Methodologies for Holistic and Objective SLAM Benchmarking and Mapping Accuracy Enhancement

Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy, March 2025.

Shengshu Liu

20200200

Supervised by

Dr. Xin Dong

Prof. Dragos Axinte

Department of Mechanical, Materials and Manufacturing Engineering

University Of Nottingham

Signature ___________________________

Date _________ / __________ / _________

Abstract

Simultaneous Localization and Mapping (SLAM), commonly referred to as concurrent mapping and localization, is a key area of study in robotics. It focuses on the dual task of building a map of an environment while simultaneously determining the robot location within it—a process that blends mapping and localization. The accuracy of localization and mapping in a SLAM system is typically measured using a SLAM benchmark. SLAM benchmarking plays a pivotal role in advancing the field by providing a common ground for performance evaluation, fostering collaboration, and promoting the development of reliable and efficient SLAM algorithms with real-world applicability. It serves as a foundation for driving progress and innovation in the broader field of robotics and computer vision.

A key challenge in SLAM benchmarking is ensuring a holistic and objective evaluation of SLAM system performance, and the goal is to achieve unbiased assessment results that accurately reflect a SLAM system’s true overall capabilities. To achieve such goal, a SLAM benchmark should assess localization and mapping together as a whole within a unified global framework. Despite this, most existing studies focus solely on evaluating localization performance while neglecting mapping accuracy, primarily due to the difficulty of obtaining a high-precision environment map which can serve as a reliable ground truth reference. Even among studies that assess both aspects, localization and mapping are often evaluated separately as unrelated tasks, leading to potentially biased assessment results. Thus, the development of a holistic and objective SLAM benchmark that produces unbiased performance measures of both localization and mapping is necessary.

The challenge of acquiring high-precision maps extends beyond SLAM benchmarking and affects various robotics applications, including navigation, path planning, and obstacle avoidance, where precise mapping is critical. Current methods often rely on complex hardware setups and labour-intensive manual processes, making high-precision map generation both costly and time-consuming. This limitation has led to an additional focus in this PhD research: devising a method for enhancing mapping accuracy with the potential to generate high-precision maps more efficiently. These maps could serve as credible ground truth references for SLAM benchmarking while also being applicable to broader robotic applications. The ultimate goal is to achieve this in a simpler, more resource-efficient, and scalable manner.

This PhD project introduces innovative approaches to tackle the previously mentioned dual challenges of creating a holistic and objective SLAM benchmark while enhancing mapping accuracy. The proposed novel SLAM benchmarking method transforms all localization and mapping data into a unified global coordinate frame, where localization and mapping errors are systematically measured. Its holism lies in evaluating these two components together as a whole rather than separately, while its objectivity stems from recognizing their interdependence by maintaining the original spatial relationships among all reference frames throughout the transformation process. By leveraging the benchmark results as feedback, mapping accuracy is improved through an optimization process that minimizes localization errors. This is achieved by first optimizing the alignment between the estimated trajectory and the ground truth trajectory and then applying the resulting transformation to the estimated map to enhance its accuracy.

The optimization that minimizes the localization error is achieved through a newly proposed point cloud registration technique called Centre Point Registration-Iterative Closest Point (CPR-ICP). This enhanced variant of the Iterative Closest Point (ICP) algorithm begins by aligning two point clouds using their centroids and least-square planes, followed by the classic ICP method to further reduce discrepancies between them.

The proposed methodologies were rigorously validated through both simulation-based and real-world experiments. Results from both types of experiments demonstrated the effectiveness of the proposed SLAM benchmarking framework in accurately reflecting a SLAM system's true performance. Furthermore, the experiments confirmed the feasibility of using benchmark feedback to improve mapping accuracy. Statistical analyses consistently showed that the CPR-ICP method outperformed the classic ICP approach in enhancing mapping accuracy. Additionally, the results also revealed a correlation between the performances of both methods in improving mapping accuracy and the size of the scene.

Therefore, it can be concluded that the proposed SLAM benchmarking method surpasses existing approaches in accurately representing the genuine overall performance of a SLAM system. Additionally, the proposed mapping accuracy enhancement method offers an efficient way to generate high-precision maps that have the potential to serve as reliable ground truth references for SLAM benchmarking and be effectively utilized in various robotic applications.

