightforward motion patterns, such as in the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) raw

dataset (Geiger et al., 2013), it fails to capture subtle movements and is generally ineffective for data gathered with handheld devices or Unmanned Vehicles (UVs) in both indoor and outdoor environments. Therefore, aiming to address challenging aggressive movement situations, X. Zheng and Zhu (2023) propose Traj-LO. This method demonstrates that by employing a continuous-time perspective and parameterizing LiDAR movement with a continuous trajectory, LiDAR alone can achieve robust and effective performance, even outperforming methods that rely on additional inertial sensors.

3.2.2 LiDAR-Inertial Approaches

Similar as discussed previously in V-SLAM (Subsection 3.1.2), a trend in LiDAR odometry is integrating IMU data (Bai et al., 2022b; K. Chen et al., 2023a, 2023b; Shan et al., 2020b; Z. Wang et al., 2023; Wu et al., 2023; W. Xu & Zhang, 2021; W. Xu et al., 2022). This integration is named LiDAR-Inertial Odometry (LIO). For example, LIO-SAM (Shan et al., 2020b) advanced the field with a tightly coupled, factor-graph optimized framework, enhancing odometry accuracy in dynamic environments.

One of the main advantages of incorporating IMU measurements is the possibility of undistorted single LiDAR scans, which were captured during fast motion. For example, Direct LiDAR Inertial Odometry (DLIO) (K. Chen et al., 2023a, 2023b) draws inspiration from Forster et al., 2016 to allow for parallel point-wise motion correction incorporating a motion model with constant jerk and angular acceleration, leveraging IMU measurements.

3.2.3 Other Advancements in LiDAR-based Pose Estimation

Some researchers have also integrated camera measurements, leading to approaches known as LiDAR-Inertial-Visual Odometry (LIVO) (Lin & Zhang, 2022; Lin et al., 2021; C. Zheng et al., 2022). For example, Boche et al. (2024) presents a tightly-coupled LiDAR-Visual-Inertial SLAM system and 3D mapping framework. Their approach introduces a novel probabilistic formulation of LiDAR residuals for scalable large-scale environments, achieving SOTA pose accuracy and producing globally consistent volumetric occupancy submaps. While integrating multiple sensor modalities into a single system has proven robust in scenarios where single sensor modalities fail, this tight integration significantly

increases system complexity and computational demands, necessitating also calibrated, time-synchronized data from the different sensors.

For further details on the development of LiDAR-based SLAM, refer to the literature reviews by Huang (2021), Nam and Gon-Woo (2021), Tee and Han (2021), and Y. Zhang et al. (2024).

These advancements highlight the growing sophistication of SLAM and mapping technologies, pushing the boundaries of autonomous navigation and large-scale mapping with innovative solutions in both LiDAR and visual-inertial domains.

3.3 Map-based Pose Estimation

This section will provide an overview of the SOTA approaches that intend to align sensor measurements, such as those acquired with LiDAR or camera sensors, to prior building information, such as BIM models, floor plans, or point clouds.

3.3.1 Visual Approaches

Several studies have approached the alignment of measurements from camera (depth or monocular) sensors with BIM models or reference maps in two main ways: (1) as a global localization problem and (2) as a pose-tracking problem.

In the global localization problem, Acharya et al. (2022) introduced BIM-PoseNet, utilizing synthetic images from a 3D indoor model to achieve a 2-meter accurate camera pose without an initial position. Haque et al. (2020) localized an Unmanned Aerial Vehicle (UAV) in the coordinate system of a BIM models by detecting doors and windows in red, green, and blue (RGB) images, using You Only Look Once (YOLO) for object detection and ORB-SLAM2 (Mur-Artal et al., 2015) for 3D mapping.

In the pose-tracking approach, Kropp et al. (2018) focused on image-to-4D BIM model registration using line segments as features, with manual intervention for initial registration. Asadi et al. (2019) proposed an augmented monocular SLAM algorithm for real-time localization; however, it is limited to constant velocity acquisition. Boniardi, Valada, et al. (2019) proposed a clutter-handling method using a convolutional neural network for

layout prediction and a particle filter algorithm for pose tracking using a floor plan as a reference map. Acharya et al. (2019) introduced BIM-Tracker, achieving real-time camera pose tracking with an accuracy of over 10 cm in dynamic environments.

Other methods addressed the challenge of creating a coherent 3D map of the environment aligned with a given reference map. Sokolova et al. (2022) presented the Floorplan-Aware Camera Poses Refinement (FACaP) method, aligning Visual-SLAM maps with floor plans using semantic segmentation and an optimization model considering geometric, floor-to-plane and wall-to-floorplan terms for map correction.

3.3.2 2D LiDAR-based Approaches

Follini et al.2020 show how the standard Adaptive Monte Carlo Localization (AMCL) technique may be utilized to obtain the transformation matrix between the robot reference system and an extracted 2D map from the BIM model. They also state that the AMCL algorithm could overcome small objects that are not present in the BIM model due to the probability distribution of its beam model.

AMCL (Pfaff et al., 2006) is the current de facto standard localization algorithm for estimating the pose of mobile robots within known 2D environments. It is implemented in the navigation stack of ROS and is widely used for robot localization in 2D OGMs. AMCL leverages Kullback-Leibler distance (KLD)-sampling PF to sample the particles adaptively; in this manner, the error of the pose-estimated distribution stays within an acceptable range, and the method remains computationally efficient (Fox et al., 1999).

The same technique was applied by Prieto et al. (2020), S. Kim et al. (2021), Karimi et al. (2021), and K. Kim and Peavy (2022) to localize a wheeled robot in a 2D OGM produced from a BIM model. The primary distinction between these strategies is how they extract the OGM from the BIM model.

An OGM discretizes the environment into 2D square cells with a predetermined resolution; the value in each cell reflects the likelihood that an obstacle occupies the cell. Thus, an OGM allows distinguishing whether a space is free, occupied, or undiscovered.

Prieto et al. (2020) make use of the geometry of the spaces in the Industry Foundation Classes (IFC) file as well as the location and size of each opening, in contrast to Follini

et al. (2020), who use the vertices of elements that intersected a horizontal plane and the Open CASCADE viewer to create an OGM in *pgm* format.

Karimi et al. (2020) created Building Information Robotic System (BIRS), an ontology that allows the generation and transfer of topological, semantic, and metric maps from a BIM model to ROS. An optimal path planner was included in the tool in (Karimi et al., 2021), incorporating crucial elements for the evaluation of the construction. However, this method still does not incorporate Mechanical, Electrical, and Plumbing (MEP) equipment.

A technique to transform an IFC file into a ROS-compliant Simulation Definition Format (SDF) world file appropriate for robot job planning was implemented by S. Kim et al. (2021). They evaluated their strategy for an automatic painting of interior walls. The prototype includes a converter that generates a ROS-compliant world file from IFC file and subprocesses that perform localization, navigation, and motion planning.

Later, a method to turn an IFC model into an Universal Robot Description Format (URDF) building environment was proposed by K. Kim and Peavy (2022) in order to add dynamic objects and for the purpose of door inspection. From this point, a robot may directly access lifecycle information from the BIM model for job planning and execution. Once they have the URDF model, they use PgmMap (H. Yang, 2018) to extract an OGM from it.

For 2D-LiDAR localization, Hendrikx et al. (2021) propose a method that uses a robot-specific world model representation taken directly from an IFC file rather than from an OGM. In their factor graph-based localization strategy, the system receives information about the lines, corners, and circles in the immediate environment of the robot and builds data linkages between those items and the laser readings. They updated and assessed their approach for global localization in (Hendrikx et al., 2022), producing superior results when compared to AMCL.

Boniardi et al. (2017) uses an architectural floor plan based on Computer-aided Design (CAD) rather than a BIM model. They use a Generalized ICP (GICP) implementation for scan matching together with a pose graph SLAM system in their localization and mapping system. They transform a CAD floor plan into a 2D binary image and use it for robot localization in a wearhouse-like scenario. Later, they suggested an improved pipeline that

outperformed Monte Carlo Localization (MCL) in the pose tracking problem for long-term localization and mapping in dynamic situations Boniardi, Caselitz, et al. (2019).

General Monte Carlo Localization (GMCL) demonstrated better performance than AMCL (Alshikh Khalil & Hatem, 2021). It implements three additional particle filters: Optimal, Intelligent, and Self-Adaptive. The Optimal PF reduces the computational complexity of the rejection sampling method by adding auxiliary particles, while the Intelligent PF recalculates the weights of a third of the particles with small weights to better estimate the pose and orientation of the robot. The Self-Adaptive PF computes a Similar Energy Region (SER), which represents a set of energy cells whose energy is similar to the sensor reading's energy, spreading particles over the map. GMCL achieves a 13% improvement in pose tracking and a 74% success rate in global localization, compared to AMCL's 28%.

3.3.3 3D LiDAR-based Approaches

Other approaches investigated 3D LiDAR localization using 3D reference maps.

Gawel et al. (2019) presented a very accurate robotic building construction system. They use ray tracking with three *laser distance sensors*, a 3D CAD model, and a robust state estimator that merges IMU, 3D LiDAR, and wheel encoders to locate the end-effector with subcentimeter precision. They did this by taking several orthogonal range measurements while the robot was static.

In the technique proposed by Ercan et al. (2020) and Blum et al. (2020), the 3D LiDAR scan is aligned with the BIM model using the ICP algorithm.

While Ercan et al. (2020) limits the alignment to a few carefully chosen reference-mesh faces to overcome ambivalence, Blum et al. (2020) uses picture information to separate the foreground and background in the point cloud and uses only the latter for registration. The pipeline was then extended to provide a self-improvement semantic perception technique that can better handle environmental clutter and increase accuracy (Blum et al., 2021).

To take advantage of the high performance of Google Cartographer (Hess et al., 2016) for localization, Moura et al. (2021) suggest a method to create *.pbstream* maps from BIM models. Although this approach is quite practical, since they only employ Cartographer in localization mode, their method does not create a map of the environment if the robot is

not localized and inside the boundaries of the reference map. This means that the robot's initial position must be inside the boundaries of the prior reference map (i.e., the BIM model) in order for it to be localized and a new map to be created.

Oelsch et al. (2021) propose Reference-LOAM (R-LOAM), a technique that uses a combined optimization that includes point and mesh characteristics for 6-DoF UAV localization. Later, in (Oelsch et al., 2022), they improved their approach using pose-graph optimization to decrease drift even when the reference object is not visible.

A semantic ICP approach was presented by Yin et al. (2023). This method uses the 3D geometry and semantic data of a BIM model to achieve a reliable 3D LiDAR localization method. Their system suggests a BIM model-to-Map conversion, turning the 3D model into a point cloud that is semantically enhanced. Their research demonstrates that a 3D LiDAR-only localization can be accomplished using an BIM model in uncluttered environments.

Another exciting strategy, suggested by Shaheer et al. (2022), relies on geometric and topological information in the form of walls and rooms rather than object semantics for localization. They build Situational Graphs (S-Graphs) using these data, which are subsequently used for precise pose tracking. Later, they improved their technique by allowing the acquisition of a map before localization, as well as the posterior matching and merging with an A-graph (extracted from BIM models). The combined map's ultimate designation was an informed Situational Graph (iS-Graph) (Shaheer et al., 2023).

Direct LiDAR localization (DLL) is a fast localization method introduced by Caballero and Merino (2021). They use a registration method based on non-linear optimization of the distance between the points and a reference point cloud. Their method does not require feature extraction to achieve an accurate and fast registration. By correcting the anticipated pose using odometry, the technique can follow the robot's pose with sub-decimeter precision in real-time. Their technique performed better compared to AMCL 3D (Perez-Grau et al., 2017).

3.4 Research gap

Numerous methods have been developed that use reference 2D and 3D maps for LiDAR localization and mapping. Most of them have concentrated on real-time localization without enabling pose-graph-based optimization approaches to provide a more accurate estimation of the calculated poses.

Additionally, practically every method requires the scanning to begin in a known initial pose that must be inside the boundaries of the reference map.

This requirement means that for several methods, there is no chance of localization or the generation of an aligned map if the sensor starts from a location where the reference map is not visible or from where there are large Scan-Map deviations, like in a cluttered environment.

Furthermore, rather than retrieving a posterior accurate, updated, and extended map of the environment and detecting environmental changes, most researchers focused only on improving the accuracy of the pose-tracking process.

In this dissertation, particularly in Chapter 4, some contributions are made to improving pose-tracking performance in changing environments. However, the additional contributions (Chapters 5 and 6) go beyond the real-time constraints and provide strategies that demonstrate the feasibility of creating an aligned, optimized map and calculating accurate 6-DoF poses that closely approximate the ground truth poses. In summary, in this dissertation, the emphasis shifts from faster, less accurate methods to approaches that prioritize precision and correctness in the registration process rather than speed.

Chapter 4

Real-time LiDAR and Image Localization

This chapter presents two novel methodologies aimed at contributing to real-time localization using BIM models or reference 3D/2D maps. The first method delves into a comparative analysis of existing 2D LiDAR real-time localization algorithms. Furthermore, it proposes a framework for transitioning from Occupancy Grid Maps (OGMs) to Pose Graph-based Maps (PGBMs). This framework facilitates the utilization of PFs for rapid global localization, followed by Graph-based Localization (GBL) algorithms for subsequent pose tracking. Here, *"global localization"* refers to the initial process of determining the robot's position within the entire map. At the same time, *"pose tracking"* signifies the continuous refinement of the robot's position and orientation as it moves. The second proposed method to advance real-time localization systems focuses on refining camera poses using a 3D BIM model. The solution involves the extraction of Vanishing Points (VPs) and Vanishing Lines (VLs) from camera images and synthetically generated images from a BIM model.

4.1 Motivation

Real-time localization is a fundamental capability for robots operating in dynamic environments. It allows robots to maintain an accurate understanding of their position relative to the surrounding world on a given prior map.

For effective autonomous robot navigation within a mapped environment, fast and accurate localization is of prime importance. Only with a clear understanding of its current location can a robot efficiently compute a path and autonomously navigate toward its objective while avoiding obstacles.

Sensor selection plays a critical role in achieving real-time localization with cost-effectiveness in mind. Cameras and 2D LiDAR (laser scanners) are often preferred due to their affordability and widespread availability on robots. 2D LiDAR sensors with laser

beams effectively measure distances to surrounding obstacles, offering valuable data for navigation. Meanwhile, camera sensors provide rich visual information, including texture and color details, along with a vertical FoV, which can further enhance both localization and navigation capabilities.

Real-time localization with 2D LiDAR plays a crucial role in enabling robots to navigate efficiently.

Despite the widespread adoption of the Adaptive Monte Carlo Localization (AMCL) algorithm (a PF method) for real-time 2D LiDAR localization (often due to its inclusion in the Robot Operating System (ROS) navigation stack), there is a scarcity of research dedicated to advancing the robustness of such algorithms. This includes improvements in both localization accuracy and the ability to handle discrepancies between sensor scans and the reference map (here referred to as Scan-Map deviations). In contrast, Simultaneous Localization and Mapping (SLAM) systems have witnessed significant advancements. A crucial development was the transition from particle filter approaches to pose-graph optimization with pose-graph maps. This shift enabled SLAM systems to achieve higher precision and robustness, particularly when correcting for substantial drifts that can occur after the scanning of large areas. This research proposes a novel method that facilitates the transition of real-time 2D LiDAR localization from particle filter-based approaches to pose-graph optimization. It is also demonstrated that this transition has the potential to enhance the robustness and accuracy of real-time localization, leading to improved navigation performance for robots.

Cameras offer rich visual information for robot navigation, making them a valuable tool for localization. The second area of research in this chapter explores the refinement of camera poses using a 3D BIM model. While the proposed method does not yet achieve real-time performance, it represents a significant step towards that goal by leveraging the rich information available in 3D BIM models. This advancement paves the way for the future development of fully real-time image-based localization solutions.

4.2 Research Questions

This chapter aims to contribute to the field of robot localization and navigation by providing an answer to the following main research question:

RQ 1. *How can 3D BIM models be leveraged for real-time 2D LiDAR and image localization systems?*

The following are the specific three sub-research questions addressed in this chapter:

RQ 1.1 Transformation of 2D OGMs to PGBMs:

- How can a 2D Occupancy Grid Map (OGM) be transformed into a Pose Graph-based Map (PGBM) to facilitate the transition of localization algorithms into the optimization paradigm?
    - *Rationale:* While generating an OGM from a given map can be straightforward (depending on the reference map), creating an accurate Pose Graph-based Map (PGBM) requires specific expertise. This transformation is crucial for allowing the easy transition of localization algorithms and leveraging their highest performance for specific localization tasks.

RQ 1.2 Performance Evaluation of State-of-the-art (SOTA) 2D LiDAR localization Algorithms:

- How do various SOTA algorithms perform in different environments, characterized by varying levels of clutter (Scan-Map deviations) and dynamic conditions, in the context of pose tracking and global localization tasks?
    - *Rationale:* Assessing the performance of multiple Particle Filters (PFs) and Graph-based Localization (GBL) algorithms under different environmental conditions will provide insights into their robustness and applicability for real-time localization.

RQ 1.3 Correction of Camera Poses Using 3D Models:

- How can poses obtained from a Visual-SLAM system be corrected with the assistance of a 3D model?

- *Rationale:* Integrating 3D BIM models to correct camera poses can enhance the accuracy of the localization of low-cost sensors, thereby improving the overall map precision.

By addressing these questions, this research aims to equip robotic systems with more reliable and accurate real-time localization capabilities, ultimately leading to improved autonomous navigation performance in diverse environments. In the following two sections, a method to enhance 2D-LiDAR real-time localization and another aiming for near real-time camera image localization with 3D models will be presented.

4.3 Real-time 2D LiDAR Localization[1]

Our method to explore and enhance real-time 2D LiDAR localization can be divided into three main steps: **Step 1:** Creation of an OGM from an Industry Foundation Classes (IFC) file (3D BIM model) employing IfcConvert and OpenCV. **Step 2:** Automatic generation of a PGBM out of an OGM with a combination of image processing, coverage path planner, and ray casting. **Step 3:** Robust localization using particle filter algorithm and graph-based localization system.

4.3.1 Step 1: OGM Generation from a BIM Model

For the creation of suitable 2D OGM for robot localization and navigation from complex multi-story IFC models, the IfcConvert tool of IfcOpenSchell (Krijnen, 2015) and image processing techniques are used.

IfcConvert allows the creation of a 2D map in Scalable Vector Graphics (SVG) format with the desired elements in the IFC model that cross a plane at the desired height.

In this case, non-permanent entities such as spaces, windows, and doors are excluded from the resulting 2D OGM by ignoring the corresponding entity names. This exclusion is essential to filter only structural information about the building, enabling further autonomous navigation between the rooms that want to be explored. Besides having the permanent structures in the OGM, and with the aim of global localization and posterior correct pose

[1]Portions of this section were previously published in (M. A. Vega-Torres, Braun, & Borrmann, 2022)

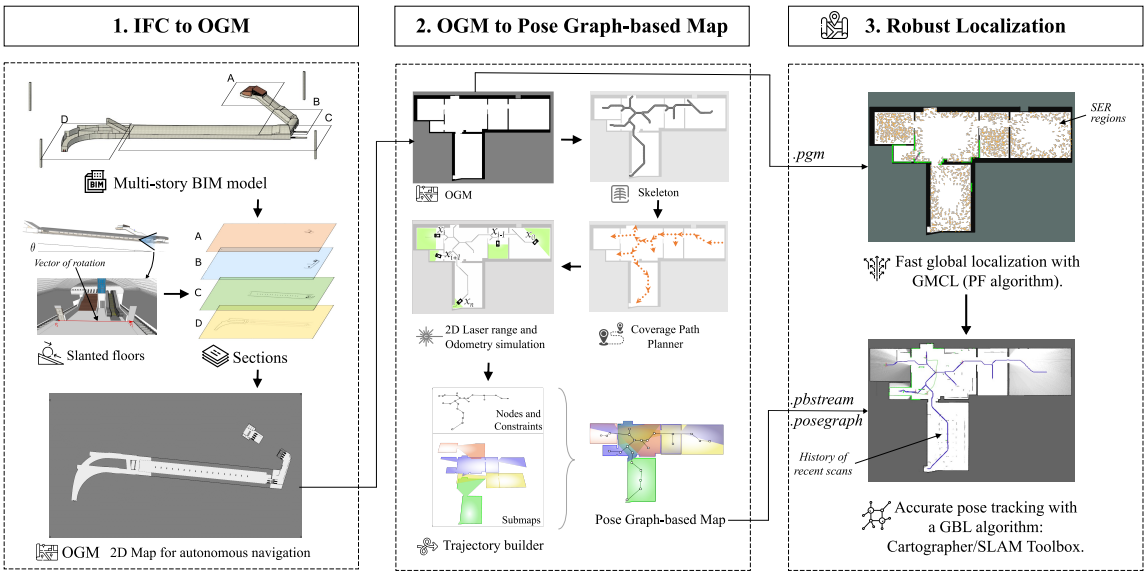


Figure 4.1: Proposed IFC to PGBM for robust 2D-LiDAR localization. In the first step, an OGM is created from multi-story non-convex BIM models, which can have slanted floors; this map is suitable for path planning and autonomous robot navigation. In the second step, a PGBM is generated from the OGM. Finally, in the third step, these maps allow fast global localization and robust pose tracking in changing and dynamic environments.

graph map generation, it is crucial to differentiate between outdoor (unknown) and indoor (navigable) spaces in the OGM. This distinction can be automated by creating a second OGM with all the entities in the IFC file (i.e., with doors, windows, and spaces).

The final separation of outdoor (gray color), indoor (white), and obstacle (black) is done based on the contours in the SVG image. OpenCV allows the processing of the contours depending on their hierarchy, i.e., depending on whether they are inside (child contours) or outside another contour (parent contours).

The resulting file is finally converted to *.pgm*, which, together with its properties (the resolution and origin) in a *.yaml* file can then be loaded into the robotic system as prior environment information, allowing robot localization, path planning, and autonomous navigation.

A similar procedure can be followed for multi-story level buildings. In the particular case of non-overlapping stories, the different OGMs can be merged into a single one if the relative position between them is known. To maintain this spacial relationship, while obtaining the OGMs, reference auxiliary elements with a height equal to the maximum building's height can be included in its surroundings (four of these elements are visible in

the surroundings of the building in the left upper part of Fig. 4.1). With these additional elements, all the OGMs will have the same dimensions, allowing its merging.

Creating 2D OGMs with IfcConvert is relatively straightforward when the desired section is horizontal (parallel with the XY plane). However, if the model has a ramp or a slightly slanted floor, the model must be rotated before the occupancy map is generated.

Favorably, IfcConvert also allows the model to rotate at the desired angle, given a quaternion calculated from the vector of rotation.

4.3.2 Step 2: OGM to Pose Graph-based map Conversion (OGM2PGBM)

The automatic generation of data suitable for GBL methods from BIM models implies the simulation of sequential laser data in the entire navigable space in the model with the corresponding odometry data.

For this aim, the previously generated 2D OGM is used. Applying the skeleton method proposed in (Lee et al., 1994) enables the interconnection of all the rooms in a smooth trajectory.

Subsequently, a Wavefront Coverage Path Planner (Zelinsky et al., 1993) is applied over the navigable area inside a dilated version of the skeleton, allowing finding the waypoints over which the laser will be simulated.

Then, using a ray casting algorithm and without a real-time simulation engine (such as Gazebo), laser sensor data and odometry are simulated following the waypoints found in the previous step. Finally, a trajectory builder merges these sensor data, creating an accurate pose graph-based map, serialized as a *.pbstream* file for Cartographer (Hess et al., 2016) or as a *.posegraph* file for SLAM Toolbox (Macenski & Jambrecic, 2021). For a brief explanation of the basics of Cartographer and SLAM Toolbox systems, refer to Section 3.2.

This pipeline allows the automatic efficient generation of pose graph-based maps (with submaps, nodes, and constraints) from a 2D OGM. As the OGM2PGBM workflow does not require a Gazebo for data simulation, it is faster and more portable than a Gazebo-based pipeline, allowing its execution in an isolated manner. Moreover, since the technique does not consider the complete 3D model but only a 2D OGM, it is very efficient. In

addition, it can be used from any given OGM, which, besides being generated from a BIM model (with the method presented in the previous section), can be generated out of a floor plan or a previously scanned map.

4.3.3 Step 3: Robust Localization

Once the different needed map representations (OGM and pose graph-based maps) are generated from a BIM model, they can be used for robust localization in changing environments. In this contribution it is proposed to take advantage of the Self-Adaptive PF of GMCL to spread particles only in the SER regions and solve the global localization problem efficiently [2]. As it is shown later (in Subsection 4.3.5), PF algorithms being able to represent non-Gaussian distributions can solve the global localization faster than graph-based algorithms. Once an estimated pose is found with a covariance smaller than 0.05, the nodes of GMCL are stopped, and a GBL algorithm can be started. For example, to track the pose of the robot accurately, Cartographer can be activated with the *start_traj* service at the time when GMCL converges and using the *.pbstream* map generated with the method proposed in Subsection 4.3.2. Similarly, SLAM Toolbox can be started with an initial pose; however, with a prior *.posegraph* map.[3].

4.3.4 Experiments

This section presents the evaluation scenarios designed to evaluate the various techniques and details of the implementation and evaluation.

Evaluation Scenarios

As illustrated in Figure 4.2, three different scenarios were conceived to evaluate the different methods. Each scenario increases the level of clutter present in the environment and, therefore, decreases the level of overlap that a perception sensor would have with permanent building objects (such as walls, columns, floors, and ceilings). The latter are the elements that are usually present in a BIM model.

[2]For a more detailed explanation of the fundamentals of AMCL and GMCL systems, please refer to Subsection 3.3.2.

[3]For a more detailed explanation of the fundamentals of Cartographer and SLAM toolbox systems, please refer to Subsection 3.2.1.

(a) (b) (c)

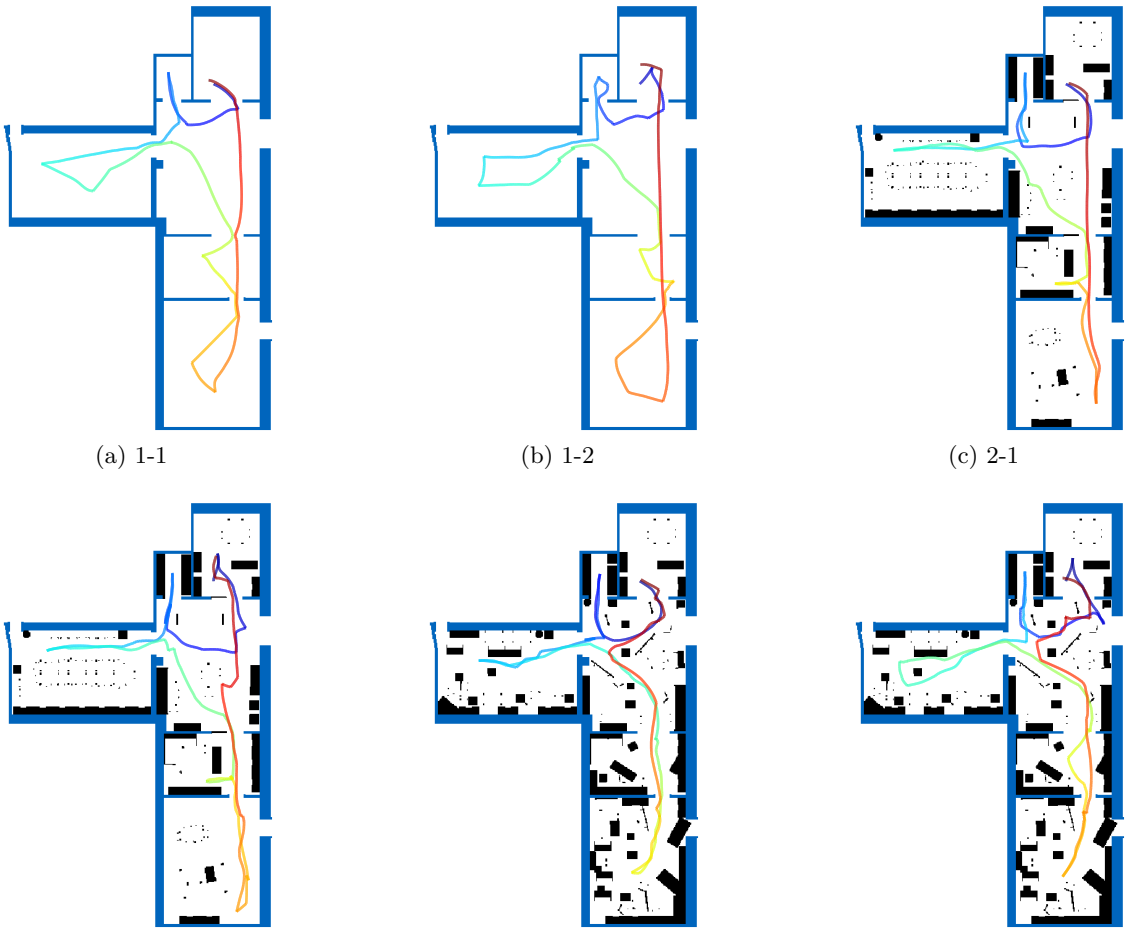
Figure 4.2: Evaluation Scenarios. (a) Empty Room: represents a typical BIM model, without furniture; (b) Reality: represents a standard office environment and is based on real-world TLS data; (c) Disaster: is an environment after a simulated disaster with large Scan-Map deviations.

Additionally, to increase the simulation's realism level, animated walking human models (also called dynamic agents) moving in the environment were added. In scenarios 1 and 2, five humans walk from each room to the closest exit of that room. In the scenario Nr. 3 ("Disaster"), a total of six people move faster, trying to escape through the main door. Once the agents reach their goal, they start again, moving from their initial planned position in an infinite loop.

Gazebo Simulation

To simulate the experimental data Gazebo was used. Once the IFC model is converted to Collada format using IfcConvert, it can be imported into Gazebo. Importing complex IFC models in Gazebo is essential to ensure that every element has its geometric representation. One way to avoid instantiating multiple objects from the same data is using the export capabilities of Blender.

For trustworthy data simulation the collision and visual models were separated. Since LiDAR sensors cannot perceive glass materials, windows and glass doors were removed in the collision models.



(a) 1-1 (b) 1-2 (c) 2-1

(d) 2-2 (e) 3-1 (f) 3-2

Figure 4.3: Sequences of data with the respective OGMs. (a) and (b) correspond to an empty environment (i.e., without furniture) with and without dynamic agents resp.; (c) and (d) similar to the pair (a) and (b) but in a scenario with the furniture as it is in a real-world office; (e) and (f) in a simulated disaster environment. To better visualize the different levels of Scan-Map deviations, the OGM of the empty environment is presented over the other OGMs in blue color. The change in color of the trajectory represents the initial and end position of the robot, with dark blue being the start and red the endpoint.

Robot Simulation

The robot used for the simulated experiments was the holonomic Robotnik SUMMIT XL equipped with a 2D LiDAR Hokuyo UST-10LX. It was commanded with stable linear and angular velocity of at approximately 1 m/s and 1 deg/s, respectively.

Using the URDF model of this robot, it is possible to leverage the different packages of the ROS Navigation Stack for ROS visualization (RViz). One of these packages is NAVFN, which assumes a circular robot and allows it to plan a path from a start point to an endpoint in a grid based on a cost map.

A cost map is an inflated version of the given 2D OGM with a specified amplification radius created to avoid the robot colliding with obstacles while navigating through the environment.

To speed up the usage of the OGM for robot simulation, the Gazebo Plug-in PgmMap creator (H. Yang, 2018), was also implemented, allowing the creation of maps with known origin position. In practice, this step is not required since the alignment between the real world and the map can be retrieved as a localization system result.

It is worth mentioning that using navigational goals instead of single movement commands is very convenient for data simulation since it significantly reduces the probability of collisions, which can make the entire sequence useless.

Following this approach, 2D LiDAR, IMU, Wheel odometry, and ground truth odometry were simulated in the six scenarios (three models with and without dynamic agents). The resulting trajectories of the simulation are presented in Figure 4.3.

Implementation details

Due to the stochastic nature of PF algorithms (AMCL and GMCL) and similarly as done by (Alshikh Khalil & Hatem, 2021), these methods were executed 30 times in each sequence, and the average values were calculated.

Similarly as (Zimmerman et al., 2022), it is considered that a method converges when its pose estimate is within a distance of 0.5 m from the ground truth pose. If after the first 95 % of the sequence, convergence does not happen, then it is considered a failure.

Unfortunately, SLAM Toolbox could not be evaluated for global localization since it does not provide this service. The lifelong mapping mode of SLAM Toolbox was also tested for the matter of completeness; however, it yields unwanted results with poor performance.

4.3.5 Results and Analysis

The libraries provided by (Grupp, 2017a) and (Z. Zhang & Scaramuzza, 2018) were used to calculate the error metrics of the various methods on the different sequences.

Table 4.1: Summary of the quantitative evaluation results for sequences 1-1, 1-2, 2-1, and 2-2. Translational Root Mean Square Error (RMSE) in centimeters and angular RMSE in degrees, respectively.

Method	1-1		1-2		2-1		2-2	
AMCL	8.49	0.44	8.47	0.50	33.68	2.71	37.44	3.26
GMCL	8.27	0.24	7.86	0.24	24.27	2.57	52.38	4.37
SLAM Toolbox	**3.69**	**0.17**	**3.95**	**0.17**	28.69	1.50	23.57	1.50
Cartographer	4.01	0.24	3.96	0.25	**7.19**	**0.15**	**4.11**	**0.21**

Table 4.2: Summary of the quantitative evaluation results for sequences 3-1 and 3-2. Translational RMSE in centimeters and angular RMSE in degrees, respectively.

Method	3-1		3-2	
AMCL	63.04	3.29	65.12	3.37
GMCL	66.60	3.70	126.91	4.46
SLAM Toolbox	**37.84**	**1.34**	**37.96**	**1.70**
Cartographer	**-**	**-**	**-**	**-**

Pose tracking

Tables 4.1 and 4.2 present the translational and rotational RMSE for each sequence for each method evaluated on the pose tracking problem with the ground truth from the simulation. Figure 4.4 shows the resulted trajectories by the different methods in the multiple scenarios. Figure 4.5 presents a summary of the statistics of the translational errors for all the methods in all sequences.

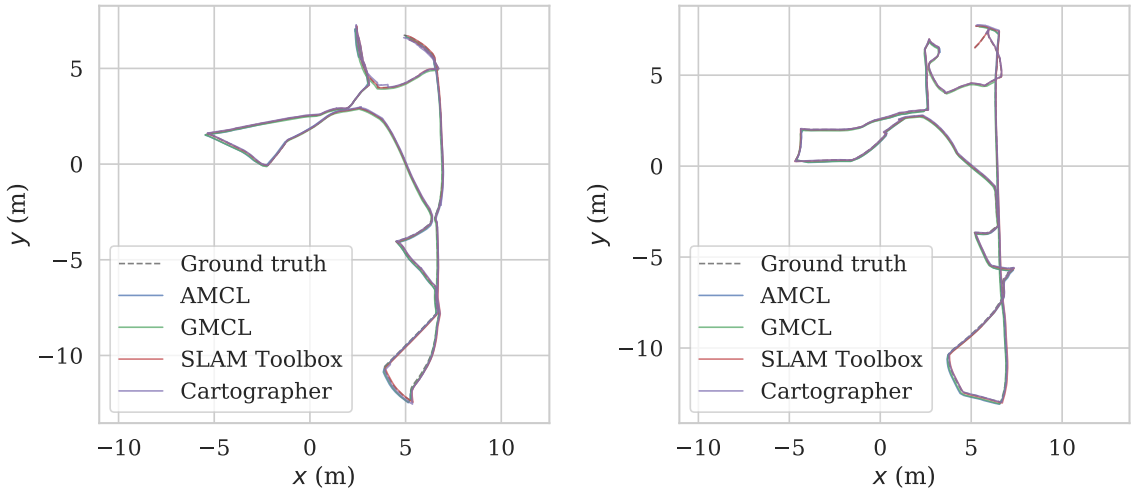
Overall, it can be seen that GBL methods always perform better than PF algorithms in the pose tracking problem.

Among the tested PF algorithms, GMCL performs most of the time better than AMCL. Only in the scenarios 2-2, 3-1, and 3-2, AMCL achieves lower RMSE. In scenarios 2-2 and 3-2 GMCL has a very high translational RMSE.

This shows that the additional filters of GMCL cause the method to be more sensitive to dynamic environments in changing environments.

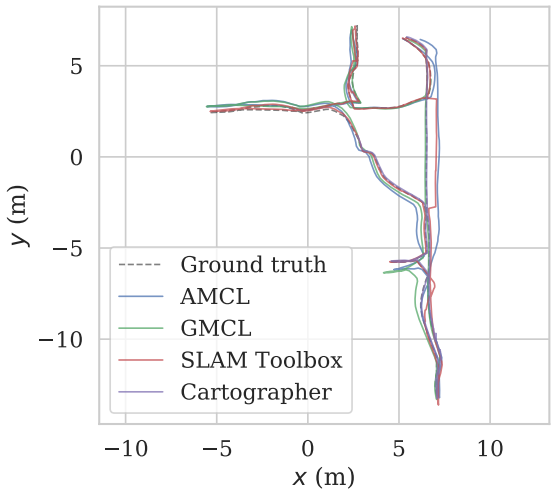
Regarding the GBL algorithms, SLAM Toolbox achieves the best performance in scenarios 1 and 3. As expected, scenario 3 (with the most significant Scan-Map deviations) was the most challenging scenario for all the methods. On top of that, in this scenario, the pure localization mode of Cartographer always found wrong data associations, resulting in

wrong relative constraints that cause localization failure. Therefore Cartographer could not be quantitatively evaluated in this environment, even when an initial approximated pose was provided. Nonetheless, Cartographer achieved an impressive performance in scenario 2 (real-world scenario), accomplishing a translational RMSE four times lower than SLAM Toolbox in the environment without dynamic agents (7.19 cm and 28.69 cm respectively) and almost six times lower in the scenario with dynamic agents (4.11 cm and 23.57 cm respectively).

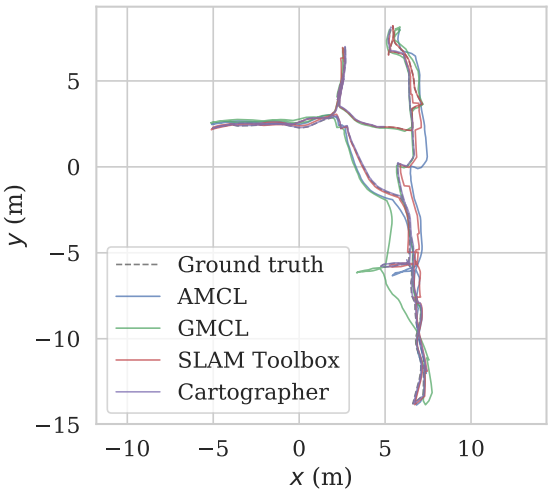


(a) 1-1

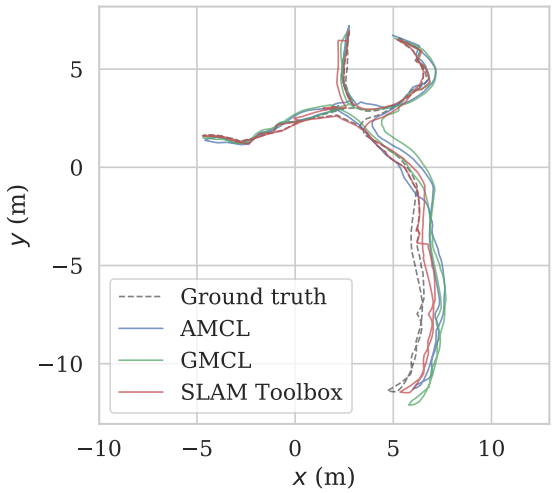
(b) 1-2



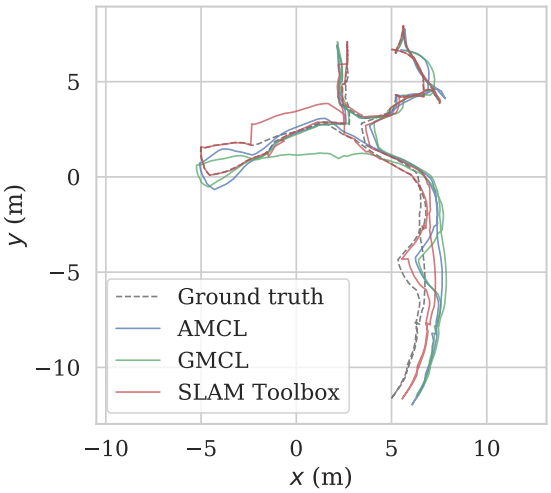
(c) 2-1



(d) 2-2

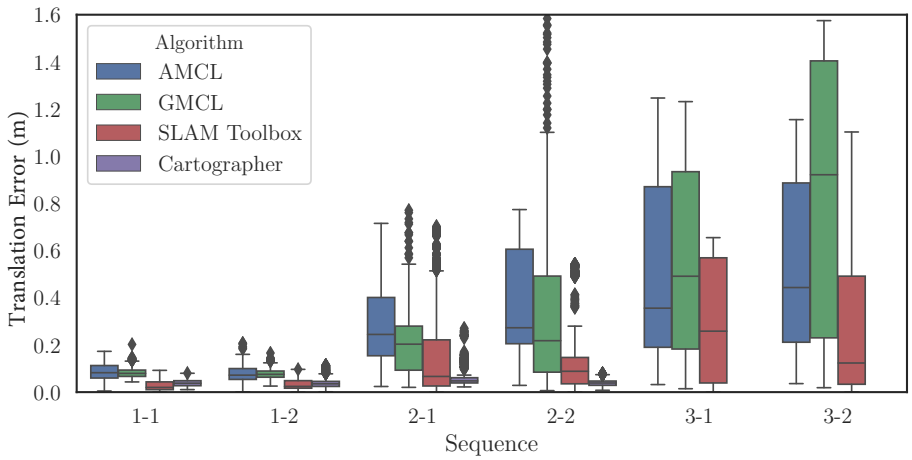


(e) 3-1

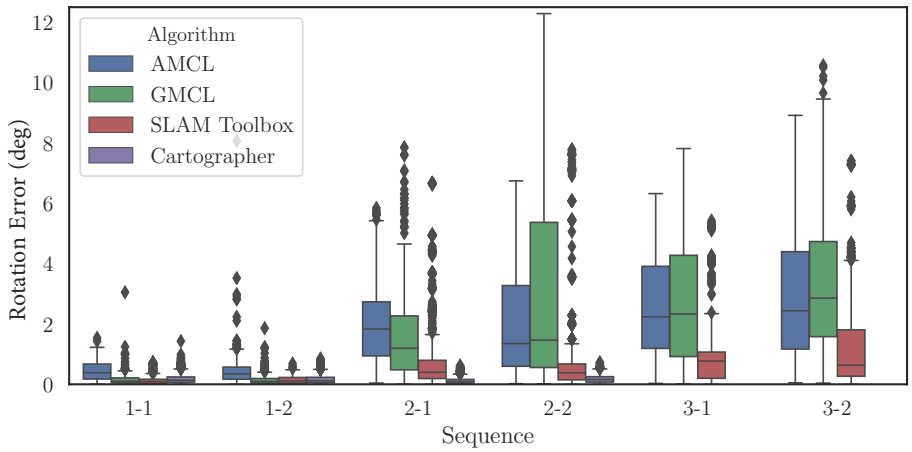


(f) 3-2

Figure 4.4: Trajectories estimated by the different methods in each sequence together with the respective ground truth trajectories. All methods perform relatively well in scenario 1 (sub-figures (a) and (b)). In scenario 2 (sub-figures (c) and (d)) Cartographer manifests a clear superiority. However, the same method is not able to track the robot's pose in scenario 3 (sub-figures (e) and (f)). In this scenario, SLAM Toolbox performs the best.