Publications

The research in this thesis has contributed in part or full for the following publications:

[1]. Liu, S., Lei, Y., & Dong, X. (2022, July). Evaluation and Comparison of Gmapping and Karto SLAM Systems. In *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)* (pp. 295-300). IEEE.

[2]. Liu, S., Sun, E., & Dong, X. (2024). SLAMB&MAI: a holistic methodology for SLAM benchmark and map accuracy improvement. Robotica, 42(4), 1039-1054.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my family for their unwavering support, encouragement, and generosity in sponsoring my PhD journey. Their belief in me has been the foundation of my perseverance and success.

To my wonderful wife, I cannot thank you enough for standing by my side through every challenge and triumph. Your patience, love, and encouragement have been my greatest source of strength. Most importantly, you have given me the most precious gift I could ever receive—our son—who has brought immeasurable joy and inspiration into my life during this PhD journey.

I am immensely grateful to my first supervisor, Dr. Xin Dong, for his invaluable guidance, insightful advice, and continuous support throughout my research. His expertise and mentorship have played a crucial role in shaping this work. I would also like to sincerely thank my second supervisor, Prof. Dragos Axinte, for his wisdom, constructive feedback, and encouragement, which have been instrumental in refining my research and expanding my academic perspective.

Furthermore, I would like to extend my appreciation to the post-doctoral researchers, technicians, and fellow students from the UTC. Their collaboration, technical assistance, and shared discussions have enriched my research experience and made this journey even more fulfilling.

Finally, I would like to thank everyone who has supported me, directly or indirectly, throughout this endeavour. This thesis is the result of collective effort, and I am truly grateful to all who have contributed to it.

Table of Contents

List of Figures

List of Tables

List of Symbols

N	Scalar quantity
$\boldsymbol{R}$	Rotation matrix
$\boldsymbol{t}$	Translation vector
$\boldsymbol{T}$	Transformation matrix
$\boldsymbol{O}$	Coordinate frame
$\boldsymbol{I_m}$	$m \times m$ identity matrix
$\mathbf{1}$	Vector $[1, \dots, 1]^T$
$\boldsymbol{Sc}$	SLAM camera frame
$\boldsymbol{Sca}$	SLAM Camera Attachment Frame
$\boldsymbol{FSc}$	SLAM camera frame $\boldsymbol{Sc}$ of the first image in the captured sequence
$\boldsymbol{G}$	Global frame
$\boldsymbol{P}$	Estimated trajectory
$\boldsymbol{p_i}$	The i-th point in the estimated trajectory $\boldsymbol{P}$
$\overline{\boldsymbol{p}}$	Centroid of the Estimated trajectory $\boldsymbol{P}$
$\boldsymbol{P'}$	Normalization of the estimated trajectory $\boldsymbol{P}$
$\boldsymbol{p_i}'$	The i-th point in $\boldsymbol{P'}$
$\overline{\boldsymbol{p}'}$	Centroid of $\boldsymbol{P'}$ which is $[0\ 0\ 0]^T$
$\boldsymbol{P}^*$	Ground truth trajectory
$\boldsymbol{p_i^*}$	The i-th point in the ground truth trajectory $\boldsymbol{P}^*$
$\boldsymbol{Q}$	Estimated map
$\boldsymbol{q_i}$	The i-th point in the estimated map $\boldsymbol{Q}$
$\widehat{\boldsymbol{Q}}$	Refined estimated map with improved accuracy
$\boldsymbol{Q}^*$	Ground truth map
$\boldsymbol{q_i^*}$	The i-th element (point or mesh surface) in the ground truth map $\boldsymbol{Q}^*$
$\boldsymbol{\hat{u}_1}$	Unit normal vector of the first reference plane
$\boldsymbol{\hat{u}_2}$	Unit normal vector of the second reference plane
$\boldsymbol{\hat{u}_3}$	Unit normal vector of the third reference plane

Chapter 1 Introduction

1.1 Background

Simultaneous Localization and Mapping (SLAM), also referred to as Concurrent Mapping and Localization (CML), is a fundamental challenge in robotics. It involves a robot building a map of an unfamiliar environment while simultaneously determining its own position within it—a dual process of mapping and localization. As illustrated in Figure 1-1, SLAM integrates these two processes seamlessly. Localization entails estimating the robot position using a pre-existing map with known landmark locations, whereas mapping involves creating a map by calculating the positions of landmarks based on the robot's known positions.

There are scenarios where SLAM might not be necessary. For instance, service robots working in restaurants might rely on pre-defined reference maps with known landmarks to navigate itself, or robots with GPS access could use it as a known landmark for localization. Under such circumstances where the landmark positions are already known, the robots can localize itself without needing to map the environment. However, SLAM is crucial for scenarios involving robots exploring unknown environments where no prior knowledge of map or landmarks are available. In these scenarios, both the map and the robot positions start as unknowns. As the robot explores the environment, it must estimate its positions and map the surroundings simultaneously, using natural or artificial landmarks. SLAM enables the robot to thrive in such challenging, unknown settings.

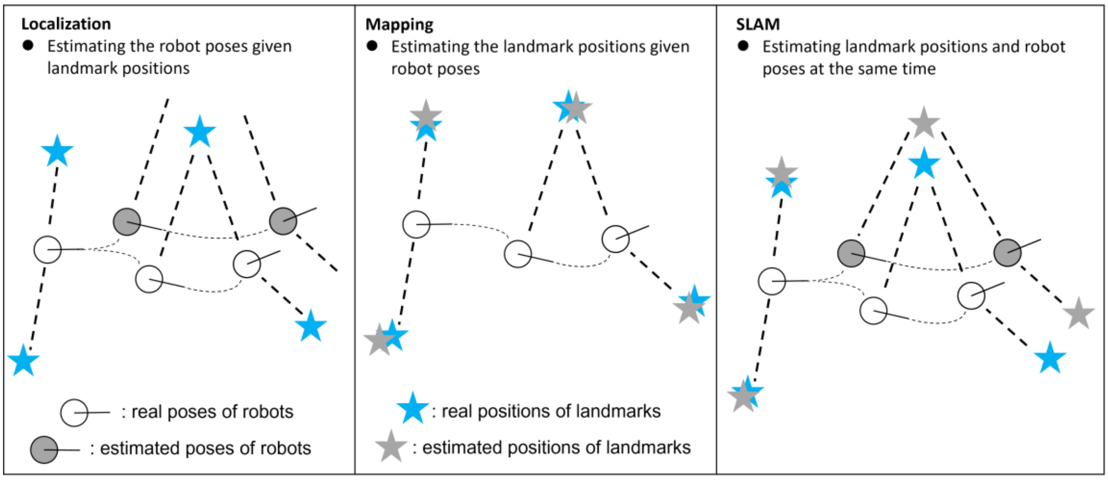

Localization
● Estimating the robot poses given landmark positions
Mapping
● Estimating the landmark positions given robot poses
SLAM
● Estimating landmark positions and robot poses at the same time
: real poses of robots
: estimated poses of robots
: real positions of landmarks
: estimated positions of landmarks

Figure 1-1. Illustration of the SLAM problem. Blue stars represent the actual landmark positions, white circles represent the robot's true poses, and grey stars and circles represent the estimated positions of the robot and landmarks, respectively.