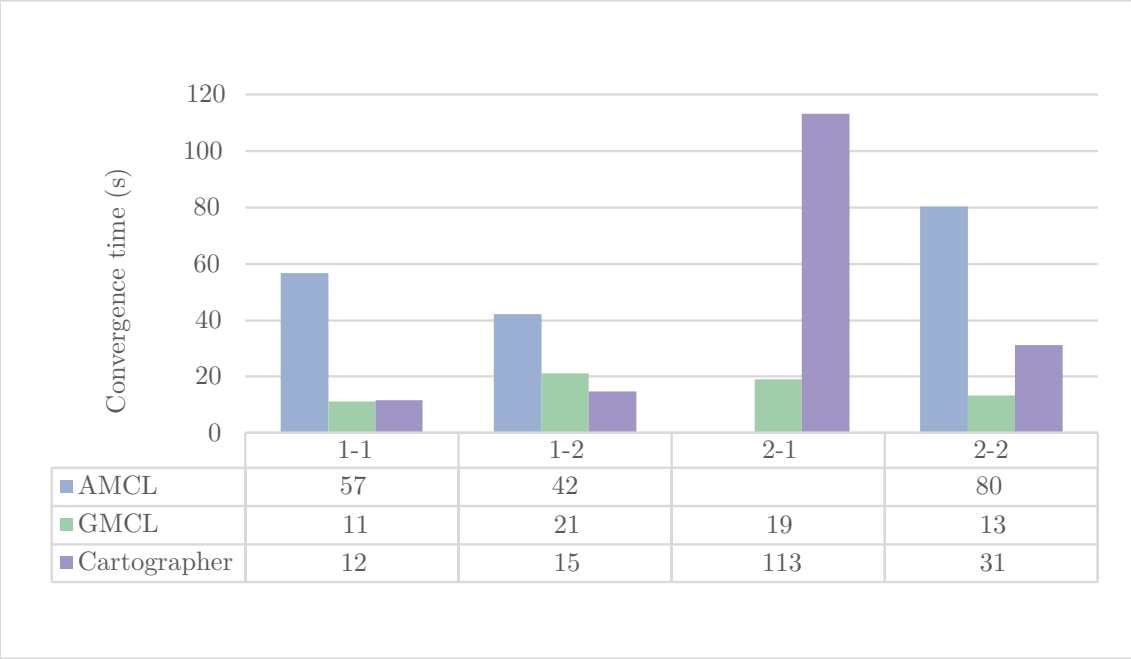
(a) Translational errors



(b) Rotational errors

Figure 4.5: Statistics of the pose error estimates in (a) translation and (b) orientation for each method on the six evaluation scenarios.

Global localization



	1-1	1-2	2-1	2-2
AMCL	57	42		80
GMCL	11	21	19	13
Cartographer	12	15	113	31

Figure 4.6: Convergence time in seconds for the various methods in the different scenarios.

The performance of the different methods regarding convergence time is presented in Figure 4.6.

GMCL, thanks to its Self-Adaptive PF, performs the best in the global localization problem. Only in scenario 1-2 Cartographer shows a slight superiority. Meanwhile, AMCL always takes at least twice as long compared to the other methods to converge to a good pose. In addition, it does not converge in scenario 2-1.

Due to the high level of Scan-Map deviations, none of the implemented methods converge while trying to solve the global localization problem in scenario 3.

In general, Graph-based Localization (GBL) methods (such as SLAM Toolbox or Cartographer) perform best for pose tracking and Particle Filter (PF) methods (in particular GMCL) for global localization. This distinction arises because PF algorithms, which rely solely on the most recent observation to update the belief of the current pose, exhibit robustness in highly ambiguous scenarios; for example, in scenario 3, the PF methods do not fail such as Cartographer. However, this same reliance can lead to significant inaccuracies when facing medium levels of Scan-Map deviations (e.g., scenario 2). Conversely, GBL algorithms leverage a recent history of observations, enabling them to better navigate real-

world scenarios and maintain more accurate pose tracking. Therefore, it is recommended to use a GBL algorithm for accurate Building Information Modeling (BIM)-based (or floor plan-based) 2D LiDAR pose tracking in real-world environments and GMCL for global localization.

The previously presented technique, results, and analysis showed how it is possible to improve 2D-LiDAR real-time localization in changing environments. Rather than relying on map construction with an SLAM algorithm, it is possible to use a reliable map with only permanent elements of the environment (such as a BIM model), allowing for precise pose-tracking and fast global localization even when temporal elements change their position over time.

While 2D-LiDAR systems are commonly integrated into various robotic platforms, cameras offer a more cost-effective alternative with the added advantage of capturing richer information, including texture and vertical geometries within the vertical FoV that 2D-LiDAR cannot measure. Consequently, the following section will delve into a methodology for iteratively refining camera poses using a 3D model as a reference.

4.4 Real-time Image Localization[4]

The proposed methodology to achieve image localization with a 3D model can be divided into three main steps: **Step 1.** Point cloud acquisition with SLAM; **Step 2.** Perspective detection in the BIM view and keyframes, and **Step 3.** Localization improvement to find the fine camera poses. An overview of the workflow can be seen in Figure 4.7.

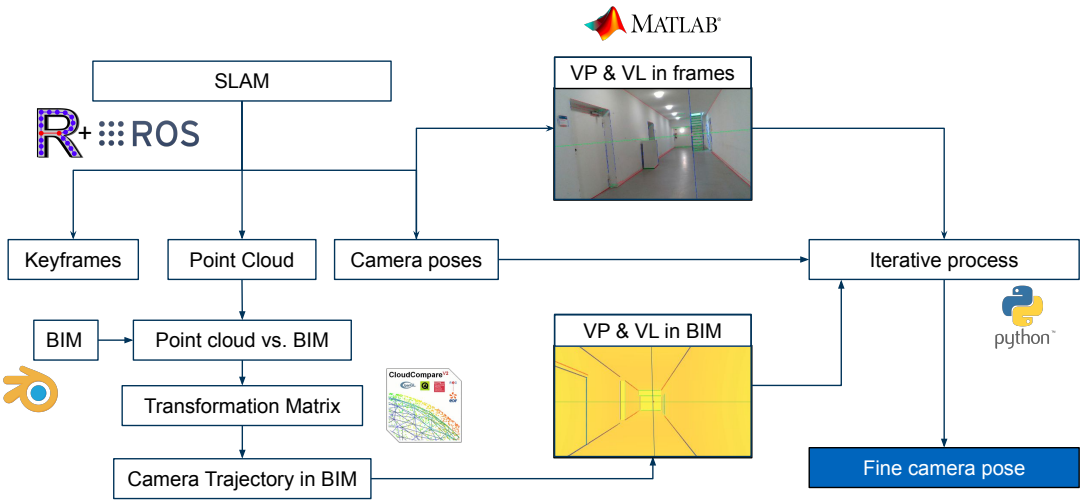


Figure 4.7: Proposed workflow for fine camera pose correction with a BIM model.

4.4.1 Step 1: Point Cloud Acquisition and First BIM Alignment

In order to obtain the rough camera poses, a point cloud acquisition was performed using a RealSense D435i camera with the RTAB-MAP SLAM framework (Labbé & Michaud, 2019). Once a bag file is recorded, a database is created with the help of the "2019-UGRP-DPoom" project (Shinkansan, 2021).

The point cloud database is then opened in RTAB-MAP where the poses in the trajectory are optimized, and from where the final rough camera poses and frames can be exported. Besides that, creating a point cloud database is important since it will later be used as one part of the input data in CloudCompare (CC) for the alignment between the point cloud and the BIM model.

[4]The method described in this section was developed in collaboration with Rafaella Dantas, Simone Peter, and Xingzhou Wang, as part of a seminar project (called Software Lab), supervised by the author of this dissertation. Portions of this section were previously published in (Dantas et al., 2022).

In order to find the camera poses in the BIM coordinate system, the resulting point cloud has to be aligned with the model. This first alignment is done in a semi-automatic manner using CC.

The BIM model, originally in IFC format, needs to be exported as a triangulated mesh (in stereolithography (STL) format), which can be conducted using Revit. Afterward, the BIM model can be imported into CC. Additionally, to render the BIM keyframes, the model is imported into Blender, where the camera object and scripting tools are primarily used. The keyframes are frames extracted every second from the video stream.

The transformation matrix that aligns the point cloud with the BIM ensures that the initial BIM keyframe of the executed trajectory is in the scope of the one from SLAM.

The alignment in CC is conducted after manually selecting three reference points in the point cloud and their correspondences in the BIM model. Finally, a transformation matrix from point cloud to BIM is obtained.

Once the correct transformation matrix and scale of the point cloud are found with CC, the coordinates of the rough camera trajectory obtained previously with RTAB-MAP are transformed into the BIM coordinate system.

4.4.2 Step 2: Perspective Detection

One significant task in the computer vision community is to extract 3D information from 2D images (Asadi et al., 2019). The estimation of VPs is important for performing the localization improvement step. In this contribution, the algorithm proposed in (Hedau et al., 2010)was applied in order to conduct this step. The vanishing points are points on the image plane, where 2D perspective projections of mutually parallel lines in 3D space intersect. Hedau et al. (2010) propose to compute the parallel lines by the edge detection approach. Afterward, the triplet of orthogonal vanishing points and two vanishing lines are calculated.

In order to obtain the VPs and VLs in the BIM view, a virtual camera is simulated in Blender at the rough camera poses. The parameters of the camera object are set according to the intrinsic parameters of the D435i real-depth camera. An adaptor was written to

generate frames in the BIM view directly in Python and avoid a manual step in Blender. Blender is called in the background using the command line rendering.

The script extracts the location, rotation, and frame destination from the passed command and renders the respective keyframe in Blender. Then, the module "Data Access" is used to obtain Blender's internal data and set the new camera information.

4.4.3 Step 3: Camera Pose Improvement

The localization improvement is based on the distance error and the angular error (see Δd and $\Delta\theta$ respectively in eq. 4.2 and eq. 4.1) between the VPs and VLs of the keyframe and BIM, which are depicted in Figure 4.8.

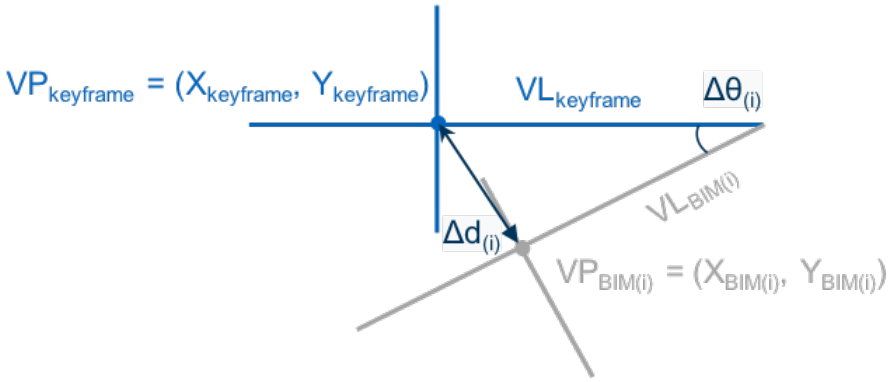


Figure 4.8: Distance error Δd and angular error $\Delta\theta$ between the VPs and VLs of the current key frame and the corresponding BIM frame. The BIM frame has to be corrected until it matches the current real-world frame; in this way, the real camera pose is found in the BIM coordinate system (own illustration based on (Asadi et al., 2019)).

$$\Delta d = \sqrt{(X_{BIM} - X_{keyframe})^2 + (Y_{BIM} - Y_{keyframe})^2} \tag{4.1}$$

$$\Delta\theta = tan^{-1}\left(\frac{VL_{BIM} - VL_{keyframe}}{1 + VL_{BIM} * VL_{keyframe}}\right) \tag{4.2}$$

At first, the location values of the camera poses are corrected. The two corresponding VPs located in the image frame are backward projected from 2D pixel (u, v) to 3D camera coordinates (X_C, Y_C, Z_C) using the intrinsic parameters of the camera.

A correction based on the obtained difference is applied. However, it turns out that the influence on the VPs is relatively small compared to the influence of the VLs. The second part is the improvement of the rotation angles, based on Euler's definition.

An iterative, stepwise correction process conducts this step until a defined threshold is reached. The Euler rotation around the y-axis, also referred to as *pitch*, is directly related to the in-plane computed angle $\Delta\theta$, so that it can be directly applied as a correction for this value.

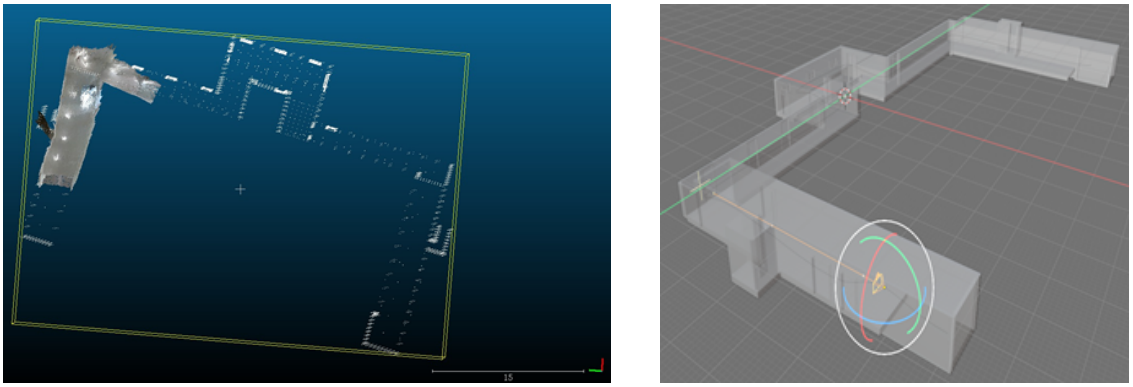
The *yaw* rotation around the z-axis mainly influences the x-coordinate of the VP.

This coordinate is changed stepwise until a threshold of 3 % for the difference between the x-coordinates of both VPs is reached.

After each correction step, a new BIM frame is generated with Blender. The VPs and, thereby, the updated distance and angular error are computed. A similar approach is used for the y-coordinate and the *roll* rotation around the x-axis.

4.4.4 Experiments and Results

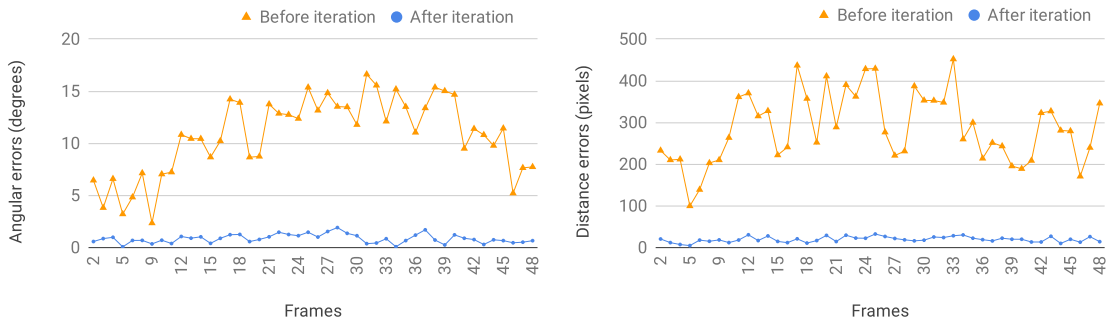
A section of an uncluttered corridor was selected to test the image localization system. The resulting point cloud and the corresponding BIM model are illustrated in Figures 4.9a and 4.9b, respectively.



(a) (b)

Figure 4.9: Evaluation Scenario: (a) Sparse point cloud reconstructed with RTAB-MAP for a corner with the real-world camera frames used to evaluate the proposed image localization system. (b) Corresponding BIM model of the whole corridor.

Figures 4.10a and 4.10b show the angular and translational errors of the initial rough camera poses (estimated with RTABMAP) and the improved errors after the proposed image localization pipeline is executed. Figure 4.11 presents the average computational time of each step of the pipeline.



(a) (b)

Figure 4.10: Visualization of the results: (a) Performance of the image localization system angular errors in degrees. (b) Corresponding distance errors in pixels.

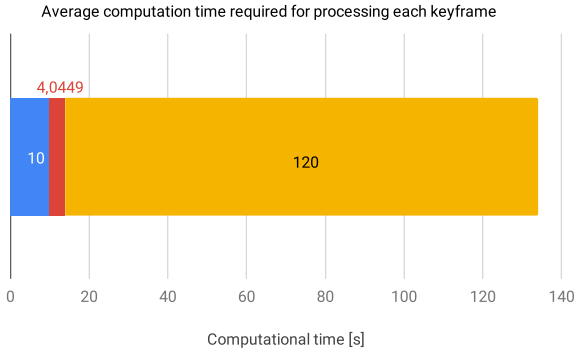


Figure 4.11: Average computational time performance of the image localization system. In red is the time for estimating the rough camera pose with SLAM, in blue for perspective detection (computation of VPs and VLs), and in yellow is the time to recover the fine camera pose through localization improvement.

The proposed image localization algorithm achieved a reduction of the average distance error between the VPs of the video and BIM of 93 % and of the average angular error of 92 %, with an average processing duration of 134 s per frame.

4.5 Contributions and Limitations

4.5.1 Contributions

This chapter presented significant contributions to the field of robot localization and navigation, specifically focusing on the integration of 3D BIM models with real-time 2D LiDAR and image localization systems. The key contributions are as follows:

C 1.1 Transformation of 2D Occupancy Grid Maps (OGMs) to Pose Graph-based Maps (PGBMs) (RQ 1.1):

- A novel open-source code was developed for transforming 2D OGMs into PGBMs, which facilitates the transition of localization algorithms into the optimization paradigm (M. Vega-Torres, 2022). Link to the repository.
- The method involves skeletonization of the OGM, coverage path planning, and ray casting. Up to this stage, the method operates very fast and independently of ROS or Gazebo for LiDAR simulation.
- The approach provides an automatic and systematic framework for generating PGBMs, which is of great advantage for specific pose-tracking tasks, especially in complex, cluttered, changing, and dynamic environments. SOTA SLAM techniques have switched from using particle filters to graph-based optimization approaches; based on the conducted experiments, it is possible to conclude that it will be analogously advantageous for most localization systems.

C 1.2 Performance Evaluation of State-of-the-art (SOTA) 2D LiDAR Localization Algorithms (RQ 1.2):

- A comprehensive evaluation was conducted of four SOTA 2D LiDAR localization algorithms, including two particle filters (AMCL and GMCL) and two pose-graph-based methods (Cartographer and SLAM Toolbox).
- The evaluation was performed in six different environments characterized by varying levels of Scan-Map deviations (i.e., clutter and dynamic conditions), providing valuable insights into the robustness and applicability of these algorithms for real-time localization.

- The results offer a detailed understanding of the performance trade-offs and the conditions under which each algorithm excels, guiding the selection of appropriate localization methods for different scenarios. For example, Graph-based Localization (GBL) methods (such as SLAM Toolbox or Cartographer) perform best for pose tracking and Particle Filter (PF) methods (in particular GMCL) for global localization. In particular, it is advisable to employ a GBL algorithm for precise BIM-based (or floor plan-based) 2D LiDAR pose tracking in real-world settings and utilizing GMCL for global localization.
- In the open-sourced repository (M. Vega-Torres, 2022), a straightforward and effective approach was presented to combine the strengths of a PF algorithm for global localization with a GBL method for pose tracking. This is achieved by initializing the GBL method once the PF algorithm has converged.
- The simulated data used for the experiments and evaluation of the different methods, as well as the reference maps, was made open-access to enable easy benchmarking of new localization or pose estimation methods. The data can be found here (M. A. Vega-Torres, Braun, & Borrmann, 2024a)[5].

C 1.3 Correction of Camera Poses Using 3D Models (RQ 1.3):

- An innovative approach was introduced for correcting poses obtained from SLAM systems using cameras by integrating 3D BIM models.
- The proposed method intends to minimize the distance between VPs and VLs extracted from real-world RGB images and synthetically generated images from a BIM model.
- The method demonstrates the potential of leveraging 3D models to augment the localization capabilities of camera-based SLAM systems, contributing to more accurate and reliable navigation.

[5]For a comprehensive list of all open contributions, refer to Section 1.14.

4.5.2 Limitations

This chapter has demonstrated significant progress and notable contributions; however, it is also important to recognize the limitations of the two proposed methodologies:

L 1.1 Complexity of PGBM Generation:

- The process of transforming 2D OGMs to PGBMs requires significant computational resources and expertise. The methodology, while effective, does not work in real-time, and the complexity is very high; therefore, it is not directly suitable for large-scale maps.
- The transformation process is sensitive to the quality and resolution of the initial 2D OGM, potentially affecting the accuracy of the resulting PGBM.
- The final step, which entails the use of the trajectory builder, necessitates the deployment of Cartographer or SLAM Toolbox, both of which are based on a ROS system and can require high computational resources and certain expertise for their correct implementation. Nonetheless, a Docker was provided for easy deployment on any machine, which now also supports ROS2 (M. Vega-Torres, 2022).

L 1.2 Disregarding mapping accuracy:

- While real-time sensor localization offers the benefit of facilitating the creation of an aligned environmental map, there are inherent trade-offs between speed and accuracy. Real-time algorithms often fail to converge on the most precise robot poses due to time and computational constraints. Higher accuracy can be achieved by allocating more processing time for pose calculation, allowing for a more optimal registration of sensor measurements with the existing map.
- Global localization algorithms, while effective in addressing the initial alignment problem, can suffer from excessively long convergence times in large-scale environments. Additionally, these algorithms require the sensor to be positioned within the pre-existing map for successful convergence. This limitation hinders their ability to extend the existing map or initiate scanning in entirely unmapped areas. Consequently, real-time localization algorithms are not well-suited for these scenarios.

- Although many real-time 2D LiDAR localization algorithms demonstrate robustness to Scan-Map deviations, significant mismatches between the prior map and actual observations in small regions can cause the calculated poses to diverge drastically or even to a system failure. Consequently, the reconstructed map from these poses becomes unusable.

L 1.3 Image Localization Performance:

- The developed image localization pipeline does not work in real-time due to inefficiencies in its implementation, mainly in the optimization algorithm to retrieve the corrected poses. By definition, if the speed of processing each keyframe is higher than the speed of the frame sampling, the process can be considered real-time. The proposed method requires, on average, 134 s to process one keyframe. As it can be seen in Figure 4.11, it does not work in real-time (considering each keyframe is extracted every 1 s). Nonetheless, it is essential to consider that the required time per frame varies depending on the quality of the initial pose estimation. Alternative methods like ORB-SLAM3 (Campos et al., 2021) or OKVIS2 (Leutenegger, 2020; Leutenegger, 2022) may be more advantageous than using RTAB-MAP.
- The method is also highly dependent on the environmental conditions; it requires an environment with a low level of Scan-Map deviations, such as clutter and the presence of dynamic elements; otherwise, the method will not be able to extract correctly the VPs and VLs for the alignment. To address this issue, one approach may consider room layout prediction methods (such as proposed by Lukierski et al. (2017) or by Boniardi, Valada, et al. (2019)); however, with the limitation of working only on Manhattan World environments. To contribute to the solution of this problem, a dataset with over 200 labeled images of the ConSLAM dataset with the corresponding layout was provided. Here is the link to the dataset.

This chapter introduced novel methodologies that contribute to the field of real-time localization and navigation for robots utilizing BIM models or 3D maps. While limitations, particularly the speed-accuracy trade-off inherent to real-time algorithms, restrict their ability to achieve optimal pose estimation, the presented methods offer significant ad-

vancements in addressing the core challenge of sensor-map alignment and provide valuable insights for further development.

However, considering the motivations outlined in Section 1.3, achieving a high-fidelity updated aligned 3D map may ultimately be more crucial and in alignment with the primary objectives (such as supporting decision-making and progress monitoring) than achieving real-time performance in localization or alignment. This understanding informs the focus of subsequent chapters, which will prioritize advancements in accurate map generation and alignment over real-time localization performance.

Chapter 5

Aligning Integrated Mobile 3D LiDAR-inertial Session Data with a Reference Map[1]

This chapter describes a novel method to address some of the most critical limitations of real-time localization algorithms presented in the previous chapter. This enhancement is achieved by focusing on the creation of an updated alignment map rather than emphasizing real-time alignment.

The proposed framework, called SLAM2REF, incorporates innovative feature descriptors based on the widely used SCD for place recognition and introduces a novel YawGICP registration algorithm. Additionally, motion distortion correction for individual scans is integrated by incorporating IMU measurements to create continuous-time trajectories. These components are holistically integrated into a multi-session anchoring framework that enables the registration of drifted SLAM session data with a reference map. The primary objective is to align and correct a given distorted map with a reference 3D BIM model or point cloud, even in the presence of Scan-Map deviations, as typically encountered in construction sites or buildings during emergency scenarios (e.g., post-disaster). Furthermore, a module to analyze environmental changes following the alignment process is presented. This comprehensive approach aims to enhance the accuracy and reliability of localization and mapping in dynamic and challenging environments.

[1]Significant parts of this chapter have been previously published in the Journal of Construction Robotics (M. A. Vega-Torres, Braun, & Borrmann, 2024c), which is an extension of a conference paper (M. A. Vega-Torres et al., 2023). The paper can be accessed at: https://doi.org/10.1007/s41693-024-00126-w

5.1 Motivation

Creating a high-fidelity, updated, and aligned 3D map is essential for the primary objectives of this dissertation, which include increasing situational awareness and supporting construction site progress monitoring. This priority surpasses the need for absolute real-time performance in localization or alignment, as discussed in the previous chapter (Chapter 4).

To address some of the critical limitations of real-time localization systems, this chapter introduces a new methodology. Unlike localization algorithms, SLAM algorithms have been the focus of intense research over the past decades, leading to rapid development. However, even the best algorithms are susceptible to drift, particularly in long trajectories within dynamic or cluttered environments or under the presence of fast sensor motion (L. Zhang et al., 2023).

While the goal is not to develop a new SLAM algorithm, the aim is to utilize the output of any LiDAR-based SLAM or odometry system as an initialization step to approximate sensor poses in a local coordinate system. The proposed framework will then handle automatic alignment and drift correction with the reference map.

Given these requirements, the proposed framework is built on top of Lt-SLAM. This SOTA open-source module enables the alignment of multiple sessions created from specific SLAM systems enhanced with a *key-information-saver* (G. Kim & Kim, 2022).

Session Data (SD), as will be described more formally later, is a sequence of sensor data acquired from a particular place during a specific period of time. In this case, it can be simplified as a set of sequential timestamped LiDAR scans with known positions. The *key-information-saver* is the module that allows this data to be retrieved.

While several improvements in the pipeline were introduced, one of the most critical changes is the focus on aligning session data with a reference map of an indoor building rather than another session. This reference map can contain various levels of Scan-Map deviations and might not be the same size as the acquired session.

The proposed method begins by creating a map of the environment using a SOTA LiDAR-Inertial-Odometry or SLAM algorithm, which is then aligned and corrected with

a reference map. Both maps can have small levels of overlap, and the initial scanning position should not need to be within the reference map.

The less drift the initial poses have, the better the resulting alignment and correction will be. However, a major advantage of the proposed approach is its ability to handle trajectories with significant drift and still correct them using the reference map.

Moreover, unlike the methods discussed and introduced in the previous chapter, the approach presented here has been tested on real-world (not simulated), open-access data of large-scale maps. Additionally, instead of using 2D LiDAR measurements, it leverages 3D LiDAR scans undistorted with fused IMU measurements. Another very important requirement of the presented method is to be able to retrieve very accurate poses (close to the ground truth poses), given an accurate reference map.

5.2 Research Questions

In line with the motivation and requirements stated above, this chapter aims to answer the following main research question.

RQ 2. *How can reference 3D BIM models or point clouds be utilized for alignment and correction of session data from 3D LiDAR and IMU measurements?*

RQ 2.1 Automatic Generation of Accurate Occupancy Grid Maps and 3D Session Data from a Reference Map:

- How can an automatic method be developed to create accurate OGMs and 3D session data from large-scale BIM models or point clouds?
    - *Rationale:* This question addresses the need for a robust and automated process to convert building data into usable formats for localization and alignment, ensuring high accuracy and reliability.

RQ 2.2 Alignment and Correction of Drifted Sessions with a Reference Map:

- How can fast place recognition and multi-session anchoring be leveraged to align and correct drifted sessions acquired with SLAM or LiDAR-inertial odometry systems in indoor environments?

· *Rationale:* Investigating this question aims to enhance the accuracy and stability of 6-DoF pose retrieval and map extension, surpassing current SOTA methods by minimizing drift and improving alignment with the reference map.

RQ 2.3 Analysis and Change Detection in Aligned Data:

- How can a module be developed to analyze acquired aligned data, providing both positive and negative difference detection for updated 3D map visualization?

  · *Rationale:* This question focuses on creating tools for detailed analysis of aligned data, enabling the detection of changes and updates in the environment to maintain an accurate and up-to-date 3D map.

To address the previously described research questions, SLAM2REF is introduced, a novel framework that integrates 3D LiDAR data and IMU measurements with a reference map to achieve precise pose estimation, enabling also map extension and long-term map management.

The effectiveness of SLAM2REF will be demonstrated through extensive experiments in various large-scale indoor GPS-denied real-world scenarios, showcasing its ability to achieve centimeter-level accuracy in trajectory estimation and robust map alignment over extended periods. Additionally, it will be demonstrated that the method enables the robust automatic alignment of the data with a reference BIM model, which does not contain clutter, furniture, or dynamic elements as the real-world data.

This is achieved through innovative feature descriptors based on the widely used Scan Context descriptor (G. Kim et al., 2021) and a novel YawGICP registration algorithm built based on the Open3D GICP method. Additionally, motion distortion correction of individual scans is incorporated by integrating IMU measurements to create continuous-time trajectories inspired by the Direct LiDAR Inertial Odometry system (K. Chen et al., 2023b). These elements are holistically integrated into a multi-session anchoring framework that enables the registration of drifted SLAM session data with a reference map.

Our framework, while drawing significant inspiration from LT-SLAM G. Kim and Kim, 2022, advances beyond existing methods by retrieving ground truth poses when an accu-

rate reference map is available. Additionally, the proposed approach incorporates motion distortion correction and is particularly effective for indoor scenarios. It is versatile in its ability to utilize various types of 3D maps, such as point clouds or BIM models, as references. This flexibility ensures that the proposed method is not confined to the registration of session data pairs alone.

5.3 SLAM2REF and Change Detection Methodology

Our approach can be broken down into three key components, as shown in Figure 5.1. In **Step 1**, synthetic reference SD is generated automatically from large-scale 3D reference BIM models or point clouds.

Then, in **Step 2**, a real-world undistorted LiDAR SD acquired using a SOTA LIO algorithm is aligned and corrected using the reference 3D map.

Finally, in **Step 3**, the aligned map is further automatically analyzed, allowing the creation of an updated 3D map, which considers the detection of positive and negative environmental changes.

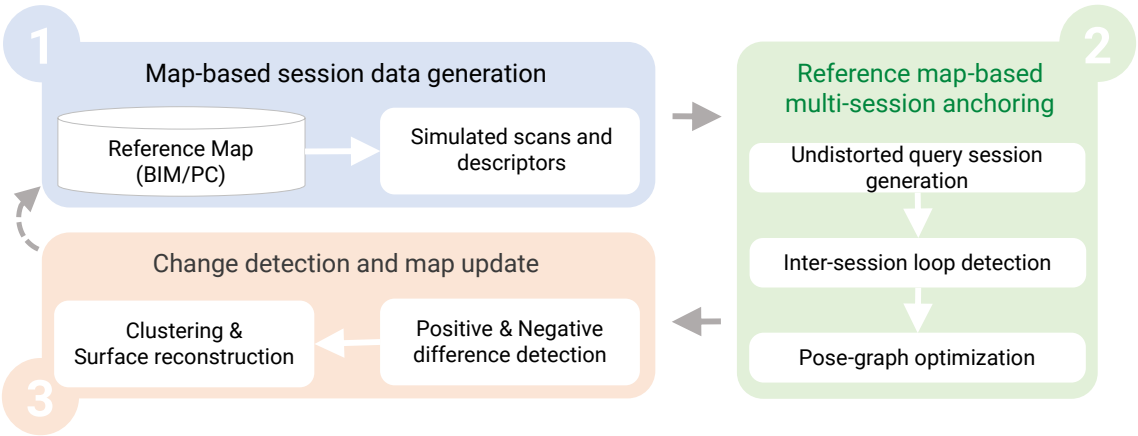


Figure 5.1: Overview of **SLAM2REF**. The pipeline consists of three steps: map-based session data generation, Reference map-based multi-session anchoring, and Change detection and map update.

5.3.1 Step 1: Map-based Session Data Generation (Map to Session Data)[2]

In this step, the objective is to encapsulate the geometry of the reference 3D map—whether it is a BIM model or a point cloud—into individual LiDAR scans with their corresponding feature descriptors. These descriptors serve to encode the visible geometry from the origin of the scan within the reference map, enabling us to rapidly find the correct alignment of real-world session data with a reference map.

In real-world data acquisition, Session Data (SD) refers to consecutive sensor data acquired from a particular place at different periods (Cramariuc et al., 2022). Nonetheless, since the goal is to convert a reference map to synthetic SD, these data can be considered a set of LiDAR scans (with known carefully selected positions) and their corresponding descriptors.

Formally[3], a session $\mathcal{S}$ is defined as follows:

$$\mathcal{S} := \Big(\mathcal{G}, \{(\mathcal{P}_i, d_i)\}_{i=1,\dots,n}\Big) \tag{5.1}$$

Here, $\mathcal{G}$ is a pose-graph map that contains the coordinates of the pose nodes, odometry edges, and optionally recognized intra-session loop edges with uncertainty matrices. These matrices represent how certain the positions of these edges are. This map can be saved in a text file, usually in *.g2o* format.

The $(\mathcal{P}_i, d_i)$ are the pairs of 3D LiDAR scans with their corresponding global descriptors of the i^{th} keyframe and n is the total number of equidistantly sampled keyframes.

Generating synthetic SD (simulated scans and descriptors) from a reference map can be subdivided into three substeps. First, an OGM is extracted from the reference map. This extraction is achieved in an automated manner, taking as input only the IFC model or the reference point cloud and the floor level (z coordinate value) from where the OGM should be generated. In a second substep, the OGM is used to find the poses in which the LiDAR scans will be simulated. In a third and final substep, LiDAR scans are rapidly simulated

[2]The open source code for this step can be found here: https://github.com/MigVega/Map2SessionData

[3]For a comprehensive list of all mathematical variables used, please refer to Appendix A.

in the positions calculated in the previous step, and the corresponding descriptors are calculated.

These substeps have been optimized so that it is possible to efficiently simulate data from large-scale 3D BIM models and point clouds. The following subsections provide a more detailed explanation of each substep.

OGM from Reference Map

Initially, and for convenience, the 3D geometry of the reference map is reduced into a 2D OGM. This dimensional reduction has been demonstrated to be very computationally efficient, allowing the implementation of the pipeline in complex, large-scale models.

Moreover, a 2D OGM (with known scale and origin) allows the direct usage of the map with the ROS navigation stack for autonomous navigation (Macenski, Moore, et al., 2023). Besides path planning, cost maps, and navigational algorithms, the ROS navigation stack includes several SOTA features, such as the regulated pure pursuit algorithm to adjust the robot's speed depending on the path with a particular focus on safety in constrained and partially observable spaces (Macenski, Singh, et al., 2023).

The approach for generating an OGM depends on the type of input data. The following sections detail the procedures for creating OGMs from BIM models and point clouds.

OGM from IFC model (BIM2OGM). The proposed automated generation of OGMs from BIM models builds upon prior work described in (M. A. Vega-Torres, Braun, & Borrmann, 2022). However, the key distinction lies in the enhanced automation of the pipeline.

For this purpose, the IfcConvert (IfcOpenShell Contributors, 2023b) tool is leveraged, and image-processing techniques are employed akin to previous related works. IfcConvert, a command-line interface application within the open source IfcOpenShell project (Krijnen, 2015), facilitates the versatile conversion of a 3D BIM model from the *.ifc* file format to various other formats such as 3D meshes (*.obj*, *.dae*) or 2D layers (*.svg*). Detailed documentation for the IfcConvert functionality is available (Gopee et al., 2022; IfcOpenShell Contributors, 2023a).

The input 3D IFC model is first converted to SVG format and then processed with the OpenCV library to output different layers as Portable Network Graphics (PNG) files. These layers will then be merged to produce the final Portable Gray Map (PGM).

To ensure compatibility with the ROS navigation stack and facilitate accurate scan simulations, the 2D PGM map must adhere to specific guidelines. It should represent unknown (external) regions in gray, navigable space (floor) in white, and potential collision-causing objects (e.g., walls and columns) in black.

IfcConvert is used to convert the 3D IFC model into 2D SVG files with the desired elements intersecting a plane at the chosen height. Furthermore, the resolution and size of the output SVG image are modified to only include the elements of interest.

To generate the OGM, the semantics of the BIM model are leveraged, focusing on extracting permanent elements like walls, ceilings, columns, and floors. This process excludes non-permanent features and objects invisible to LiDAR sensors, such as spaces, doors, windows, and curtain walls.

Filtering just permanent structural information about the building enables finding reliable correspondences between the geometry from the BIM model and real-world 3D LiDAR data. In this letter, it is assumed that the permanent structures in the BIM model are reliable features for localization and scan-matching. In the presence of open doors and windows, their exact placement in the space is unknown (open, closed, or semi-open) and is not provided in the BIM model; therefore, those elements should not be considered while creating the 2D OGM or any source of information used for alignment or localization.

A critical consideration in the conversion of SVG files to PNG format is the choice of units utilized within the original SVG file. By default, IfcConvert assigns millimeters as the unit of measurement for the SVG files. However, these millimeters do not undergo a direct one-to-one transformation to pixels during the conversion to PNG. Consequently, it becomes imperative to eliminate explicit unit specifications within the SVG file to ensure consistent scaling and preservation of the established coordinate origin during the conversion to PNG.

Additionally, it is critical to consider the effect of displacement while creating sections at different heights. While the scale will be maintained, the values of the coordinates of the geometry (saved in *paths*) in the SVG file will be adjusted according to the elements that

intersect that specific height. To counteract this effect and have all the PNG images in the same coordinate system, the images are shifted according to the x and y values saved in the *data matrix* of the SVG generated with IfcConvert.

Automating the creation of the OGM involves producing the following two sections:

1. In the indoor layer, the floor area is designated as white. This section is generated at the z-coordinate corresponding to the upper surface of the slab of interest, i.e., at the floor level where the alignment should happen. Subsequently, the resultant gray-scale PNG image from this SVG is converted to binary. Then, its inverted version represents the indoor layer, in which the floor is represented as white pixels and the rest as black.

2. In the collision layer, permanent elements like walls and columns are extracted while excluding non-permanent structures such as doors and windows. The creation of this layer occurs slightly above (1 m) the z-coordinate of the previous layers. It is crucial to note that the coordinate system of this image deviates from the preceding layer due to its creation at a different height. Therefore, it is imperative to compensate for this offset, as previously explained, before converting it into PNG format.

Subsequently, the indoor layer is placed over a gray image of the same size, allowing to distinguish outdoor (unknown) and indoor areas.

Finally, the pixels in the black color of the collision layer are transferred to the indoor layer. Given this, the final OGM is created and saved in the rasterized ROS standard PGM format. Figure 5.2 illustrates the layers and the final 2D map.

Additionally, a corresponding YAML Ain't Markup Language (YAML) configuration file is generated, containing crucial details such as the origin and resolution of the 2D map, extracted from the *data-matrix* of the initial SVG file.

Besides being an essential step in the proposed pipeline, accurately creating a 2D OGM holds significant potential for SOTA localization algorithms, facilitating rapid and collision-free autonomous navigation. This has been exemplified by M. A. Vega-Torres, Braun, and Borrmann, 2022 (see section 4.3) and corroborated by numerous other studies (refer to chapter 3).

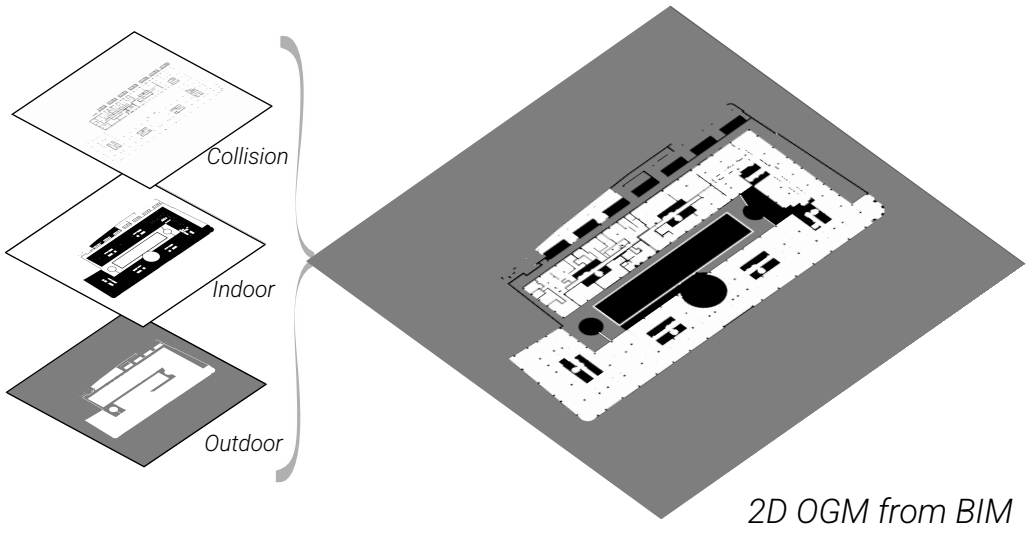


Figure 5.2: Generated OGM from the BIM model. On the left, the different layers, and on the right, the merged final OGM.

OGM from a Point Cloud The steps involved in creating a OGM from a point cloud are as follows: First, a 2D grid is created to the length and width of the point cloud and scaled given a grid resolution. Each cell within this grid is initially assigned a gray color.

Then, and as discussed in (M. A. Vega-Torres, Braun, Noichl, et al., 2022), the points are projected onto the XY plane, considering the resolution of the grid and its origin (the minimum XY coordinate of the point cloud). If points within a cell are found to be near the floor level (within a range of ± 0.5 m), the cell is colored white, signifying navigable space.

On the other hand, cells are colored black if points are detected at a height 1 m above the floor level, assuming that this region predominantly consists of walls, columns, and other permanent elements.

Locations for Data Simulation

Once a correct OGM is generated from the reference map, this is utilized to find proper locations where LiDAR scans will be simulated. These locations should be equally separated coordinates ordered by proximity, aiming to closely replicate real-world data acquisition with full coverage of the map. To this aim, first, the skeleton of the image is extracted, which gives a smooth path similar to the one a person would follow during acquisition with a mobile LiDAR or scanning device. Then, points are sampled over this path uniformly.

Similarly, as proposed in (M. A. Vega-Torres, Braun, & Borrmann, 2022), the process extracts a skeleton from the OGM. This skeleton is derived using the approach outlined by Lee et al., 1994, producing a smooth trajectory over the free space that interconnects all rooms and open areas within the OGM.

In a previous version of the pipeline (M. A. Vega-Torres et al., 2023), a Wavefront Coverage Path Planner (PP) (Zelinsky et al., 1993) was used over this skeleton to find the waypoints in which the 3D LiDAR will be simulated. However, the Wavefront Coverage PP approach is inherently intricate, making it unfeasible to be applied over large OGMs without consuming large amounts of computational resources.

Therefore, to handle large-scale reference maps, the following method is proposed instead, which tries to sample uniformly key points over the path created with the skeleton approach:

1. The scan locations are initially extracted using image processing techniques. This involves generating masks with equally spaced vertical and horizontal lines, isolating only the white pixels intersecting these masks and the previously generated skeleton. The idea behind this is that only isolated pixels will remain rather than elongated lines present in the skeleton.

2. Subsequently, the corresponding center points of the remaining pixels are extracted using a contour detection algorithm. To ensure a minimum distance between points, the spatial distribution of these coordinates is downsampled.

3. Finally, the coordinates are sequentially ordered using the nearest neighbor algorithm.

Figure 5.3 shows the calculated scan locations for an OGM of a large building.