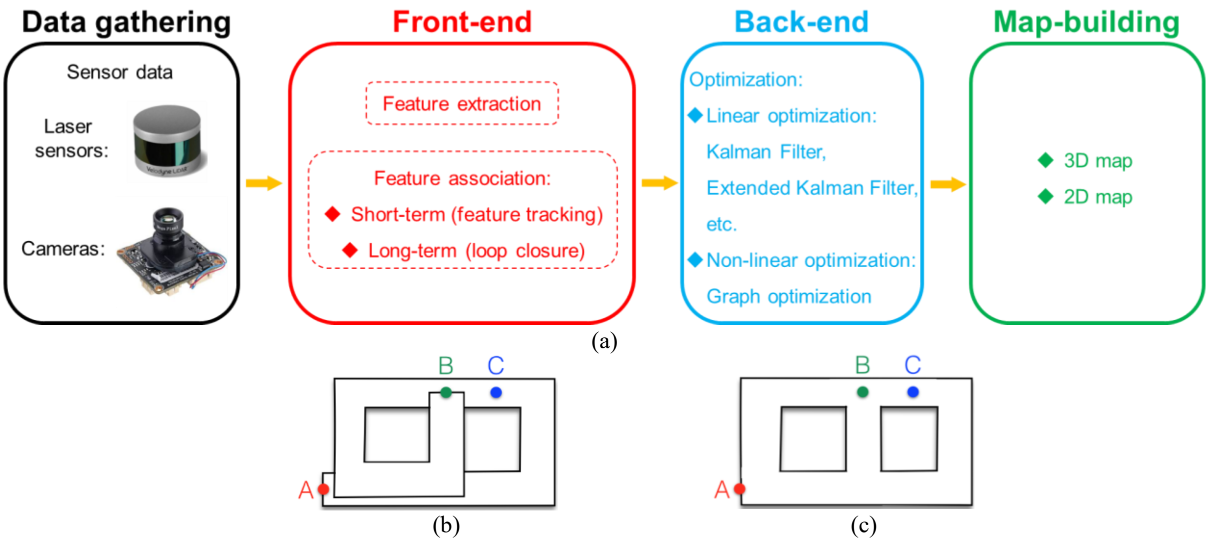


Figure 1-2. Illustration of the SLAM system. (a) Structure and workflow of the SLAM system. (b) Without loop closure, depending only on odometry, SLAM misinterprets the topology as an infinite corridor. (c) With loop-closure, the SLAM system is able to identify the "shortcut" and reconstruct the true topology [29].

Figure 1-2 (a) illustrates the structure and workflow of the SLAM system. As depicted, the system consists of four components: the sensor, the front-end, the back-end, and map-building. The sensors are responsible for collecting map data from the environment, and they could be divided into two major categories which are laser sensors and cameras. Accordingly, the SLAM systems can be categorized as visual SLAM (V-SLAM) system and laser-based SLAM system. The data captured by the sensor is fed into the front-end for feature extraction and association. Feature extraction first identifies and isolates key elements from the captured data, then feature association matches corresponding extracted features from different batches of the captured data to establish correlations between them.

For laser-based SLAM, the commonly employed feature extraction techniques fall into geometric, statistical, and deep learning-based methods. Geometric methods focus on extracting local and global geometric properties from 3D points, such as normals and curvature [1], eigenvalue-based descriptors [2], and bounding box [3], etc. Statistical methods analyse the point cloud's statistical characteristics, including distribution [4], point density [5], and spatial relationships [6], etc. Deep learning-based methods, which are gaining popularity, leverage pre-trained neural network models for feature extraction. Widely used models include point-based approaches like PointNet [7] and PointNet++ [8], voxel-based VoxelNet [9], graph-based GNNs [10], and transformer-based Point Transformer [11].

For visual SLAM, feature extraction techniques are divided into keypoint-based, dense feature-based, and learning-based methods. Keypoint-based methods detect distinct, stable image points across transformations, with popular options including the Harris Corner Detector [12], SIFT [13], SURF [14], ORB [15], and BRIEF [16]. Dense feature-based methods utilize all image pixels for direct SLAM instead of sparse keypoints. Examples include direct image alignment [17] and optical flow-based methods [18]. Learning-based methods, which leverage deep learning for feature extraction, include SuperPoint [19], LF-Net [20] and D2-Net [21].

Feature association techniques can be classified into short-term association and long-term association. Short-term association matches corresponding features across consecutive data batches, enabling sensor pose tracking and simultaneous triangulation of map points. In laser-based SLAM, the most widely used method is the ICP (Iterative Closest Point) algorithm, which aligns two point clouds by iteratively reducing the distance between corresponding points. This PhD research introduces a novel ICP variant: CPR-ICP. It is implemented alongside classic ICP for point cloud manipulation (see Section 4.4 for details). Beyond ICP, various feature association methods are employed in both laser-based and visual SLAM, including nearest-neighbour [22], k-nearest neighbours [23], maximum likelihood estimation [24], Joint Compatibility Branch and Bound (JCBB) [25], Multiple Hypothesis Tracking (MHT) [26], etc.

Long-term association, in contrast, matches newly captured features with much older ones, primarily to support loop closure detection with back-end optimization. The widely adopted technique for this in both laser-based and visual SLAM is Bag-of-Words (BoW) [27, 28]. Loop closure plays a critical role in SLAM by recognizing when the robot revisits a previously mapped place, even after a long trajectory drift. By doing so, it allows the system to establish global constraints that correct accumulated errors and enforce global consistency. This process is essential for reconstructing the true environment topology, as it prevents the map from diverging due to local odometry errors and drift. Loop-closure leverages back-end feedback to reveal the environment’s true topology. To illustrate this, consider a scenario where a SLAM system is mapping a corridor, as shown in Figure 1-2 (b) and (c). Without loop closure, Figure 1-2 (b) depicts the system misinterpreting the corridor as an infinite path while the robot loops. With loop closure, as shown in Figure 1-2 (c), the SLAM system is able to identify the "shortcut" and correctly reconstruct the environment's true topology.

The front-end of a SLAM system, also known as odometry, enables short-term pose estimation and mapping with reasonable accuracy. However, small errors accumulate over time, significantly degrading localization and mapping accuracy. To mitigate this drift, back-end optimization is required to constraint errors within acceptable levels, using either linear or non-linear methods.

Linear methods, such as the Kalman filter and its variants (e.g., Information filter, Iterated Kalman filter, Unscented Kalman filter, Sliding Window filter and Particle filter, etc.), rely on the Markov assumption. This assumes the state at time k only depends on the state at time $k - 1$, ignoring earlier states. Because of this assumption, linear methods are also known as incremental approaches, and they have two limitations: (1) It fails in scenarios where the current state depends on states before $k - 1$, such as in loop closure. (2) The first-order Taylor approximation often poorly captures nonlinear motion and observation models, leading to high error rates.

In contrast, non-linear methods, also known as batch methods, collect multiple measurements and optimize them simultaneously, offering superior accuracy and robustness. Prominent techniques include bundle adjustment (BA) and graph optimization. Bundle adjustment refines map point positions and sensor poses by minimizing errors, yielding optimal results. However, continuously optimizing all map points throughout the SLAM process can be computationally intensive with diminishing results. Graph optimization, a streamlined version of BA, mitigates this by fixing converged map positions after initial refinements and using them as constraints to optimize sensor poses, saving both time and resources. Figure 1-3 shows a graphical illustration of pose graph. As depicted, in a pose graph, the converged positions of landmarks and map positions are treated as constraints linking robot poses, which streamlines the optimization process.

Map-building is the final stage of the SLAM workflow, as illustrated in Figure 1-2. The two most commonly used map types are the point cloud map [30] and occupancy grid map [31, 32]. As depicted in Figure 1-4 (a), the point cloud map consists of 3D coordinate points representing sensor measurements of the object's external surface. For SLAM benchmarking, the point cloud is compared against other point clouds or 3D models in different formats using certain evaluation metrics, and this is discussed in depth in Chapter 4 where the innovative SLAM benchmarking methodology is presented.

Occupancy grid maps are also used widely in SLAM. The concept of occupancy grid is to represent the environment as an uniformly spaced grid where each grid cell is assigned a binary value indicating the presence or absence of obstacle at that location [24]. As shown in Figure 1-4 (b), the occupancy grid map consists of a set of voxel cubes corresponding to grid cells. Each grid cell has a probability indicating whether that location is filled with obstacles or empty, and this probability is constantly updated by occupancy grid algorithms after new observations. The occupancy grid map is particularly valuable for robot navigation and obstacle avoidance, guiding the robot toward cells more likely to be empty while steering clear of those probable to contain obstacles.

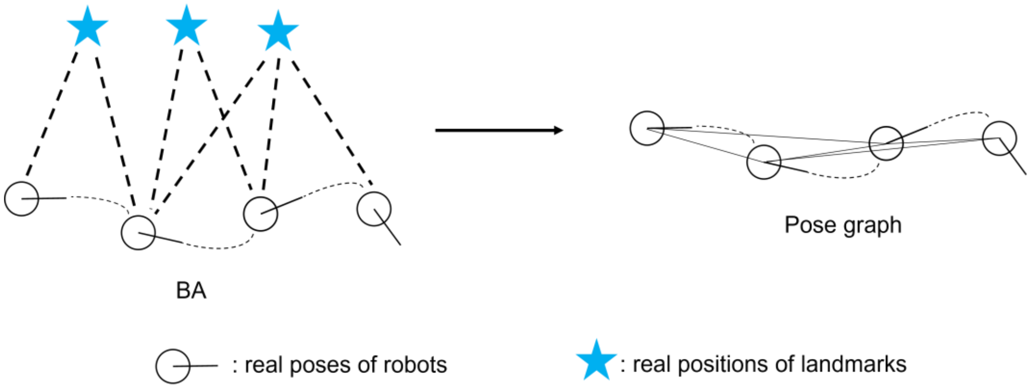
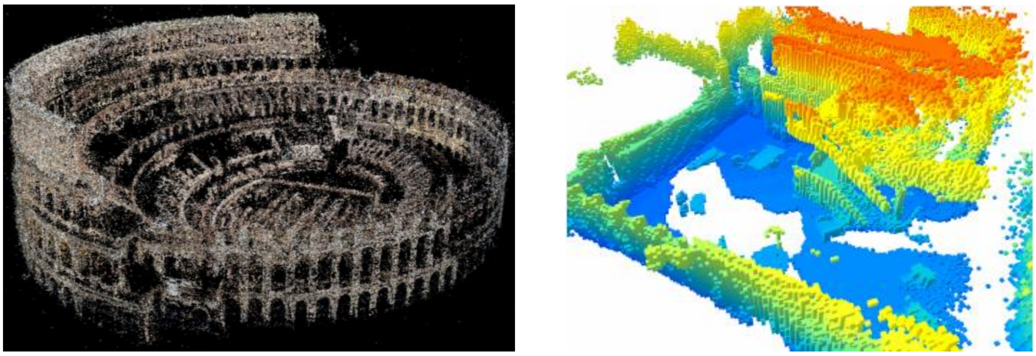