LiDAR data Simulation

In previous work (M. A. Vega-Torres et al., 2023), the identified waypoints were utilized to set navigational goals for a robot operating autonomously within the ROS navigation stack, simulated in the Gazebo physics engine (Koenig & Howard, 2004). Then, a sequence of simulated 3D LiDAR scans was produced with Gazebo and saved in rosbag files. Here,

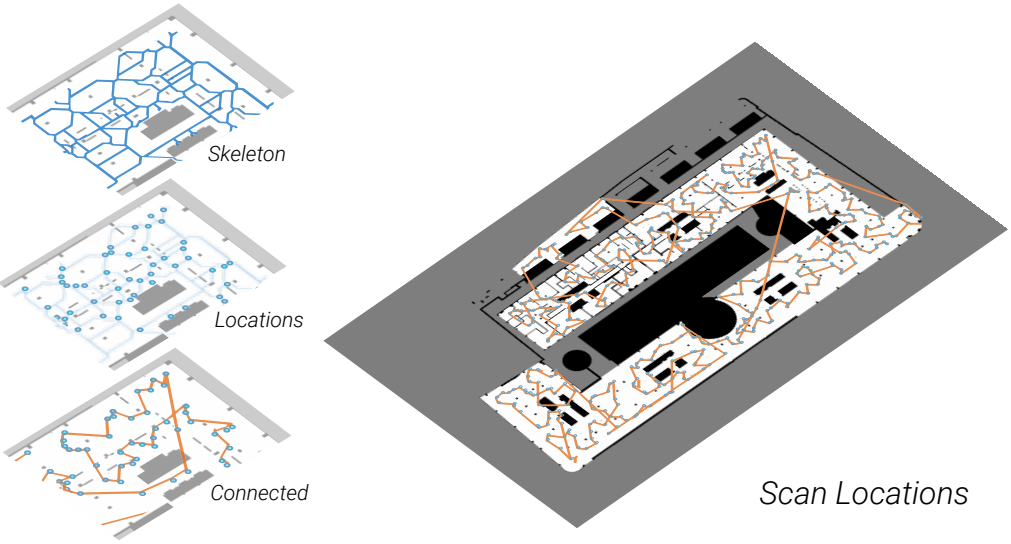


Figure 5.3: Calculated locations for scan simulation. On the left are the main steps, and on the right are all the calculated positions in the entire OGM.

an enhanced approach eliminates the need for ROS or Gazebo is presented; by such means, the creation of large rosbag files containing redundant information is avoided.

Instead, it is proposed to leverage Blender Sensor Simulation Toolbox (BlenSor), a versatile software designed for simulating various range scanners (Gschwandtner, 2013; Gschwandtner et al., 2011). With the BlenSor Application Programming Interface (API), it is possible to automatically load the coordinates for simulating LiDAR scans (calculated in the previous step), streamlining the simulation process.

The process of simulating LiDAR data can be subdivided into three main steps:

1. The reference map is converted to an STL mesh. In the case of a BIM model, this involves conversion to Wavefront .obj file (OBJ) format after filtering only permanent structures using IfcConvert, similar to the process employed in creating the 2D OGM. However, instead of generating an SVG file, the proposed method creates an OBJ file containing the 3D geometry of the model described explicitly. To ensure precise 3D conversion, the proposed approach selectively *includes* required permanent elements (e.g., walls, columns, floors, and slabs) rather than *excluding* entities. The conducted experiments revealed that the *exclusion* command does not consistently produce satisfactory results for this 3D conversion. Subsequently, the generated OBJ file is converted to STL format for seamless integration of the geometry into BlenSor.

When dealing with a point cloud as the reference map, the ball pivoting method has consistently demonstrated reliability in reconstructing mesh surfaces from 3D point clouds. Before applying this method, the process involves estimating the normals of the point cloud and calculating an optimal radius based on the average nearest neighbor distance, facilitating accurate and efficient surface reconstruction.

2. Later, the coordinates determined in the preceding steps, where the data will be simulated, are transformed from pixels (in 2D) to meters (in 3D). This conversion utilizes the scale and origin information specified in the YAML file of the corresponding OGM.

3. Subsequently, the simulated LiDAR properties are adjusted to align with those employed in real-world scanning. Then, a sub-process initiates the parallel simulation of 360° LiDAR scans at these coordinates using BlenSor.

Finally, and after the simulation, Scan Context (SC) descriptors are created for each simulated scan. More information about these descriptors will be provided in the following section 5.3.2 (Step 2.1).

Following the steps above, the geometry of the reference map or the permanent objects in the BIM model is now established as a reference session, denoted as $\mathcal{S}_{\mathcal{R}}$, and is illustrated in Figure 5.4.

In the subsequent step, this synthetic Session Data, encompassing descriptors and simulated scans, will be leveraged for fast place recognition and data alignment. However, before this process, it is necessary to generate session data from real-world datasets.

5.3.2 Step 2: Reference Map-based Multi-Session Anchoring[4]

To derive a globally consistent map aligned with the reference map from real-world sequential LiDAR data, the following three substeps are executed: (1) Creation of the real-world motion-undistorted query session $\mathcal{S}_{\mathcal{Q}}$, which is similar to the synthetic reference session $\mathcal{S}_{\mathcal{R}}$ (created as explained in the previous section); however, from real-world data. (2) Place recognition for inter-session loop detection between $\mathcal{S}_{\mathcal{Q}}$ and $\mathcal{S}_{\mathcal{R}}$. (3) Pose graph optimization with multi-session anchoring and pose refinement with K-nearest neighbors

[4]The open source code for this step can be found here: https://github.com/MigVega/SLAM2REF

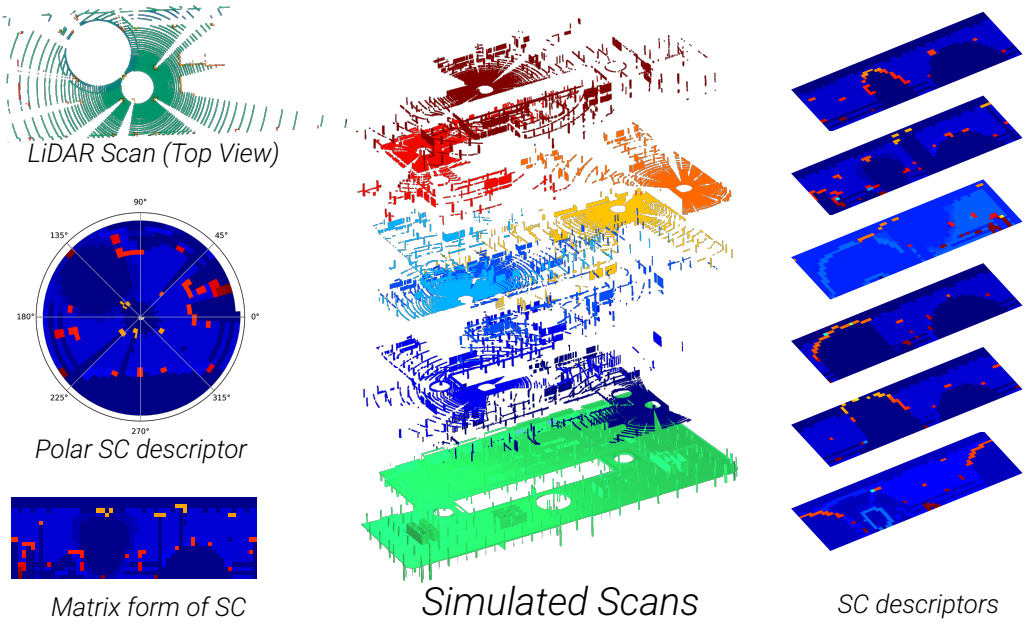


Figure 5.4: Synthetic session data from the reference map. On the left, from top to bottom: Top view of one LiDAR scan, its corresponding polar SC descriptor, and the descriptor in the matrix form. In the middle, a set of simulated scans and the STL mesh from the BIM model are used. Right, corresponding SC descriptors for the simulated scans.

(KNN) loops and a final ICP registration. These substeps are described in detail in the following subsections.

Figure 5.5 illustrates a flowchart outlining the complex multi-session anchoring process in the **SLAM2REF** framework.

Following the generation of SD from the reference map $\mathcal{S}_{\mathcal{R}}$ (Step 1 presented in Section 5.3.1) and the construction of the real-world query session $\mathcal{S}_{\mathcal{Q}}$ (Section 5.3.2), the alignment procedure can be initiated. This involves an inter-session loop detection phase employing ISC and YawGICP (Section 5.3.2), which identifies encounters $\mathbf{c}$ denoting correspondences between the sessions. These encounters, along with initial odometry constraints, are integrated into a factor graph problem. Subsequent to optimization, pose refinement is carried out using KNN loops (Section 5.3.2) and a *final ICP* process. The resulting information comprehends the following elements attributed to the query session: the anchor node Δ_Q^*, which facilitates the global alignment to the reference map, the optimized 6-DoF poses of each scan $\mathbf{x}_Q^*$, and a confidence level list ν_Q providing the reliability of each pose after scan registration.

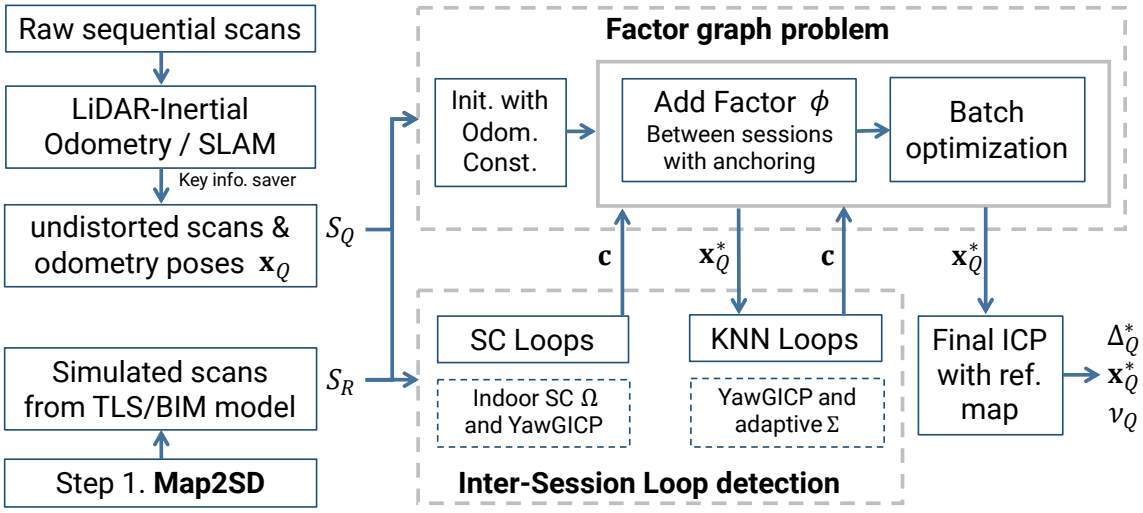


Figure 5.5: Comprehensive flowchart illustrating the multi-session anchoring process within **SLAM2REF**. This process includes the generation of session data from the reference map $\mathcal{S}_\mathcal{R}$, creation of the real-world query session $\mathcal{S}_\mathcal{Q}$, inter-session loop detection using Indoor Scan Context and YawGICP, and pose refinement with KNN loops and *final ICP*. The outcome includes the anchor node Δ_Q^*, optimized 6-DoF poses $\mathbf{x}_Q^*$, and a confidence level list ν_Q for each pose in the query session.

Real-world Query Session Generation

The correct generation of a query session $\mathcal{S}_\mathcal{Q}$ from real-world data involves three primary substeps, elaborated upon as follows.

Motion Distortion Correction. Point clouds acquired from mobile spinning LiDAR sensors often experience motion distortion because the rotating laser array collects points in various instances during a sweep, leading to inaccuracies. Therefore, one of the main issues using LiDAR-only algorithms is the difficulty in correcting motion-distorted LiDAR scans in the presence of fast motion.

In some SOTA LiDAR-only SLAM algorithms, the authors have assumed constant velocity models to overcome this issue, as done in KISS-ICP (Vizzo et al., 2023b). Although this assumption can hold for data acquired with LiDAR placed over autonomous cars and simplistic motion patterns, as in the KITTI raw dataset (Geiger et al., 2013), the constant-velocity model cannot capture subtle movements and generally does not hold for data acquired with handheld devices or UVs in indoor or outdoor scenarios (X. Zheng & Zhu, 2023).

Therefore, the MDC of one SOTA LIO system is used to generate undistorted scans before alignment with the reference map.

In particular, the MDC implementation in DLIO (K. Chen et al., 2023b) is leveraged, which, inspired by Forster et al., 2016, generates continuous-time trajectories. Their approach considers a motion model characterized by constant jerk and angular acceleration compensated with IMU measurements. This enables fast and parallelizable point-wise motion correction.

Once the scans are deskewed with the information from the IMU, keyframe scans can be extracted with timestamps and odometry calculated poses. This process is explained in the subsequent section.

Key Information Saver. The goal here is to save equally spaced undistorted scans (i.e., after a specific variation of time, translation, or rotation) with respective odometry estimated poses from a sequence of data that was previously recorded in a ROS *bagfile* during acquisition with a mobile mapping system device.

To extract keyframes and construct the real-world query session $\mathcal{S}_{\mathcal{Q}}$, the methodology proposed by G. Kim et al., 2022 presents a viable approach. The authors implemented loop closure mechanisms and keyframe information-saving capabilities as an extension in several SOTA algorithms.

In general, the approach can vary depending on the available data. When dealing with LiDAR-only data, SC-A-LOAM (G. Kim et al., 2022), an enhanced version of A-LOAM (J. Zhang & Singh, 2014) is a valid technique; however, it assumes constant velocity for MDC. For an additional calibrated 9-axis IMU, the corresponding enhanced version of LIO-SAM (Shan et al., 2020a) can be used.

If the data contains 9-axis or only 6-axis IMU measurements, which are typical for the internal IMUs of LiDAR and camera sensors, the proposed open source keyframe information saver[5] together with almost any LIO pipeline can be used (e.g., FAST-LIO2 (W. Xu et al., 2022), FASTER-LIO (Bai et al., 2022a) or iG-LIO (Z. Chen et al., 2024)). Something essential to consider is that the LIO pipeline should publish (i.e., make available) the ROS

[5]https://github.com/MigVega/Key-Info-Saver-SLAM

topic with the undistorted scan in the local coordinate system. This last characteristic is not standard and depends on the used MDC strategy.

Given that DLIO demonstrated the best MDC results in the conducted experiments, the corresponding enhanced version, which transforms the deskewed scan to the correct local pose after undistortion, was implemented and made open-source.[5]

After saving the keyframe scans along with odometry information (i.e., time-stamped approximate 6-DoF poses), the final step to generate the query session involves feature descriptor extraction to encode the geometric information of the scans. This process will facilitate efficient comparison with reference session descriptors later.

Indoor Scan Context Descriptor. For place recognition, the new Indoor Scan Context Descriptor (ISCD) is introduced. This variant diverges from the original Scan Context descriptor by focusing exclusively on indoor scans, as opposed to outdoor scans typically encountered in autonomous car environments, for which SC was originally conceived. With ISCD, the objective expands beyond merely eliminating ceiling points, which are notably common in indoor scans, especially in acquisitions with significant variations in pitch and roll angles, as usually encountered in handheld systems. Moreover, the objective is to selectively filter permanent vertical building elements perpendicular to the XY-plane, characterized by visible vertical surfaces of considerable length.

Inspired by G. Kim and Kim, 2018; G. Kim et al., 2021, and by the formal definitions in (L. Li et al., 2021; H. Wang et al., 2020), the creation of ISCD is as follows: Azimuthal and radial bins split the 3D scan from the top view following an equally spaced arrangement (for reference, see an example on the left side of Figure 5.4).

In the Cartesian coordinate system, a LiDAR scan is defined with n points as $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n\}$ with each point $\mathbf{p_k} = [x_k, y_k, z_k]$. Each point p_k can be converted into a polar coordinate system, as follows:

$$\begin{aligned} \mathbf{p_k} &= [r_k, \theta_k, z_k]\,, \\ r_k &= \sqrt{x_k^2 + y_k^2}, \\ \theta_k &= \arctan \frac{y_k}{x_k}. \end{aligned}$$

The point cloud is then segmented into N_s sectors and N_r rings by equally diving polar coordinates in azimuthal and radial directions. Each block is represented by:

$$B_{ij} = \Bigg\{ \mathbf{p_k} \in \mathcal{P} \mid \frac{(i-1) \cdot R_{\max}}{N_r} \leq r_k < \frac{i \cdot R_{\max}}{N_r}, \\ \frac{(j-1) \cdot 2\pi}{N_s} - \pi \leq \theta_k < \frac{j \cdot 2\pi}{N_s} - \pi \Bigg\},$$

where $i \in [1, N_s], j \in [1, N_r]$, and $R_{\max}$ is the maximum radius considered to create the descriptor. In contrast with the original SCD, instead of taking only the z value of the highest point in the bin b_{ij}, in ISCD, if there are a minimum of $ISC_{\min}$ points in the bin, a value equal to 1 is assigned to that bin, and 0 otherwise. Formally:

$$b_{ij} = \begin{cases} 1 & \text{if count}(\mathbf{p_k} \in B_{ij}) \geq ISC_{\min} \\ 0 & \text{otherwise} \end{cases}$$

The final ISCD $\Omega \in \mathbb{R}^{N_r \times N_s}$, can be generated by:

$$\Omega(i, j) = b_{ij}.$$

The global signature Ω is a 2D matrix that efficiently encodes the geometry of mainly permanent elements (e.g., walls and columns) visible from the position of the sensor.

Note that if $B_{ij} \in \varnothing$, $\Omega(i, j) = b_{ij} = 0$, i.e., if in the bin there are no scan data because the bin is free or occluded, the bin will have a value of zero and will be visible as a blue color in the image representation of the descriptor (as shown in 5.4 and 5.6).

In the following section, these descriptors are exploited to rapidly determine the rough alignment between the query and reference sessions.

Place Recognition for Inter-session Loop Detection [6]

Having $\mathcal{S}_\mathcal{Q}$ (real-world query session) and $\mathcal{S}_\mathcal{R}$ (session from the reference map), the goal is to align these two sessions.

To achieve this, correspondences are sought by comparing the previously generated ISCDs between sessions to identify inter-session loop closures. This task is also known as *place*

[6]The open source code for this step can be found here: https://github.com/MigVega/SLAM2REF

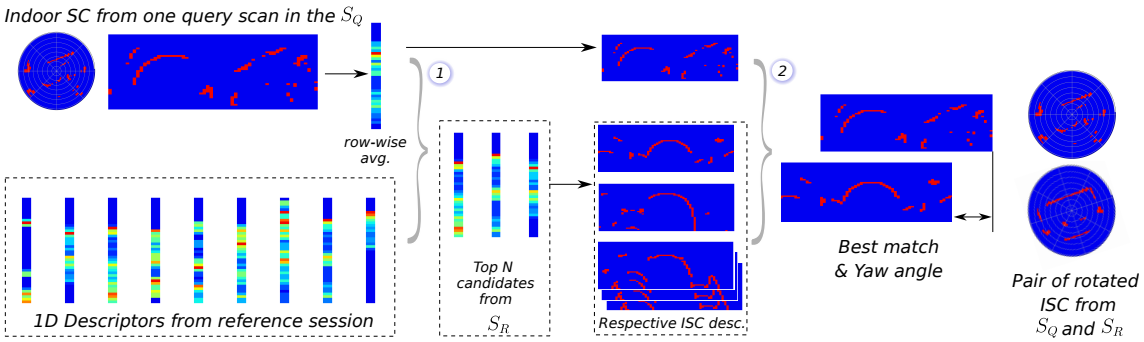


Figure 5.6: Indoor Scan Context loop detection: The query session's scan is converted into 1D rotational invariant descriptors. These descriptors are quickly compared with those from the reference session to select the top N_c candidates (see number 1). In the second phase, the 2D descriptors of these candidates are compared using cosine similarity while systematically varying the column position to identify the best match and optimal yaw angle alignment.

recognition, in which one aims to identify or determine the specific location or place of sensor measurements (in this case, single LiDAR scans) within a given map.

In order to facilitate quick comparison, the 2D descriptor is condensed into a one-dimensional vector. This vector is generated by calculating the average of the rows in the 2D descriptor. This average ensures rotation invariance, meaning that if a scan is in a location that is approximately the same but with a different yaw angle, the resulting 1D descriptor will remain unchanged.

The comparison between the query scan (from $\mathcal{S}_\mathcal{Q}$) and the scans from the $\mathcal{S}_\mathcal{R}$ is facilitated by employing a KNN search in a KD-Tree and using the L2-norm metric.

Subsequently, the corresponding 2D descriptors of the N_c closest 1D descriptors are compared using the *column-wise cosine distance*.

This column-wise cosine distance is calculated to identify the similarity between two ISCDs Ω^q and Ω^r. Let $\mathbf{v}_i^q$ and $\mathbf{v}_i^r$ be the i^{th} column of Ω^q and Ω^r; the score can be found by:

$$\varphi_i\left(\Omega^q, \Omega^r\right) = \frac{1}{N_s} \sum_{i=0}^{N_s-1} \left(\frac{\mathbf{v}_i^q \cdot \mathbf{v}_i^r}{\|\mathbf{v}_i^q\| \cdot \|\mathbf{v}_i^r\|} \right).$$

A comparison conducted column by column is beneficial for handling dynamic entities or slight differences between the reference map and the query session (e.g., new furniture or clutter) since although some columns of the 2D descriptor may show variations, the remaining columns will exhibit similarities. However, relying solely on this comparison

overlooks the possibility of revisiting the exact location from a different perspective. To tackle this limitation and ensure rotational invariance in the matching process, the method computes distances using a range of column-shifted scan contexts. Then, it identifies the shift that yields the minimum distance. This procedure resembles the coarse alignment of two sets of points, focusing mainly on aligning the yaw angle. By implementing this approach, the optimal number of column shifts (i.e., optimal yaw angle) for alignment and the corresponding minimum distance can be determined.

Formally, if Ω_k^q and Ω^r are compared where Ω_k^q is Ω^q shifted by k^{th} column. The final score is calculated as follows:

$$\Phi_i \left(\Omega^q, \Omega^r\right) = \underset{k}{\operatorname{argmin}} \ \varphi_i \left(\Omega_k^q, \Omega^r\right).$$

The matched pairs are subsequently refined through a filtering process employing an empirical threshold, denoted as ϵ, applied to the calculated minimum distance metric, Φ_i.

After detection of ISC loop closures, a 6D relative constraint is established between two keyframes if there is a successful alignment between a sub-map from the reference session, denoted as $\mathcal{P}_{R,i}$ (which comprises the three closest scans to the one that matched the scan in the query session), and the single undistorted scan from the query session, denoted as $\mathcal{P}_{Q,j}$.

The correctness of the alignment between these two keyframes is essential for the subsequent steps in the pipeline, as it dictates the effectiveness of the initial global registration between sessions.

To achieve this alignment robustly, YawGICP is introduced, an improved variant of the GICP algorithm. YawGICP primarily focuses on translational changes and yaw angle adjustments, thereby mitigating significant pitch or roll rotations commonly induced by conventional GICP alignment procedures. This precaution prevents instances where standard GICP may accidentally rotate the source point cloud by 90 degrees (in pitch or roll), leading to erroneous associations between wall, ceiling, or floor points.

The YawGICP is initialized with the yaw angle calculated in the previous step.

Consistent with prior work (G. Kim & Kim, 2022; M. A. Vega-Torres et al., 2023), only ISC loops exhibiting a satisfactory fitness score, indicating a high percentage of inliers are

considered. These loops are then incorporated into the factor graph problem with low covariance Σ_c, serving as factors between sessions with anchoring. Further elaboration on the factor graph problem will be provided in the subsequent section (5.3.2). Figure 5.7 illustrates the detected ISC loop closures, which are then classified into correct and incorrect using YawGICP.

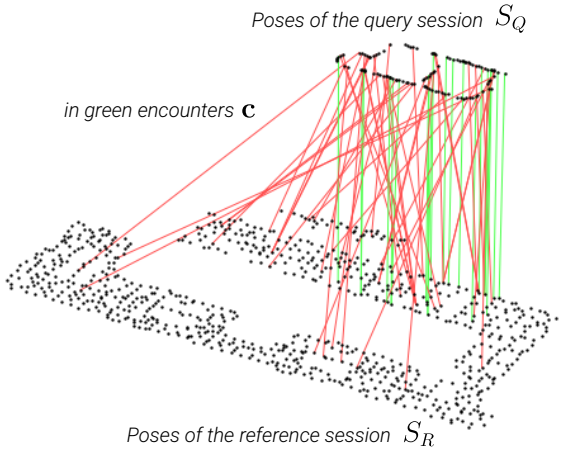


Figure 5.7: Detection of Indoor Scan Context loop closure between sessions. On the top are the poses from the query session, and on the bottom are the poses from the reference session, in this case, created from a BIM model. Correct correspondences are represented by green lines, while erroneous ones as red. After the YawGICP step, the erroneous correspondences are effectively discarded.

Pose Graph Optimization and Data Alignment [7]

In this substep, the initial odometry constraints derived from the preserved session data (referenced in substep 5.3.2) and previously identified inter-session ISC loop closures (introduced in substep 5.3.2) are leveraged to achieve the data alignment.

The objective is to first roughly align the entire query session with the reference session from the reference map. Consequently, even if some scans within the query session's keyframes do not have any correspondence with the reference session, they are still aligned to the most cohesive pose based on the identified correspondences (SC loops) with adjacent scans and the provided odometry constraints.

Formally, in this contribution, the alignment between the sessions is done using *multi-session anchoring*. This method was originally introduced by B. Kim et al., 2010 and was further developed by McDonald et al., 2013, Ozog et al., 2016, G. Kim and Kim, 2022. One of the main motivations behind these projects is to solve the so-called *multi-robot*

[7]The open source code for this step can be found here: https://github.com/MigVega/SLAM2REF

mapping problem. In this context, and as explained in Section 2, maps generated by different robots commonly have distinct reference coordinate systems, which require the merging of these maps to form a globally consistent map with a unified global coordinate system.

Formally the problem can be defined as follows: Given two sessions, $\mathcal{S}_\mathcal{Q}$ and $\mathcal{S}_\mathcal{R}$, each provided with odometry constraints, and in the case of $\mathcal{S}_\mathcal{Q}$, potentially equipped with intra-session loop closure constraints identified by a SLAM algorithm with a key information saver (as explained in Section 5.3.2), the objective is to determine the optimal poses for the nodes in $\mathcal{S}_\mathcal{Q}$. These poses should effectively align the measurements within $\mathcal{S}_\mathcal{Q}$ with those of $\mathcal{S}_\mathcal{R}$, considering the existence of inter-session loop closure constraints between the two sessions.

As explained in 2, multi-session anchoring can be formulated as a factor graph MAP optimization problem.

To properly consider the encounter measurements ($\mathbf{c}$) in the MAP formulation in Eq. (2.4), it is needed to redefine the relative measurement model $h(.)$ in the global frame with the help of the anchor nodes.

This adjustment is needed, considering that the encounter is a global assessment between two trajectories. However, the pose variables for each trajectory are defined in the session's local coordinate frame. With the anchor nodes, the poses of the respective sessions are transformed into a global frame, where a comparison with the measurement becomes possible.

The measurement model $h(.)$ is modified to $h'(.)$, to incorporate the anchor nodes, and therefore, the respective term in Eq. (2.4) is changed to:

$$\sum_{j \in N_e} \left\| h'_j \left(\mathbf{x}_{R,j}, \mathbf{x}_{Q,j}, \Delta_Q, \Delta_R \right) - \mathbf{c}_j \right\|^2_{\Sigma_c}$$

The difference $\mathbf{c}$ in the global frame between a pose $\mathbf{x}_R$ and a pose $\mathbf{x}_Q$ is estimated by $\mathbf{c} = (\Delta_R \oplus \mathbf{x}_R) \ominus (\Delta_Q \oplus \mathbf{x}_Q)$, where $\oplus$ and $\ominus$ are the SE(3) pose composition operators (Blanco-Claraco, 2021; Smith et al., 1990).

The operation $\Delta_Q \oplus \mathbf{x}_Q$ represents concatenating the transformation of $\mathbf{x}_Q$ (the second pose) to the reference system already transformed by the anchor node Δ_Q. In SE(3), the operator $\oplus$ is equivalent to matrix multiplication (Blanco-Claraco, 2021).

Hence, the subsequent *factor between sessions with anchoring* will properly integrate the encounters in the pose graph optimization. It achieves this by initially transforming the poses of each session into the global frame using the anchor nodes.

$$\begin{aligned} &\phi\left(\mathbf{x}_{R,i}, \mathbf{x}_{Q,j}, \Delta_R, \Delta_Q\right) \\ &\propto \exp\left(-\frac{1}{2}\left\|\left(\left(\Delta_R \oplus \mathbf{x}_{R,i}\right) \ominus \left(\Delta_Q \oplus \mathbf{x}_{Q,j}\right)\right) - \mathbf{c}\right\|^2_{\Sigma_c}\right) \end{aligned} \tag{5.2}$$

While initializing the factor graph, the odometry constraints from both sessions and the constraints after ISC loop detection are added to the optimization problem, the first as *between factors* and the latter as *factors between sessions with anchoring*.

Considering that in this scenario, the objective is to use the coordinate system of $\mathcal{S}_\mathcal{R}$ as the global system for alignment, the anchor node Δ_R of the reference session should be assigned an insignificantly small covariance (Σ_P). Conversely, for the anchor node Δ_Q of the query session, a significant covariance is assigned (Σ_L).

Moreover, the odometry poses are also added to the factor graph. However, since $\mathcal{S}_\mathcal{R}$ comes from the reference map, its poses $\mathbf{x}_R$ are treated as fixed and should not be altered by the optimization. To avoid changes to these poses, they are added to the factor graph optimization problem as *prior factors* with very low covariance (Σ_P) in its noise model.

Following batch optimization, the intermediate optimized values of the anchor node Δ_Q^* and the poses $\mathbf{x}_Q^*$ are obtained. However, these poses are expressed in the local coordinate system of $\mathcal{S}_\mathcal{Q}$. To convert them from this local coordinate system (denoted as ${}^{Q}\mathcal{G}_Q^*$) to the global coordinate system ${}^{W}\mathcal{G}_Q^*$ of the reference map, the following transformation is applied to each pose $\mathbf{x}$ in the graph:

$$ {}^{W}\mathbf{x}_Q^* = \Delta_Q^* \oplus {}^{Q}\mathbf{x}_Q^*, $$

where W is the global coordinate system, or in this case, the coordinate system of the reference session.

After the previous step, the query session roughly aligns with the reference session. To further refine the poses of the query session, a rapid KNN loop detection method with adaptive covariance is introduced. Initially, submaps are generated by selecting KNN scans from the scan to be aligned within the query session, along with the k-nearest scans from the reference session. Subsequently, the YawGICP algorithm (see 5.3.2) is employed to register these two submaps, and the quality of registration is assessed based on a predefined fitness threshold, classifying the alignment as either good, acceptable, or unacceptable.

Upon acceptance of the alignment, the constraints are added to the optimization problem as factors between sessions with anchoring with adaptive covariance. This adaptive covariance strategy assigns a very low covariance in the noise model to constraints originating from well-registered keyframe submaps, while constraints from just acceptable registrations receive a higher covariance. This approach allows the pose graph optimization to appropriately weigh the influence of these constraints in calculating optimized poses.

After conducting batch optimization one more time with incorporated odometry, ISC, and KNN constraints in the factor graph problem, the resulting poses undergo further refinement through a *final ICP* registration. Unlike previous steps that relied on registration with simulated scans from the reference map, this stage utilizes a one-centimeter-dense point cloud obtained from the reference map as the registration target. In case the reference map is a BIM model, this point cloud is created by sampling uniformly points over a mesh of permanent elements in the building (i.e. without doors and windows similarly as done in Step 1, section 5.3.1)

Due to the high density of the target point cloud, GICP fails to offer any significant advantage over Point-to-Point (P2P)-ICP (Besl & McKay, 1992). In fact, in specific scenarios, GICP yields inferior results. Therefore, P2P-ICP is preferred, which not only produces competitive results but also operates considerably faster.

To speed up computations and avoid the time-intensive KNN search associated with registrations involving a large target point cloud, scans within the query session are allocated into proximity-based groups. Subsequently, for each group, a target point cloud is created, dynamically cropping the reference map into spheres. The individual source scans within each group are then registered concurrently, leveraging parallel computing techniques.

The registration results are evaluated using three metrics. One metric is the RMSE, and the other two correspond to fitness scores calculated at two distinct maximum P2P distances: F_1 and F_2. The fitness score is the percentage of source inliers, considering a maximum P2P distance threshold to classify points as inliers after registration.

These metrics are computed explicitly for points located within 30 cm from the target point cloud after registration. This approach ensures the exclusion of points outside the reference map or those influenced by significant environmental changes, such as the addition of new walls or large pieces of furniture.

Depending on the metric values, the resulting aligned scans are categorized into four classes: Perfect, Good, Bad, and Outside the Map. The result is saved on a list, denoted as ν_Q.

The resulting poses will be used in the subsequent step to create the final aligned map and compare it accordingly with the reference map.

5.3.3 Step 3: Change Detection and Map Update

Following the completion of the prior steps, the two sessions have been precisely aligned, and they now share a unified coordinate system. Subsequently, a comprehensive 3D map of the most up-to-date environmental state can be generated by placing the keyframes $\mathcal{P}_{Q,i}$ from the query session $\mathcal{S}_\mathcal{Q}$ in the estimated poses ${}^W\mathbf{x}^*_{Q,i}$, which are now in the global coordinate system.

If desired and to ensure the integrity and fidelity of the final map representation, it is recommended to exclusively incorporate scans classified as ”perfectly” or ”good” aligned within ν_Q during the map construction process.

However, it is essential to note that although the remaining poses may not meet the strict alignment criteria with the reference map, they have already undergone significant optimization through odometry and loop closure constraints. Consequently, they can be utilized to generate the final map and even extend the reference map if the scan extends beyond its boundaries.

Since both maps are now aligned, a comparison of the two 3D maps becomes feasible. The comparison process involves categorizing the elements in the map into three distinct

types: Positive Differences (PDs) denote instances where new objects have been introduced compared to the reference map; Negative Differences (NDs) signify the removal of objects previously documented in the reference map; and Unaltered Elements (UEs) denote features that remain constant across both maps.

This categorization is facilitated with the OctoMap library (Hornung et al., 2013). OctoMap, a widely-used library in robotics and 3D mapping, operates by dynamically updating voxel occupancy status within its octree structure as new point clouds are integrated. The analysis of measurement densities in OctoMap enables us to distinguish between occupied and free space, facilitating reliable 3D mapping.

Additionally, also the probabilistic capabilities of OctoMap during measurement accumulation is leveraged to facilitate the automatic removal of dynamic elements from the final point cloud. This removal is done based on occupancy patterns across multiple scans. The resulting map is the one used to detect PDs) and UEs in the preceding step. Moreover, OctoMap calculates free space by identifying regions where the sensor fails to detect objects; this free space will be leveraged for NDs detection later.

To detect PDs and UEs, a P2P distance threshold is used between a point cloud from the reference map (also used in the previous final ICP step) and the newly created map with OctoMap, similar to what was presented in (M. A. Vega-Torres et al., 2023). A signed distance computation allows the distinction of points that are near and far from the reference map. Near points allow for the confirmation of UEs, whereas distant points are regarded as PDs.

The point cloud of identified PDs is passed through an outlier removal process. Subsequently, the point cloud undergoes a segmentation process through Density-Based Spatial Clustering of Applications with Noise (DBSCAN). This step is based on a neighbor-distance threshold and a minimum number of points per cluster.

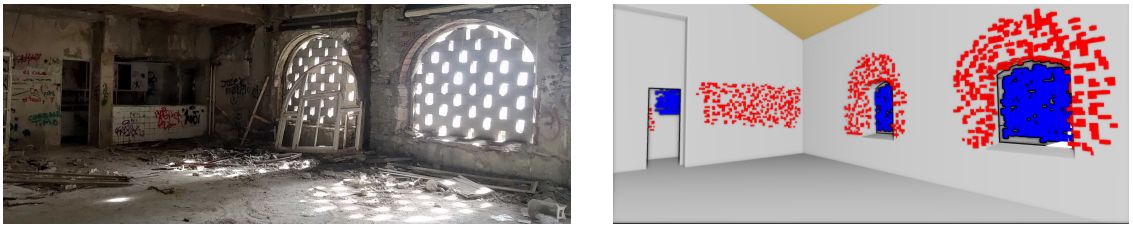
Lastly, for each PD cluster, a mesh is created using cubes from a Voxel Grid (VG) of the point cloud.

Voxels, in contrast to other surface reconstruction approaches, capture the actual geometry of objects present in the scene. This leads to improved visualization of the new elements in conjunction with the reference map, providing a better understanding of the scene.

The process of detecting NDs involves conducting a visibility analysis using individual scans from the query session ($\mathcal{P}_{Q,i}$). As mentioned before, the OctoMap library facilitates this analysis by calculating the free space, i.e., areas where the LiDAR did not detect any objects from its origin point. Similarly, as with the PDs, this free space is used together with a P2P distance threshold against a point cloud sampled from the reference map to identify the NDs.

The regions at the intersection between the reference map and the free space are the NDs. These are then passed through the outlier removal and clustering process, removing isolated points and small clusters.

The final voxels are transformed into meshes and are colored blue for PDs and red for NDs. An exemplary result is depicted in 5.8.



(a) (b)

Figure 5.8: Positive and negative differences between the point cloud and the reference BIM model are illustrated as follows: (a) A picture of the real-world scene. (b) Visualization of the detected changes in the form of voxelized clustered meshes with positive differences depicted in blue and negative differences in red. Particularly, it is visible that the windows in the model are smaller compared to the real-world windows.

5.4 Experiments

This section presents the data used to evaluate the efficacy of the proposed strategies. Comprehensive implementation details, such as the values of the essential parameters, are meticulously outlined to ensure a thorough understanding of the proposed approach.

5.4.1 ConSLAM Dataset

To ensure reproducibility and benchmarking, the approach was evaluated by applying it to the recently released open-access ConSLAM dataset (Trzeciak et al., 2023a, 2023b).

The ConSLAM dataset consists of four sequences of a construction site captured with a handheld system. It incorporates synchronized timestamped LiDAR scans, 9-axis IMU measurements, and RGB and Near-infrared (NIR) camera images.

Given the TLS point cloud of sequence number two, a half-centimeter-accurate BIM model was elaborated.

Moreover the OA-LICalib library (Lv et al., 2020, 2022) was used to retrieve the extrinsic calibration parameters (rotation and translation) between the LiDAR and the IMU sensors.

5.4.2 Implementation Details

While *Step 1* and *3* were implemented in Python, *Step 2* was written in C++.

Step 1: Reference Session Generation[8]

In *Step 1*, to generate the reference session data ($\mathcal{S}_{\mathcal{R}}$), the vertical FoV of the simulated LiDAR scans can be customized according to preferences. To achieve alignment with a TLS point cloud as a reference map, the simulated LiDAR scans encompass a range from -45 degrees to 45 degrees in the vertical FoV. However, in the conducted experiments, while aligning the data with a BIM model, it can be observed an improved ISC loop detection when no ceiling points were present in the simulated scans. Consequently, the scans are adjusted to cover only from 0 to -25 degrees in the vertical direction. In Blensor, during the LiDAR simulation process, the noise was set to a mean of zero with a standard deviation of 0.03 m, an angular resolution of 0.1728 degrees, and a maximum distance of 15 m.

Step 2: Query Session Generation, Alignment, and Correction[9]

Step 2.1: Query Session Creation. In *Step 2*, to generate the query session from the real-word data ($\mathcal{S}_{\mathcal{Q}}$), for the MDC step, it was opted for using DLIO, because, in contrast to FAST-LIO2 W. Xu et al. (2022), it does not require heavy downsampling of the point cloud for deskewing and registration. Hence, clean, undistorted scans with DLIO

[8]The open source code for this step can be found here: https://github.com/MigVega/Map2SessionData

[9]The open source code for this step can be found here: https://github.com/MigVega/SLAM2REF and here: https://github.com/MigVega/Key-Info-Saver-SLAM

allow dense map reconstruction. As suggested by L. Zhang et al. (2022), the data in the *bagfiles* was reproduced at a low rate (half of the original speed) to avoid errors during the distortion process.

Regarding the key information saver, while it is possible to wait for a minimum variation in translation or rotation between consecutive scans, instead, the scans were saved either according to a list of timestamps or after a specific interval had passed. This approach is advantageous, as it allows for comparison with existing ConSLAM GT poses, where specific frames with known timestamps are of primary interest. For the creation of the ISCD, the parameters $N_s = 60$, $N_r = 20$ (as recommended by G. Kim et al. (2021)), $ISC_{\text{min}} = 40$, and a maximum radius of 10 m were selected. This means that the 10 m will be divided into 20 radial sections (since $N_r = 20$) each of 0.5 m.

Step 2.2: Inter-session Loop Detection with ISC. Nanoflann (Blanco & Rai, 2014) is used to create a KD-tree of 1D rotational invariant descriptors. A total of 100 (N_c) top candidates were chosen to evaluate in 2D after the 1D descriptor comparison; it is worth mentioning that the retrieval of correct correspondences is very sensitive to this value. A cosine similarity threshold $\epsilon = 0.3$ is used to filter out pairs of 2D descriptors that passed with the minimum distance among the possible column shifts k. Only column shifts of 10% of the total number of columns (i.e., 36 deg) are considered for the alignment. All YawGICP registrations in the ISC and KNN loops are done using parallel computations with OpenMP. Unlike conventional ICP implementations, when employing YawGICP, it is imperative to express the target point cloud (i.e., from the reference map) in the local coordinate system of the source scan (i.e., the point cloud to be aligned). Otherwise, the process will yield undesirable results. This shift is critical because the resulting transformation matrix is relative to the origin of the source scan, with the aim of rotating the point cloud from its local origin rather than the origin of the global coordinate system.

Step 2.2: KNN loops, Pose-graph Optimization, and Final ICP. The K-nearest neighbors used to create the submaps for KNN loop detection in the second step of optimization is 5. To ensure correct alignment with the BIM model as the reference map, the KNN loop detection process was omitted. This decision was made because this process tends to induce erroneous correspondences. Meanwhile, in Step 2.3.2 (Section 5.3.2), the

pose-graph optimization is done with GTSAM using iSAM2; the following are the values of the variances of the different noise models: $\Sigma_L = \pi^2$ (significant noise for query session's anchor node); $\Sigma_P = 1 \times 10^{-102}$ (prior noise for reference map poses and initial poses); $\Sigma_O = 1 \times 10^{-4}$ (for odometry constraints); $\Sigma_c = 0.5$ (robust noise for encounters, i.e., loop closure constraints). The parallel creation of spheres for target point cloud registration and the P2P ICP of the single source scans is done using OpenMP in C++. Here, the following two maximum distances to calculate the fitness scores were used: $F_1 = 1$ cm and $F_2 = 3$ cm. This means that points that are farther apart than 3 cm will be considered outliers. Therefore, if the reference map, for example, the BIM model has Scan-Map deviations larger than 3 cm on large permanent structures (such walls), the method will most likely not be able to find correct correspondences. Only if the majority of the elements are within this range is a good registration possible.

Step 3: Change Detection and Map Update

In *Step 3*, the process is performed with Trimesh, OctoMap, and Open3D.

A P2P distance threshold of 0.3 m was used to calculate the positive and negative differences.

OGM2PGBM (M. A. Vega-Torres, Braun, & Borrmann, 2022), Scan Context (G. Kim & Kim, 2018), and LT-SLAM (G. Kim & Kim, 2022) are projects that were used as a reference and that are freely available online.

For the evaluation of the results, presented in the following section, the trajectories were compared against the ground truth using the EVO library (Grupp, 2017b) in Technical University of Munich (TUM) format (Sturm et al., 2012) and using the Umeyama alignment (Umeyama, 1991).

5.5 Results and Analysis

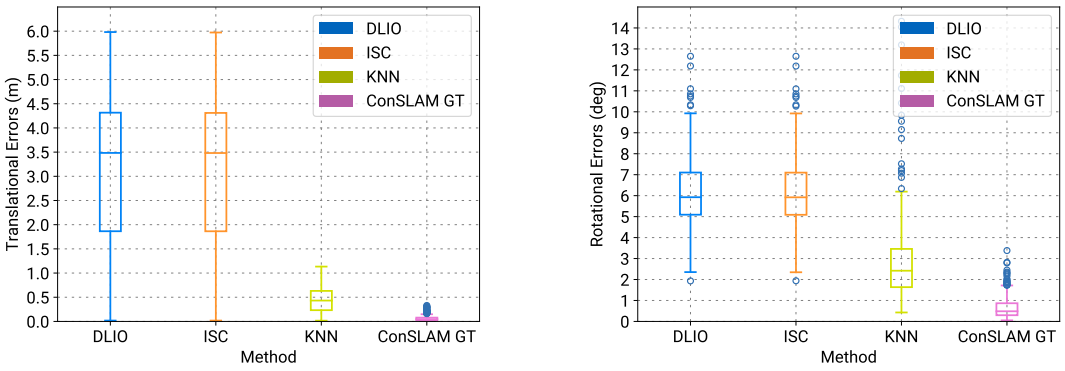
This section provides the results of the proposed framework with respect to the alignment with an accurate TLS point cloud and with a BIM model as a reference map.

Table 5.1 shows the Absolute Pose Error (APE) summary of the different methods in each sequence of the ConSLAM dataset after alignment with the corresponding TLS point clouds. In the table, the performance of DLIO (with Umeyama alignment) is compared against the results of the porposed framework after improving the DLIO trajectory with ISC loop detection and after KNN loop detection and optimization. The results after the *final ICP* step correspond to the current ground truth (used to evaluate the methods); therefore, they are not numerical values for this step. Moreover, the results are compared against the original ConSLAM ground truth poses provided by the authors together with the dataset.

Furthermore, figures 5.9, 5.10, 5.11 illustrate the distribution of errors (translational and rotational) for each method in the various sequences. Figure 5.13 provides a visual representation of the resulted trajectories and 3D maps.

Method	S2 (225 m)		S3 (340 m)		S4 (275 m)		S5 (320 m)		Average	
DLIO	20.2	2.2	21.4	2.6	359.0	6.4	17.4	2.3	104.5	3.4
ISC	20.1	2.2	24.3	2.6	358.6	6.4	18.7	2.3	105.4	3.4
KNN	9.0	1.4	34.6	4.3	53.4	3.5	11.2	2.5	27.1	2.9
ConSLAM	5.2	0.7	4.2	0.7	9.3	0.9	12.1	1.1	7.7	0.8

Table 5.1: Quantitative comparative results for each ConSLAM sequence (S2, S3, S4 and S5). Translational and angular APE RMSE in centimeters and degrees, respectively. Additionally, the length of each sequence is given in meters. ISC refers to the results of DLIO after Indoor Scan Context loop detection and optimization, similarly KNN refers to the results after KNN loops. ConSLAM refers to the ground truth poses provided together with the dataset.



(a) (b)

Figure 5.9: Translational (a) and rotational (b) errors for sequence 4 after alignment with the respective TLS point cloud.

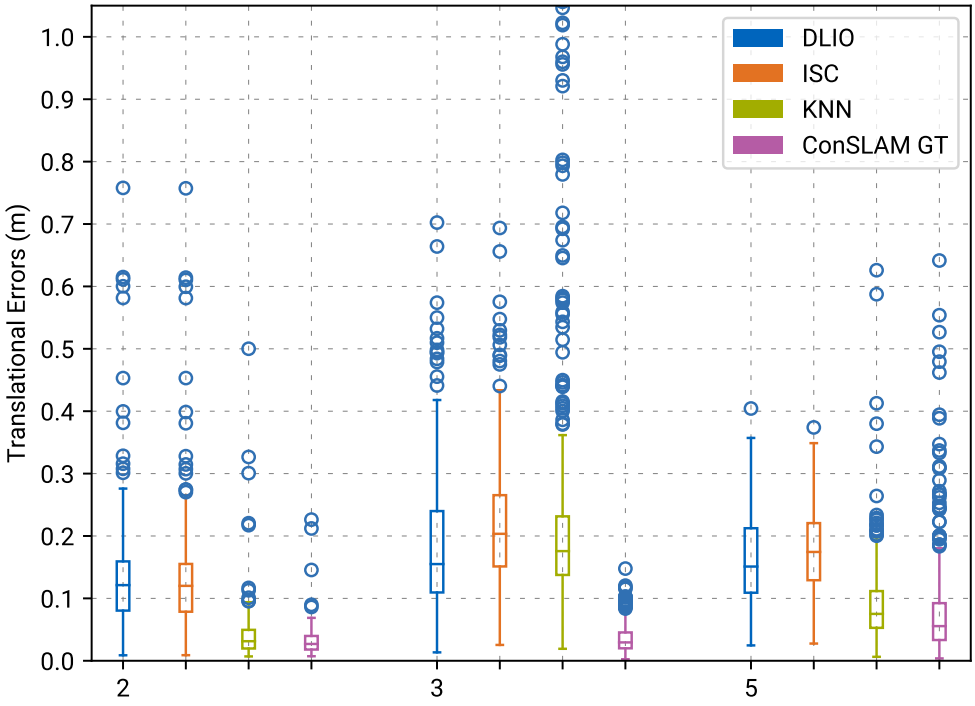


Figure 5.10: Translational errors for sequences 2, 3, and 5 after alignment with the respective TLS point clouds.

Notably, the errors exhibit an evident reduction across almost all sequences while the pipeline evolves.

The ISC loops primarily allow the critical first rough alignment between the query and reference sessions. Since only a few ISC loops are retained due to rigorous threshold criteria, the outcomes after ISC loop detection and optimization exhibit minimal alteration in trajectory accuracy when compared to the initial results derived from DLIO.

On the other hand, the subsequent KNN loops exhibit a more pronounced impact on the results after ISC loops.

While the average rotational error, as depicted in Figure 5.11, experiences a significant decrease in sequences 2 and 4, it exhibits apparent stability or even an increase in sequences 3 and 5.

Regarding the GT poses provided with the ConSLAM dataset, although the RMSE for APE remains below 8 cm and 1 degree for translation and angular errors, respectively (as shown in the last column of Table 5.1), the maximum errors escalate to 20 cm or even 60 cm in sequences 2 and 5 (see Figure 5.10). While these significant discrepancies are

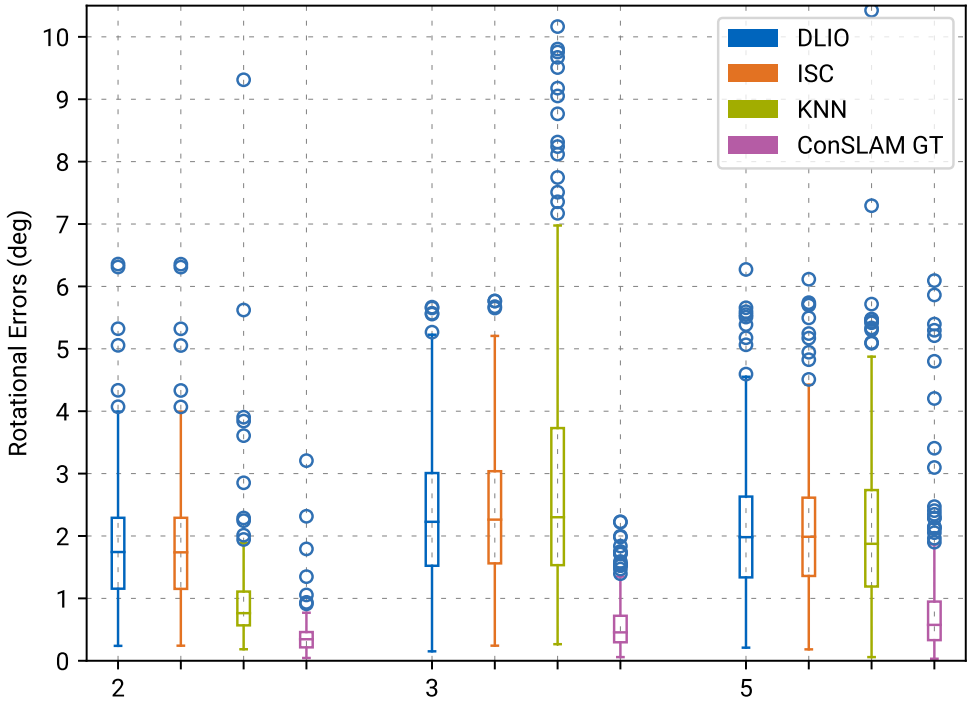


Figure 5.11: Rotational errors for sequences 2, 3, and 5 after alignment with the respective TLS point clouds.

in relatively small sections of the trajectories, it is also essential to recognize that for a LiDAR-based SLAM dataset, ground truth poses should ideally exhibit accuracy levels of at least one centimeter across the entire trajectory. This level of accuracy is now achievable in a highly automated manner with the proposed SLAM2REF framework.

Additionally, it was demonstrated that it is possible to align and correct a 3D map using a BIM model as a reference map, despite significant deviations between the current map and the reference BIM model (Scan-Map deviations of the types 1 and 2 as stated in the introduction, see Section 1). This significant level of deviation is particularly evident in the context of the ConSLAM construction site. Figure 5.12 and 5.13 depicts the results after alignment with the BIM model. Here, the error values after the final ICP step are visible since they do not coincide with the ground truth poses anymore.

Similar to the alignment process with the TLS point cloud presented previously, the error does not decrease after ISC loops; however, it notably reduces after the final ICP step. The translational RMSE of the APE decreases to 14.8 cm, while the rotational RMSE is 0.56 degrees.

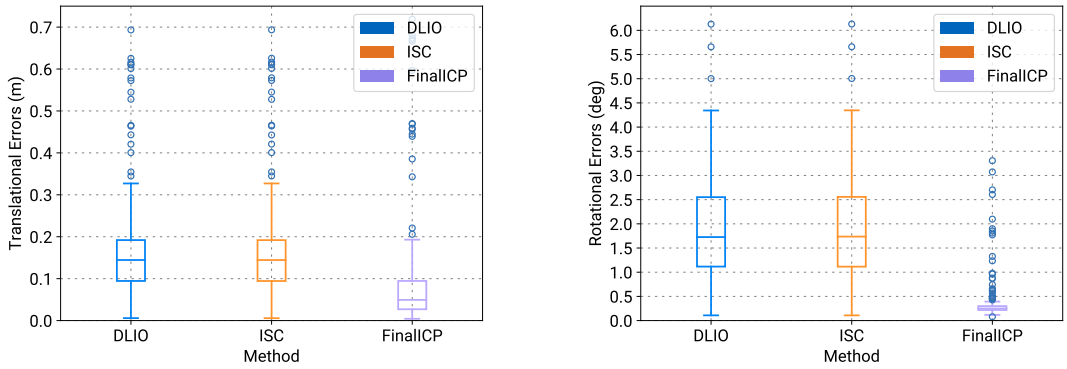


Figure 5.12: Translational (a) and rotational (b) errors for the sequence 2 after alignment with the BIM model.

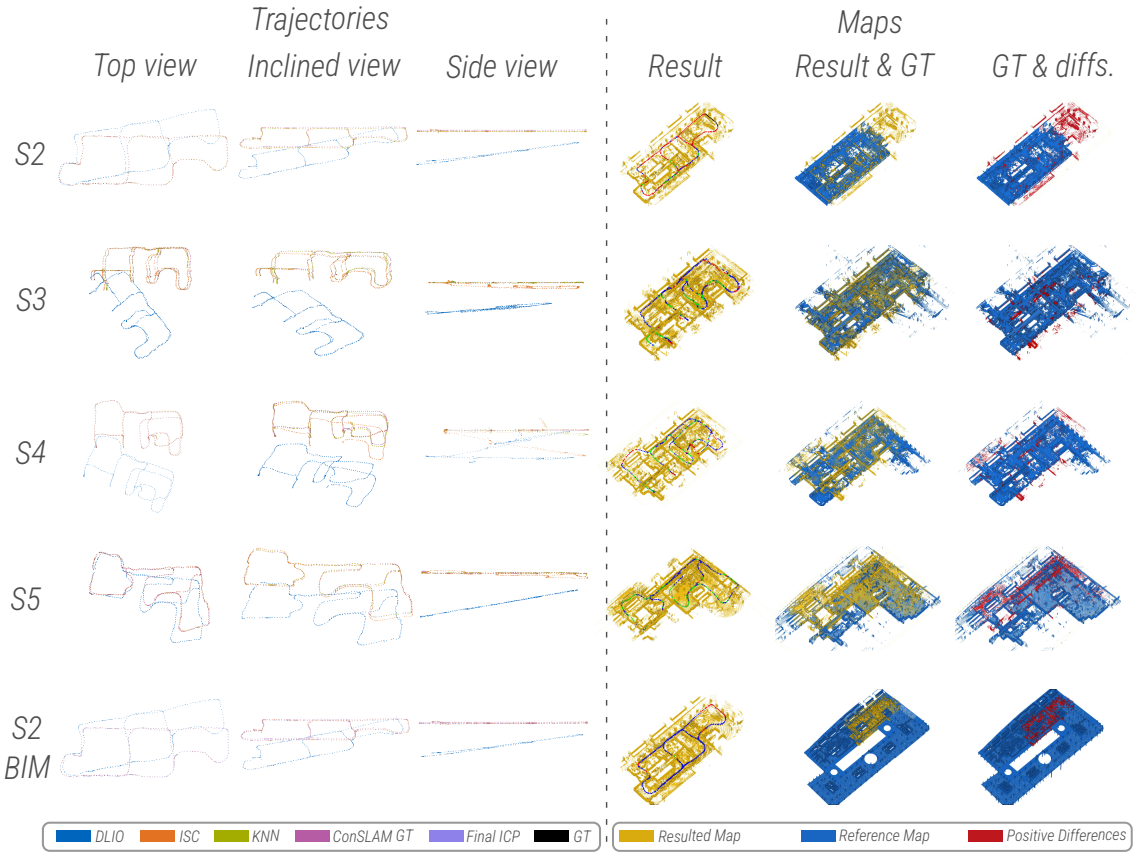
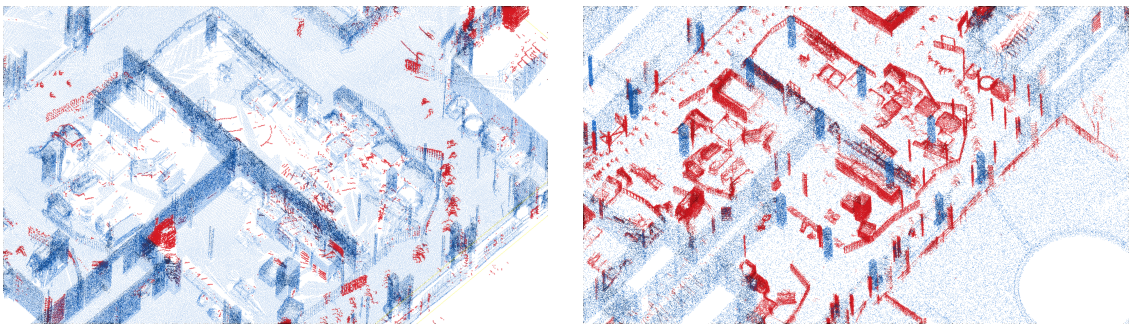


Figure 5.13: Trajectories and maps for sequences 2, 3, 4, and 5 after alignment with the respective TLS point clouds, and for sequence 2 after alignment with the BIM model. The trajectories of the first three columns correspond to the results of the different methods/steps, which have the same label colors as in Fig. 5.10. Additionally, the ground truth trajectory is shown in black. The trajectory in the fourth column displays points in different colors to indicate registration results: perfect (green), good (blue), bad (red), or outside of the map (black). In the fourth and fifth columns, the resulting map is shown in yellow, and the reference target map is shown in blue. In the last column, the differences (new elements in the resulting map) are depicted in red.

Regarding the results of *Step 3*, the identified positive changes are highlighted in red within the final column of Figure 5.13. Furthermore, Figures 5.14a and 5.14b provide detailed visualizations of the discrepancies observed in sequence 2 following alignment with both the TLS point cloud and the BIM model, respectively. While the disparities with the TLS point cloud are relatively minor, involving slight shifts in the positions of certain fences and construction resources, the distinctions when compared to the BIM model (Figure 5.14b) are notably substantial. This serves to exhibit the robustness of the proposed alignment methodology in effectively accommodating considerable levels of Scan-Map deviations.



(a) (b)

Figure 5.14: Change detection after map alignment. Both images correspond to the results of sequence 2: in (a), the sequence was compared against the respective TLS point cloud, and in (b) against the BIM model. The positive differences, i.e., new elements in the resulting map, are depicted in red.

5.6 Discussion

This section contains a more detailed interpretation of the results reported previously. Furthermore, it looks into the motivation for the methodology and how it contributes to progress in this field of research, explaining the enhancements of the approach compared to prior works and outlining directions for future studies.

The apparently contradictory pattern of the rotational errors in sequences 2 and 5 can be attributed to the Umeyama alignment process (Umeyama, 1991). In certain regions, the actual trajectory after ISC loop detection (without Umeyama alignment) deviates approximately 1.5 meters from the ground truth in the Z and X directions, leading to erroneous identification of KNN loops. Nonetheless, these erroneous loops are effectively identified and filtered out during the final ICP step.

One potential source of error for the ConSLAM GT poses lies in the MDC step. Contrary to common practice, the authors extracted the scans directly from the recorded *bagfiles*, omitting the deskewing process Trzeciak et al. (2023a). Avoiding the undistortion process can mislead any registration method, particularly affecting the accuracy of the calculated poses in sections where the trajectory was recorded under rapid motion.

The reason why KNN loops tend to yield incorrect correspondences during alignment with the SD from a BIM model can be attributed to Scan-Map deviations, as well as the absence of ceiling points in simulated scans from the BIM model. These facts complicate the registration of small sub-maps from the real world with sub-maps from a BIM model, particularly given that elements on the construction site have corners and features sometimes misinterpreted by the YawGICP registration process as permanent elements. Nonetheless, the final ICP method overcomes this challenge by utilizing a dense point cloud from the BIM model and relying solely on P2P correspondences, thus avoiding estimating tangent planes for the alignment.

An alternative to simulating LiDAR scans (as done in section 5.3.1) could involve cropping a point cloud from the reference map within spheres as performed for the final ICP step (section 5.3.2). However, simulating scans offers a critical advantage: it enables the incorporation of only the geometry of elements visible from the scan's origin. This visibility filter is crucial for ensuring the robustness of descriptor-based alignment in the ISC loop detection step, as only the information of single scans is compared here. Furthermore, when registering real-world scans with simulated ones, the process not only demonstrates quickness but also mitigates potential interference from double surfaces, such as from walls, as only the visible surfaces from the sensor origin are considered.

In comparison to prior research (M. A. Vega-Torres, Braun, & Borrmann, 2022) and other localization algorithms, SLAM2REF presents notable advantages. Since it enables the creation of a map and subsequent alignment with a reference map, unlike typical localization methods, the framework does not require the sensor to initiate mapping from within the map itself. Instead, it allows the sensor to start from any location, ensuring that the resulting map aligns with some overlapped regions of the reference map. Therefore, SLAM2REF also supports the extension of the reference map. This means that even if

sensor measurements expand beyond the map boundaries, they are still aligned with the existing map in the most coherent manner.

Additionally, the proposed pipeline does not necessitate any manual intervention to align the first keyframe, a process typically required by methods utilized to generate the ground truth poses in some of the latest datasets, such as in (Ramezani et al., 2020; Trzeciak et al., 2023a, 2023b; L. Zhang et al., 2022). Moreover, due to the initialization of the proposed pipeline with SLAM or odometry-calculated poses and the optimized parallel registrations, the proposed pipeline also enables the rapid retrieval of GT poses utilizing dense, accurate reference maps.

When contrasting with BIM-SLAM (M. A. Vega-Torres et al., 2023), SLAM2REF showcases several distinct advantages: Firstly, it is compatible with large-scale reference maps, encompassing not only large BIM models but also dense high-quality point clouds. Secondly, it effectively considers motion distortion in LiDAR data and mitigates it by leveraging IMU measurements. Thirdly, it achieves significantly improved accuracy in 6-DoF pose retrieval through the final ICP step and a TLS point cloud as a reference map. Lastly, the proposed pipeline enables the alignment in the presence of Scan-Map deviation, such as with a BIM model, leveraging the proposed enhanced version of the Scan Context descriptor tailored for indoor environments.

Additionally, the proposed pipeline operates independently of ROS or Gazebo (used previously for scan simulation). Another remarkable characteristic of the proposed method is its adaptability, as it is not restricted to Manhattan-world environments with enclosed rooms, as the method proposed by Shaheer et al. (2023).

5.7 Conclusions

This contribution presents SLAM2REF, a modular framework to allow automatic 3D LiDAR data alignment and change detection with a reference map, which can be a BIM model or a point cloud.

The framework operates independently of the sensor's initial position, eliminating the necessity for the scanning process to start within the provided map boundaries. Conse-

quently, the proposed framework enables map alignment and extension even when the reference map is outside the sensor's FoV or only a portion of the map has been scanned.

Moreover, if an accurate TLS point cloud is available, it can serve as a reference map to correct the poses of a query session and even retrieve centimeter-accurate ground truth poses.

In conclusion, SLAM2REF offers a novel solution to the challenges of lifelong mapping by integrating 3D LiDAR data and IMU measurements with a reference map, enabling automatic alignment, precise 6-DoF trajectory estimation, map extension, and change detection.

By allowing Scan-Map deviations, SLAM2REF offers a robust solution for automated 3D data alignment, even with as-designed BIM models that typically have significant deviations from as-built environments.

Our approach provides indirect support for the development of Digital Twin for buildings, allowing the automatic alignment of newly acquired data with digital models. These models require continuous data integration to maintain its accuracy and relevance.

Practical applications are found in areas such as construction site monitoring, emergency response, disaster management, and others, where fast-updated digital 3D maps contribute to better decision-making and productivity.

Furthermore, since the proposed method is capable of exploiting BIM models that are semantically enhanced or point clouds as reference maps for localization, it can be used to support the development of autonomous robotic activities.

Another advantage of SLAM2REF is that it advances SLAM research by enabling the automatic retrieval of centimeter-level accurate 6-DoF GT poses for large-scale indoor and outdoor trajectories.

5.8 Contributions and Limitations

5.8.1 Contributions

This chapter presents significant contributions to the field of multi-session anchoring, specifically focusing on the integration of 3D BIM models and point clouds with 3D LiDAR-based SLAM systems. The key contributions are the following:

C 2.1 Automatic Generation of Accurate Occupancy Grid Maps and 3D Session Data from a Reference Map (RQ 2.1):

- A module was developed to create fully automated, reliable, and accurate **OGMs** from BIM models and 3D point clouds, enabling direct usage for localization and autonomous navigation tasks using the ROS navigation stack.
- A method was devised to generate **3D session data** from large-scale BIM models or point clouds, allowing rapid place recognition and localization within a reference map.
- The Map2SessionData code to facilitate reproducibility and enable further development by the research community was made open-source. Link to the repository.

C 2.2 Alignment and Correction of Drifted Sessions with a Reference Map (RQ 2.2):

- **Indoor Scan Context (ISC)** was introduced, a new LiDAR scan descriptor built on the widely-used Scan Context (SC) family of descriptors, specialized for place recognition in indoor environments rather than the original SC descriptor's focus on outdoor autonomous car localization.
- Additionally, **YawICP** was introduced, a simple yet powerful and efficient algorithm for point cloud registration, primarily addressing variations in yaw angles (Z-axis angles). This type of registration is crucial for finding the best transformation to align scans from 360-degree LiDAR measurements.
- The final step in the alignment method includes a **refined P2P ICP** procedure to achieve 3 cm accurate and reliable 6-DoF pose retrieval.

- All these and some additional components were integrated into **SLAM2REF**, an open-source **holistic multi-session anchoring system** that enables the alignment and correction of drifted sessions acquired with SLAM or LiDAR-inertial odometry systems in indoor or outdoor environments. Link to the repository.
- The code needed to create SD from any LiDAR-based SLAM or odometry framework were made open-source.
  Link to the repository.[10]
- The BIM model created for the experiments along with the computed GT poses, has been made openly accessible at: (M. A. Vega-Torres, Braun, & Borrmann, 2024b).

C 2.3 Analysis and Change Detection in Aligned Data (RQ 2.3):

- A module was developed to analyze the acquired aligned data. Using visibility analysis with the OctoMap library and point-to-point distance measurements, the module can detect both **positive** (new elements in the scene from the real-world's most up-to-date data) **and negative differences** (elements removed from the scene).
- A method to visualize the updated 3D map in the form of color-coded voxels was proposed.

[10]For a comprehensive list of all open contributions, refer to Section 1.14.

5.8.2 Limitations

Although this chapter highlights significant advancements and contributions, it is important to recognize also the limitations of the proposed methodology:

L 2.1 Lack of Reliable Landmark Extraction for Registrations:

- A significant challenge for any registration algorithm is the identification of accurate correspondences. In this scenario, the presence of Scan-Map deviations intensifies this challenge. The method's success with the current BIM model (in the conducted experiment) does not necessarily indicate it will perform well in environments with higher levels of clutter. Although the semantics of the BIM model are leveraged to generate session data, the method does not semantically enrich the collected real-world data. Consequently, the method relies heavily on scans from locations with minimal Scan-Map deviations, as only in such locations will the extracted features from the BIM model align with those of the LiDAR scan. One possible solution to this issue is the application of point cloud semantic segmentation algorithms to individual 3D LiDAR scans. However, since most existing methods are designed for outdoor environments, adapting these algorithms for indoor construction settings would require labeled datasets specifically tailored to indoor construction environments.

L 2.2 Complexity and Scalability Issues in SLAM2REF Method:

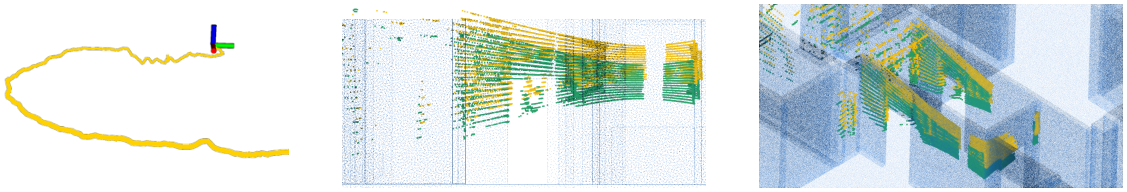
- Due to sensor noise, potential inaccuracies in the undistortion process with IMU measurements, and Scan-Map deviations, achieving a 1 cm precision 6-DoF pose for all keyframe scans remains challenging. For applications requiring this level of precision in every single timestamped scan, laser trackers and motion capture systems are used. However, considering that some SLAM researchers even consider it acceptable to use the end-to-end translational error measurement, the proposed method provides a sufficient number of 1 cm precision 6-DoF poses to study SLAM algorithms along complex trajectories.
- Additionally, the complexity of the alignment method escalates with the increase in the number of keyframes, particularly from the query session. This is because the final ICP registration must be conducted for each individual

scan of the query session using a dense segment of the point cloud from the reference map.

- The method operates offline (not in real-time) and necessitates fine-tuning of specific parameters, particularly when using a BIM model as a reference map instead of a point cloud. Enhancing the efficiency and robustness of the SLAM2REF method to support a real-time framework offers promising potential for various applications, such as collaborative robot mapping and localization (Cramariuc et al., 2022; Lajoie & Beltrame, 2024). This advancement could effectively address challenges like the kidnapping robot problem in indoor environments using the proposed ISCD. Furthermore, a critical aspect of achieving more robust alignment involves utilizing deep-learning-based place recognition algorithms, which are expected to become increasingly reliable for indoor scenarios with adequate training data in the future.

L 2.3 Special Challenging Cases:

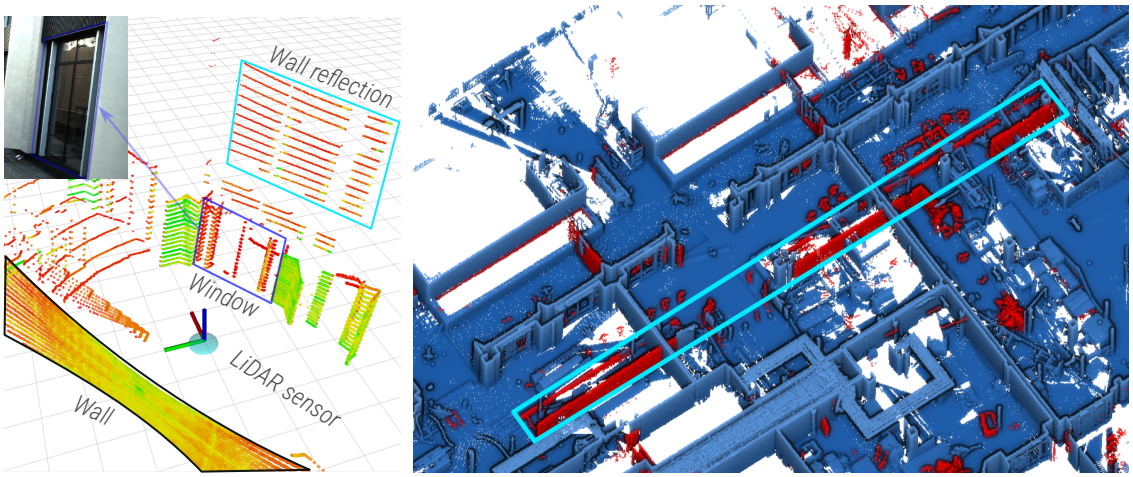
- **Height Retrieval in Narrow Corridors:** The developed alignment method struggles to accurately retrieve the height (Z-coordinate) of scans located in narrow corridors that lack points on the ceiling or the floor, as depicted in Figure 5.15. While the correct X and Y coordinates can likely be obtained, the Z coordinate may be erroneous, particularly when the SLAM or LiDAR-Odometry algorithm fails to provide an accurate initial height estimate, often due to rapid motion. In particular, when encountering a Z-drift within a narrow corridor, the limited information provided by individual scans regarding horizontal elements (such as floors or ceilings) can sometimes make automatic height retrieval very challenging. One possible approach to mitigate this issue involves the utilization of the *free space*, which can be calculated using OctoMap (as outlined in Section 5.3.3). Assuming most transition elements, such as doors and windows, are open during scanning, they could serve as reference elements for height retrieval by utilizing their frames as a feature for registration.
- **Reflections from Window Elements:** Since the method does not account for semantics in real-world measurements, there is a potential for the creation of reflections from window elements in the final map. Figures 5.16a and 5.16b



(a) (b) (c)

Figure 5.15: Limitation regarding the inaccuracy of initial poses: If the initially calculated poses drift in the Z direction, as depicted in (a), the final scan might not be automatically registered in the correct poses if they are located in narrow corridors. This occurs because specific scans, like the yellow one in (b) and (c), lack the ceiling or floor points necessary to determine the correct sensor height. (b) and (c) are the side and perspective views of the same scene. The green scan represents the manually correctly registered scan, while the blue depicts the reference map, in this case, a point cloud sampled from the BIM model.

illustrate how windows cause reflections in LiDAR measurements, resulting in fictitious reflected elements within detected changes. Notably, reflected walls are visible in sequences 3, 4, and 5 (refer to the last column of Figure 5.13), whereas sequence 2 is unaffected due to the absence of windows and the scan trajectory being confined within walls without windows. To mitigate this issue, one approach is to use camera measurements to selectively filter out LiDAR data collected near windows. Alternatively, a more manual and labor-intensive method involves physically occluding windows prior to scanning.



(a) (b)

Figure 5.16: Limitation regarding wrongly detected changes following map alignment. In (a), the presence of reflective surfaces, such as windows, can lead to the generation of fictitious walls due to LiDAR's inability to filter out reflected measurements. This is illustrated in the top left, where an image of the actual window causing reflections is depicted, while at the bottom, the LiDAR measurement captures the wall, window, and the reflected fictitious wall. In (b), these reflections result in inaccurately detected changes in sequences 3, 4, and 5.

- **Constraints in ISCD and Loop Detection:** The proposed ISCD and the associated loop detection step are constrained by their design, which targets LiDAR scans from sensors with a 360-degree horizontal FoV. This design choice aims to facilitate the creation of rotationally invariant 1D descriptors. Consequently, the pipeline is not directly compatible with data from sensors with a reduced FoV, such as solid-state LiDARs and depth cameras.

This chapter introduced significant contributions to creating aligned, updated 3D maps of the environment from even heavily drifted session data. Utilizing novel place recognition descriptors and registration algorithms, this comprehensive method also detects positive and negative changes in the environment. However, the proposed method has several limitations, such as the dependency on a low level of Scan-Map deviations to find correct correspondences, the reflections caused by windows in the scanned environment, and the constraints with respect to the FoV of the sensor measurements.

In the upcoming chapter, some of these limitations will be addressed through the integration of camera information, semantic extraction from real-world data, and the application of a bundle adjustment algorithm to refine sensor poses derived from reduced FoV measurements. This approach is designed to significantly improve the accuracy and robustness of the updated 3D maps, aligning it with a reference map. Specifically, it prioritizes working with reduced FoV data and ensures better-filtered landmarks, thereby enhancing map alignment accuracy, rather than relying on non-semantically enriched, undistorted 360-degree LiDAR scans.

Chapter 6

AI-supported Integration of LiDAR and Camera Data with BIM Models and Reference Maps

This chapter introduces two innovative approaches designed to enhance the alignment of LiDAR data when fused with camera data, utilizing BIM models or semantic reference maps.

The first method addresses the challenge of globally registering a SLAM-based reconstructed point cloud with a BIM model. This approach assumes that the real-world data was acquired with a low level of drift and has been semantically enriched.

The second proposed method refines sensor poses by integrating LiDAR and camera data, enhanced with semantic information. This method aligns elements such as walls, columns, floors, and ceilings from real-world measurements with their corresponding components in a 3D BIM model. The solution employs novel neural depth completion and image semantic segmentation algorithms to achieve accurate alignment.

6.1 Motivation

In the method proposed in the previous Chapter 5, a comprehensive approach was presented to align SLAM-based session data with a reference map, even in cases of significant drift, such as in sequence 4 of the ConSLAM dataset (Trzeciak et al., 2023a, 2023b) (see Fig. 5.13).

However, in many scenarios, it is reasonable to assume that only a minimal percentage of drift will be present in the acquired map. This assumption is based on the expectation that future SLAM algorithms will become more robust, sensor noise will be reduced, and

techniques for mitigating distortion due to motion will improve. Additionally, if the data is acquired carefully with slow-speed motion, the drift is expected to be low as well.

Moreover, given the fast advancements in AI, it is reasonable to aim for methods that exploit acquired data not only with low drift but also with enhanced semantic information.

Considering the previously mentioned aspects, it is reasonable to conceive a method that allows offline global registration of point clouds with low drift and a reference semantic map, such as a BIM model. Such a method would allow the first alignment with less computational demand in comparison with the SLAM2REF method introduced in the previous chapter (Chapter 5) mainly because it will avoid the comparison of every single scan from the query to the central session, which in a global registration problem would be reduced to the comparison of extracted features from the source to the target point cloud.

Once the initial alignment is complete, the sensor poses should ideally be further refined to eliminate any remaining drift in the map. To address this challenge, the BIMCaP framework is proposed. This framework integrates mobile reduced FoV sparse LiDAR data with camera measurements to filter semantic landmarks and refine sensor pose using BA and pre-existing BIM models or semantic 3D maps. Filtering semantic landmarks, such as walls and columns, is essential to manage the issue of Scan-Map deviations. This step ensures that only permanent elements from the real world are matched with those in the reference map, thereby avoiding false registrations caused by clutter or temporary objects. Through this approach, BIMCaP aims to overcome some of the limitations inherent in the SLAM2REF method.

6.2 Research Questions

In line with the motivation and requirements stated above, this chapter aims to answer the following main research question.

RQ 3. *How can semantics and LiDAR-camera fusion be utilized to create a robust alignment and correction method of SLAM-acquired real-world 3D data with a BIM model or a semantic 3D map?*

The following are the specific three sub-research questions addressed in this chapter:

RQ 3.1 Cross-source global registration for aligning SLAM-reconstructed point clouds with BIM models:

- Given a low percentage of drift and a semantically enriched SLAM-reconstructed point cloud from the real, cluttered world, how is it possible to find the transformation matrix that roughly aligns that point cloud with the corresponding BIM model?
    - *Rationale:* Assuming low drift and semantic enrichment is reasonable in many practical scenarios. A robust global registration algorithm can significantly reduce computational costs by efficiently aligning the entire map instead of individual, separated scans. However, this cross-source registration process is challenging due to anomalies in scans, incomplete real-world data, and different levels of Scan-Map deviations, necessitating advanced techniques to manage these complexities.

RQ 3.2 Semantic enrichment of 3D maps in real-world construction sites with LiDAR-camera fusion:

- How can camera and LiDAR data be effectively fused to create semantically enriched 3D point clouds of indoor construction site environments?
    - *Rationale:* Semantic segmentation techniques have advanced rapidly; however, their application in indoor construction site environments remains under-explored. Similarly, the integration of sparse LiDAR and camera data poses significant challenges to creating accurate, dense maps of indoor construction site environments.

RQ 3.3 Refinement of drifted data with a reference BIM model:

- How is it possible to improve pose refinement accuracy by integrating semantic, textural, and geometrical features from camera images and LiDAR scans, using a BIM model as a reference?
    - *Rationale:* This investigation builds upon the challenges identified in **L 2.1**. It aims to enhance the precision and reliability of 6-DoF pose retrieval by leveraging semantic information from both the reference BIM

model and real-world data. By minimizing drift and enhancing alignment, the goal is to improve the quality of the updated 3D map, even under large levels of Scan-Map deviations. Ultimately, this contributes to advancing SOTA methods in indoor construction site mapping and monitoring

To address the aforementioned research questions, a cross-source global registration methodology for approximate alignment of real-world point clouds with BIM models is proposed. Additionally, BIMCaP is introduced, a novel framework that integrates 3D LiDAR data and RGB camera measurements with a reference map. This framework aims to further refine single sensor poses and enhance the accuracy of updated aligned maps.

6.3 Global Registration of Cross-Source Data[1]

This section introduces the cross-source global registration methodology for approximate alignment of real-world low-drift SLAM-created point clouds with BIM models.

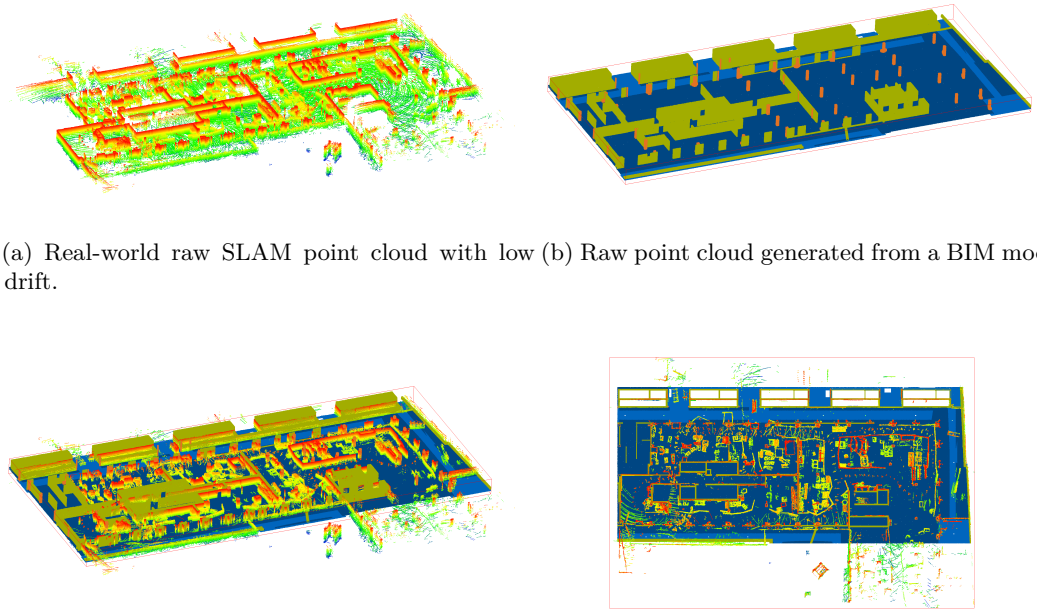
Besides the low percentage of drift in the SLAM point cloud, the proposed global registration pipeline is based on the following assumptions:

- It is assumed that there are sufficient orthogonal walls in most buildings, allowing for the use of methods based on right-angled intersections and corners.
- Both maps (i.e., the SLAM-acquired point cloud with low drift and the reference semantic map, such as a BIM model) include semantic information. This means that the data captured by the scanners are not only geometric but also inherit semantic information either by automatic or manual segmentation by using the respective BIM model to obtain these.
- The gravity direction is pointing upwards and set as the Z-axis, which can be either obtained by a modern laser scanner or set manually. This definition is essential for consistency across different datasets and dataset sources.

[1]The methods described in this section were developed in collaboration with Armin Kamberi as part of his master's thesis, supervised by the author of this dissertation.

For easier understanding, the following terminology is established: The SLAM reconstructed point cloud is the *source* point cloud. The BIM reconstructed point cloud is referred to as the *target* point cloud, i.e., the reference map to which the alignment is required.

Fig. 6.1 presents the raw maps that the proposed methodology seeks to align. The real-world SLAM point cloud includes extra elements (such as clutter and materials) that must be filtered out to successfully align the map with the reference BIM model which only contains the permanent elements of the building (such as walls and columns).



(a) Real-world raw SLAM point cloud with low drift.

(b) Raw point cloud generated from a BIM model.

(c) SLAM & BIM maps aligned and overlapped.

(d) Top down view of SLAM & BIM maps.

Figure 6.1: Point cloud dataset overview

The methodology is divided into three main steps as follows: **Step 1.** The BIM model is sampled to a point cloud representation, transforming semantic labels to distinct colored points, and the SLAM reconstructed point cloud is enriched with semantic labels through a manual alignment of the BIM and SLAM point cloud.

Step 2. The initial angle alignment step uses orthogonal walls to align source and target point clouds, aiming to find the correct rotational angle for the alignment. It creates a histogram of normal angles, using wall and column points, and identifies main wall orientations. After this first rotational alignment, there are only *four possible directions* in which both point clouds align correctly. The *four directions* refer to the cardinal directions (0°, 90°, 180°, and 270°).

Then, in **Step 3.**, two methods are implemented to find the final best transformation for global registration; these methods are described as follows:

Method 1 constructs an Occupancy Grid Map (OGM) by projecting each point from the point cloud onto the XY-plane, resulting in a 2D OGM. Subsequently, 2D features are extracted using the Features from Accelerated Segment Test (FAST) algorithm, as implemented within the Oriented FAST and Rotated BRIEF (ORB) framework. The Binary Robust Independent Elementary Features (BRIEF) descriptors are then generated based on these extracted features. For matching, the *Hamming distance* metric[2] is employed to assess the similarity between descriptors from the source and target datasets. Based on the identified correspondences, the median length of the matches is computed, which serves as the basis for estimating the translation. To ensure robustness, the registration process is evaluated across the four cardinal directions within the 2D space. The final transformation is selected based on the registration with the highest Fitness Score (which will be explained in 6.3.4), ensuring the best alignment.

Method 2 differs from *Method 1* as it builds correspondences based on the median length of the center points of the extracted columns. In this approach, the points representing columns are extracted using the semantic labels, which are then projected into 2D. The final transformation is estimated following the same principles as in *Method 1*.

Initially, the number of correspondences within specific length ranges of the histograms (derived from all built correspondences) is counted. Subsequently, the median length of all distances is utilized to estimate the translation. Finally, a final transformation is determined after iterating through all four possible rotation directions. Figure 6.2 presents an overview of the presented framework.

6.3.1 Step 1: Preprocessing

To create a 3D semantic point cloud from a BIM model, first, it is converted from IFC format into a triangulated mesh representation in OBJ format using ifcConvert (IfcOpenShell Contributors, 2023b). Following this, distinct OBJ files are generated for each entity within the model (e.g., walls, columns, floor, ceiling, windows, and doors). Then, uniform

[2]The Hamming distance is a measurement of the (dis)similarity between two strings or vectors of equal length, and provides the number of positions at which the corresponding symbols are different.