Figure 1-3. Illustration of BA and pose graph. Positions of landmarks are considered as constraints between poses in the pose graph, saving both computational time and resources.



(a) (b)

Figure 1-4. Examples of a point cloud map and an occupancy grid map. (a) Point cloud map of the Colosseo [33]. (b) Occupancy grid map of an environment built by a drone [34].

SLAM has undergone remarkable growth over recent decades, with numerous methods developed and applied across various domains, including space exploration [35-42], autonomous vehicle [43, 44], underwater [45, 46], UAVs [47, 48], service robots [49-62], and AR/VR applications [63-70]. For example, in field robotics, SLAM plays a crucial role in path planning and obstacle avoidance [35-62], while in wearable devices, it is widely used to enhance virtual and augmented reality experiences [63-70]. The rapid development of SLAM has stimulated the advancement of SLAM benchmarking. SLAM benchmarking is vital for advancing robotics and computer vision by providing a standardized platform for performance evaluation, encouraging collaboration, and fostering the creation of reliable, practical SLAM algorithms. It lays the groundwork for innovation and progress in these fields. However, a significant challenge is ensuring a holistic and objective assessment of SLAM system performance, with the aim of delivering unbiased assessment results that genuinely reflect a SLAM system's true overall capabilities.

Ideally, a SLAM benchmark should evaluate localization and mapping together as a whole within a unified global framework, rather than as separate entities. Yet, most existing studies emphasize localization accuracy, often overlooking mapping due to the challenge of obtaining a high-precision map as a reliable ground truth map for comparison. Even when both aspects are considered, they are typically assessed independently, leading to biased outcomes. This underscores the need to develop a holistic, objective benchmarking methodology that integrates localization and mapping to produce unbiased results.

Obtaining high-precision maps remains a persistent obstacle in SLAM benchmarking and many other robotic applications, as current methods rely on costly hardware and labour-intensive processes. This limitation has prompted an additional key focus in this PhD research: developing a method to improve mapping accuracy with potential capability to generate high-precision maps more efficiently. These maps can serve as reliable ground truth references for SLAM benchmarking while also being applicable to a wide range of robotic applications. The overarching goal is to achieve this in a more streamlined, resource-efficient, and scalable way.

While this research focuses on SLAM benchmarking, it is worth noting that another related concept is 3D reconstruction benchmarking. Although both aim to evaluate spatial accuracy, they differ in important ways. SLAM benchmarking assesses a system's ability to perform simultaneous localization and mapping in real-time and dynamic environments, typically without prior knowledge of sensor trajectories. In contrast, 3D reconstruction benchmarking often focuses on the geometric fidelity of static scene reconstruction, assuming known or post-processed camera poses. This distinction highlights the unique challenges of SLAM benchmarking, particularly in evaluating the coupled performance of localization and mapping. A more detailed discussion on 3D reconstruction benchmarking and its relation to SLAM benchmarking is provided in Section 2.2 of the literature review.

1.2 Problem definition

As noted in the previous section, existing SLAM benchmarks face two key limitations:

Limitation 1: Lack of holism and objectivity. The first limitation is a lack of holism and objectivity in current SLAM benchmarks. In this thesis, a holistic and objective SLAM benchmark is defined as one that jointly evaluates localization and mapping in an integrated manner, rather than in isolation, and provides an unbiased measure of performance by preserving the intrinsic correlations between localization and mapping. Existing SLAM benchmarks either focus solely on the assessment of localization while neglecting mapping—due to the difficulty of obtaining a reliable ground truth map—or evaluate localization and mapping separately rather than as an integrated process, compromising a holistic evaluation. Here, benchmarking refers to the systematic evaluation of SLAM performance against reference data, while ground truth (GT) denotes the most accurate available representation of the robot trajectory or environment, used as a reference for such evaluation.