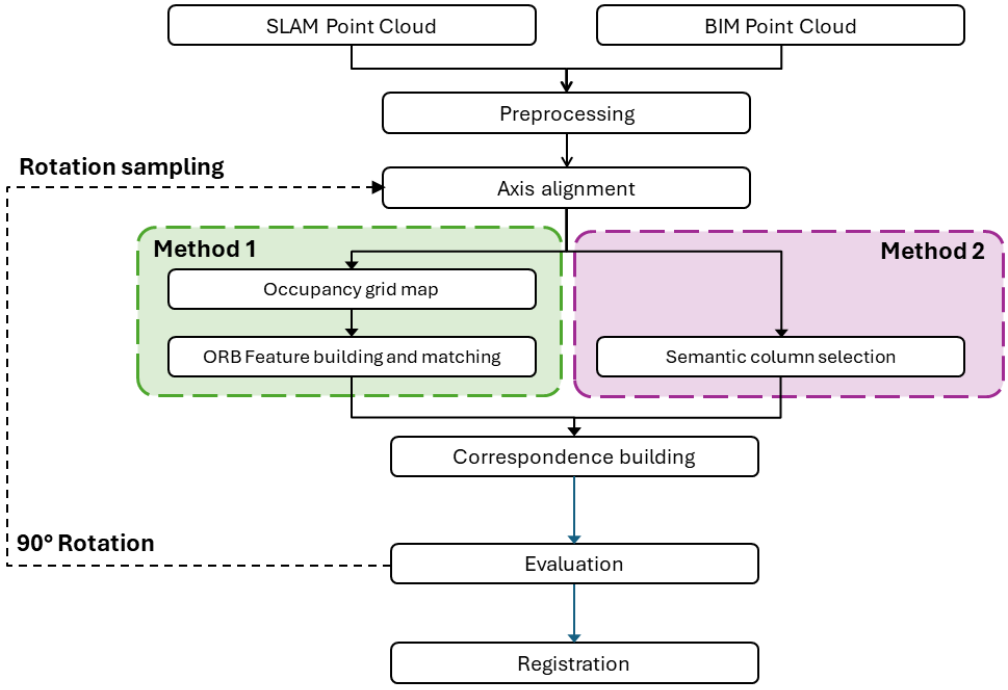


Figure 6.2: Pipeline overview.

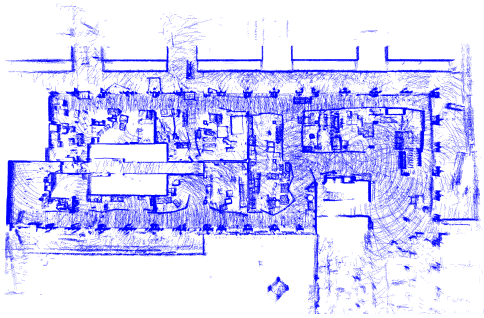
point cloud sampling is applied to each OBJ file, and the resulting semantically enriched synthetic point clouds are merged into a single one.[3]

To semantically enrich the source point cloud, it is manually aligned with the target BIM point cloud using CloudCompare (CC). Then, a labeling algorithm is used to determine points in close proximity between the source and the target. The target semantic labels, stored in the point cloud as color or scalar field information, are then assigned to the source point cloud.

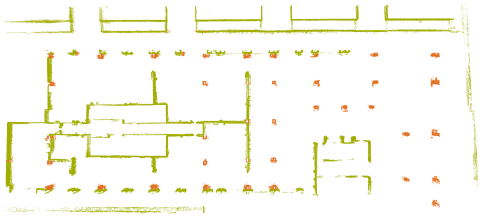
This process also helps to remove occlusions and noise in the source point cloud by only keeping source points that are in close proximity to a target point in the BIM point cloud. Figure 6.3 shows an example of an initial low-drifted SLAM point cloud and the result after semantic enrichment and cleaning.

Both point clouds are downsampled using a voxel-based method to reduce the number of points and decrease computational costs. This approach averages the coordinates of points within each voxel, producing a uniformly sparse dataset that retains the key features and shapes of the original data. However, careful selection of the voxel size is crucial to balance data reduction with the preservation of geometric details.

[3]The code to convert BIM models into semantically enriched point clouds can be found here: Link.



(a) Raw low-drifted SLAM reconstructed point cloud before cleanup and labelling.



(b) Cleaned and semantically labeled SLAM point cloud.

Figure 6.3: Semantic enrichment of SLAM point cloud with low drift (part of Step 1, see Section 6.3.1).

Standardizing the density between the source and target point clouds facilitates accurate comparison and alignment while significantly lowering the computational load of subsequent registration steps. A uniform point cloud density enhances the reliability of feature extraction and normal estimation, ensuring that the overall density of both point clouds is approximately equal, even though inherently sparse regions remain sparse.

Additionally, normals for the source point cloud are estimated using the nearest neighbor algorithm, which is essential for further processing. Consistent normal orientation is particularly important for surface reconstruction and point cloud alignment algorithms.

6.3.2 Step 2: Initial Angle Alignment

As stated previously, it is assumed that a sufficient amount of orthogonal walls are present in most buildings. Therefore, it is possible to use this consideration to align the point clouds to each other or the principal axis. By aligning the principal axis of the source and target point clouds, the solution space of the transformation for alignment is significantly reduced, which improves the performance of subsequent registration steps. The method for retrieving the primary alignment angle is similar to approaches previously proposed in the literature, such as those by M. A. Vega-Torres, Braun, Noichl, et al. (2022) and M. A. Vega-Torres et al. (2021) or by Sokolova et al. (2022).

To find the principal axis for each point in the point cloud, the angle of its normal vector, which is perpendicular to the surface at that point, is calculated in relation to the X and Y directions. These angles are analyzed within a range of 0° to 180°, treating angles 180°

apart as equivalent. This approach is taken because normal vectors pointing in opposite directions (e.g., 0° and 180°) are considered identical in this context.

All point normals that point up or down above 30° are considered floor or ceiling points. As the proposed pipeline relies on walls and columns, the floor and ceiling are not of interest and are filtered out.

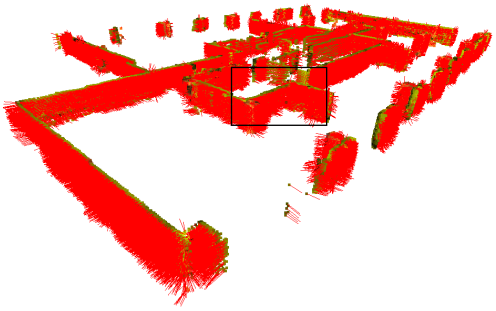
Given that, it follows that in a planar wall, the wall normals mainly point in one of the two orthogonal main directions. These orthogonal directions are determined by analyzing the histogram peaks of the normal vectors calculated for each point in the point cloud. The main direction of the point cloud walls is derived from these peaks, which represent the primary orientations of the building structure.

This step is performed for both source and target point clouds to obtain their respective orthogonal orientations and is illustrated in Fig. 6.4. Figures 6.4a and 6.4b show a 3D visualization of the calculated normals for the source point cloud. Figures 6.4c and 6.4d present histograms of the normal angles of the source and target point clouds, respectively. With the help of the determined angles, the source point cloud is rotated to align with the target point cloud so that they are either orthogonal or parallel to each other, depending on the initial rotation of the point clouds.

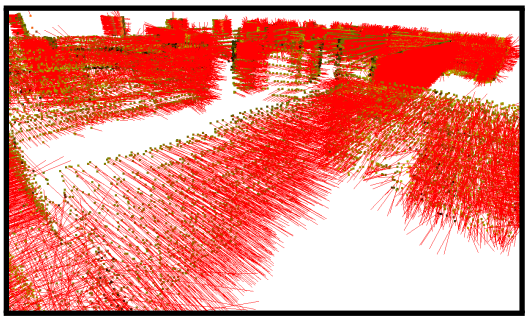
6.3.3 Step 3: Transformation Estimate

After preprocessing the data and finding the initial angle for alignment, two methods were developed to find the correct rotation and translation for the alignment. The first approach uses a 2D projection of the top-down view of the point cloud to build features of corners, points, or squares. The second approach builds correspondences directly through the center points of columns.

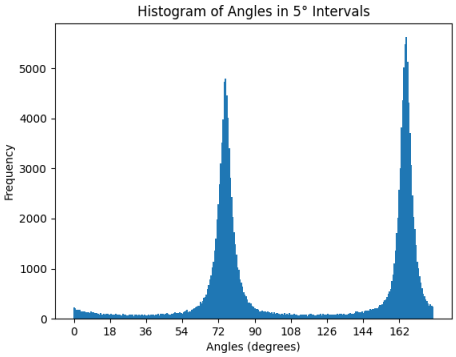
In both methods, the algorithm operates by assuming there are only four distinct possible solutions for the rotational angle (after Step 2, see previous Section 6.3.2). Among these four possible rotations, there is exactly one scenario where the lengths of the connections between recognized features are equal. This outcome follows from the property of translation, where correct correspondences are linearly dependent when the correct rotational solution is identified.



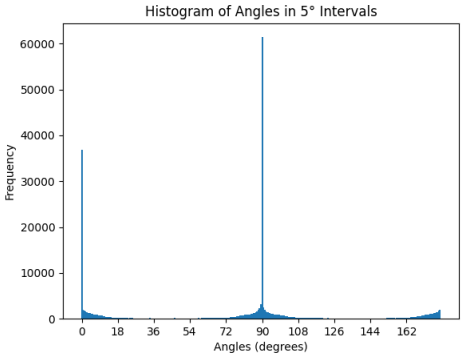
(a) Normal Estimation of SLAM point cloud.



(b) Close up view of wall normals in subfigure (c).



(c) SLAM Point cloud histogram wall normal angels. From this histogram it is possible to determine that the SLAM Point cloud has to be rotated around 73 deg.



(d) BIM Point cloud histogram wall normal angels. From this histogram, it is possible to determine that the BIM Point cloud does not need to be rotated.

Figure 6.4: Detailed overview of the axis alignment Process (Step 2, see Section 6.3.2).

Similarly, in both methods the algorithm iteratively finds the best solution through a process of rotation, translation, and evaluation. For each rotation candidate, the corresponding transformation is assessed, and the one with the highest fitness score is selected as the optimal solution. Fig. 6.2 illustrates this iterative process.

Method 1. ORB Feature-based Correspondence Building

First, the two-point clouds are discretized into a common 2D OGM with a predefined grid resolution. The accuracy of the conversion process depends on the size and density of the point cloud, and different grid resolutions may result in varying levels of accuracy. Section 6.3.4 discusses the optimal grid resolution. First, FAST features are generated based on the 2D image projection. Then, the BRIEF descriptors using the FAST features in both maps are determined. Using the Hamming norm, descriptors with low Hamming distances are matched, indicating high similarity between features. This approach ensures that

only corresponding features, present in both maps and representing the same underlying characteristic, are utilized for establishing mutual correspondence.

A histogram of all correspondence lengths matched using BRIEF is created to estimate the translation. Assuming the point cloud is correctly oriented, the maximum number of distances with the same length range equals the maximum inlier set. The translation is estimated and evaluated using the median of the histogram bin with the most correspondences.

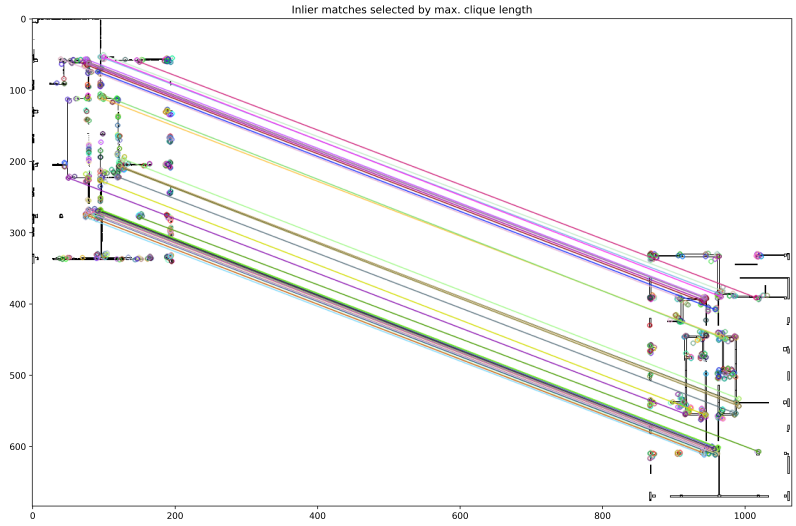
The optimal rotation candidate is selected based on several criteria, including fitness, inlier RMSE, Translational Error (TE), and Rotational Error (RE). These metrics are detailed in subsection 6.3.4. Figure 6.5 provides an overview of Method 1 and illustrates a representative registration result.

Method 2. Semantic Column Correspondence Building

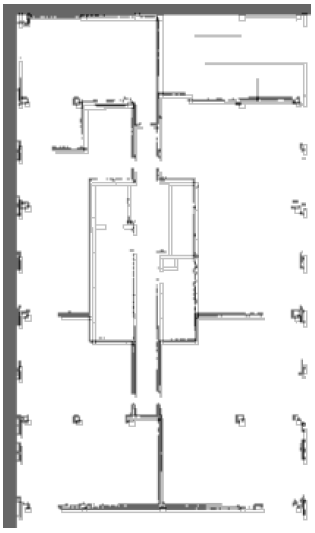
The second technique estimates correspondences from only the columns of the point clouds, similarly as done by Qiao et al. (2023). The process begins with applying DBSCAN to cluster column points within the semantically labeled point cloud data. A bounding box is then generated to enclose the column's width, accommodating variations in point density and addressing potential gaps in the source point cloud.

Subsequently, the center of the bounding box for each partial column is calculated and used to establish all-to-all correspondences with the columns in the target point cloud (derived from the BIM model). The term "partial" refers to the SLAM-reconstructed point cloud potentially lacking complete column data due to limited scanning perspectives and angles. Additionally, faster scans result in lower point density, and column edges may be sparsely captured if only briefly recorded while navigating through a room

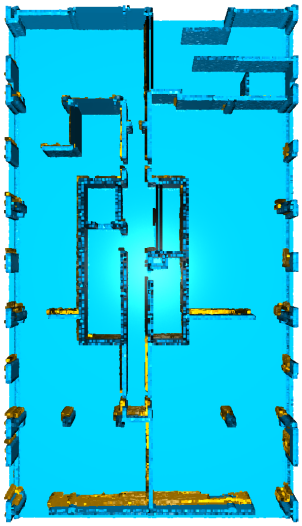
A histogram of all correspondences is constructed, following a process similar to Method 1, and the median of the bin with the most frequent distances is selected. The assumption is that if the rotation candidate is accurate, the bin with the highest frequency will correspond to the maximum clique, as discussed in Section 6.3.3. This procedure is repeated for each rotation candidate.



(a) left 2D occupancy grid map created form source point cloud; right the same OGM from the target point cloud; in the middle lines representing the correspondences which will be used to calculate the translation



(b) Both 2D OGMs overlapped after transformation

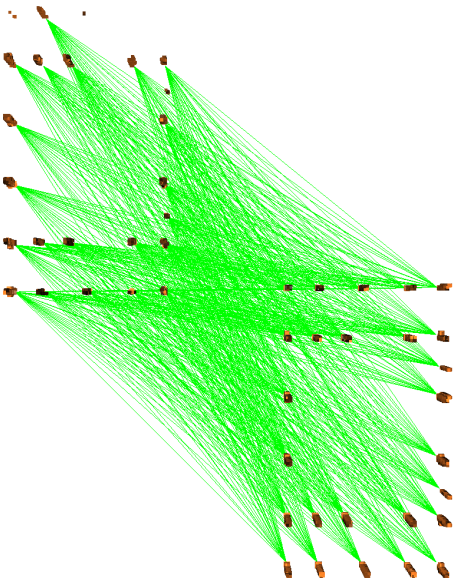


(c) Source (yellow), target (blue) after successful registration displayed as point clouds

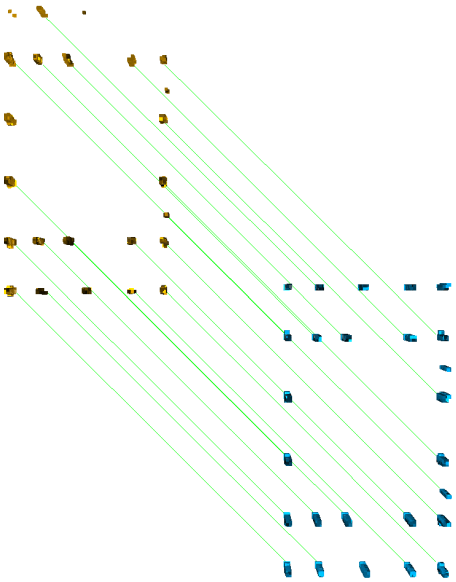
Figure 6.5: Overview of Method 1

Figure 6.6 provides an overview of the Method 2 workflow, along with an example registration result.

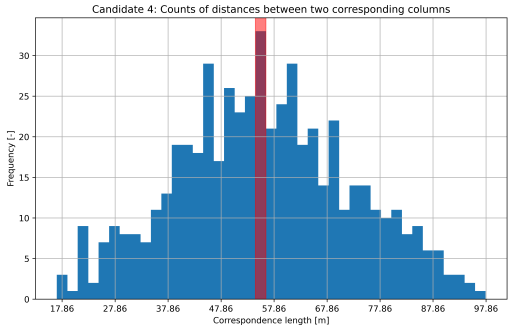
In both Method 1 and 2, the process iterates through all four possible rotation candidates. Each rotation candidate is evaluated, and the transformation estimates are compared. The best rotation is selected based on the fitness score, and the point clouds are registered accordingly.



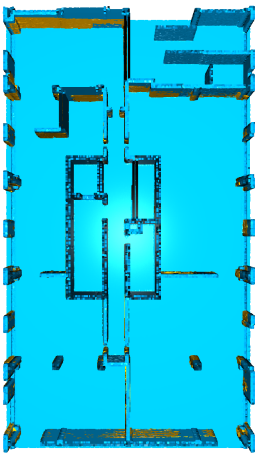
(a) All-to-all correspondences.



(b) Correspondence building based on column center.



(c) Histogram of all correspondence lengths with peak highlighted.



(d) Transformation estimate.

Figure 6.6: Method 2 overview showcasing the building of all-to-all correspondences on the top left. Selection of the correct inliers by median distance on the top right. Registration result on the bottom right.

6.3.4 Experiments and Results

This section presents the validation and testing of the proposed global registration algorithm across various dataset variations. Through three experiments, the algorithm's performance was assessed using standard metrics such as translational and rotational errors, fitness score, and inlier RMSE. Each test is designed to evaluate various aspects and scenarios of real-world applications, including same-size maps with small and large point clouds (Experiments 1 and 2) and registration involving maps with varying size, where the

entire BIM model is utilized as target (as in Experiment 3). Multiple tests were performed with random initial misalignments to assess robustness. This subsection concludes with a comprehensive analysis of the method's strengths and limitations.

Dataset

The ConSLAM dataset was obtained from the works of Trzeciak et al. (2023a, 2023b).

The data was collected using a LiDAR scanner (Velodyne VLP-16), and a map (of the Sequence Nr. 2) was created using the HDL-graph SLAM algorithm (Koide et al., 2019).

As illustrated in Fig. 6.1, this point cloud exhibits a high level of noise and significant areas of interference due to the placement of material and construction site tools. In some cases, walls were only scanned from one side, as sometimes the room behind them was not entered. Additionally, some of the columns have only been scanned on one side. It is also important to note that some points are outside the building. Most importantly, and as explained in Step 1 (see Subsection 6.3.1), the SLAM scans were cleaned from noise and clutter by manually aligning them and using the BIM model together with the nearest neighbor algorithm to label points that are in the vicinity of the BIM point cloud and labeled accordingly. This way, only the walls and columns were preserved and labeled. The dataset is cropped to specific sizes to test various SLAM and BIM configurations. An overview of the dataset sizes is presented in Table 6.1.

Table 6.1: Datasets used to evaluate the methods proposed in this research.

No.	**Name**	**Nr. Of Points**	**L x W x H [m³]**
1	Cropped SLAM	185.807	41 x 23 x 2.2
2	SLAM	545.051	80 x 35 x 2.7
3	Cropped BIM	135.068	41 x 23 x 3.9
4	BIM	283.386	80 x 34 x 3.9
5	Full Model BIM	910.588	166 x 86 x 3.9

Metrics for Validation Results

This section discusses the metrics and presents the results of the experiments conducted to evaluate the performance of the proposed methodology. The following metrics were used to quantify the accuracy and precision as well as evaluate the performance of the algorithms.

The **Translational Error (TE)** measures the difference in meters between the actual and estimated positions in the xy-plane. It is computed as the Euclidean distance between the actual translation (x_a, y_a) and the estimated translation (x_e, y_e), formally:

$$TE = \sqrt{(x_a - x_e)^2 + (y_a - y_e)^2}.$$

The **Rotational Error (RE)** assesses the difference in degrees between the actual and estimated rotations around the z-axis. It is calculated as the absolute value of the difference between the estimated rotation θ_e and the actual rotation θ_a, formally:

$$RE = |\theta_a - \theta_e|.$$

The **Inlier Root Mean Square Error (RMSE)** evaluates the average squared difference between the actual and estimated point clouds. It is defined as

$$\text{Inlier RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i)^2},$$

where N represents the total number of point correspondences and d_i is the Euclidean distance between corresponding points i. The advantage of RMSE is that it shows how far apart actual and estimated data points are. Squaring the differences makes it sensitive to large errors, which is useful when larger errors need to be weighted more. It is easy to compute and understand, so it is often used to measure the accuracy of estimation models.

The **Fitness Score** considers the number of corresponding points between the two clouds within a certain distance threshold, thereby measuring the degree of alignment. Given by

$$\text{fitness} = \frac{n}{N},$$

where n is the number of points within the specified threshold and N is the total amount of points in the source point cloud.

The **Success Rate (SR)** was defined in accordance with Qiao et al. (2023), where registration is considered successful if the resulting TE is within 1 m and the RE is within 3 degrees.

For each experiment, 100 registration tests were performed, and the SR was calculated by assessing the number of successful registrations according to the established criteria. The values for the other metrics were derived from tests that met the criteria and achieved the highest fitness scores. If a method did not have any successful registration, only the metrics of the test with the highest fitness scores were used.

The registration accuracy in each experiment was assessed under two scenarios: **Case 1** involves using the same SLAM point cloud as both the source and target (as a classical point cloud registration problem), while **Case 2** involves using the SLAM point cloud as the source and the corresponding BIM point cloud as the target.

The proposed methods were evaluated against SOTA algorithms, such as Truncated Least Squares Estimation And SEmidefinite Relaxation (TEASER)++ with Fast Point Feature Histograms (FPFH) features and TEASER++, after semantic column center extraction, which will be abbreviated as *TEASER (Sem)* in the rest of this section. Moreover, the results of the Random Sample Consensus (RANSAC) algorithm (after 100k iterations) are also provided.

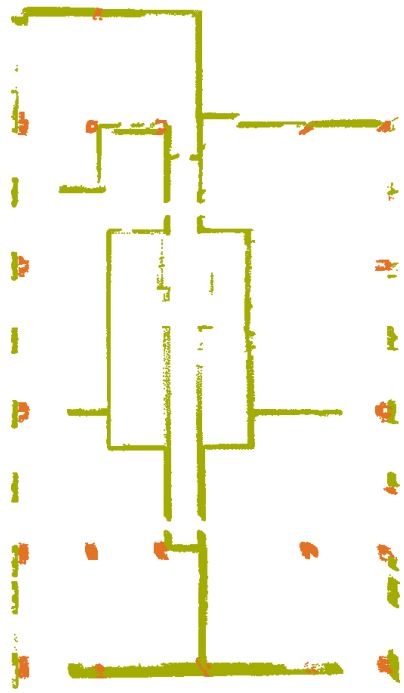
Experiment 1: Small Subsection

Exp. 1 was performed on a selected subset of the entire available SLAM point cloud.

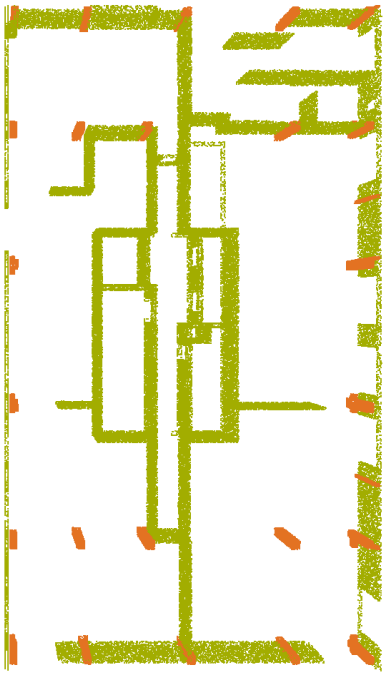
Fig. 6.7 illustrates the point clouds source and target to be registered, as well as both evaluation cases, i.e., when the point clouds are the same in target and source (same-source), and when the SLAM point cloud is registered with the BIM point cloud (cross-source).

In all presented experimental result tables of this section, the data is marked in **bold** for the best result according to its category and underlined for the second-best result.

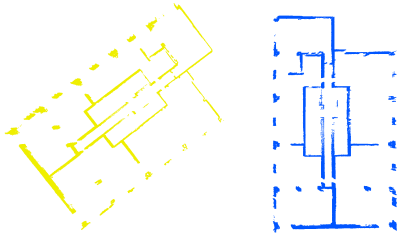
Table 6.2 illustrates that in **Case 1** (same-source case), the TEASER++ method with FPFH features and the TEASER++ method using only semantic column centers both achieved exceptional accuracy. Both methods produced highly precise registrations, evidenced by fitness scores of 1 and minimal translational and rotational errors. Both of the proposed methods (*Method 1 & 2*) delivered the next-best performance among all algo-



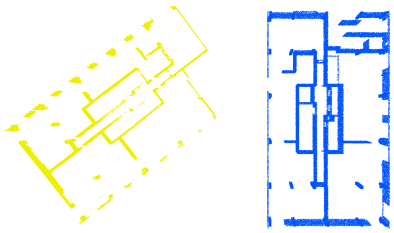
(a) Preprocessed semantically enriched SLAM point cloud (source).



(b) semantically enriched BIM point cloud (target).



(c) Exemplary Initial Situation Case 1 (same-source): rotation of -125° and translation in x = 40 and y = 5 applied to source.



(d) Exemplary Initial Situation Case 2 (cross-source): same angles as in Case 1.

Figure 6.7: Setup for Exp. 1.

rithms, also achieving a fitness score of 1, though with slightly higher translational and rotational errors.

The TEASER++ algorithm (with all points and with only semantic columns) performs exceptionally well due to its decoupled solving mechanism. The algorithm generates so-called Translational and Rotational Invariant Measurements (TRIMs), estimating correspondence that stays the same length when rotated and translated. These measurements are selected and compared to ensure a correspondence set with the maximum number of inliers, which enables successful registration. As the source and target are identical, the algorithm accurately identifies the same features with FPFH and a high number of

Table 6.2: Results of Exp. 1.

Case	Method	TE [m]↓	RE [°]↓	Fitness↑	RMSE↓	SR [%]↑
	RANSAC (FPFH)	8.55	35.95	0.14	0.19	3
No. 1	**TEASER (FPFH)**	**0.01**	**0.03**	**1.00**	0.08	**100**
Same-	**TEASER (Sem)**	**0.01**	**0.03**	**1.00**	**0.01**	**100**
Source	Ours (Method 1)	0.05	0.24	**1.00**	0.04	**100**
	Ours (Method 2)	**0.01**	0.24	**1.00**	0.04	**100**
	RANSAC (FPFH)	6.85	36.11	0.17	0.18	5
No. 2	TEASER (FPFH)	11.71	32.97	0.23	0.18	8
Cross-	TEASER (Sem)	16.34	45.76	0.10	0.16	0
Source	**Ours (Method 1)**	**1.09**	**23.64**	**0.94**	**0.10**	**87**
	Ours (Method 2)	11.74	88.44	0.22	0.16	1

Translational and Rotational Invariant Measurements (TRIMs), thus matching the same descriptors to yield the correct solution.

A comparable outcome can be observed in the case of Method 1. As the ORB algorithm can identify the identical FAST features in both the source and target, it can classify them using the same descriptors using BRIEF. The calculated Hamming norm is identical for all the detected features, resulting in an accurate correspondence matching.

The results of the RANSAC algorithm, when applied in conjunction with FPFH features, exhibit random performance as anticipated. Due to the iteration limit of 100,000, the algorithm was unable to identify the correct correspondences and estimate an accurate transformation.

As for **Case 2** (cross-source), where the target is now the corresponding BIM recreated point cloud, Method 1 and Method 2 demonstrate a clear advantage. The ORB algorithm in Method 1 can identify sufficient features with a low enough Hamming distance to match them correctly. Method 2 was able to estimate the correct rotation and translation by comparing the correspondence angles and distances between column centers.

TEASER++ with semantic column center correspondences was unable to achieve correct registration in this case. It is hypothesized that the substantial differences in distance between the source and target point clouds hindered the algorithm's ability to establish consistent measurements. Previous attempts to fine-tune the algorithm's parameters did not result in successful optimization. Additionally, the non-uniform density of columns, with greater density on one side, caused a shift in the column center. This shift affected the correspondence distance between source and target columns.

Regarding TEASER++ with FPFH features, it is assumed that the discrepancies observed in feature descriptors are similar to those seen with the semantic method. Given that the SLAM point cloud inherits drift from the reconstruction algorithm, it is hypothesized that these discrepancies lead to significant variations in correspondence distances between extracted features. This, in turn, complicates the process of building TRIM.

Experiment 2: Full SLAM point cloud

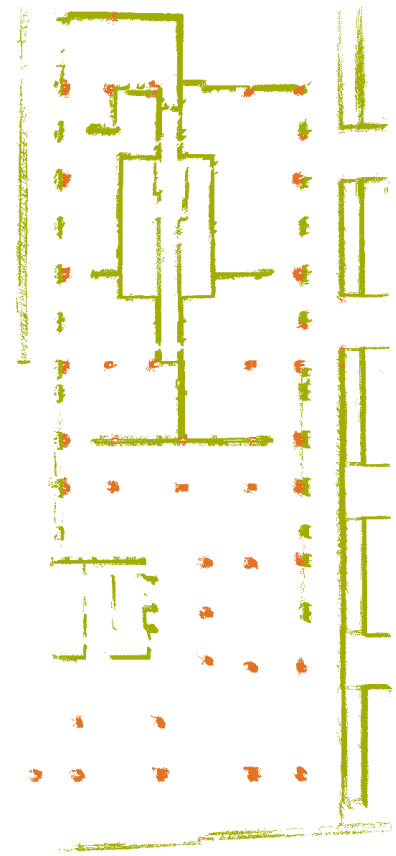
Exp. 2 was conducted using the entire available SLAM point cloud, first analyzing the same-source case followed by the cross-source scan. The results were systematically analyzed and classified. It is noteworthy that the point cloud's size increased substantially, leading to a significant rise in computational time. This increase also resulted in the detection of a considerably larger number of features in the same-source scenario. Similarly, as for Exp. 1, Fig. 6.8 illustrates both point clouds to be registered (source and target), as well as both evaluation cases, i.e., same-source and cross-source.

FPFH generated 1000 - 30000 putative correspondences depending on the number of points present. For this reason, the data was downsampled for the TEASER++ algorithm. The number of semantic column center points only increased by less than double the amount of Exp. 1, as these depend on the number of columns present. The steep increase in computational time aligns with the work by B. Yang et al. (2016), which stated their algorithm does not scale well with an increased number of correspondences.

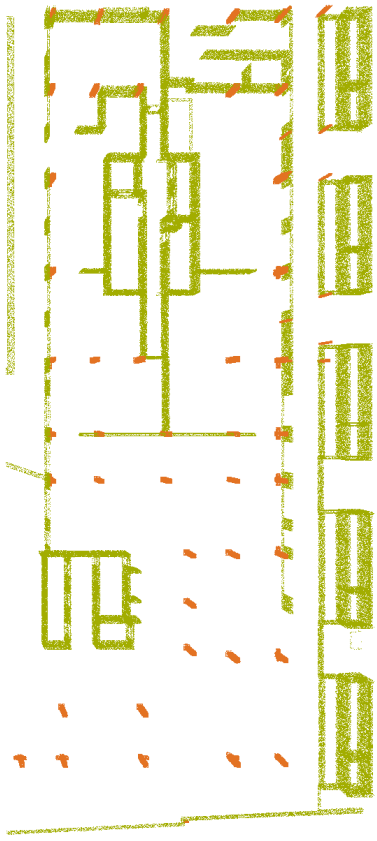
In Exp. 2, **Case 1**, as presented in Table 6.3, semantic TEASER++ achieves the lowest error rate (TE, RE and RMSE) of all the algorithms tested. Additionally, a fitness value of 1.0 was achieved, indicating that each source point is within the selected threshold distance from a corresponding target point.

TEASER++ with FPFH features is in second place for Exp. 2 for the same-source case. Here, too, the translation error and rotation error are close to zero.

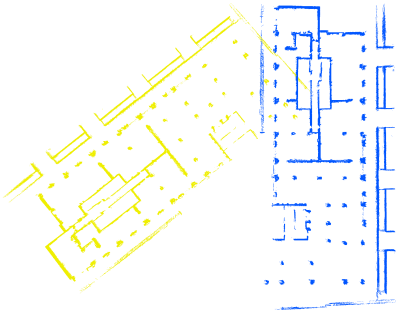
In this experiment (and contrary to Exp. 1), semantic correspondences outperformed FPFH features. This improvement is likely due to fewer correspondences, which facilitated faster convergence to the correct solution. TEASER++ terminates after a specified



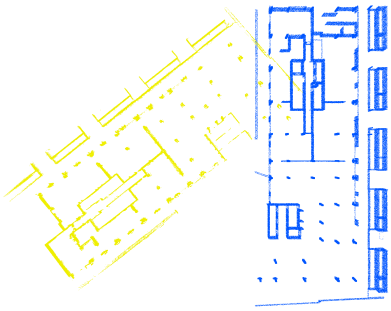
(a) Preprocessed semantically enriched SLAM point cloud (source).



(b) semantically enriched BIM point cloud (target).



(c) Exemplary Initial Situation Case 1 (same-source).



(d) Exemplary Initial Situation Case 2 (cross-source).

Figure 6.8: Setup for Exp. 2.

number of iterations or when predefined error bounds are met, contributing to this enhanced performance.

As for **Case 2**, Method 1 shows steady performance for the same-source and cross-source cases, allowing for a subsequent ICP algorithm to further enhance the registration in the cross-source case.

Table 6.3: Results of Exp. 2.

Case	Method	TE [m]↓	RE [°]↓	Fitness↑	RMSE↓	SR [%]↑
	RANSAC (FPFH)	9.10	54.11	0.77	1.48	3
No. 1	TEASER (FPFH)	**0.01**	0.01	**1.00**	0.11	**100**
Same-	**TEASER (Sem)**	**0.01**	**0.00**	**1.00**	**0.01**	**100**
Source	Ours (Method 1)	0.29	6.76	0.97	0.42	73
	Ours (Method 2)	10.46	45.46	0.83	0.92	30
	RANSAC (FPFH)	14.25	53.00	0.64	1.52	0
No. 2	TEASER (FPFH)	16.79	28.11	0.48	1.53	9
Cross-	TEASER (Sem)	25.66	47.97	0.48	1.53	0
Source	**Ours (Method 1)**	**0.89**	**13.06**	**0.97**	**0.52**	**72**
	Ours (Method 2)	14.17	173.26	0.82	1.29	0

Considering the SR, TEASER with FPFH features stays in second place. This low SR result is attributed to the varying distances between semantic landmarks, as the column center is influenced by the scan density at each column. In some cases, one face of the column has high resolution, while the opposite face is sparsely scanned. This discrepancy causes a significant shift in the column's center, resulting in inconsistent measurement distances across correspondences. Consequently, the registration process is impaired, and the TEASER++ algorithm struggles to resolve a solution for rotation and translation.

Within its iteration threshold, RANSAC could not produce a sufficient registration within the ICP range. As the point cloud gets more extensive, obtaining a good solution takes significantly more iterations as it is a random-based method.

Our Method 2 failed to correctly register the point clouds due to several factors. First, the parameters require precise fine-tuning. The angle threshold for distances, as illustrated in 6.6b, is crucial, particularly as the point cloud becomes larger, given its sensitivity. Similarly, accurate binning of the histogram is essential. As more columns are included, the number of all-to-all correspondences increases significantly, necessitating a more precise binning threshold. Notably, the TE approximates 180°, suggesting that the registration was misoriented by 180°. This issue likely arises from the limited number of distinct columns and the symmetry within the portion of the building represented in the point cloud.

The initial angle alignment calculation is also very computationally intensive, as it increases linearly with the number of point normals and the histogram bins for the angle.

As the point cloud normals are calculated and counted within the histogram, a smaller bin width results in a very sharp subdivision and, thus, a more accurate initial alignment.

The discrepancy in rotational errors between Methods 1 and 2 is significant despite both methods using the same algorithm for rotation determination. This difference can be attributed to the fact that a high fitness value was achieved for one of the rotation candidates during the translation determination process. Although the rotation angle itself was incorrect (usually by $\pm$ 180°), the high fitness value led to this rotation candidate being erroneously selected as the optimal solution.

Experiment 3: Room to entire BIM model

Regarding Exp. 3, the SLAM scan covered approximately three rooms, including hallways and a larger room. This data was then aligned with the BIM model of the entire floor, encompassing a space approximately three times the size of the SLAM point cloud. The experiment aimed to simulate a scenario where only a portion of the construction site is scanned rather than the entire site.

In contrast to Exp. 1 and 2, in Exp. 3, the **Case 1** utilizes the BIM point cloud rather than the SLAM point cloud to evaluate the algorithms within the same source case. For this purpose, the BIM point cloud was cropped to a smaller section, matching the size of the SLAM point cloud used in Exp. 2. This small section is used as the source and the entire BIM point cloud as the target. Fig. 6.9 illustrates the point clouds used for this experiment.

As shown in Table 6.4, TEASER++, in combination with FPFH, shows the smallest translational and rotational errors for Case 1. The fitness score is the highest and inlier RMSE the lowest of all the examined algorithms. In addition, all runs yielded a successful registration, i.e., within the SR thresholds. Our Method 1 was able to achieve the second lowest translational and rotational error as well as the best inlier RMSE of 0.1088.

In this experiment, TEASER++ demonstrated its efficacy in accurately registering a small subset of the BIM point cloud with the complete BIM point cloud. Since the source and target point clouds were identical within the region of interest, the algorithm effectively generated TRIMs to resolve the rotation and translation, as the majority of FPFH features

(a) Exemplary Initial Situation Case 1 (same-source). Notice that both point clouds come from the original BIM model.

(b) Exemplary Initial Situation Case 2 (cross-source). SLAM point cloud to the left, and entire BIM point cloud in the right

Figure 6.9: Setup for Exp. 3.

Table 6.4: Results of Exp. 3.

Case	Method	TE [m]↓	RE [°]↓	Fitness↑	RMSE↓	SR [%]↑
	RANSAC (FPFH)	33.63	36.32	0.81	1.50	0
No. 1	**TEASER (FPFH)**	**0.03**	**0.03**	**1.00**	0.24	**100**
Same-	TEASER (Sem)	NA	NA	NA	NA	NA
Source	Ours (Method 1)	0.14	0.29	**1.00**	**0.11**	**100**
	Ours (Method 2)	44.66	32.11	0.86	1.35	0
	RANSAC (FPFH)	38.97	46.08	0.50	1.60	0
No. 2	TEASER (FPFH)	40.15	18.97	0.56	1.56	0
Cross-	TEASER (Sem)	NA	NA	NA	NA	NA
Source	**Ours (Method 1)**	**2.50**	**23.92**	**0.98**	**0.57**	**71**
	Ours (Method 2)	57.24	17.56	0.78	1.39	0

were consistent. The same effectiveness was observed with Method 1, where the identical nature of the source and target point clouds allowed for precise feature matching and 100 % SR.

Method 2 found correspondences but not the correct ones, thus resulting in a catastrophic failure. This failure is due to the large number of columns and their similar distribution in different parts of the building.

Case 2 was successfully registered only using Method 1. To achieve this, the grid resolution was increased to 0.8. However, this resolution was not chosen arbitrarily; a detailed discussion on the optimal grid resolution is presented in Subsubsection 6.3.4.

Method 2 frequently assigned a high fitness score to candidates rotated 180° from the correct orientation, leading to entirely incorrect registrations. This failure occurred because the symmetric distribution of columns in the target floor plan and the source point cloud created two peaks in the histogram instead of one, leading to an inaccurate median translation estimate.

Optimal Parameters for Method 1 - Exp. 3

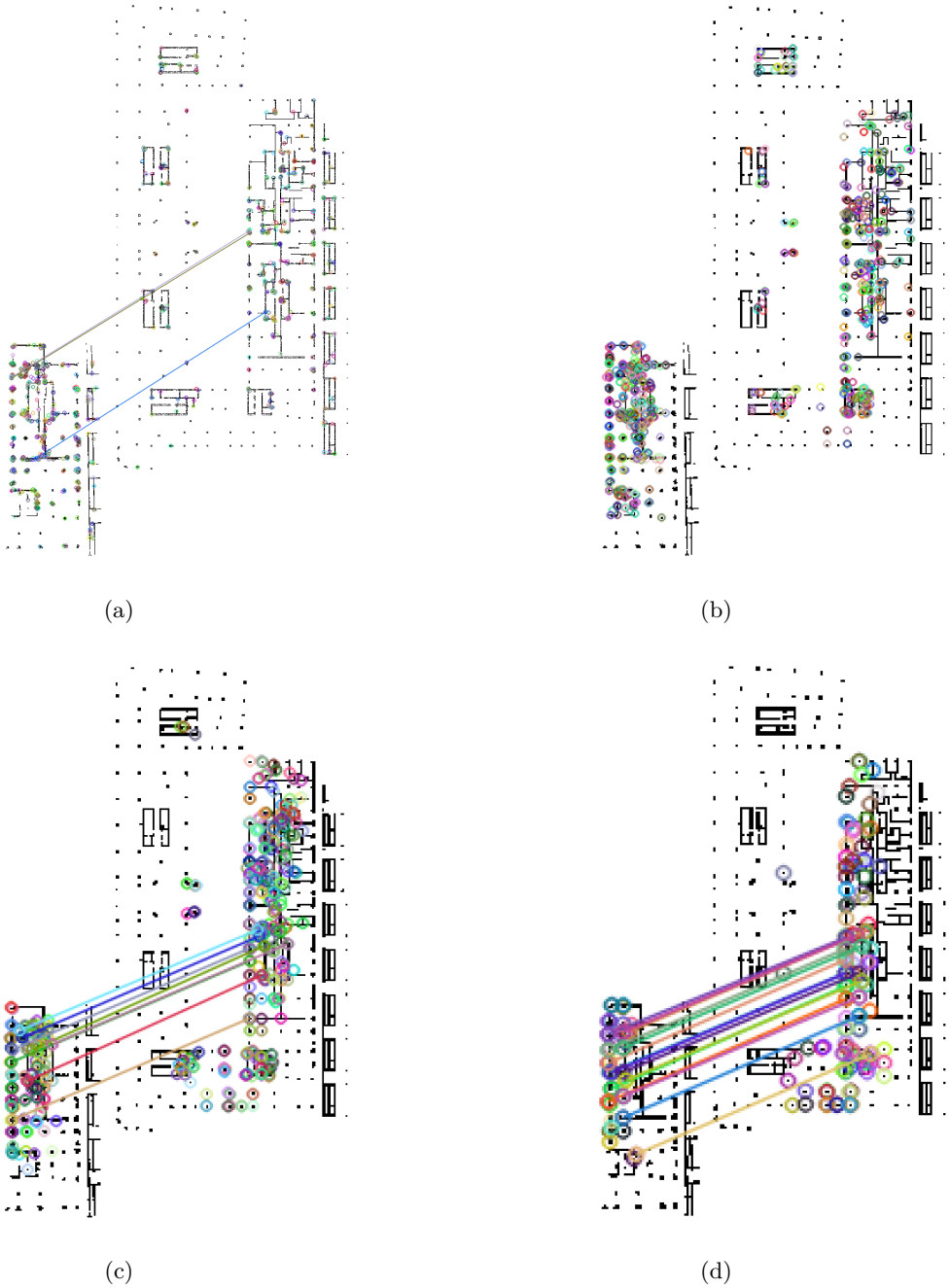
In the proposed Method 1, the grid resolution of the OGM is crucial for the detection of matching FAST features. As the results presented in Table 6.5 show, the optimal grid resolution lies around 0.8 for the setup of Exp. 3.

Table 6.5: Results of Method 1 with different grid resolutions.

Grid Resolution	**TE [m]↓**	**RE [°]↓**	**Fitness [-]↑**	**RMSE [m]↓**	**Time [s]↓**
0.1	30.74	0.60	<u>0.99</u>	1.37	7.86
0.2	15.84	0.60	0.97	1.01	6.88
0.3	14.17	0.60	0.96	1.07	6.98
0.4	8.25	0.60	0.97	1.05	6.80
0.5	19.66	0.60	0.97	0.95	6.85
0.6	1.52	0.60	**1.00**	0.88	6.92
0.7	<u>0.34</u>	0.60	**1.00**	<u>0.28</u>	6.79
0.8	**0.08**	0.60	**1.00**	**0.21**	**6.56**
0.9	0.41	0.60	**1.00**	0.32	<u>6.70</u>
1	0.80	0.60	**1.00**	0.58	6.77

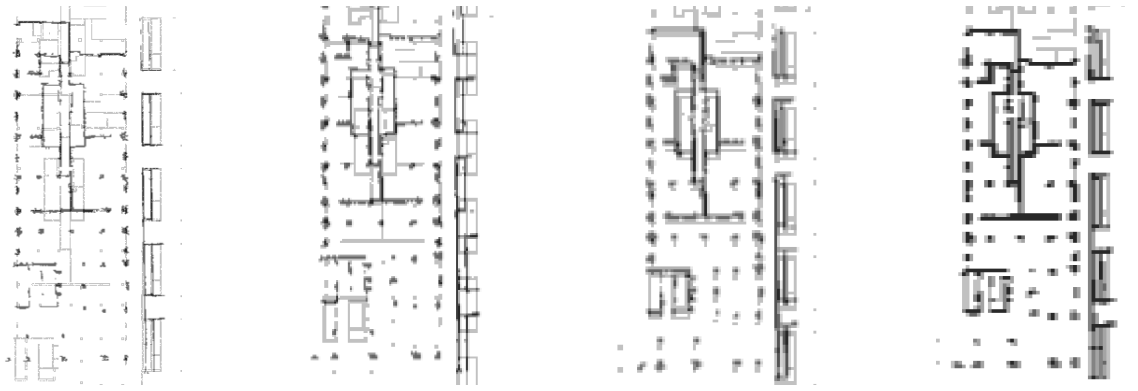
As shown in Figures 6.10 and 6.11, a smaller grid resolution results in smaller distinct squares to be occupied by the algorithm that creates the OGM. For higher resolutions, small, sparse elements will be merged as one. Depending on the size of the point cloud, a different grid resolution is needed.

With an optimal grid resolution, the registration process is notably robust. However, the accuracy of the initial angle calculation—described in Step 2 of Subsection 6.3.2—is critical. If the rotational angle is sufficiently accurate (i.e., if at least one of the four candidates is within 3 degrees of the true angle), then both the translation and the overall registration accuracy will be significantly improved.



(a) (b) (c) (d)

Figure 6.10: Feature matching as a function of grid resolution. Grid resolutions shown are: (a) 0.2, (b) 0.4, (c) 0.6, and (d) 0.8. Subfigures (a) through (d) demonstrate that as grid resolution increases, image resolution decreases due to point merging, leading to a higher similarity between FAST features and BRIEF descriptors.



(a) (b) (c) (d)

Figure 6.11: Registration accuracy as a function of grid resolution. Subfigures (a) through (d) illustrate the registration results for various cases using different grid resolutions, corresponding to those shown in Fig. 6.10. Accuracy improves with increasing grid resolution. As grid resolution increases, walls become more uniform, with fewer holes and greater similarity, allowing the ORB algorithm to more effectively match corresponding features as finer details are merged.

Based on extensive experimentation, it can be concluded that Method 1 provides the best performance for cross-source registration when employing the optimal grid resolution. However, it is important to note that this method has not achieved a 100% SR, meaning that it can still fail in approximately 20% of cases, as demonstrated by the three documented experiments.

Additionally, for registration involving the same-source point cloud, TEASER++ remains a highly effective choice, particularly when using FPFH features. Relying solely on semantic column centers may lead to significant failures, as evidenced by the results from Exp. 3.

With the low-drifted SLAM point cloud now properly aligned with the reference map, the next objective is to further refine the sensor poses. This refinement ensures that the permanent elements of the newly acquired map accurately correspond to those in the reference map, ultimately leading to the creation of a correct, up-to-date, and well-aligned map representation. In the upcoming section, the optimization process of adjusting sensor poses using either a 3D BIM model or a semantic reference map will be elaborated.

6.4 BIMCaP: BIM-based AI-supported LiDAR-Camera Pose Refinement[4]

A framework designed to align a sequence of synchronized LiDAR scans and RGB images with a 3D BIM model is proposed, thereby refining the initial approximated camera poses, which inherently suffer from drift owing to the characteristics of SLAM algorithms. The proposed framework can be divided into three significant steps.

Step 1. The initial step of the proposed methodology involves fusing camera images and sparse LiDAR scans in precise depth maps. This process is facilitated through a hybrid approach employing interpolation and a Deep Learning (DL) technique, which then allows the projection of the pixel information (such as semantic information) into the 3D space.

Step 2. Subsequently, in the second step, semantic segmentation is applied to the images, enabling the detection of permanent elements such as walls, columns, and floors within the reconstructed 3D map. Simultaneously, a point cloud and a vectorized floor plan with semantic information are created from the BIM model. This vectorized semantic floor plan will be used as a reference map for the alignment of the real-world data.

Step 3. In the third step, a statistical approach is employed to generate initial synthetic camera poses. These poses are then refined through a BA module, which integrates custom cost functions. These functions are designed to iteratively enhance the accuracy of sensor poses, thereby ensuring optimal alignment between the generated map and the semantically vectorized floor plan from the BIM model. This refinement process selectively considers only permanent elements, which are identified through semantic segmentation in real-world images and projected into three-dimensional space using the previously estimated depth maps.

Fig. 6.12 illustrates the proposed semantic-aware pose optimization framework.

[4]The methods described in this section were developed in collaboration with Anna Ribic as part of her interdisciplinary project, and Shaowen Qi as part of his master thesis; both supervised by the author of this dissertation. Portions of this section were previously published in (M. A. Vega-Torres, Ribic, et al., 2024)

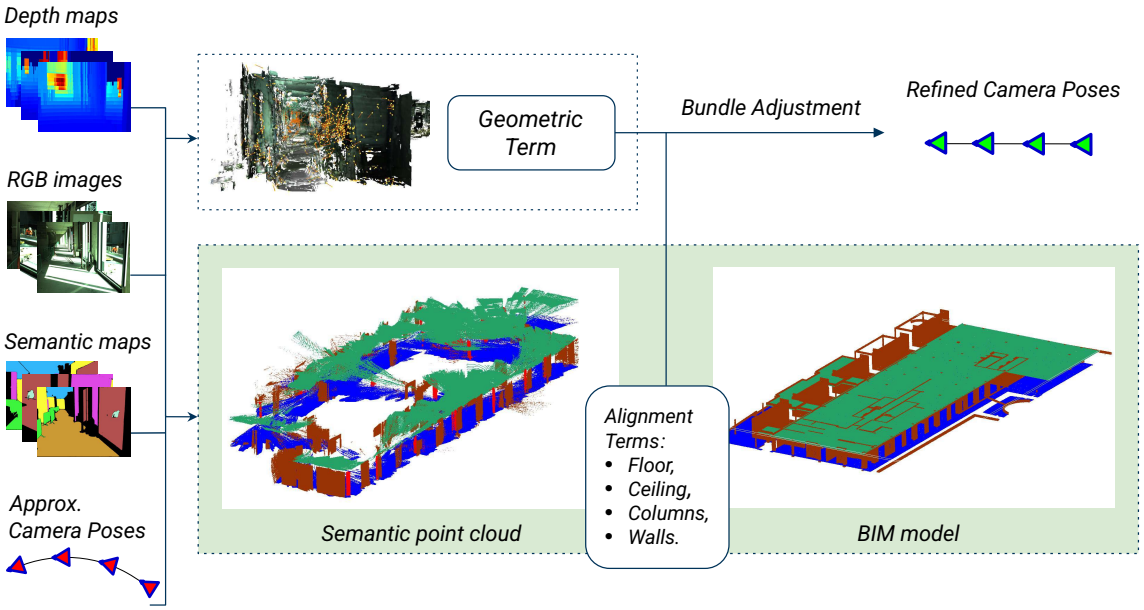


Figure 6.12: Overview of the proposed BIMCaP framework for sensor pose refinement. The depth maps are used to project the semantic maps (created from the images) into the 3D space using the approximated initial poses (drifted due to SLAM). Different terms aim to correlate the data measured over permanent elements (i.e., reliable landmarks) with the BIM model. Moreover, a geometric term ensures geometric consistency among real-world images.

6.4.1 Step 1: LiDAR and Camera Fusion

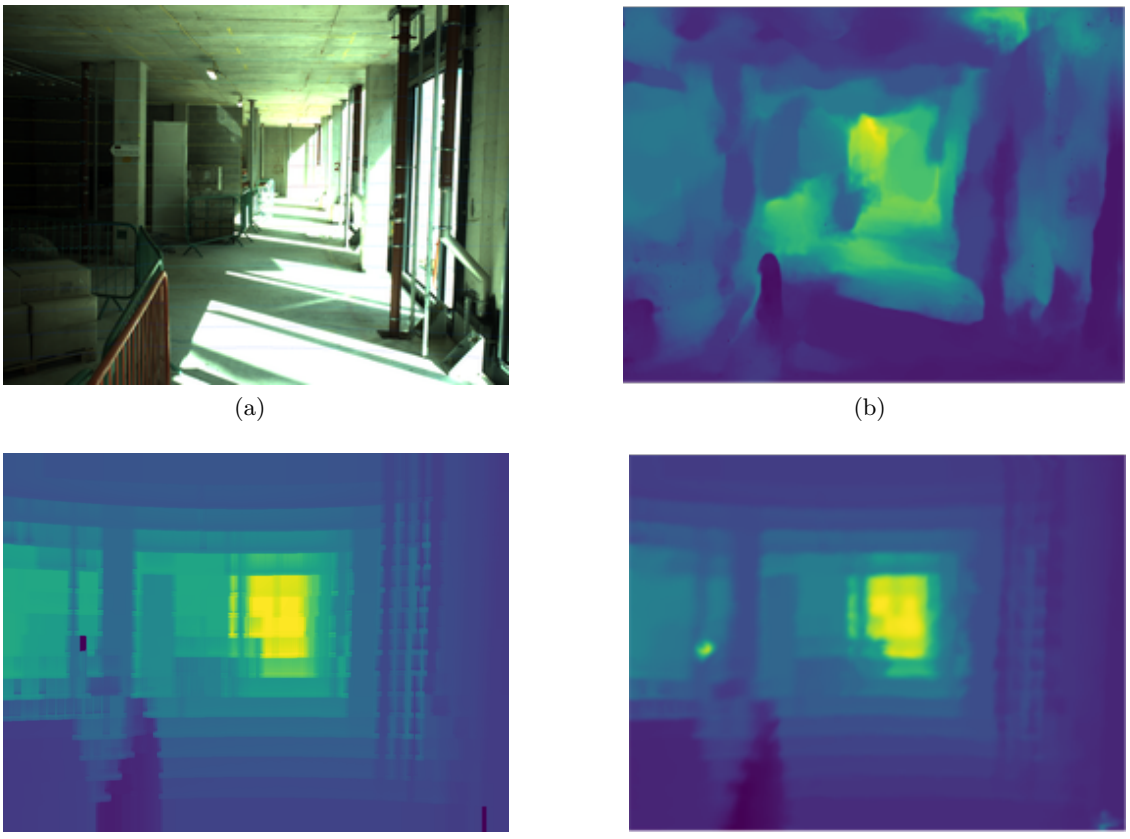
To fuse the information from the LiDAR and the camera, the visible point cloud within the camera's FoV is first projected onto the image plane. The goal is then to generate a dense map that aligns coherently with both the image and LiDAR data.

The projection of LiDAR points onto the camera image is carried out using the camera's intrinsic and extrinsic parameters, as well as the package provided by Trzeciak et al. (2023a). This package ensures that the camera image is undistorted and that only the corresponding LiDAR points, which are timestamp-synchronized with the small FoV of the camera, are projected from the 3D space onto the 2D image. As a result, depth information is obtained for several pixels within the image. However, this depth information remains sparse due to the use of a 360° LiDAR with only 16 rays in the vertical direction.

It is important to note that to ensure proper functionality with sensors having a reduced FoV (such as solid-state LiDARs or RGB-D cameras), only the LiDAR data within the camera's FoV is utilized.

The sparsity of the point cloud would not be sufficient to leverage all the information from the image in the 3D space; therefore, subsequently, the goal is to create a dense depth map using the point cloud and the corresponding camera image.

Currently, numerous DL methods serve for depth estimation, yet many of them are optimized for outdoor environments (such as the KITTI dataset). Therefore, their accuracy tends to decline in indoor settings, which constitutes the focus of this investigation. Following extensive experimentation with various methodologies, a hybrid method was adopted that combines linear interpolation with CompletionFormer (CF) (Y. Zhang et al., 2023). Fig. 6.13 illustrates the results of this hybrid approach, showing the original CF output alongside the refined outcome involving an initial linear interpolation.



(a) (b) (c) (d)

Figure 6.13: Depth completion with sparse LiDAR point cloud: (a) original image from the ConSLAM dataset with original sparse projected LiDAR scan; (b) depth map using only CompletionFormer; (c) depth map using only linear interpolation and (d) using linear interpolation and CompletionFormer. It is evident that (d) yields the best results since it is smother than (c) and more coherent with the measurements than (b).

6.4.2 Step 2: Semantically Enriched Maps

In this step, the goal is to create maps that will allow the sensor pose correction in the subsequent step. This step is divided into two sub-steps: Firstly, a reference semantic vectorized floor plan is created from the BIM model, and secondly, the 3D map created with real-world data is enriched with semantic information. This semantic enrichment serves a pivotal role in distinguishing permanent elements within real-world data, such as walls, columns, and floors, which can be reliably aligned with the BIM model.

Reference Map

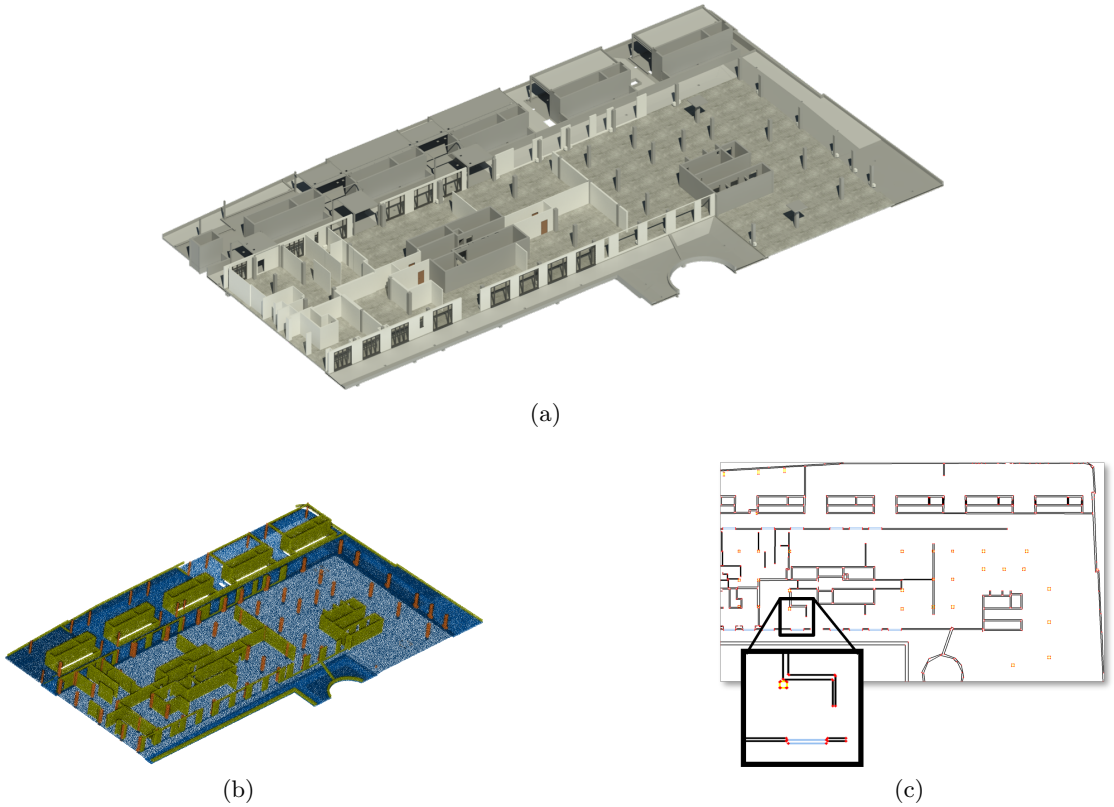
To prepare for implementing the pose correction module, the 3D BIM model's geometry is simplified into a 2D semantic vectorized floor plan. Since walls and columns are perpendicular to the XY plane, this reduction not only retains all vertical structural element information but also allows efficient pose optimization in subsequent stages.

To generate the 2D semantic vectorized floor plan, the BIM model undergoes conversion from IFC format to a 3D semantic point cloud as explained in section 6.3.1[5]. An illustration of such a point cloud can be observed in Fig. 6.14b.

The created synthetic 3D point cloud is projected vertically into 2D images within a specified height range, typically within $\pm$ 20 cm from the floor level. Semantic labels are utilized to filter each element in the point cloud. Subsequently, image processing methods such as contour and line detection are employed to identify line segments representing individual elements in the 2D projection. These detected lines are then consolidated, including their start and end points, to form the vectorized semantic floor plan. A resulting floor plan is depicted in Fig. 6.14c.

In previous contributions, such as those in Sections 4.3 and 5.3.1, it was demonstrated the feasibility of generating vectorized floor plans directly from BIM models using IfcConvert. Here, the purpose is to develop a more general method that enables the generation of semantic vectorized floor plans from semantically enriched point clouds, thereby eliminating the limitation of relying solely on semantic BIM models as required by IfcConvert.

[5]The code to convert BIM models into semantically enriched point clouds can be found here: Link



(a)

(b)

(c)

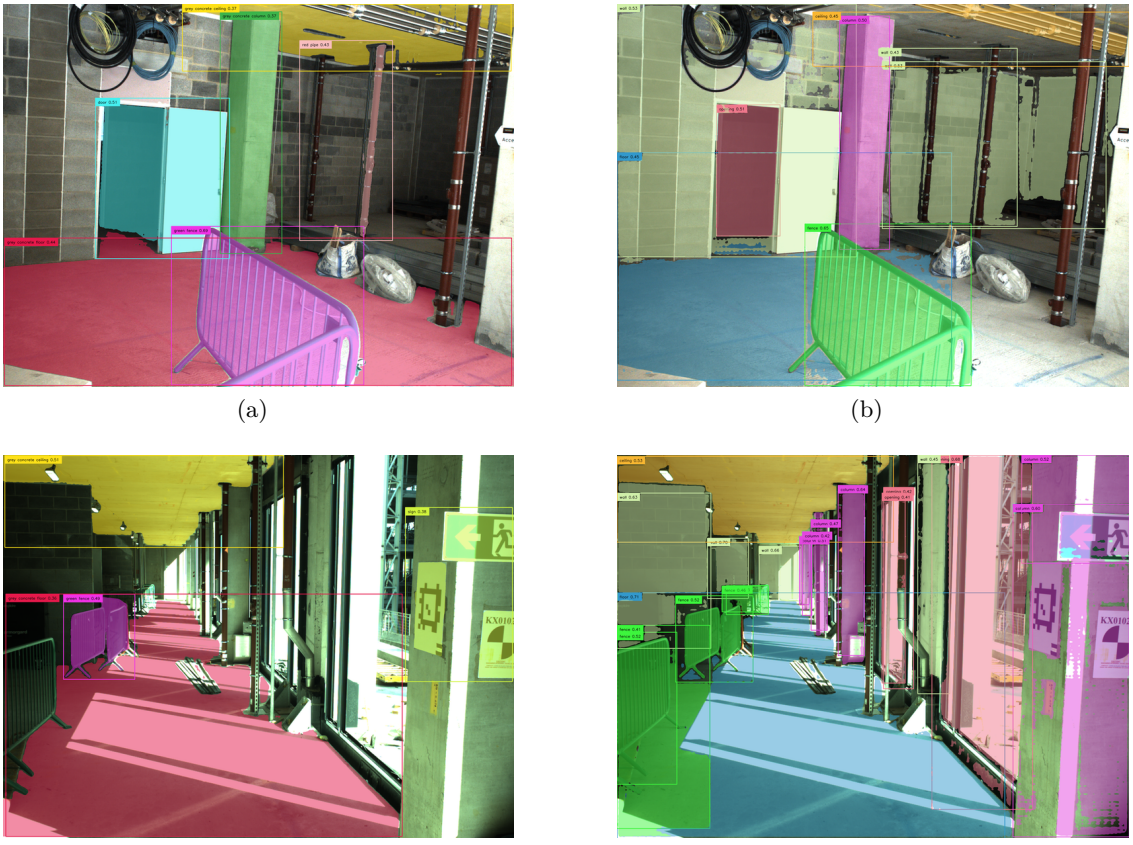
Figure 6.14: Reference map preparation: (a) original 3D BIM model (without ceiling); (b) uniformly sampled 3D point cloud with semantic information from the BIM; and (c) vectorized semantic floor plan, from which the walls and columns (in black and yellow) are used for pose refinement in the subsequent pose optimization step.

Semantic Segmentation of Real-world Data

To filter permanent elements that are possible to be matched from the real-world data with the BIM model, SOTA image semantic segmentation algorithms are leveraged. More specifically, a modified version of Grounding-DINO (S. Liu et al., 2023) is used. However, for the object detection task, the DINO algorithm (F. Li et al., 2022) was replaced with a tiny version of the RTMDet algorithm (Lyu et al., 2022) pre-trained with the Common Objects in Context dataset (COCO) dataset and 250 labeled images of the ConSLAM dataset, which contains custom classes typical of a construction site. These images were labeled semi-automatically using the Computer Vision Annotation Tool (CVAT) (CVAT.ai Corporation, 2023).

Thus, the proposed approach enables the detection of objects of interest, expanding beyond the foreground elements identified by the original Grounding DINO version. Fig. 6.15 illustrates the results of the semantic enrichment before and after the proposed en-

hancement, and Fig. 6.16a shows the top view of the resulting semantically enriched 3D point cloud after projecting the semantic labels to the 3D space with the previously generated depth maps. In this last figure, it is also visible that it is possible now to filter walls, columns, floor, and ceiling points in the depth maps, which can reliably be used for registration with the BIM model and, therefore, for camera pose optimization. It is worth mentioning that the floor and ceiling predicted labels were also used to optimize the depth maps, creating smoother surfaces in these regions with blurring operations in the 2D depth maps.



(a) (b)

(c) (d)

Figure 6.15: Semantic segmentation over 2D images of the ConSLAM dataset: (a) inference with the original Grounding DINO algorithm (b) inference result after replacing DINO with pre-trained RTMDet for object detection. (b) comprehends predicted labels for the walls in the background, which are critical for the proposed camera pose refinement framework. Similarly, (c) and (d) are, respectively, the results of Grounding DINO and the results of the proposed pipeline.

6.4.3 Step 3: Sensor Pose Calculation and Refinement

The initial approximations of sensor poses are ideally determined using a Visual-SLAM framework. However, the experimentation with cutting-edge SLAM algorithms such as

DROID-SLAM or Go-SLAM yielded unsatisfactory results when applied to the ConSLAM dataset, functioning correctly only for limited segments of the trajectory. Despite these limitations, advancements in odometry systems suggest that addressing this challenge will become feasible in the future. Therefore, and since the goal is to refine slightly drifted poses and experiment under different magnitudes of drift, a synthetic trajectory was created instead, simulating the output of a SLAM framework. This process is explained in the following subsection. Subsequently, the method that is used to improve the accuracy of these poses is introduced.

Synthetic Pose Calculation.

To ensure alignment with the typical trajectory patterns observed in existing SLAM systems and to provide the flexibility needed to assess the stability of the method under varying initial positions, synthetic trajectories were carefully engineered. The creation of these trajectories focused on replicating the gradual drift characteristic of SLAM-generated trajectories, where errors at each pose incrementally increase over time.

Accordingly, the translation offset from the original sensor pose ΔT_{i+1} was modeled as a normally distributed random variable with mean Δt_i and variance σ_t^2. Formally, $\Delta T_{i+1} \sim \mathcal{N}\left(\Delta t_i, \sigma_t^2\right)$ with $\Delta T_1 \sim \mathcal{N}\left(0, \sigma_t^2\right)$ where Δt_i is the previously sampled offset value, and the variance σ_t^2 is an adjustable hyper-parameter which would determine the offset of the trajectory from the ground truth poses. Regarding the camera rotation, degree offsets $\Delta\phi \sim \mathcal{N}\left(0, \sigma_p^2\right), \Delta\theta \sim \mathcal{N}\left(0, \sigma_{th}^2\right)$ were randomly sampled around pitch and yaw directions. Fig. 6.16b presents the resulting map using a synthetically drifted trajectory.

Pose Optimization.

Inspired by the FACaP framework (Sokolova et al., 2022), several terms are incorporated into the cost function to refine the sensor pose with the BIM model using BA. To construct a consistent 3D map from real-world sequential images, a geometric term is employed to capture the divergence between 3D point estimations from two distinct viewpoints. This geometric term integrates photogrammetric constraints; in this case, COLMAP was utilized to obtain features and correspondences among sequential images. Fig. 6.16b and 6.16c visualize some of these features.

Additionally, a floor term is utilized to ensure that segmented points corresponding to the floor are confined to a single plane and are aligned with the floor surface defined in the BIM model. Similarly, a ceiling term is introduced to incorporate information from the model's ceiling for optimization purposes. Wall and column elements are also included, as they are crucial for correcting rotations around the vertical axis (yaw variations) and horizontal translations. These adjustments are vital for the precise alignment of the reconstructed environment, thereby ensuring its geometric consistency.

The following equation contains all the main terms:

$$\begin{aligned} L \quad = \quad & \sum_{(p,p')\in M} \left\| p - p' \right\|_2 \quad + \quad \lambda_{\mathrm{F}} \sum_{p\in P_F} \mathrm{dist}\left(p, \pi_F\right) \quad + \quad \lambda_{\mathrm{Ce}} \sum_{p\in P_{Ce}} \mathrm{dist}\left(p, \pi_{Ce}\right) \\ & + \quad \lambda_{\mathrm{W}} \sum_{p\in P_W} \mathrm{dist}\left(p, \pi^f(q(p))\right) \quad + \quad \lambda_{\mathrm{Co}} \sum_{p\in P_{Co}} \mathrm{dist}\left(p, \pi^f(q(p))\right), \end{aligned}$$

where (p, p') denotes a pair of 3D back projected key points grouped in M; π and P represent planes and points; λ is the weight given to each lost function; and the sub-indices F, W, Co, and Ce represent floor, walls, columns, and ceiling; $q(p)$ is the nearest floor plan point; and the distance to the corresponding wall or column line in the vectorized floor plan is denoted as $\pi^f(q(p))$.

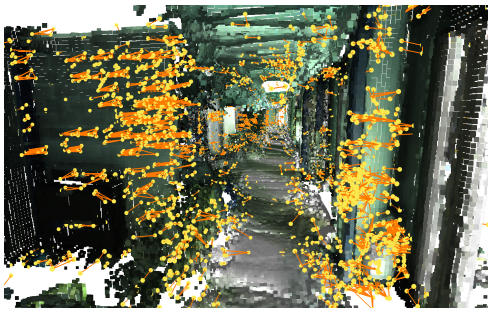
6.4.4 Experiments and Results

Dataset and Evaluation Details

To ensure reproducibility and enable benchmarking, the developed method was tested on the ConSLAM dataset (Trzeciak et al., 2023a, 2023b). ConSLAM represents a pioneering effort, offering the first open-access dataset acquired in an indoor cluttered construction site. This dataset encompasses sequences of RGB and LiDAR data together with Terrestrial Laser Scanner (TLS) point clouds. The latter was leveraged as a resource for the generation of a BIM model with centimeter-level accuracy. The GT poses of ConSLAM were calculated using SLAM2REF (M. A. Vega-Torres, Braun, & Borrmann, 2024c), an enhanced version of BIM-SLAM (M. A. Vega-Torres et al., 2023) and OGM2PGBM (M. A. Vega-Torres, Braun, & Borrmann, 2022) for large-scale maps, which is robust to LiDAR

(a)

(b)



(c)

Figure 6.16: Features used for optimization (a) Top view semantic segmented map generated with the ground truth poses and the segmentation results of walls (in red) and floor (in blue) as explained in Section 6.4.2; (b) map created with synthetic poses of Exp. 1 (obtained as explained in Section 6.4.3), here the COLMAP features are visible; (c) view from an indoor observer's perspective of the point cloud with highlighted Scale-Invariant Feature Transform (SIFT) features used in the geometric term for optimization.

motion distortion and Scan-Map deviations. These data are also open-access (M. A. Vega-Torres, Braun, & Borrmann, 2024b).

To quantify the quality of the whole trajectory before and after pose optimization, the standardized Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE) in position (also referred to as translation) and in rotation were used, defined as in (Z. Zhang & Scaramuzza, 2018) as follows:

$$\mathrm{ATE}_{\mathrm{pos}} = \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\Delta \mathbf{p}_i\|^2 \right)^{\frac{1}{2}},$$

$$\mathrm{ATE}_{\mathrm{rot}} = \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\angle (\Delta \mathrm{R}_i)\|^2 \right)^{\frac{1}{2}}.$$

where

$$\Delta \mathrm{R}_i = \mathrm{R}_i \left(\hat{\mathrm{R}}'_i\right)^\top,$$

$$\Delta \mathbf{p}_i = \mathbf{p}_i - \Delta \mathrm{R}_i \hat{\mathbf{p}}'_i$$

and $\angle(\cdot)$ means converting the rotation matrix to an angle axis representation and using the rotation angle as the error. Moreover, for better comparison, some of the metrics introduced in (Sokolova et al., 2022) including the Map Mean Entropy (MME), Mean Plane Variance (MPV), and the Nearest Neighbor Distance (NND) were incorporated.

The MME serves to assess the quality of 3D maps, with a higher MME signifying favorable alignment between the input cloud and the reference map. The MPV evaluates the variance among planes within the map, with lower MPV values indicating more uniform and well-defined surfaces. The NND quantifies the average distance between adjacent points in the point cloud, with smaller NND values indicating denser point clouds. The MME value of 0.761, calculated using the GT poses (as shown in 6.6), represents the optimal alignment between the real-world point cloud and the BIM model. For understanding, this value would be zero if no deviations between the actual environment and the model (Scan-Map deviations) exist.

Pose Refinement Results

The results of the proposed framework are compared against the SOTA FACaP pipeline (Sokolova et al., 2022) and evaluated meticulously with three different experiments. The first experiment consists of a synthetic trajectory that has an offset of around 1.4 meters in translation and 10 degrees in rotation (Exp. 1), the second one has an offset of only 30.3 cm in translation and 8.82 deg in rotation (Exp. 2), and the third one only has rotation offset of 9.6 degrees (Exp. 3).

Table 6.6 shows initial metrics based on ground truth and synthetic poses for Exp. 1, along with results after pose optimization using various terms of the FACaP pipeline and the proposed BIMCaP framework. Fig. 6.17 illustrates the evaluation of the error while optimizing Exp. 3.

The findings from Exp. 1 (Tab. 6.6) emphasize the efficacy of utilizing all terms for optimizing translational errors. However, this approach may not consistently yield optimal results when addressing rotational errors. Notably, while BIMCaP demonstrates a superior

Table 6.6: Comparison of validation measurements for Exp. 1 using the different methods. All values are in meters except for the rotational Absolute Trajectory Error (ATE), which is given in degrees. The best overall results are highlighted in bold, while the best results per method are underlined. G, F, W, Co, and Ce stand for the geometric, floor, wall, column, and ceiling terms, respectively.

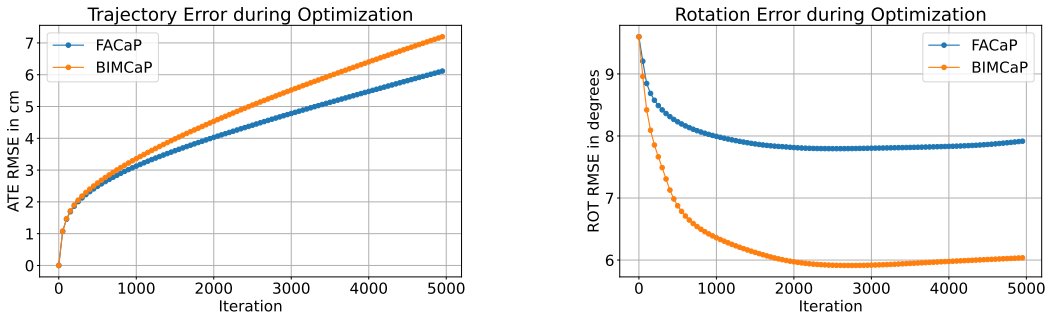
Source	**G**	**F**	**W**	**Co**	**Ce**	**MME↓**	**MPV↓**	**NND↓**	**ATE_{pos}↓**	**ATE_{rot}↓**
GT poses	-	-	-	-	-	0.761	0.040	0	0	0
Exp. 1	-	-	-	-	-	1.027	0.059	0.557	1.391	9.99
FACaP	✓	✓	✓	-	-	0.979	0.054	0.503	1.321	15.40
	✓	-	-	-	-	1.013	0.058	0.566	1.385	**8.84**
	-	✓	-	-	-	0.966	0.053	0.503	1.358	16.50
	-	-	✓	-	-	1.031	0.059	0.545	1.378	9.82
BIMCaP	✓	✓	✓	✓	✓	**0.956**	**0.052**	0.460	**1.281**	11.84
	✓	✓	✓	-	✓	0.959	0.052	**0.456**	**1.281**	11.81
	✓	✓	✓	-	-	0.975	0.053	0.505	1.311	12.58
	-	✓	-	-	-	0.966	0.054	0.519	1.351	13.63
	-	-	✓	✓	-	1.034	0.059	0.549	1.378	9.75
	-	-	-	-	✓	0.982	0.054	0.480	1.387	12.71

reduction in translational error by 4 cm compared to FACaP, both methodologies become trapped in a local minimum, impeding the accurate optimization of the poses. This issue can be attributed to the substantial difference between the synthetic poses and the ground truth.

Table 6.7: Pose optimization results for Exp. 2 and 3: given a small translational and rotational offset. In addition to the ATE_{pos} and ATE_{rot}, the RMSE for the Yaw, Pitch, and Roll axes separately in degrees is provided.

Source/Method	**ATE_{pos} (cm)↓**	**ATE_{rot} (deg)↓**	**Yaw↓**	**Pitch↓**	**Roll↓**
Exp. 2 before optim.	30.3	8.82	4.31	4.31	0.29
FACaP	**30.2**	7.70	3.79	3.79	**0.21**
BIMCaP	30.4	**5.61**	**2.73**	**2.75**	0.23
Exp. 3 before optim.	0	9.60	4.70	4.72	0.29
FACaP	**6.2**	7.92	3.91	3.90	**0.19**
BIMCaP	7.2	**6.04**	**2.96**	**2.96**	0.22

Exp. 2 and 3 results (Table 6.7 and Fig. 6.17) indicate superior performance of both methods in optimizing rotational errors over translational errors. Notably, BIMCaP significantly enhances yaw and pitch angles during the pose optimization process. Fig. 6.18 illustrates how BIMCaP aligns the floor and ceiling points to the correct planes, contrary to FACaP, which tries to fit a plane among the given measurements without any reference.



(a) (b)

Figure 6.17: Development of the translational (a) and rotational error (b) given only a rotational offset as described for Exp. 3.

Exp. 3 exposes a limitation in the proposed approach, as optimizing trajectory with only rotational offsets resulted in unintended translations. This could be due to the simultaneous optimization of both translation and rotation, causing discrepancies. Additionally, the challenge of accurately calculating sensor poses is intensified by the reduced FoV and the sparse ground truth poses.

(a) (b)

(c) (d)

Figure 6.18: Side views of the different maps. (a) ground truth map; (b) map created with synthetic poses of Exp. 1; (c) map after FACaP optimization and (d) after BIMCaP optimization. The BIMCaP result shows better alignment with the real floor and ceiling planes.

6.5 Contributions and Limitations

6.5.1 Contributions

This chapter presented significant contributions to the field of cross-source point cloud global registration and sensor pose refinement with a particular focus on the integration of 3D LiDAR and camera measurements and 3D BIM models. The key contributions are as follows:

C 3.1 Cross-source global registration for aligning SLAM-reconstructed point clouds with BIM models (RQ 3.1):

- A method was provided to convert a BIM model into a semantically enriched point cloud, which is then as a target point cloud for registration.[6]. As well as a source of information for automatic real-world point cloud labeling.
- Rapid and accurate estimation of primary wall angles is achieved using the principal normal direction count method. This technique allows for precise initial angle computation for point cloud alignment based on the structure's main axes.
- The research demonstrated the feasibility of using OGMs as a viable method for cross-source point cloud registration using ORBs 2D features, providing insights into their strengths and weaknesses and achieving over 70% SR in all the conducted cross-source and same-source registration experiments.
- Comprehensive analysis is provided to expose the limitations and advantages of existing SOTA algorithms (such as TEASER++ and RANSAC as well as a proposed method based on column centers) in same-source and cross-source point cloud registration.

C 3.2 Semantic enrichment of 3D maps in real-world construction sites with LiDAR-camera fusion (RQ 3.2):

- A combination of linear interpolation and a neural approach for depth completion tasks is proposed, which enables the retrieval of a smooth depth map

[6]The code to convert BIM models into semantically enriched point clouds can be found here: Link

from a very sparse LiDAR scan (16 rays in the vertical direction) and an RGB image.

- A new methodology for semantic segmentation of images of indoor construction sites was developed. In particular, the method is not only able to detect objects present in the construction environment but also detect not only elements from the foreground but also the background of the image.
- The object detection method was made open-source (Link to pre-trained RTMDet model), as well as the labels of the 250 very unique images of the open-access ConSLAM dataset. Link to the data.[7].[8].

C 3.3 Refinement of drifted data with a reference BIM model (RQ 3.3):

- BIMCaP was introduced, an open-source framework that enables alignment and correction of a sequence of camera and reduced FoV LiDAR measurements with a semantic vectorized floor plan, which is automatically created from a BIM model. Link to the repository.
- BIMCaP considers only reliable selected semantic landmarks (such as floor, walls, columns, and ceiling) for drift correction while disregarding other elements (such as clutter, windows, and doors). This filtering process enables a reliable registration that is even robust to large levels of Scan-Map deviations.
- The technique was evaluated using the open-access ConSLAM dataset, ensuring reproducibility and benchmarking by comparing it against a SOTA method.
- BIMCaP was compared against the SOTA FACaP method (Sokolova et al., 2022) in several experiments, demonstrating an improvement of 4 cm in translational error.

[7]Here, there is another dataset with more classes; however, only 88 labeled images. These labels were not used since they have proved to be excessive and imbalanced. Nonetheless, they could be further expanded: Link to dataset

[8]For a comprehensive list of all open contributions, refer to Section 1.14.

6.5.2 Limitations

This chapter demonstrated substantial progress and key contributions; however, it is also crucial to acknowledge the limitations of the proposed methodologies:

L 3.1 Limitations of the global registration method:

- The cross-global registration method demonstrated effective results for initial rough alignment in the conducted experiments. However, it is dependent on the point cloud being enriched with semantic information, which may not always be readily obtainable. Another significant limitation arises if the point cloud contains a high percentage of drift, as the method assumes minimal drift for accurate alignment.
- The method's performance is highly sensitive to the chosen parameters. Incorrect parameter selection can lead to suboptimal results, necessitating a thorough parameter study to identify the optimal values (e.g., grid size) for generalizing the algorithm. Currently, the algorithm's efficacy is constrained by the specific dimensions and parameters used in the presented experiments.
- The method partially relies on the Manhattan-World assumption, which postulates that buildings' primary structures align with orthogonal axes. While this assumption generally holds, exceptions exist. The method can still function if the building's main axis contains a rectangular pattern, even if the facade is of a different geometry, e.g., circular.
- The translation estimate is derived from the median distances of correct correspondences. This approach may lack robustness, particularly in scenarios with few structural columns, significant deviations in column positions, or highly occluded elements. Such situations can undermine the reliability of the translation estimate, affecting the overall accuracy of the registration. A potential approach to addressing the problem of occluded and partially scanned elements could involve neural point cloud completion (such as proposed by Zhou et al. (2022) or by P. Cai et al. (2024)). By completing elements such as walls and columns, registration could be improved by using only their centroids.

L 3.2 BIMCaP limitations:

- While all the main processes of the framework, such as BA for pose refinement and the creation of a vectorized floor plan from a BIM model or a point cloud, are fully automated, they must be executed separately. This requires manual placement of data in a specific order and structure. Integrating these steps into a unified system would streamline the process and reduce manual intervention.
- Despite the proposed BIMCaP method achieving better results than the SOTA, the numerical values remain significantly distant from the ground truth. This discrepancy is likely due to the limited number of keyframes used in the optimization process. Utilizing more keyframes would provide additional features for various terms, including geometric ones, thereby improving convergence to the correct poses. However, the conducted experiments were limited to the number of keyframes for which the ground truth positions were available.
- The dataset used to train the semantic segmentation is relatively small, limiting its generalizability. As a result, the method may not perform well across all construction sites. A larger dataset, comprising samples from diverse construction sites worldwide, would enhance the method's applicability and robustness.

This chapter presented innovative methodologies that advance the field of cross-source global point cloud registration, sensor pose refinement with a BIM model, and construction site image semantic segmentation. Despite certain limitations—such as sensitivity to specific parameters, limited generalizability of semantic segmentation, and a notable disparity between optimized poses and ground truth—the proposed methods represent significant progress in addressing the primary challenge of aligning real-world data, characterized by clutter and small SLAM drift, with a BIM model or reference map. These contributions offer important insights and lay a foundation for further research and development in this domain.

Chapter 7

Conclusions and Further Development

This dissertation addressed the challenging problem of developing comprehensive methods for long-term localization and mapping that leverage LiDAR, IMU, and camera sensors and that utilize reference 3D BIM models or point clouds for alignment and correction of the sensor poses. This dissertation has a particular focus on correcting drifted data acquired in changing and cluttered environments.

This chapter restates the three main research questions (elaborated in Chapter 1) and answers them, thereby providing a comprehensive overview of the research. Simultaneously, it provides a summary of the contributions, underscoring the breadth and depth of the work undertaken in this dissertation.

Subsequently, this chapter presents recommendations for future work that could be investigated to enhance map-based long-term localization and mapping methodologies.

7.1 Conclusions on the Map-based Long-term Localization and Mapping Techniques

This section presents a summarized answer to the main research questions, drawing the main findings and conclusions of this dissertation. At the end of this section, an overall conclusion statement that synthesizes and underscores the importance of this research and its implications is provided.