Most existing mapping benchmark studies focus on evaluating the 3D reconstruction of small to medium-sized objects using available CAD models as ground truth. In these studies, estimated maps are assessed as standalone entities. A typical approach [71-75] involves aligning the estimated map with the ground truth map using a 3D shape registration technique [76-83], followed by error calculation based on a specific error metric [84-87].However, this method has a critical flaw: Localization and mapping are inherently correlated, as they rely on each other during the SLAM process, as illustrated in Figure 1-1. Therefore, an objective SLAM benchmark must account for these correlations to accurately reflect the SLAM system's true performance. This requires preserving the original pose relationship between the estimated and ground truth maps during evaluation, as it embodies the interplay between localization and mapping. By doing so, the resulting error calculation can provide an objective, unbiased measure of the SLAM system performance. Despite this, as noted earlier, most existing studies evaluate mapping in isolation, treating the estimated map as a standalone entity and using manual alignment techniques. This approach obfuscates the critical correlations, leading to biased evaluation results.

Limitation 2: Difficulty of obtaining high-precision ground truth. Another limitation in SLAM benchmarking and various robotic applications, as previously mentioned, is the difficulty of obtaining a high-precision map. Whether used as a reliable ground truth reference or for other purposes, generating such maps is a time-consuming and resource-intensive process. This limitation has sparked interest in developing a more efficient, less resource-intensive approach to generating decent maps that show promise to serve as ground truth references for SLAM benchmarking or be used for other robotic applications.

Achieving this goal depends on reducing mapping errors to align estimated maps more closely with ground truth, underscoring the need for holistic and objective SLAM benchmarks. Because only such SLAM benchmarks are able to produce unbiased error calculations suitable for improving mapping accuracy and potentially generating high-precision maps—unlike the biased error calculations from existing benchmarks, which fall short for this purpose.

1.3 Aim and objectives

The aim of this research is to address the two key limitations of existing SLAM benchmarks—namely, the lack of holism and objectivity, and the difficulty of obtaining high-precision ground truth maps. To this end, the research develops a holistic and objective SLAM benchmarking methodology and establishes a feedback mechanism that leverages the benchmark results to enhance mapping accuracy. This unified framework seeks not only to provide unbiased performance evaluation of SLAM systems but also to enable the generation of more accurate maps, potentially serving as reliable ground truth references for benchmarking and other robotic applications.

The objectives for attaining the aim are as follows:

1. Identify the limitations of existing SLAM benchmarks
    - Critically review current benchmarking approaches and highlight the lack of holism and objectivity, as well as the difficulty of acquiring high-precision ground truth maps.
2. Develop a holistic and objective SLAM benchmarking framework

- Design a methodology that evaluates localization and mapping jointly within a unified global coordinate frame, ensuring that their intrinsic correlations are preserved for unbiased performance assessment.

3. Leverage benchmark results as feedback for mapping accuracy enhancement
    - Establish the principle that unbiased localization error obtained from benchmarking can be systematically exploited to improve mapping accuracy.
4. Propose a novel point cloud registration technique (CPR-ICP)
    - Introduce and validate an enhanced variant of ICP that provides robust alignment, serving as the core algorithm for the feedback-based mapping improvement process.
5. Validate the proposed unified methodology through experiments
    - Conduct both simulation-based and real-world experiments to demonstrate the effectiveness of the benchmarking framework and its feedback-driven mapping accuracy enhancement in diverse environments.

1.4 Thesis structure

Chapter 2 presents the literature review covering SLAM benchmarking and a related field—3D reconstruction benchmarking. The SLAM benchmarking review is structured around three key components: sensory data, ground truth, and evaluation metrics. Meanwhile, the review of 3D reconstruction benchmarking focuses on the commonly used techniques: two-view stereo, multi-view stereo, and Structure-from-Motion (SfM). While 3D reconstruction shares similarities with SLAM, it prioritizes mapping—often termed structure reconstruction—over localization. As a result, insights from 3D reconstruction benchmarking can contribute to the development of mapping benchmarks in SLAM. The chapter concludes by identifying research gaps and challenges in the field.