RQ 1. ***How can 3D BIM models be leveraged for real-time 2D LiDAR and image localization systems?***

In Chapter 4, it was demonstrated the feasibility of generating Occupancy Grid Maps (OGMs) from BIM models and converting these OGMs into Pose Graph-based Maps (PGBMs). This conversion enhances pose tracking accuracy in environments character-

ized by significant deviations between the real world and the reference map (Scan-Map deviations). The proposed method holds substantial promise for mobile robot developers using cost-effective 2D LiDAR sensors and operating in dynamic and changing environments. It facilitates the transition from traditional Particle Filter (PF) algorithms (such as AMCL or GMCL) to advancedGraph-based Localization (GBL) methods (such as Cartographer or SLAM Toolbox), thereby ensuring more precise pose tracking of the mobile platform, even in the presence of clutter or environmental changes. Additionally, a thorough comparison of SOTA algorithms for pose tracking and global localization tasks was provided.

Moreover, a method aimed at achieving rapid camera pose tracking with a reference 3D model was introduced. This method leverages Vanishing Points (VPs) and Vanishing Lines (VLs) extracted from real-world images and BIM-simulated views to correct camera poses effectively. Real-time camera pose tracking can be particularly useful for mixed and virtual reality applications, in which precise alignment between the virtual content and the real world is crucial. An enhanced pose tracking ensures seamless integration of virtual elements with the physical environment, thereby improving user immersion and interaction.

Additionally, accurate camera or LiDAR pose estimation is crucial for mapping using autonomous mobile systems. Knowledge of the sensor's exact position and orientation enables the creation of precise and reliable maps, which are essential for navigating and interacting effectively in dynamic environments.

While the methods presented in Chapter 4 are designed for real-time pose estimation, based on the quantitative results, it is evident that these time constraints compromise the accuracy of the tracked pose. Therefore, in line with the motivations and objectives outlined in Chapter 1, the subsequent chapters prioritize developing a highly accurate, updated, and aligned map, emphasizing precision over real-time performance.

RQ 2. ***How can reference 3D BIM models or point clouds be utilized for alignment and correction of session data from 3D LiDAR and IMU measurements?***

Considering the limitations of real-time systems, Chapter 5 presents major advancements in generating aligned, updated 3D maps of environments using even heavily drifted session data obtained from SLAM or LiDAR-inertial odometry algorithms. The method,

named SLAM2REF, integrates innovative place recognition descriptors and registration algorithms. This comprehensive approach also identifies both positive and negative changes in the environment.

The approach first transforms reference maps into session data, i.e., single sequential LiDAR scans with known positions. Extracted feature descriptors from these scans, specifically the proposed Indoor Scan Context Descriptor (ISCD), which enables the usage of place recognition algorithms to quickly achieve a first alignment of the real-world data and the reference map. Subsequently, the alignment is refined with K-nearest neighbors (KNN) loop detection and a final Iterative Closest Point (ICP) registration with dense target point clouds from the reference map.

SLAM2REF represents a particularly important contribution for SLAM, localization, and pose estimation researchers, enabling them to calculate reliable ground truth poses of self-captured data given a reference map without the need for expensive sensors or specialized indoor setups for accurate pose tracking, as elaborated more in detail in Chapter 1. Moreover, the method could also be leveraged by surveyor practitioners (e.g., in construction sites or emergencies) to align and correct drifted maps with prior reference maps, being able to detect changes and create a comprehensive updated representation of the environment.

However, the proposed method has several limitations, including a comparatively small accepted level of Scan-Map deviations to establish correct correspondences for the initial alignment and constraints related to the FoV of the sensor measurements, requiring not less than 360-degree 3D LiDAR measurements.

In the subsequent chapter, these limitations were addressed by incorporating camera data, extracting semantics from real-world data, and applying a bundle adjustment algorithm to improve sensor poses derived from restricted FoV measurements.

RQ 3. ***How can semantics and LiDAR-camera fusion be utilized to create a robust alignment and correction method of SLAM-acquired real-world 3D data with a BIM model or a semantic 3D map?***

Chapter 6 introduces novel methodologies for seamlessly registering point clouds from diverse sources, overcoming the challenge of aligning noise and slightly drifted real-world data with a BIM model or reference semantically enriched map. The proposed regis-

tration methodologies effectively simplify the alignment process, assuming Manhattan World and only a small percentage of drift in the acquired map. Moreover, semantic landmarks were leveraged, in particular columns and walls, to reliably align the point clouds, avoiding wrong alignments because of false positive correspondences with clutter or dynamic elements.

To refine the sensor poses after global registration, BIMCaP was proposed. BIMCaP not only shows superior performance than a SOTA algorithm in sensor pose refinement but also shows how it is possible to fuse sparse LiDAR scans with camera images using a combination of classical and neural depth completion methods. Moreover, a method was proposed to create vectorized semantic floor plans from BIM models or semantically enriched 3D maps, simplifying the geometric and semantic information available in the model.

Another important contribution in Chapter 6 represents the data and models that enable semantic segmentation on images of real-world indoor construction sites. By pre-training an object detection algorithm with the created labeled data and leveraging a SOTA segmentation algorithm, it is possible to detect permanent elements in the construction site (such as walls and columns), which, as stated before, represent critical elements not only for cross-source global registration but also for sensor pose refinement.

7.2 Contributions to the Field

The following list summarizes the main contributions of this dissertation to the field of map-based long-term localization and mapping.

- Developed a novel methodology for transforming 2D Occupancy Grid Maps (OGMs) into Pose Graph-based Maps (PGBMs), called OGM2PGBM.
- Conducted comprehensive evaluations of SOTA 2D LiDAR localization algorithms.
- Developed automated methods for generating accurate OGMs and 3D session data from large-scale BIM models and point clouds.
- Introduced Indoor Scan Context (ISC) and YawICP for fast indoor place recognition and point cloud registration.

- Introduced SLAM2REF, a holistic multi-session anchoring system for aligning and correcting drifted sensor poses using 3D BIM models or point clouds, achieving high reliability and accuracy.

- Developed methods for analyzing and detecting changes in aligned 3D data.

- Leveraged principal normal direction count and feature matching for cross-source global point cloud registration using semantic landmarks.

- Proposed BIMCaP framework for aligning and correcting sensor measurements with BIM models using reliable semantically filtered landmarks and also advancing semantic segmentation on images of construction sites.

- The majority of the proposed methods were evaluated using open-access datasets and have been made open-source to ensure reproducibility and facilitate benchmarking. This initiative aims to contribute to the rapid advancement of more effective and resilient methodologies by future researchers. For a comprehensive list of all open-source contributions, refer to Section 1.14.

7.3 Practical Implications

The following list highlights the key practical implications of the developed methods for researchers and practitioners in the mapping and autonomous navigation industry.

- Researchers will be less limited while obtaining accurate ground truth poses for the evaluation ofSLAM or localization algorithms, not depending on highly cost sensors or specialized indoor setups for pose tracking.

- People working in the survey industry might be able to correct wrongly scanned regions with the help of a reference 3D map and the proposed SLAM2REF framework.

- Developers and users working with autonomous mobile vehicles and 3D LiDARs might leverage the proposed Indoor Scan Context Descriptor (ISCD) for fast place recognition within 3D reference maps, and those working with 2D LiDARs might benefit from a robust real-time pose tracking algorithm for which the proposed OGM2PGBM pipeline can be of great use.

- With the help of the created labeled dataset and segmentation methods, future researchers on autonomous mobile systems on construction sites might benefit in terms of semantic segmentation on real-world images of indoor construction sites, for which currently there is a very scarce amount of open-access data.

- With the provided open-access datasets and open-access alignment methods, it is hoped that a more automated, efficient, high-quality, and less human-risky construction process will be achieved.

- Similarly, first responders might find it very helpful to have clearly visible 3D information of the incident area aligned and corrected with a reference map for them to rescue victims effectively, avoiding dangerous areas and not putting themselves into high-risk situations.

- The proposed innovative handheld mapping and mounting systems (M. A. Vega-Torres & Borrmann, 2024; M. A. Vega-Torres & Pfitzner, 2023), specifically developed for the Go1 robot but adaptable to any robot with a flat surface, are designed to serve as comprehensive tools or guides for future researchers. These systems aim to facilitate the development of cost-effective and robust mobile mapping and autonomous robotic systems, thereby accelerating advancements in this field. For additional details, please refer to Appendix B.

- The comparison of available legged robots, detailed in Appendix B, can help possible future users make informed decisions when considering the purchase of one of these systems.

- In the field of Mixed and AR, the proposed advancements in precise sensor pose correction and robust localization can significantly enhance the accuracy and reliability of AR experiences. By aligning real-world data with virtual elements using the proposed SLAM2REF or BIMCaP methods, developers can create more immersive and contextually accurate AR applications. This can be particularly beneficial in sectors such as architecture, construction, and maintenance, where precise overlay of digital information on physical environments is crucial.

- Another significant practical implication lies in mobile robotics within the retail industry, where there is a constant need for periodic map updates and robustness to

environmental changes. Using the proposed BIMCaP framework, these systems can align their data exclusively with permanent elements of the reference map (such as columns, walls, floors, and ceilings). This capability enables the creation of accurate, updated maps and the precise detection of changes. Consequently, this will lead to improved inventory management and more efficient operations within retail stores.

This dissertation has advanced the field of map-based long-term localization and mapping by addressing critical challenges and providing innovative solutions. By leveraging 3D BIM models and point clouds, developing new methodologies for pose refinement and alignment, and integrating semantic data and camera information, it was demonstrated that it is possible to significantly improve the accuracy of maps acquired with SLAM systems in changing environments. Our contributions, including the SLAM2REF and BIMCaP frameworks, demonstrate significant potential for real-world applications in mapping challenging environments, such as construction sites.

Moreover, this research underscores the importance of combining traditional techniques with modern deep-learning-based algorithms to enhance landmark filtering, depth completion, and map alignment. These advancements not only provide valuable insights for researchers but also offer practical tools for industry practitioners, enabling more precise and reliable mapping in various contexts.

The implications of this work extend beyond the immediate scope of this dissertation, paving the way for future developments in collaborative robot mapping, robust pose tracking, and efficient construction site management. By making the created datasets and algorithms available as open-source resources, the goal is to foster further innovation and facilitate the adoption of the proposed methods in diverse applications.

In summary, this research lays a solid foundation for ongoing advancements in the field, highlighting the transformative potential of integrating BIM models, point clouds, semantic data, and cutting-edge algorithms for improved localization and mapping in complex and changing environments.

7.4 Limitations and Recommendations on Future Directions

This section outlines the primary limitations of the current work and offers recommendations for future research to enhance the proposed solutions and facilitate their practical application. It is divided into six key points addressing the extension of these solutions. Although some recommendations for future studies are provided throughout the chapters, this section presents a comprehensive overview.

- The transformation of 2D OGMs to PGBMs requires substantial computational resources and expertise, making it unsuitable for real-time applications and large-scale maps. In this regard, a Docker to make the implementation on other machines much easier was provided. However, the computational time could potentially be increased by selecting only certain key positions to simulate the scans. A more sophisticated process would be to avoid running the trajectory builders of graph-based SLAM algorithms to create the respective PGBMs and instead create these maps directly while the data is being simulated. The difficulty here is that every algorithm has its way of serializing PGBMs, so each of them would require a different implementation.

- The developed image localization pipeline is highly dependent on environmental conditions, requiring a low level of Scan-Map deviations to accurately match vanishing points and vertical lines. These dependencies limit its applicability in dynamic or cluttered environments. For such environments, using neural algorithms for room layout detection can be of utility; however, with the limitation of working only on Manhattan-World environments.

- In terms of landmark extraction for registrations, identifying accurate correspondences in environments with Scan-Map deviations is challenging. The proposed SLAM2REF method relies on scans from locations with minimal deviations to ensure place recognition and alignment with features from the BIM model. Additionally, retrieving highly accurate 6-DoF poses for all keyframe scans is difficult due to sensor noise, undistortion inaccuracies, and the presence of Scan-Map deviations. One potential solution to this issue is to use point cloud semantic segmentation algorithms for individual 3D LiDAR scans. However, most existing methods have been designed for outdoor environments. To adapt these algorithms for indoor construc-

tion settings, labeled datasets specific to indoor construction environments would be necessary.

- Extending the efficiency and robustness of the SLAM2REF method towards a real-time framework represents a promising direction for various tasks, including collaborative robot mapping and localization (Cramariuc et al., 2022; Lajoie & Beltrame, 2024), such as to be able to solve the kidnapping robot problem in indoor environments with the proposed Indoor Scan Context Descriptor (ISCD). Also, an essential aspect of achieving more robust alignment involves leveraging deep-learning-based place recognition algorithms, which are anticipated to become progressively reliable for indoor scenarios with sufficient training data in the future.

- The cross-source global registration method, while effective for initial rough alignment, depends on semantically enriched point clouds, which may not always be available. High drift in point clouds can also limit the effectiveness of the method. Furthermore, the method assumes a Manhattan-World geometry, which may not hold in all scenarios. Translation estimates based on median distances may lack robustness in certain situations, e.g., in scenarios with few structural columns or highly occluded elements, affecting registration accuracy. A possible technique that can support tackling the issue of occluded and partially scanned elements might involve neural point cloud completion. Complete elements (such as walls and columns) would allow a better registration using only their centroids.

- The BIMCaP framework, although achieving better results than SOTA methods, still requires the manual execution of separate processes, limiting automation. Moreover, numerical results remain distant from the ground truth, most likely due to the limited number of keyframes used in optimization. Additionally, the dataset used for training the semantic segmentation method is relatively small, restricting generalizability. A larger, more diverse dataset would enhance the method's robustness and applicability across different construction sites.

Literaturverzeichnis

Abaspur Kazerouni, I., Fitzgerald, L., Dooly, G., & Toal, D. (2022). A survey of state-of-the-art on visual slam. *Expert Systems with Applications*, *205*, 117734. https://doi.org/https://doi.org/10.1016/j.eswa.2022.117734

Acharya, D., Khoshelham, K., & Winter, S. (2019). BIM-posenet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *150*, 245–258. https://doi.org/10.1016/j.isprsjprs.2019.02.020

Acharya, D., Tennakoon, R., Muthu, S., Khoshelham, K., Hoseinnezhad, R., & Bab-Hadiashar, A. (2022). Single-image localisation using 3D models: Combining hierarchical edge maps and semantic segmentation for domain adaptation. *Automation in Construction*, *136*, 104152. https://doi.org/10.1016/j.autcon.2022.104152

Agostinho, L. R., Ricardo, N. M., Pereira, M. I., Hiolle, A., & Pinto, A. M. (2022). A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions. *IEEE Access*, *10*, 72182–72205. https://doi.org/10.1109/ACCESS.2022.3188990

Alliez, P., Bonardi, F., Bouchafa, S., Didier, J.-Y., Hadj-Abdelkader, H., Muñoz, F. I., Kachurka, V., Rault, B., Robin, M., & Roussel, D. (2020). Real-time multi-SLAM system for agent localization and 3D mapping in dynamic scenarios. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4894–4900.

Alshikh Khalil, M. A., & Hatem, I. (2021). GMCL as a proposed replacement to AMCL in ros for mobile robots localization in known-based 2D environments.

Asadi, K., Ramshankar, H., Noghabaei, M., & Han, K. (2019). Real-time image localization and registration with BIM using perspective alignment for indoor monitoring of construction. *Journal of Computing in Civil Engineering*, *33*(5), 04019031. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000847

Azzam, R., Taha, T., Huang, S., & Zweiri, Y. (2020). Feature-based visual simultaneous localization and mapping: A survey. *SN Applied Sciences*, *2*(2). https://doi.org/10.1007/s42452-020-2001-3

Bai, C., Xiao, T., Chen, Y., Wang, H., Zhang, F., & Gao, X. (2022a). Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels. *IEEE Robotics and Automation Letters*, *7*(2), 4861–4868. https://doi.org/10.1109/LRA.2022.3152830

Bai, C., Xiao, T., Chen, Y., Wang, H., Zhang, F., & Gao, X. (2022b). Faster-LIO: Lightweight Tightly Coupled Lidar-Inertial Odometry Using Parallel Sparse Incremental Voxels. *IEEE Robotics and Automation Letters*, *7*(2), 4861–4868. https://doi.org/10.1109/LRA.2022.3152830

Behley, J., & Stachniss, C. (2018). Efficient surfel-based slam using 3d laser range data in urban environments. *Proc. of Robotics: Science and Systems (RSS)*.

Besl, P., & McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256. https://doi.org/10.1109/34.121791

Blanco, J. L., & Rai, P. K. (2014). nanoflann: a C++ header-only fork of FLANN, a library for Nearest Neighbor (NN) wih KD-trees.

Blanco-Claraco, J. L. (2021). A tutorial on **SE**(**3**) transformation parameterizations and on-manifold optimization. *CoRR*, *abs/2103.15980*. https://arxiv.org/abs/2103.15980

Blum, H., Milano, F., Zurbrügg, R., Siegwart, R., Cadena, C., & Gawel, A. (2021). Self-improving semantic perception on a construction robot. *CoRR*, *abs/2105.01595*. https://arxiv.org/abs/2105.01595

Blum, H., Stiefel, J., Cadena, C., Siegwart, R., & Gawel, A. (2020). Precise robot localization in architectural 3D plans. *arXiv preprint arXiv:2006.05137*.

Boche, S., Laina, S. B., & Leutenegger, S. (2024). Tightly-Coupled LiDAR-Visual-Inertial SLAM and Large-Scale Volumetric Occupancy Mapping. https://arxiv.org/abs/2403.02280

Boche, S., Zuo, X., Schaefer, S., & Leutenegger, S. (2022). Visual-Inertial SLAM with Tightly-Coupled Dropout-Tolerant GPS Fusion. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7020–7027. https://doi.org/10.1109/IROS47612.2022.9981134

Boniardi, F., Caselitz, T., Kummerle, R., & Burgard, W. (2017). Robust LiDAR-based localization in architectural floor plans. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3318–3324. https://doi.org/10.1109/IROS.2017.8206168

Boniardi, F., Caselitz, T., Kümmerle, R., & Burgard, W. (2019). A pose graph-based localization system for long-term navigation in cad floor plans. *Robotics and Autonomous Systems*, *112*, 84–97. https://doi.org/10.1016/j.robot.2018.11.003

Boniardi, F., Valada, A., Mohan, R., Caselitz, T., & Burgard, W. (2019). Robot localization in floor plans using a room layout edge extraction network, 5291–5297. http://arxiv.org/pdf/1903.01804v2

Borrmann, A., Biswanath, M., Braun, A., Chen, Z., Cremers, D., Heeramaglore, M., Hoegner, L., Mehranfar, M., Kolbe, T. H., Petzold, F., Rueda, A., Solonets, S., & Zhu, X. X. (2024). Artificial intelligence for the automated creation of multi-scale digital twins of the built world—ai4twinning. In T. H. Kolbe, A. Donaubauer, & C. Beil (Eds.), *Recent advances in 3D geoinformation science* (pp. 233–247). Springer Nature Switzerland.

Botín-Sanabria, D. M., Mihaita, A.-S., Peimbert-García, R. E., Ramírez-Moreno, M. A., Ramírez-Mendoza, R. A., & Lozoya-Santos, J. d. J. (2022). Digital twin technology challenges and applications: A comprehensive review. *Remote Sensing*, *14*(6). https://doi.org/10.3390/rs14061335

Braun, A., & Borrmann, A. (2019). Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning. *Automation in Construction*, *106*, 102879. https://doi.org/10.1016/j.autcon.2019.102879

Braun, A., Tuttas, S., Borrmann, A., & Stilla, U. (2020). Improving progress monitoring by fusing point clouds, semantic data and computer vision. *Automation in Construction*, *116*, 103210. https://doi.org/10.1016/j.autcon.2020.103210

Caballero, F., & Merino, L. (2021). DLL: Direct LiDAR localization. a map-based localization approach for aerial robots. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5491–5498.

Cai, D., Li, R., Hu, Z., Lu, J., Li, S., & Zhao, Y. (2024). A comprehensive overview of core modules in visual SLAM framework. *Neurocomputing*, *590*, 127760. https://doi.org/10.1016/j.neucom.2024.127760

Cai, P., Scott, D., Li, X., & Wang, S. (2024). Orthogonal dictionary guided shape completion network for point cloud. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(2), 864–872. https://doi.org/10.1609/aaai.v38i2.27845

Campos, C., Elvira, R., Gomez, J. J., Montiel, J. M. M., & Tardós, J. D. (2021). ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, *37*(6), 1874–1890.

Cao, S., Lu, X., & Shen, S. (2021). GVINS: Tightly Coupled GNSS–Visual–Inertial Fusion for Smooth and Consistent State Estimation. *IEEE Transactions on Robotics*, *38*, 2004–2021. https://api.semanticscholar.org/CorpusID:232607229

Chen, K., Nemiroff, R., & Lopez, B. T. (2023a). Direct LiDAR-Inertial Odometry and Mapping: Perceptive and Connective SLAM. https://arxiv.org/abs/2305.01843

Chen, K., Nemiroff, R., & Lopez, B. T. (2023b). Direct LiDAR-inertial odometry: Lightweight LIO with continuous-time motion correction. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 3983–3989.

Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J., & Stachniss, C. (2021). Moving Object Segmentation in 3D LiDAR Data: A Learning-Based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters*, *6*(4), 6529–6536. https://doi.org/10.1109/LRA.2021.3093567

Chen, X., Milioto, A., Palazzolo, E., Giguère, P., Behley, J., & Stachniss, C. (2019). SuMa++: Efficient LiDAR-based Semantic SLAM. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4530–4537. https://doi.org/10.1109/IROS40897.2019.8967704

Chen, Z., Xu, Y., Yuan, S., & Xie, L. (2024). iG-LIO: An incremental GICP-based tightly-coupled LiDAR-inertial odometry. *IEEE Robotics and Automation Letters*, 1–8. https://doi.org/10.1109/LRA.2024.3349915

Cioffi, G., & Scaramuzza, D. (2020). Tightly-coupled Fusion of Global Positional Measurements in Optimization-based Visual-Inertial Odometry. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5089–5095. https://api.semanticscholar.org/CorpusID:212633981

Collins, F., Pfitzner, F., & Schlenger, J. (2022). Scalable construction monitoring for an as-performed progress documentation across time. *Proceedings of 33. Forum Bauinformatik*.

Cramariuc, A., Bernreiter, L., Tschopp, F., Fehr, M., Reijgwart, V., Nieto, J., Siegwart, R., & Cadena, C. (2022). Maplab 2.0–a modular and multi-modal mapping framework. *IEEE Robotics and Automation Letters*.

CVAT.ai Corporation. (2023, November). Computer Vision Annotation Tool (CVAT) [MIT License].

Cvišić, I., Marković, I., & Petrović, I. (2022). Soft2: Stereo visual odometry for road vehicles based on a point-to-epipolar-line metric. *IEEE Transactions on Robotics*, *39*(1), 273–288.

D. Caruso, J. Engel, & D. Cremers. (2015). Large-scale direct SLAM for omnidirectional cameras. *International Conference on Intelligent Robots and Systems (IROS)*.

Dantas, R., Peter, S., Wang, X., Vega-Torres, M. A., & Dugstad, A. (2022). Towards Real-time Image Localization with BIM models. In *Proceedings of 33. forum bauinformatik*.

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1052–1067. https://doi.org/10.1109/TPAMI.2007.1049

Dellaert, F. (2021). Factor graphs: Exploiting structure in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, *4*, 141–166.

Dellaert, F., Kaess, M., et al. (2017). Factor graphs for robot perception. *Foundations and Trends in Robotics*, *6*(1-2), 1–139.

Dellenbach, P., Deschaud, J.-E., Jacquet, B., & Goulette, F. (2022). CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure. *2022 International Conference on Robotics and Automation (ICRA)*, 5580–5586. https://doi.org/10.1109/ICRA46639.2022.9811849

Dugstad, A., Dubey, R., Abualdenien, J., & Borrmann, A. (2022). BIM-based disaster response: Facilitating indoor path planning for various agents. *Proc. of European Conference on Product and Process Modeling 2022*, 265–289.

Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In *Lecture notes in computer science* (pp. 834–849). Springer International Publishing. https://doi.org/10.1007/978-3-319-10605-2_54

Ercan, S., Blum, H., Gawel, A., Siegwart, R., Gramazio, F., & Kohler, M. (2020). Online synchronization of building model for on-site mobile robotic construction. *37th International Symposium on Automation and Robotics in Construction (ISARC 2020)(virtual)*, 1508–1514.

Feng, H., Chen, Q., & Garcia de Soto, B. (2021). Application of digital twin technologies in construction: An overview of opportunities and challenges. *Proceedings of the*

International Symposium on Automation and Robotics in Construction (IAARC). https://doi.org/10.22260/isarc2021/0132

Follini, C., Magnago, V., Freitag, K., Terzer, M., Marcher, C., Riedl, M., Giusti, A., & Matt, D. T. (2020). BIM-integrated collaborative robotics for application in building construction and maintenance. *Robotics*, *10*(1), 2. https://doi.org/10.3390/robotics10010002

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2016). On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, *33*(1), 1–21.

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., & Scaramuzza, D. (2017). SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Transactions on Robotics*, *33*(2), 249–265. https://doi.org/10.1109/TRO.2016.2623335

Fox, D., Burgard, W., Dellaert, F., & Thrun, S. (1999). Monte Carlo Localization: efficient position estimation for mobile robots. *Proceedings of the National Conference on Artificial Intelligence*, (Handschin 1970), 343–349.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., & Scaramuzza, D. (2022). Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(1), 154–180. https://doi.org/10.1109/TPAMI.2020.3008413

Gawel, A., Blum, H., Pankert, J., Krämer, K., Bartolomei, L., Ercan, S., Farshidian, F., Chli, M., Gramazio, F., Siegwart, R., et al. (2019). A fully-integrated sensing and control system for high-accuracy mobile robotic building construction. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2300–2307.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*.

Gladkova, M., Korobov, N., Demmel, N., Ošep, A., Leal-Taixé, L., & Cremers, D. (2022). Directtracker: 3D multi-object tracking using direct image alignment and photometric bundle adjustment. *International Conference on Intelligent Robots and Systems (IROS)*.

Gopee, M. A., Prieto, S. A., & García de Soto, B. (2023). Improving autonomous robotic navigation using ifc files. *Construction Robotics*, *7*(3–4), 235–251. https://doi.org/10.1007/s41693-023-00112-8

Gopee, M. A., Prieto, S. A., & García de Soto, B. (2022). Ifc-based generation of semantic obstacle maps for autonomous robotic systems. *Computing in Construction*. https://doi.org/10.35490/ec3.2022.161

Grisetti, G., Kümmerle, R., Stachniss, C., & Burgard, W. (2010). A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine*, *2*(4), 31–43. https://doi.org/10.1109/MITS.2010.939925

Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, *23*(1), 34–46. https://doi.org/10.1109/TRO.2006.889486

Grupp, M. (2017a). Evo: Python package for the evaluation of odometry and SLAM. https://github.com/MichaelGrupp/evo

Grupp, M. (2017b). Evo: Python package for the evaluation of odometry and SLAM.

Gschwandtner, M. (2013). *Support framework for obstacle detection on autonomous trains* [Doctoral dissertation, Department of Computer Sciences, University of Salzburg].

Gschwandtner, M., Kwitt, R., Uhl, A., & Pree, W. (2011). Blensor: Blender sensor simulation toolbox. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, & J. Ming (Eds.), *Advances in visual computing* (pp. 199–208). Springer Berlin Heidelberg.

Gutmann, J.-S., & Schlegel, C. (1996). AMOS: comparison of scan matching approaches for self-localization in indoor environments. *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots (EUROBOT '96)*, 61–67. https://doi.org/10.1109/EURBOT.1996.551882

Halder, S., & Afsari, K. (2023). Robots in inspection and monitoring of buildings and infrastructure: a systematic review. *Applied Sciences*, *13*(4), 2304.

Haque, A., Elsaharti, A., Elderini, T., Elsaharty, M. A., & Neubert, J. (2020). UAV autonomous localization using macro-features matching with a cad model. *Sensors (Basel, Switzerland)*, *20*(3). https://doi.org/10.3390/s20030743

He, G., Zhang, F., Li, X., & Shang, W. (2021). Robust mapping and localization in offline 3D point cloud maps. *2021 6th IEEE International Conference on Ad-*

vanced Robotics and Mechatronics (ICARM), 765–770. https://doi.org/10.1109/ICARM52023.2021.9536181

Hedau, V., Hoiem, D., & Forsyth, D. (2010). Thinking inside the box: Using appearance models and context based on room geometry. *European Conference on Computer Vision*, 224–237.

Hendrikx, R. W. M., Pauwels, P., Torta, E., Bruyninckx, H. J., & van de Molengraft, M. J. G. (2021). Connecting semantic building information models and robotics: An application to 2D LiDAR-based localization. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 11654–11660. https://doi.org/10.1109/ICRA48506.2021.9561129

Hendrikx, R., Bruyninckx, H., Elfring, J., & Van De Molengraft, M. (2022). Local-to-global hypotheses for robust robot localization. *Frontiers in Robotics and AI*, 171.

Henning, D. F., Laidlow, T., & Leutenegger, S. (2022). Bodyslam: Joint camera localisation, mapping, and human motion tracking. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer vision – eccv 2022* (pp. 656–673). Springer Nature Switzerland.

Hess, W., Kohler, D., Rapp, H., & Andor, D. (2016). Real-time loop closure in 2D LiDAR SLAM. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1271–1278. https://doi.org/10.1109/ICRA.2016.7487258

Hofstadler, C. (2007). *Bauablaufplanung und Logistik im Baubetrieb*. Springer.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots*, *34*, 189–206.

Huang, L. (2021). Review on LiDAR-based SLAM Techniques. *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*, 163–168. https://doi.org/10.1109/CONF-SPML54095.2021.00040

IfcOpenShell Contributors. (2023a). Ifcconvert documentation [URL: https://blenderbim.org/docs-python/ifcconvert/usage.html]. https://blenderbim.org/docs-python/ifcconvert/usage.html

IfcOpenShell Contributors. (2023b). Ifcconvert: An application for converting IFC geometry into several file formats [URL: https://ifcopenshell.sourceforge.net/ifcconvert.html]. https://ifcopenshell.sourceforge.net/ifcconvert.html

Jia, G., Li, X., Zhang, D., Xu, W., Lv, H., Shi, Y., & Cai, M. (2022). Visual-SLAM classical framework and key techniques: A review. *Sensors*, *22*(12). https://doi.org/10.3390/s22124582

Jurić, A., Kendeš, F., Marković, I., & Petrović, I. (2021). A comparison of graph optimization approaches for pose estimation in SLAM. *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, 1113–1118. https://doi.org/10.23919/MIPRO52101.2021.9596721

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, *31*(2), 216–235.

Kaess, M., Ranganathan, A., & Dellaert, F. (2008). iSAM: Incremental Smoothing and Mapping. *IEEE Transactions on Robotics*, *24*(6), 1365–1378. https://doi.org/10.1109/TRO.2008.2006706

Karimi, S., Braga, R. G., Iordanova, I., & St-Onge, D. (2021). Semantic navigation using building information on construction sites. http://arxiv.org/pdf/2104.10296v1

Karimi, S., Iordanova, I., & St-Onge, D. (2020). An ontology-based approach to data exchanges for robot navigation on construction sites. *Journal of Information Technology in Construction*.

Kayhani, N., Schoellig, A., & McCabe, B. (2023). Perception-aware tag placement planning for robust localization of UAVs in indoor construction environments [Cited by: 1; All Open Access, Green Open Access]. *Journal of Computing in Civil Engineering*, *37*(2). https://doi.org/10.1061/JCCEE5.CPENG-5068

Kayhani, N., Zhao, W., McCabe, B., & Schoellig, A. P. (2022). Tag-based visual-inertial localization of unmanned aerial vehicles in indoor construction environments using an on-manifold extended kalman filter. *Automation in Construction*, *135*, 104112. https://doi.org/https://doi.org/10.1016/j.autcon.2021.104112

Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., & Teller, S. (2010). Multiple relative pose graphs for robust cooperative mapping. *2010 IEEE International Conference on Robotics and Automation*, 3185–3192.

Kim, G., Choi, S., & Kim, A. (2021). Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, *38*(3), 1856–1874.

Kim, G., & Kim, A. (2018). Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4802–4809.

Kim, G., & Kim, A. (2022). LT-mapper: A modular framework for LiDAR-based lifelong mapping. *2022 International Conference on Robotics and Automation (ICRA)*, 7995–8002.

Kim, G., Yun, S., Kim, J., & Kim, A. (2022). SC-LiDAR-SLAM: A front-end agnostic versatile LiDAR SLAM system. *2022 International Conference on Electronics, Information, and Communication (ICEIC)*, 1–6. https://doi.org/10.1109/ICEIC54506.2022.9748644

Kim, J., Chung, D., Kim, Y., & Kim, H. (2022). Deep learning-based 3D reconstruction of scaffolds using a robot dog. *Automation in Construction*, *134*, 104092.

Kim, K., & Peavy, M. (2022). BIM-based semantic building world modeling for robot task planning and execution in built environments. *Automation in Construction*, *138*, 104247. https://doi.org/10.1016/j.autcon.2022.104247

Kim, S., Peavy, M., Huang, P.-C., & Kim, K. (2021). Development of BIM-integrated construction robot task planning and simulation system. *Automation in Construction*, *127*, 103720. https://doi.org/10.1016/j.autcon.2021.103720

Ko, P., Prieto, S. A., & García de Soto, B. (2021). Abecis: An automated building exterior crack inspection system using uavs, open-source deep learning and photogrammetry. *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*. https://doi.org/10.22260/isarc2021/0086

Koenig, N., & Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, *3*, 2149–2154.

Koide, K., Miura, J., & Menegatti, E. (2019). A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, *16*(2), 1729881419841532.

Koide, K., Oishi, S., Yokozuka, M., & Banno, A. (2022). Scalable fiducial tag localization on a 3D prior map via graph-theoretic global tag-map registration. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5347–5353. https://doi.org/10.1109/IROS47612.2022.9981079

Konolige, K., Agrawal, M., & Sola, J. (2011). Large-scale visual odometry for rough terrain. *Robotics Research: The 13th International Symposium ISRR*, 201–212.

Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., & Vincent, R. (2010). Efficient sparse pose adjustment for 2D mapping. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 22–29. https://doi.org/10.1109/IROS.2010.5649043

Krijnen, T. (2015). Ifcopenshell. https://github.com/IfcOpenShell/IfcOpenShell

Kropp, C., Koch, C., & König, M. (2018). Interior construction state recognition with 4d BIM registered image sequences. *Automation in Construction*, *86*, 11–32. https://doi.org/10.1016/j.autcon.2017.10.027

Kumar, D., & Muhammad, N. (2023). A Survey on Localization for Autonomous Vehicles. *IEEE Access*, *11*, 115865–115883. https://doi.org/10.1109/ACCESS.2023.3326069

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). G2o: A general framework for graph optimization. *2011 IEEE International Conference on Robotics and Automation*, 3607–3613. https://doi.org/10.1109/ICRA.2011.5979949

Labbé, M., & Michaud, F. (2019). Rtab-map as an open-source LiDAR and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, *36*(2), 416–446.

Lai, T. (2022). A Review on Visual-SLAM: Advancements from Geometric Modelling to Learning-based Semantic Scene Understanding. https://arxiv.org/abs/2209.05222

Laidlow, T., Bloesch, M., Li, W., & Leutenegger, S. (2017). Dense RGB-D-inertial SLAM with map deformations. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6741–6748. https://doi.org/10.1109/IROS.2017.8206591

Laina, S. B., Boche, S., Papatheodorou, S., Tzoumanikas, D., Schaefer, S., Chen, H., & Leutenegger, S. (2024). Scalable Autonomous Drone Flight in the Forest with Visual-Inertial SLAM and Dense Submaps Built without LiDAR. https://arxiv.org/abs/2403.09596

Lajoie, P.-Y., & Beltrame, G. (2024). Swarm-SLAM: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robotics and Automation Letters*, *9*(1), 475–482. https://doi.org/10.1109/lra.2023.3333742

Landgraf, Z., Scona, R., Laidlow, T., James, S., Leutenegger, S., & Davison, A. J. (2021). Simstack: A generative shape and instance model for unordered object stacks. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 13012–13022.

Lee, T.-C., Kashyap, R. L., & Chu, C.-N. (1994). Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, *56*(6), 462–478.

Leutenegger, S. (2020). Okvis 2.0 for the fpv drone racing vio competition 2020.

Leutenegger, S. (2022). OKVIS2: Realtime Scalable Visual-Inertial SLAM with Loop Closure. https://arxiv.org/abs/2202.09199

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2014). Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, *34*(3), 314–334. https://doi.org/10.1177/0278364914554813

Leutenegger, S., Melzer, A., Alexis, K., & Siegwart, R. (2014). Robust state estimation for small unmanned airplanes. *2014 IEEE Conference on Control Applications (CCA)*, 1003–1010. https://doi.org/10.1109/CCA.2014.6981466

Leutenegger, S., & Siegwart, R. Y. (2012). A low-cost and fail-safe Inertial Navigation System for airplanes. *2012 IEEE International Conference on Robotics and Automation*, 612–618. https://doi.org/10.1109/ICRA.2012.6225061

Li, F., Zhang, H., Liu, S., Guo, J., Ni, L. M., & Zhang, L. (2022). Dn-detr: Accelerate detr training by introducing query denoising. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13619–13627.

Li, H., Chan, G., Wong, J. K. W., & Skitmore, M. (2016). Real-time locating systems applications in construction. *Automation in Construction*, *63*, 37–47.

Li, L., Kong, X., Zhao, X., Huang, T., Li, W., Wen, F., Zhang, H., & Liu, Y. (2021). SSC: Semantic scan context for large-scale place recognition. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2092–2099.

Lin, J., & Zhang, F. (2022). R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package. *2022 International Conference on Robotics and Automation (ICRA)*, 10672–10678. https://doi.org/10.1109/ICRA46639.2022.9811935

Lin, J., Zheng, C., Xu, W., & Zhang, F. (2021). R ^{2} LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping. *IEEE Robotics and Automation Letters*, *6*(4), 7469–7476. https://doi.org/10.1109/LRA.2021.3095515

Liu, H., Chen, M., Zhang, G., Bao, H., & Bao, Y. (2018). ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1974–1982. https://doi.org/10.1109/CVPR.2018.00211

Liu, J., Gao, W., & Hu, Z. (2020). Optimization-Based Visual-Inertial SLAM Tightly Coupled with Raw GNSS Measurements. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 11612–11618. https://api.semanticscholar.org/CorpusID:225039931

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al. (2023). Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Liu, Y., Fu, Y., Chen, F., Goossens, B., Tao, W., & Zhao, H. (2021). Simultaneous localization and mapping related datasets: A comprehensive survey. *arXiv preprint arXiv:2102.04036*.

Lopez-de-Teruel, P. E., Garcia, F. J., Canovas, O., Gonzalez, R., & Carrasco, J. A. (2017). Human behavior monitoring using a passive indoor positioning system: A case study in a sme [14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops]. *Procedia Computer Science*, *110*, 182–189. https://doi.org/https://doi.org/10.1016/j.procs.2017.06.076

Lukierski, R., Leutenegger, S., & Davison, A. (2022). Estimating dimensions for an enclosed space using a multi-directional camera [US Patent 11,276,191].

Lukierski, R., Leutenegger, S., & Davison, A. J. (2017). Room layout estimation from rapid omnidirectional exploration. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 6315–6322. https://doi.org/10.1109/ICRA.2017.7989747

Lv, J., Xu, J., Hu, K., Liu, Y., & Zuo, X. Targetless calibration of LiDAR-IMU system based on continuous-time batch estimation. In: In *2020 ieee/rsj international conference on intelligent robots and systems (iros)*. IEEE. 2020, 9968–9975.

Lv, J., Zuo, X., Hu, K., Xu, J., Huang, G., & Liu, Y. (2022). OA-LICalib: Observability-aware intrinsic and extrinsic calibration of LiDAR-IMU systems. *IEEE Transactions on Robotics*, *38*(6), 3734–3753.

Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S., & Chen, K. (2022). RTMDet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*.

Macario Barros, A., Michel, M., Moline, Y., Corre, G., & Carrel, F. (2022). A comprehensive survey of visual SLAM algorithms. *Robotics*, *11*(1). https://doi.org/10.3390/robotics11010024

Macenski, S., & Jambrecic, I. (2021). SLAM toolbox: SLAM for the dynamic world. *Journal of Open Source Software*, *6*(61), 2783. https://doi.org/10.21105/joss.02783

Macenski, S., Moore, T., Lu, D. V., Merzlyakov, A., & Ferguson, M. (2023). From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2. *Robotics and Autonomous Systems*, *168*, 104493. https://doi.org/10.1016/j.robot.2023.104493

Macenski, S., Singh, S., Martín, F., & Ginés, J. (2023). Regulated pure pursuit for robot path tracking. *Autonomous Robots*, 1–10.

Mantha, B., & Garcia de Soto, B. (2019). Designing a reliable fiducial marker network for autonomous indoor robot navigation. *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)*. https://doi.org/10.22260/isarc2019/0011

Mantha, B. R. K., de Soto, B. G., Menassa, C. C., & Kamat, V. R. (2020, February). Robots in indoor and outdoor environments. In *Construction 4.0* (pp. 307–325). Routledge. https://doi.org/10.1201/9780429398100-16

Mantha, B. R. K., & Garcia de Soto, B. (2022). Investigating the fiducial marker network characteristics for autonomous mobile indoor robot navigation using ros and gazebo. *Journal of Construction Engineering and Management*, *148*(10). https://doi.org/10.1061/(asce)co.1943-7862.0002378

Mantha, B. R., Jung, M. K., García de Soto, B., Menassa, C. C., & Kamat, V. R. (2020). Generalized task allocation and route planning for robots with multiple depots in indoor building environments. *Automation in Construction*, *119*, 103359. https://doi.org/https://doi.org/10.1016/j.autcon.2020.103359

McDonald, J., Kaess, M., Cadena, C., Neira, J., & Leonard, J. (2013). Real-time 6-DOF multi-session visual SLAM over large-scale environments [Selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011)]. *Robotics and Autonomous Systems*, *61*(10), 1144–1158. https://doi.org/https://doi.org/10.1016/j.robot.2012.08.008

Merrill, N., Guo, Y., Zuo, X., Huang, X., Leutenegger, S., Peng, X., Ren, L., & Huang, G. (2022). Symmetry and uncertainty-aware object SLAM for 6DoF object pose estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14901–14910.

Montemerlo, M. (2003, July). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association* (Publication No. CMU-RI-TR-03-28) [Doctoral dissertation, Carnegie Mellon University].

Moura, M. S., Rizzo, C., & Serrano, D. (2021). BIM-based Localization and Mapping for Mobile Robots in Construction. *2021 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2021*, 12–18. https://doi.org/10.1109/ICARSC52212.2021.9429779

Mourikis, A. I., & Roumeliotis, S. I. (2007). A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 3565–3572. https://doi.org/10.1109/ROBOT.2007.364024

Muhle, D., Koestler, L., Jatavallabhula, K. M., & Cremers, D. (2023). Learning Correspondence Uncertainty via Differentiable Nonlinear Least Squares. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, *31*(5), 1147–1163. https://doi.org/10.1109/TRO.2015.2463671

Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics*, *33*(5), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103

Mylonas, G., Kalogeras, A., Kalogeras, G., Anagnostopoulos, C., Alexakos, C., & Muñoz, L. (2021). Digital twins from smart manufacturing to smart cities: A survey. *IEEE Access*, *9*, 143222–143249. https://doi.org/10.1109/ACCESS.2021.3120843

Nam, D. V., & Gon-Woo, K. (2021). Solid-State LiDAR based-SLAM: A Concise Review and Application. *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 302–305. https://doi.org/10.1109/BigComp51126.2021.00064

Naumann, J., Xu, B., Leutenegger, S., & Zuo, X. (2024). NeRF-VO: Real-Time Sparse Visual Odometry With Neural Radiance Fields. *IEEE Robotics and Automation Letters*, *9*(8), 7278–7285. https://doi.org/10.1109/LRA.2024.3421192

NavVis, News, L., Magazine, L., the American Surveyor, GoGeomatics, International, G., Week, G., BIMplus, Source, S., & GeoConnexion. (2022). State of Mobile Mapping Survey 2022.

Oelsch, M., Karimi, M., & Steinbach, E. (2021). R-LOAM: Improving LiDAR Odometry and Mapping with Point-to-Mesh Features of a Known 3D Reference Object. *IEEE Robotics and Automation Letters*, *6*(2), 2068–2075. https://doi.org/10.1109/LRA.2021.3060413

Oelsch, M., Karimi, M., & Steinbach, E. (2022). Ro-loam: 3D reference object-based trajectory and map optimization in LiDAR odometry and mapping. *IEEE Robotics and Automation Letters*, 1–1. https://doi.org/10.1109/LRA.2022.3177846

Opoku, D.-G. J., Perera, S., Osei-Kyei, R., & Rashidi, M. (2021). Digital twin application in the construction industry: A literature review. *Journal of Building Engineering*, *40*, 102726.

Ozog, P., Carlevaris-Bianco, N., Kim, A., & Eustice, R. M. (2016). Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle. *Journal of Field Robotics*, *33*(3), 265–289.

Pan, Y., Xiao, P., He, Y., Shao, Z., & Li, Z. (2021). MULLS: Versatile LiDAR SLAM via Multi-metric Linear Least Square. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 11633–11640. https://doi.org/10.1109/ICRA48506.2021.9561364

Papatheodorou, S., Funk, N., Tzoumanikas, D., Choi, C., Xu, B., & Leutenegger, S. (2023). Finding Things in the Unknown: Semantic Object-Centric Exploration with an MAV. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 3339–3345. https://doi.org/10.1109/ICRA48891.2023.10160490

Perez-Grau, F. J., Caballero, F., Viguria, A., & Ollero, A. (2017). Multi-sensor three-dimensional monte carlo localization for long-term aerial robot navigation. *Inter-*

national Journal of Advanced Robotic Systems, *14*(5). https://doi.org/10.1177/1729881417732757

Pfaff, P., Burgard, W., & Fox, D. (2006). Robust monte-carlo localization using adaptive likelihood models. In H. I. Christensen (Ed.), *European robotics symposium 2006* (pp. 181–194, Vol. 22). Springer-Verlag. https://doi.org/10.1007/11681120{\textunderscore}15

Pfitzner, F., Braun, A., & Borrmann, A. (2023). Object-Detection Based Knowledge Graph Creation: Enabling Insight into Construction Processes. *ASCE International Conference on Computing in Civil Engineering 2023*, *in press*.

Platinsky, L., Davison, A. J., & Leutenegger, S. (2017). Monocular visual odometry: Sparse joint optimisation or dense alternation? *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5126–5133. https://doi.org/10.1109/ICRA.2017.7989599

Prieto, S. A., Garcia de Soto, B., & Adan, A. (2020, October). A methodology to monitor construction progress using autonomous robots. In H. Osumi (Ed.), *Proceedings of the 37th international symposium on automation and robotics in construction (isarc)* (pp. 265–289). International Association for Automation and Robotics in Construction (IAARC). https://doi.org/10.22260/ISARC2020/0210

Prieto, S. A., Giakoumidis, N., & García de Soto, B. (2024). Multiagent robotic systems and exploration algorithms: Applications for data collection in construction sites. *Journal of Field Robotics*, *41*(4), 1187–1203. https://doi.org/10.1002/rob.22316

Prieto, S. A., Xu, X., & García de Soto, B. (2024). A guide for construction practitioners to integrate robotic systems in their construction applications. *Frontiers in Built Environment*, *10*. https://doi.org/10.3389/fbuil.2024.1307728

Qiao, Z., Yu, Z., Yin, H., & Shen, S. (2023). Pyramid Semantic Graph-Based Global Point Cloud Registration with Low Overlap. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11202–11209. https://doi.org/10.1109/IROS55552.2023.10341394

Qin, T., Li, P., & Shen, S. (2018). VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, *34*(4), 1004–1020. https://doi.org/10.1109/TRO.2018.2853729

Ramezani, M., Wang, Y., Camurri, M., Wisth, D., Mattamala, M., & Fallon, M. (2020). The Newer College Dataset: Handheld LiDAR, inertial and vision with ground

truth. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. https://doi.org/10.1109/iros45743.2020.9340849

Sacks, R., Brilakis, I., Pikas, E., Xie, H. S., & Girolami, M. (2020). Construction with digital twin information systems. *Data-Centric Engineering*, *1*, e14. https://doi.org/10.1017/dce.2020.16

Schlenger, J., Pfitzner, F., Braun, A., Vilgertshofer, S., & Borrmann, A. (2023). Digitaler Zwilling Baustelle–Baustellenüberwachung zur automatisierten Zeit-und Kostenkontrolle. *Bautechnik*, *100*(4), 190–197.

Shaheer, M., Bavle, H., Sanchez-Lopez, J. L., & Voos, H. (2022). Robot localization using situational graphs and building architectural plans. *arXiv preprint arXiv:2209.11575*.

Shaheer, M., Millan-Romera, J. A., Bavle, H., Sanchez-Lopez, J. L., Civera, J., & Voos, H. (2023). Graph-based global robot localization informing situational graphs with architectural graphs.

Shan, T., & Englot, B. (2018). LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4758–4765. https://doi.org/10.1109/IROS.2018.8594299

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D. (2020a). LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping. *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 5135–5142.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D. (2020b). LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5135–5142. https://doi.org/10.1109/IROS45743.2020.9341176

Shinkansan. (2021). SLAM - 2019-ugrp-dpoom. *GitHub - https://github.com/shinkansan/2019-UGRP-DPoom/blob/master/SLAM*.

Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In I. J. Cox & G. T. Wilfong (Eds.), *Autonomous robot vehicles* (pp. 167–193). Springer New York. https://doi.org/10.1007/978-1-4613-8997-2_14

Sokolova, A., Nikitin, F., Vorontsova, A., & Konushin, A. (2022). Floorplan-aware camera poses refinement. *2022 IEEE/RSJ International Conference on Intelligent Robots*

and Systems (IROS), 4857–4864. https://doi.org/10.1109/IROS47612.2022.9981148

Soto, B. G. d., & Skibniewski, M. J. (2020, February). Future of robotics and automation in construction. In *Construction 4.0* (pp. 289–306). Routledge. https://doi.org/10.1201/9780429398100-15

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 573–580. https://api.semanticscholar.org/CorpusID:206942855

Tardif, J.-P., George, M., Laverne, M., Kelly, A., & Stentz, A. (2010). A new approach to vision-aided inertial navigation. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. https://doi.org/10.1109/iros.2010.5651059

Tee, Y. K., & Han, Y. C. (2021). Lidar-Based 2D SLAM for Mobile Robot in an Indoor Environment: A Review. *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, 1–7. https://doi.org/10.1109/GECOST52368.2021.9538731

Trzeciak, M., Pluta, K., Fathy, Y., Alcalde, L., Chee, S., Bromley, A., Brilakis, I., & Alliez, P. (2023a). Conslam: Construction data set for SLAM. *Journal of Computing in Civil Engineering*, *37*(3), 04023009.

Trzeciak, M., Pluta, K., Fathy, Y., Alcalde, L., Chee, S., Bromley, A., Brilakis, I., & Alliez, P. (2023b). ConSLAM: Periodically Collected Real-World Construction Dataset for SLAM and Progress Monitoring. In L. Karlinsky, T. Michaeli, & K. Nishino (Eds.), *Computer Vision – ECCV 2022 Workshops* (pp. 317–331, Vol. 13807). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-25082-8{\textunderscore}21

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(4), 376–380. https://doi.org/10.1109/34.88573

Vega-Torres, M. (2022, November). *Occupancy Grid Map to Pose Graph-based Map for long-term 2D LiDAR-based localization* (Version 1.0). Zenodo. https://doi.org/https://doi.org/10.5281/zenodo.7330270

Vega-Torres, M. A., & Borrmann, A. (2024). CMS Sensor Mounting System. https://doi.org/https://doi.org/10.14459/2024mp1750434

Vega-Torres, M. A., Braun, A., Bauer, H., Noichl, F., & Borrmann, A. (2021, September). Efficient Vertical Object Detection in Large High-Quality Point Clouds of Construction Sites. In *Proc. of the 2021 european conference on computing in construction*. https://doi.org/10.35490/EC3.2021.156

Vega-Torres, M. A., Braun, A., & Borrmann, A. (2022, September). Occupancy Grid Map to Pose Graph-based Map: Robust BIM-based 2D- LiDAR Localization for Lifelong Indoor Navigation in Changing and Dynamic Environments. In *Proc. of european conference on product and process modeling 2022*. https://doi.org/10.1201/9781003354222-72

Vega-Torres, M. A., Braun, A., & Borrmann, A. (2023). BIM-SLAM: Integrating BIM Models in Multi-session SLAM for Lifelong Mapping using 3D LiDAR. *Proc. of the 40th International Symposium on Automation and Robotics in Construction (ISARC 2023)*. https://doi.org/10.22260/ISARC2023/0070

Vega-Torres, M. A., Braun, A., Noichl, F., Borrmann, A., Bauer, H., & Wohlfeld, D. (2022). Recognition of temporary vertical objects in large point clouds of construction sites. *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction*, *174*(4), 134–149. https://doi.org/10.1680/jsmic.21.00033

Vega-Torres, M. A., Braun, A., & Borrmann, A. (2024a). OGM2PGBM. https://doi.org/https://doi.org/10.14459/2024mp1749236

Vega-Torres, M. A., Braun, A., & Borrmann, A. (2024b, June). ConSLAM BIM and GT poses. https://doi.org/10.14459/2024MP1743877

Vega-Torres, M. A., Braun, A., & Borrmann, A. (2024c). SLAM2REF: Advancing long-term mapping with 3D LiDAR and reference map integration for precise 6-DoF trajectory estimation and map extension. *Construction Robotics*, *8*(2), 13. https://doi.org/10.1007/s41693-024-00126-w

Vega-Torres, M. A., & Pfitzner, F. (2023, September). Investigating Robot Dogs for Construction Monitoring: A Comparative Analysis of Specifications and On-site Requirements. In *Proceedings of the 34th forum bauinformatik 2023*. https://doi.org/https://doi.org/10.13154/294-10094

Vega-Torres, M. A., Ribic, A., García de Soto, B., & Borrmann, A. (2024, July). BIMCaP: BIM-based AI-supported LiDAR-Camera Pose Refinement. In *Proc. of the 31th int. conference on intelligent computing in engineering (eg-ice)*. https://github.com/MigVega/BIMCaP

Vespa, E., Funk, N., Kelly, P. H. J., & Leutenegger, S. (2019). Adaptive-Resolution Octree-Based Volumetric SLAM. *2019 International Conference on 3D Vision (3DV)*, 654–662. https://doi.org/10.1109/3DV.2019.00077

Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., & Stachniss, C. (2023a). KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters*, *8*(2), 1029–1036. https://doi.org/10.1109/LRA.2023.3236571

Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., & Stachniss, C. (2023b). KISS-ICP: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, *8*(2), 1029–1036.

von Stumberg, L., & Cremers, D. (2022). DM-VIO: Delayed marginalization visual-inertial odometry. *IEEE Robotics and Automation Letters (RA-L) & International Conference on Robotics and Automation (ICRA)*, *7*(2), 1408–1415. https://doi.org/10.1109/LRA.2021.3140129

Wang, H., Wang, C., Chen, C.-L., & Xie, L. (2021). F-LOAM : Fast LiDAR Odometry and Mapping. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4390–4396. https://doi.org/10.1109/IROS51168.2021.9636655

Wang, H., Wang, C., & Xie, L. (2020). Intensity scan context: Coding intensity and geometry relations for loop closure detection. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2095–2101.

Wang, Y., Funk, N., Ramezani, M., Papatheodorou, S., Popović, M., Camurri, M., Leutenegger, S., & Fallon, M. (2021). Elastic and Efficient LiDAR Reconstruction for Large-Scale Exploration Tasks. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 5035–5041. https://doi.org/10.1109/ICRA48506.2021.9561736

Wang, Z., Zhang, L., Shen, Y., & Zhou, Y. (2023). D-LIOM: Tightly-Coupled Direct LiDAR-Inertial Odometry and Mapping. *IEEE Transactions on Multimedia*, *25*, 3905–3920. https://doi.org/10.1109/TMM.2022.3168423

Wu, W., Zhong, X., Wu, D., Chen, B., Zhong, X., & Liu, Q. (2023). LIO-Fusion: Reinforced LiDAR Inertial Odometry by Effective Fusion With GNSS/Relocalization and Wheel Odometry. *IEEE Robotics and Automation Letters*, *8*(3), 1571–1578. https://doi.org/10.1109/LRA.2023.3240372

Xin, Y., Zuo, X., Lu, D., & Leutenegger, S. (2023). SimpleMapping: Real-Time Visual-Inertial Dense Mapping with Deep Multi-View Stereo. *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 273–282. https://doi.org/10.1109/ISMAR59233.2023.00042

Xu, B., Li, W., Tzoumanikas, D., Bloesch, M., Davison, A., & Leutenegger, S. (2019). MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM. *2019 International Conference on Robotics and Automation (ICRA)*, 5231–5237. https://doi.org/10.1109/ICRA.2019.8794371

Xu, W., Cai, Y., He, D., Lin, J., & Zhang, F. (2022). FAST-LIO2: Fast direct LiDAR-inertial odometry. *IEEE Transactions on Robotics*, 1–21. https://doi.org/10.1109/TRO.2022.3141876

Xu, W., & Zhang, F. (2021). FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Robotics and Automation Letters*, *6*(2), 3317–3324. https://doi.org/10.1109/LRA.2021.3064227

Xu, X., & Garcia de Soto, B. (2020). On-site autonomous construction robots: A review of research areas, technologies, and suggestions for advancement. *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*. https://doi.org/10.22260/isarc2020/0055

Xu, X., & Garcia de Soto, B. (2023). Deep reinforcement learning-based task assignment and path planning for multi-agent construction robots. *Proceedings of the 2nd Future of Construction Workshop at the International Conference on Robotics and Automation (ICRA 2022)*. https://doi.org/10.22260/icra2023/0008

Xu, X., & García de Soto, B. (2022). Reinforcement learning with construction robots: A review of research areas, challenges and opportunities. *Proceedings of the 39th International Symposium on Automation and Robotics in Construction*. https://doi.org/10.22260/isarc2022/0052

Xu, X., Holgate, T., Coban, P., & García de Soto, B. (2021). Implementation of a robotic system for overhead drilling operations: A case study of the jaibot in the uae. *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*. https://doi.org/10.22260/isarc2021/0089

Yang, B., Dong, Z., Liang, F., & Liu, Y. (2016). Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS Journal of*

Photogrammetry and Remote Sensing, *113*(11), 43–58. https://doi.org/10.1016/j.isprsjprs.2015.12.005

Yang, H. (2018). Github - pgmmapcreator: Create pgm map from gazebo world file for ros localization. https://github.com/hyfan1116/pgmmapcreator

Yang, R., Yang, G., & Wang, X. (2023). Neural Volumetric Memory for Visual Locomotion Control. *Conference on Computer Vision and Pattern Recognition 2023*. https://openreview.net/forum?id=JYyWCcmwDS

Yao, L., Yu, H., & Lu, Z. (2021). Design and driving model for the quadruped robot: An elucidating draft. *Advances in Mechanical Engineering*, *13*(4), 16878140211009035.

Yin, H., Lin, Z., & Yeoh, J. K. (2023). Semantic localization on BIM-generated maps using a 3D LiDAR sensor. *Automation in Construction*, *146*, 104641. https://doi.org/https://doi.org/10.1016/j.autcon.2022.104641

Yu, Y., Gao, W., Liu, C., Shen, S., & Liu, M. (2019). A GPS-aided Omnidirectional Visual-Inertial State Estimator in Ubiquitous Environments. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7750–7755. https://doi.org/10.1109/IROS40897.2019.8968519

Zelinsky, A., Jarvis, R. A., Byrne, J., Yuta, S., et al. (1993). Planning paths of complete coverage of an unstructured environment by a mobile robot. In A. Zelinsky, R. A. Jarvis, J. C. Byrne, & S. Yuta (Eds.), *Proceedings of international conference on advanced robotics* (pp. 533–538, Vol. 13). Citeseer. http://pinkwink.kr/attachment/cfile3.uf@1354654A4E8945BD13FE77.pdf

Zhang, J., & Singh, S. (2014). LOAM: LiDAR odometry and mapping in real-time. *Robotics: Science and Systems*, *2*(9), 1–9. https://doi.org/10.15607/RSS.2014.X.007

Zhang, L., Camurri, M., Wisth, D., & Fallon, M. (2022). Multi-camera LiDAR inertial extension to the newer college dataset.

Zhang, L., Helmberger, M., Fu, L. F. T., Wisth, D., Camurri, M., Scaramuzza, D., & Fallon, M. (2023). Hilti-Oxford Dataset: A millimeter-accurate benchmark for simultaneous localization and mapping. *IEEE Robotics and Automation Letters*, *8*(1), 408–415. https://doi.org/10.1109/LRA.2022.3226077

Zhang, Y., Shi, P., & Li, J. (2024). 3D LiDAR SLAM: A survey. *The Photogrammetric Record*, *39*(186), 457–517. https://doi.org/10.1111/phor.12497

Zhang, Y., Guo, X., Poggi, M., Zhu, Z., Huang, G., & Mattoccia, S. (2023). Completionformer: Depth completion with convolutions and vision transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18527–18536.

Zhang, Z., & Scaramuzza, D. (2018). A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7244–7251.

Zheng, C., Zhu, Q., Xu, W., Liu, X., Guo, Q., & Zhang, F. (2022). FAST-LIVO: Fast and Tightly-coupled Sparse-Direct LiDAR-Inertial-Visual Odometry. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4003–4009. https://doi.org/10.1109/IROS47612.2022.9981107

Zheng, X., & Zhu, J. (2023). Traj-LO: In defense of LiDAR-only odometry using an effective continuous-time trajectory. *arXiv preprint arXiv:2309.13842*.

Zhou, H., Cao, Y., Chu, W., Zhu, J., Lu, T., Tai, Y., & Wang, C. (2022). Seedformer: Patch seeds based point cloud completion with upsample transformer. In *Computer vision – eccv 2022* (pp. 416–432). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-20062-5_24

Zienkiewicz, J., Tsiotsios, A., Davison, A., & Leutenegger, S. (2016). Monocular, Real-Time Surface Reconstruction Using Dynamic Level of Detail. *2016 Fourth International Conference on 3D Vision (3DV)*, 37–46. https://doi.org/10.1109/3DV.2016.82

Zimmerman, N., Wiesmann, L., Guadagnino, T., Läbe, T., Behley, J., & Stachniss, C. (2022). Robust onboard localization in changing environments exploiting text spotting. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 917–924. http://arxiv.org/pdf/2203.12647v1

Appendix A

List of Mathematical Variables

The table below lists the mathematical variables used in Chapters 2 and 5, along with their corresponding descriptions.

Table A.1: Explanation of Variables

Variable	**Description**
$p(X\|z)$	Posterior density of the states X given the measurements Z.
$F = (\mathcal{U}, \mathcal{V}, \mathcal{E})$	Factor graph comprising nodes (x_i or ϕ_i) connected by edges e_{ij}.
$\mathcal{E}$	Set of factor edges e_{ij}.
$\mathcal{U}$	Set of factor nodes ϕ_i.
$\mathcal{V}$	Set of variable nodes x_i.
X_i	Group of variables x_i connected to a factor ϕ_i.
$\phi(X)$	Global function factorized as $\phi(X) = \prod_i \phi_i(X_i)$.
z_i	Measurements (observed data point or variable).
$h_i(x_i, l_i)$	Mean measurement function of x_i and l_i. Represents the expected value of z_i given x_i and l_i.
Σ_i	Covariance matrix associated with z_i, representing zero-mean Gaussian noise.
$p(z_i\|x_i, l_i)$	Conditional density on the measurement z_i.
$\mathcal{N}(z_i; h_i(x_i, l_i), \Sigma_i)$	Multivariate normal distribution for the variable z_i with mean $h_i(x_i, l_i)$ and covariance matrix Σ_i.
$\|h_i(x_i, l_i) - z_i\|^2_{\Sigma_i}$	Mahalanobis distance between z_i and its mean $h_i(x_i, l_i)$.
SE(3)	Special Euclidean group.
η	Normally distributed zero-mean measurement noise with covariance Σ_c.
$\mathcal{S}_\mathcal{R}$	Synthetic reference session.

Variable	Description
$\mathcal{S}_{\mathcal{Q}}$	Real-world motion-undistorted query session.
$\mathbf{x}_R$	Set of poses of the reference session.
$\mathbf{x}_Q$	Set of poses of the query session.
$f(\cdot)$	Odometry model function.
$\mathbf{u}_i^s$	Constraints between consecutive poses $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$.
M_s	Number of poses in the session $\mathcal{S}$.
N_e	Number of encounters between sessions.
$\mathbf{p}_s$	Prior factor.
Δ_Q^*	Anchor node, which facilitates the global alignment of the query session to the reference map.
$\mathcal{G}$	Pose-graph map containing coordinates of pose nodes, odometry edges, and optionally recognized intra-session loop edges with uncertainty matrices.
$(\mathcal{P}_i, d_i)$	Pairs of 3D LiDAR scans $\mathcal{P}_i$ with their corresponding global descriptors d_i of the i^{th} keyframe.
n	Total number of equidistantly sampled keyframes.
N_c	Amount of Top descriptors candidates selected from the reference session after the comparison of the rotational invariant descriptors.
Σ_c	Covariance matrix of the detected loops or encounters incorporated into the factor graph problem as factors between sessions with anchoring.
$\mathbf{x}_Q^*$	Optimized 6-DoF poses of each scan of the query session.
$\mathbf{c}$	Loop closure detections, also called encounters denoting correspondences between the sessions.
ν_Q	Confidence level list providing the reliability of each pose after scan registration.
$h(\cdot)$	Original measurement model.
$h'(\cdot)$	Modified measurement model that incorporates anchor nodes.
$\mathbf{x}_{R,j}$	6-DoF (in SE(3)) Pose j in the reference session.

Variable	Description
$\mathbf{x}_{Q,j}$	6-DoF (in SE(3)) Pose j in the query session.
Δ_R	Anchor node for the reference session (also in (in SE(3)).
Δ_Q	Anchor node for the query session (in SE(3)).
$\mathbf{c}_j$	Difference in the global frame between poses $\mathbf{x}_R$ and $\mathbf{x}_Q$ (pose in SE(3)).
$\oplus$	SE(3) pose composition operator.
$\ominus$	SE(3) pose difference operator.
$\phi(\mathbf{x}_{R,i}, \mathbf{x}_{Q,j}, \Delta_R, \Delta_Q)$	Factor between sessions with anchoring, used in pose graph optimization.
Σ_P	Covariance assigned to the anchor node of the reference session, set to be insignificantly small.
Σ_L	Covariance assigned to the anchor node of the query session, set to be significantly large.
${}^{Q}\mathbf{x}_Q^*$	Optimized poses of the query session in the local coordinate system.
${}^{W}\mathbf{x}_Q^*$	Optimized poses of the query session transformed to the global coordinate system of the reference map.
W	Global coordinate system, same as the coordinate system of the reference session.
F_i	Fitness score distance threshold. The fitness score is the percentage of source inliers after point cloud registration, considering a maximum Point-to-Point (P2P) distance threshold.

Appendix B

Investigating Robot Dogs for Construction Monitoring[1]

B.1 Introduction

Research considering improving digitization on construction sites has increased significantly within the last years (Opoku et al., 2021). The digital twin construction, a digital representation of the construction environment, introduces a platform for many construction applications (Sacks et al., 2020). An ongoing challenge is to acquire periodic data on the entire construction site to create a comprehensive digital twin.

Robot dogs are inspired by the structure and motion of quadruped animals (Yao et al., 2021) and usually consist of a body with four mechanical legs allowing use on diverse terrains.

A time-exhaustive task of construction managers is keeping their site projects on track since regular manual site inspections are required. In this regard, autonomous robot-based construction monitoring provides a promising approach to reducing the effort. Most of the research regarding robot application in construction focuses on UAVs rather than ground-based, legged robots, which might be a better fit for construction sites (Halder & Afsari, 2023), particularly for indoor scenarios.

The potential usability of robot dogs on construction sites, focusing on their ability to facilitate data acquisition, access hard-to-reach areas of the site, and sustain with limited power supply options, has to be investigated to embrace future research directions.

This appendix contains the following: (a) a detailed comparison of available legged robots; (b) the design of a compact mapping system for Unmanned Ground Vehicles (UGVs); (c) a discussion about the potential for autonomous digital twin creation in construction and

[1] Portions of this appendix were previously published in (M. A. Vega-Torres & Pfitzner, 2023)

identification of the challenges that need to be addressed for successful implementation. These contributions support advancing the use of robotics and mapping technologies in construction monitoring and management.

B.2 Background

Construction sites are complex environments that significantly differ from controlled industrial production environments. Similar to other industries, the objective of construction is that the on-site factors are controlled in such a manner that buildings can be produced in an economically optimal way. Work, equipment, and resources are decisive in enhancing the on-site production rate (Hofstadler, 2007). Due to the unique character of construction sites, diverse, robust, and dynamic monitoring approaches are needed (H. Li et al., 2016).

The amount of research considering the application of robots in construction has grown significantly in recent years (Halder & Afsari, 2023). Several companies have emerged that try to adopt a robotic workforce to solve complex, labor-intensive problems. Or even more importantly, to relieve humans of doing dangerous work.

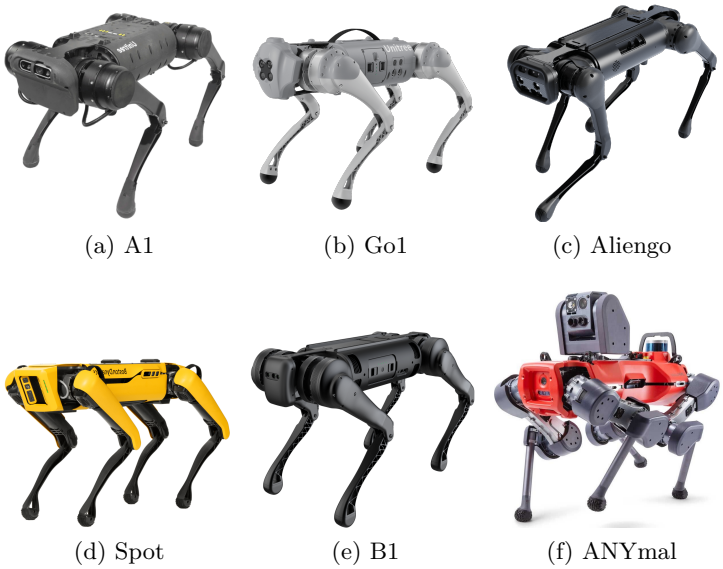
Among the different types of UGVs (wheeled, legged, or tracked), only legged robots provide the flexibility necessary to navigate by rugged hills or low-lying wet swamps, which are usually present in construction sites.

One of the few studies concentrating on legged-robot-based progress monitoring demonstrates how a robot can support safety management by tracking the location and size of scaffolding (J. Kim et al., 2022). Until now, suitability, robustness, cost, and deployment on construction sites of such legged robot systems are open topics.

Therefore, an analysis of the currently available quadruped-legged robots is conducted to inspect their practicality for construction monitoring. Moreover, a case study for automatic data acquisition with a self-developed mapping system is provided.

B.3 Quadruped robots

This section provides a comprehensive list (as of May 2023) of legged robots available in the European market, together with their main characteristics and current prices. Due



(a) A1 (b) Go1 (c) Aliengo

(d) Spot (e) B1 (f) ANYmal

Figure B.1: Currently in Europe available quadruped robots; as of May 2023.

to relevance and usability, robots heavier than 50 kg or smaller than 40 cm tall are not included.

B.3.1 Available quadruped robots

Figure B.1, tables B.1, and B.2 present the list of robots analyzed in this research together with their main properties. The table is organized by robot weight in ascending order. Some indices were calculated similarly as by Yao et al. (2021).

Table B.1: Comparison of quadruped robots. Rel. Year: Release Year; BL: body length (m); H: robot height while standing (m); W: robot weight (kg) without additional payload; PL: max payload (kg); PLC: payload capacity (%) payload/weight; IP: Ingress protection.

Name	Company	Rel. Year	BL (m)	H (m)	W (kg)	PL (kg)	PLC (%)	IP
A1	Unitree	2020	0.50	0.40	12	7	58.3	-
Go1	Unitree	2021	0.65	0.40	12	5[2]	41.7	-
Aliengo	Unitree	2019	0.65	0.60	20	13	65.0	-
Spot	Boston Dyn.	2020	1.10	0.61	32	14	43.8	54
B1	Unitree	2021	1.10	0.67	50	40	80.0	68
ANYmal	Anybotics	2019	0.93	0.89	50	23	46.0	67

[2]Some specifications say that the maximum payload of the Go1 Edu is 10 kg.

[1]All prices are for the German market as of May 2023.

[2]Some specifications say that the maximum speed of the Go1 Edu is 5 m/s.

[3]Assuming a payload of 10 kg and a speed of 5 m/s as for the Go1 Edu, this value increases to 646 Eur.

[4]The price of the Go1 depends on the version: Air, Pro, Edu, etc.

[5]This price of the ANYmal is not from an official seller, it might be wrong. Official prices are not disclosed to the public.

Table B.2: Continuation of Table 1. V: maximum speed (m/s); NS: normalized speed, maximum speed/body length; NWC: normalized work capacity, normalized speed×payload capacity; SS: maximum stairways step height (cm) recommended by the manufacturer; S: slope (in degrees) it can climb on a flat surface; T: run time range (h) with one battery provided by fabricator; M: external link to more information; Min P/Max P: minimum and maximum price (€) in Germany.

Name	V (m/s)	NS	NWC	SS (cm)	S (deg)	T (h)	M	Min P [1] (Tsd. €)	Max P (Tsd. €)
A1	3.3	6.6	385.0	12	35	1 - 2.5		13.5	-
Go1	3.7[2]	5.7	239.0[3]	12	35	1 - 2.5		5.6	23.1[4]
Aliengo	1.5	2.3	150.0	18	25	2.5 - 4.5		44.4	-
Spot	1.6	1.5	63.6	22	30	1.5		75.0	-
B1	1.8	1.6	130.9	20	35	2 - 4		70.0	86.9
ANYmal	1.3	1.4	64.3	25	30	2		150.0[5]	-

It is necessary to mention that companies like Tencent, Xiaomi Cyberdog, and Deep-Robotics were not considered here due to the current unavailability of their products in Europe.

B.3.2 Suitability analysis for construction site monitoring

Depending on diverse on-site conditions, one robot can be more suitable than another.

Table B.1 shows that it is possible to separate the listed robots into two groups: the ones with and those without water (ingress) protection. Assuming that the robot will not be exposed to heavy rain or submerged in the water, an ingress protection (IP) of 54 is suitable for construction site monitoring. Only the robots with Ingress Protection (IP) of 67 or above can be exposed to heavy rain or submerged in the water.

For the use case presented here, the payload is not a critical point. Considering that a mapping system weighs less than five kilograms, every robot listed here can carry this payload.

The maximum stairway step height is decisive when developing a completely autonomous system for construction monitoring on multiple stories. In general, high staircase steps remain a challenge. While some studies have been trying to push to the maximum limits of the capabilities of different robots, like the A1 (R. Yang et al., 2023), these attempts still need to improve stability. Assuming a standard step size of 19 cm, which is not guaranteed

to have during the construction phase, only the Spot, B1, and ANYmal robots would be suitable candidates.

Another essential aspect to consider is the weight of the robot when the robot needs to be repositioned manually. Since specific robot dogs are significantly heavier, at least two people are required to carry the Spot, B1, and ANYmal robots.

The Normalized Work Capacity (NWC) considers the robots' maximum speed, body length, and payload. Among the compared robots, the A1 archives the highest NWC. However, considering the maximum specifications of the Go1 Edu, having an NWC of 646, five times more than every IP-protected robot.

The index suggests that these robots are suitable choices for quickly and effectively carrying out the scanning process. Assuming the robot's maximum step size limitation, a possible solution is to place a robot for each level on the construction site.

As Table B.2 indicates, robot dogs' battery capacities averages are at 2.25 hours, which represents a limitation for scanning large on-site environments. Battery capacities depend on the use case and, therefore, must be tested thoroughly on-site.

Based on the analysis presented in this section, the Go1 Edu robot emerges as a good trade-off between the requirements of the proposed use case and price. Its impressive normalized work capacity of 646, which is five times higher than any IP-protected robot, indicates that the Go1 Edu can efficiently perform the scanning process. Moreover, when considering the maximum step size limitation, deploying a Go1 Edu robot on each level of the construction site can provide a potential solution for effective and timely monitoring.

B.4 Data acquisition process

Among the different current available legged robots presented, the case study was conducted with the *Go1 Edu* from the Unitree company.

Since it has four legs and 12-DoF, this robot can handle various terrains, even stairs up to 12 cm in height. It comes with a drive system, which enables a speed of up to 3.7 m/s (or 11.88 km/h). In addition, its power management system allows an operating time of up to 2.5 hours. Furthermore, the motors have a torque of 23.70 N m at the body/thighs

and 35.55 N m at the knees, allowing jumps or backflips. The Go 1 Edu has three nano processors, one Raspberry Pi, five RGB-D cameras, and four ultrasonic sensors. Moreover, it has a payload of up to 5 kg and comes with a research programming API.

As the main purpose is to leverage the capabilities of the robot for autonomous navigation and mapping, a mapping system that can be used independently of the robot was developed. In this way, the system can also be operated as a handheld or over any other robot.

B.4.1 Mapping system

Since having different sensor modalities contributes to achieving accurate pose estimation and, therefore, a more accurate map acquisition, a system that integrates LiDAR, Camera, and IMU sensors was developed. For the proposed robot-independent system, the ASRock 4x4 BOX-5800U mini personal computer equipped with 32 GB of RAM, together with two batteries XTPower XT-27000 DC-PA, were selected.

The selection of the PC prioritized two key criteria: high CPU performance and low power consumption. High CPU performance was essential for handling the intensive multi-threaded computations required by SOTA SLAM systems, while low power consumption was crucial to support efficient onboard processing.

Moreover, if the PC is not used for mapping (while connected to the system), it is also very suitable to be used as a standard workstation once connected to a display, keyboard, and mouse.

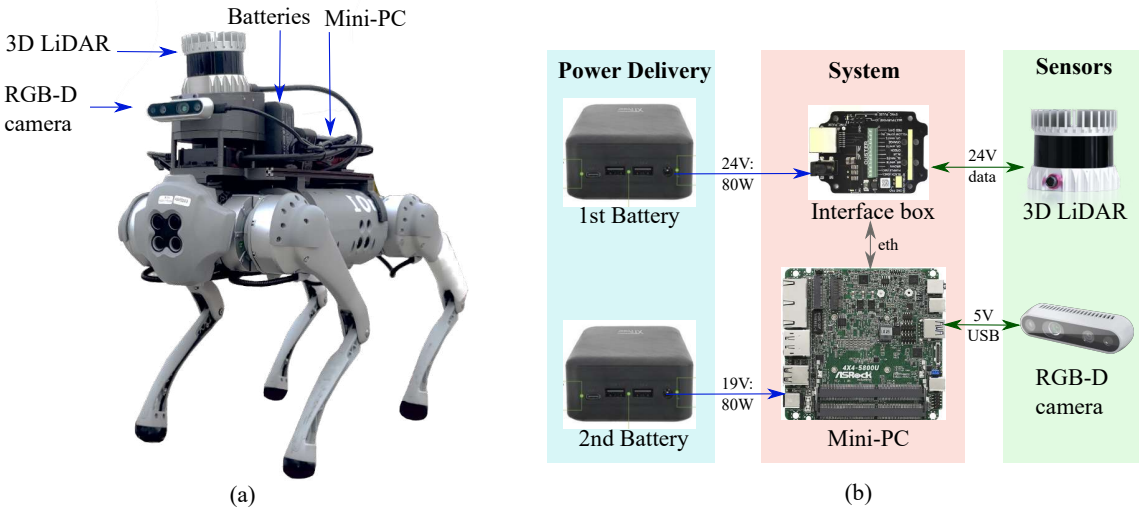
Figure B.2 shows the Go1 robot dog equipped with the developed mapping system[3] and the schematic connection between the different components.

B.4.2 Mounting System

The mounting system for attaching the sensor to the robot was designed with three primary requirements in mind. First, it needed to maintain a low center of gravity to ensure the robot's stability was not compromised. Second, the design aimed to minimize material usage, keeping the system lightweight (under 5 kg) and portable for both the robot and

[3]The 3D parts of this system are accessible here (M. A. Vega-Torres & Borrmann, 2024)

Figure B.2: Developed mobile mapping system. (a) The system is placed over the Go1 robot, with the help of the custom-designed mounting system which allows the montage on any robot with a flat surface and also allows the usage of the system as a handheld; (b) corresponding connection and data transfer diagram of the mapping system



handheld operation. Third, the rear section of the robot's loin was intentionally left unobstructed to ensure access to the robot's plugs, which can be utilized in an extended version of the system, for example, for autonomous navigation.

The system consists of twelve custom-designed 3D-printed components, two metal Maker-Beams, and various screws and inserts. Its modular design not only facilitates easy maintenance but also allows for quick replacement of individual parts if they become damaged, enhancing the system's durability and flexibility.

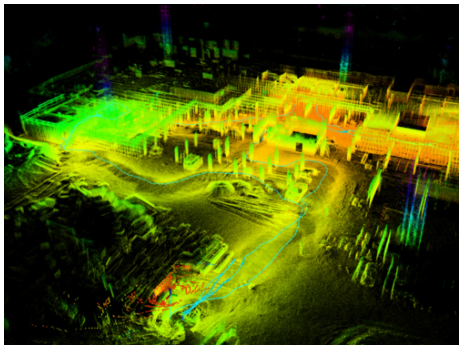
Moreover, the shape of the different parts considers the system's possible usage above other robots (with a flat surface) or as a handheld system. To use it as a handheld system, one can easily separate the LiDAR and camera and add them to a handle, and the mini-PC and battery could be placed on a backpack.

B.4.3 Acquisition process

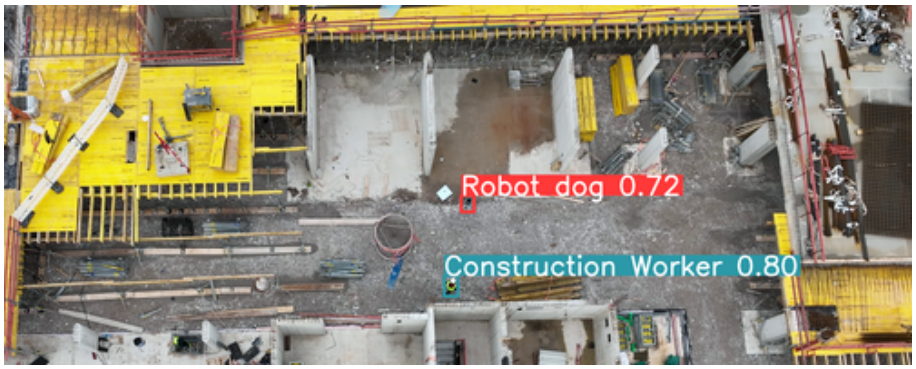
For the data acquisition process, the next steps were followed:

1. All the sensors and mini-PC are connected to the power delivery;
2. A remote connection with the mini-PC through a remote desktop application is established;

(a) Robots view using object detection to understand the environment.



(b) Point cloud reconstructed from the measurements of the sensors over the robot with an SLAM system.



(c) Bird-view of the robot and construction worker detected by the UAV's camera.

Figure B.3: Real-world on-site experiments with the robot dog.

3. The Software Development Kit (SDK) of each sensor is launched using nodes of the ROS;

4. A synchronization process between the images and the LiDAR data allows the recording of LiDAR scans and camera images with the same time stamps at 10 Hz of frequency.

5. A SLAM system, specifically an enhanced version of FAST-LIO (W. Xu et al., 2022) with loop closure capabilities, is leveraged to create 3D maps of the environment in real-time.

The data acquisition was conducted on a building construction site within the Munich area, covering a total area of 12.500 square meters. Figure B.3b illustrates the resulting 3D point cloud.

The acquired data consist of sequential images, LiDAR scans, and IMU measurements.

While total autonomy is still in development, a BIM model can be leveraged to navigate the robot autonomously in controlled environments using the ROS navigation stack as explained in (M. A. Vega-Torres, Braun, & Borrmann, 2022).

B.4.4 Analysis of acquired data

Once the 3D data is acquired, it can subsequently be aligned, corrected, and analyzed with the support of a BIM model, as explained in (M. A. Vega-Torres, Braun, & Borrmann, 2024b; M. A. Vega-Torres, Ribic, et al., 2024; M. A. Vega-Torres et al., 2023). After this process, all the data, even if they were acquired at different time stamps, should be in the same coordinate system. This means that the camera images, as well as the LiDAR scans, should have known poses aligned with the BIM model. Further, automatic semantic enrichment of 3D point cloud is possible with the method proposed here (M. A. Vega-Torres, Braun, Noichl, et al., 2022). This method would allow the detection of cranes, scaffolding, and formwork, which are elements that are very often present on construction sites.

On the other hand, the gained imaged data can be processed further by object detection pipelines and linked to other components in the construction environment, as explained more in detail here (Pfitzner et al., 2023). An example image from the robot's view with a detected crane is shown in figure B.3a. In this case, the robot dog extends SOTA monitoring methods for indoor areas, which are not covered by crane cameras or UAVs (Collins et al., 2022). Considering the unsolved imperfections of current digital twins aiming to cover the entire construction site (Schlenger et al., 2023), the robot dog presents a promising data acquisition extension.

B.5 Discussion

In general, to bring robot dogs, specifically the Go1 robot to construction sites, the following challenges should be overcome. For manual operation, the reliability of the Go1 robot dog was demonstrated to be sufficient in general. However, when aiming at autonomous navigation, small problems appeared specifically when launching software updates. Even

though full customer support is provided, the issues proved to be time-consuming, indicating the prototypical state of the robot.

On the construction site, many moving objects and a dynamically changing environment are encountered. These complex conditions lead to inaccuracies in the mapping process, which in turn can cause issues for the robot's autonomous localization and navigation. Furthermore, due to the presence of high stairway steps or significant slopes, some areas were not reachable for the robot at all and required carrying the robot. Nonetheless, in these cases, the lighter weight of the Go1 compared to other robot dogs proved to be advantageous.

As most of the construction site is outside, the robot dog has to face diverse temperatures, dust, and rain conditions. Unfortunately, the Go1 lacks adequate resistance to these conditions, thereby impeding the ability to conduct further experiments in harsh weather conditions.

The battery capacity of the robot dog showed a noteworthy limitation. After around only 30 minutes of exhaustive use, the robot dog's battery was almost empty. Currently, the Go1 cannot be charged automatically; this requires manual effort. Specifically, at a large construction site like the presented one, comprehensive autonomous scanning can become an issue, considering the limited battery capacity.

At the current state of development of robot dogs, semi-autonomous employment of a quadruped robot is feasible when fulfilling the following requirements:

1. Comprehensive and reliable 3D BIM models or 3D maps must be available to ensure autonomous localization and navigation.

2. Major changes to the construction site must continuously be updated on the 3D map.

3. The deployment of the robot dog should be handled level-wise, avoiding as many barriers as possible.

4. Precise and fast path planning algorithms are required to allow the autonomous navigation of the robot.

5. The manufacturers should significantly improve the battery capacity, and a method for autonomous charging should be employed.

B.6 Conclusion

This study explored the practicality of employing the currently available robot dogs in construction sites, with a specific emphasis on their effectiveness in enabling data acquisition.

In addition, a real-world experiment on a large-scale construction site using a quadruped robot equipped with a self-developed mapping system was conducted.

It is possible to conclude that robot dogs are suitable for scanning construction sites on a frequent basis, specifically indoor environments. Robot dogs can extend current monitoring solutions by providing valuable semantic and geometric data, facilitating the creation of a digital twin. However, the following limitations must be addressed prior to a feasible deployment: their limited battery capacity, their lack of adaptability to dynamic and harsh environments, and their prototypical condition. It is arguable that the deployment of multiple robot dogs can overcome some of the current limitations. In summary, this contribution demonstrates that robot dogs can be a valuable tool for monitoring complex construction environments in the future, specifically when technical improvements diminish their limitations.