Chapter 3 lays out the mathematical foundation for the proposed methodologies of SLAM benchmarking and mapping accuracy enhancement, exploring the core principles underlying the Euclidean transformation and the Iterative Closest Point (ICP) algorithm in detail.

Chapter 4 presents the unified methodology for holistic and objective SLAM benchmarking and its extension to mapping accuracy enhancement. It begins by identifying the two key limitations of existing

SLAM benchmarks: the lack of holism and objectivity, and the difficulty of obtaining high-precision ground truth maps. To address these limitations, an innovative benchmarking approach is proposed, which transforms all estimated data into a unified global coordinate frame to preserve spatial relationships and provide unbiased performance evaluation. Building upon this framework, the chapter then introduces the novel concept of leveraging benchmark results as feedback to improve mapping accuracy. In particular, it highlights the correlation between localization and mapping errors and proposes Centre Point Registration-ICP (CPR-ICP), a variant of the classic ICP algorithm, as a means of reducing residual alignment errors between point clouds. The chapter provides a detailed explanation of CPR-ICP and its role in enabling benchmark-driven mapping accuracy enhancement.

Chapter 5 validates the proposed methodologies for SLAM benchmarking and mapping accuracy enhancement through simulation-based experiments. It begins by introducing the software setup used for the simulations. Next, the proposed methods, along with the classic ICP method, are tested across multiple trials in various environment settings to assess their effectiveness. The chapter then presents the experimental results and analysis, comparing the performance of the proposed CPR-ICP with the classic ICP in enhancing mapping accuracy. Additionally, it explores the correlations between performance and the geometries of the scenes and trajectories involved.

Chapter 6 validates the proposed SLAM benchmarking and mapping accuracy enhancement method through real-world experiments. It starts by detailing the hardware setup, focusing on the instruments used to capture ground truth and the environments selected for mapping. The proposed methods and the classic ICP approach are then tested across multiple trials in diverse settings to evaluate their effectiveness further. Finally, the chapter analyses the experimental results, comparing the mapping accuracy performance of the proposed CPR-ICP against the classic ICP, while also examining correlations with the geometries of the scenes and trajectories involved.

Chapter 7 concludes the thesis by summarizing this PhD research's motivations and key contributions, while highlighting potential future work to advance the field further.

1.5 Highlights and contributions

- A holistic and objective SLAM benchmarking method that unbiasedly reflecting the genuine overall performance of a SLAM system is proposed. The proposed method distinguishes itself from existing approaches in that it evaluates both localization and mapping together as an integrated whole instead of separately as two unrelated tasks, ensuring objective and unbiased measures of the SLAM system performance, offering a more accurate representation of a SLAM system's true capabilities.

  This contribution is developed in Chapter 4, where the innovative benchmarking framework is presented and explained in detail.

- A novel concept of mapping accuracy enhancement is proposed. The primary motivation behind this is to generate high-precision maps that can potentially be used in SLAM benchmarking as ground truth references or for other applications, such as robot navigation, path planning, and obstacle avoidance. Current methods for obtaining high-precision maps are often time-consuming and resource-intensive. In contrast, the proposed method leverages unbiased assessment results from the holistic and objective SLAM benchmark to improve mapping accuracy in a more efficient, scalable, and resource-effective manner.

  This contribution is also established in Chapter 4, where benchmark-driven feedback is presented as a natural extension of the proposed evaluation framework, and is further validated in Chapter 5 and Chapter 6 through simulation and real-world experiments.

- An innovative ICP variant CPR-ICP is proposed to more effectively reduce discrepancies between point clouds by enhancing the alignment between them. The classic ICP method struggles with alignment when there is no reliable initial feature matching between point clouds. The proposed CPR-ICP addresses this issue by first pre-aligning two point clouds by their centroids and least-square planes, providing a more accurate initial alignment, then further refining this alignment with the classic ICP algorithm to reduce discrepancies.

  This contribution is introduced in Chapter 4 as the core algorithm enabling the mapping accuracy enhancement concept, and its effectiveness is validated against classic ICP in Chapter 5 and Chapter 6.

Chapter 2 Literature Review

This chapter primarily reviews existing research on SLAM benchmarking in Section 2.1 . Additionally, it explores a related field—3D reconstruction benchmarking—in Section 2.2 . Unlike SLAM, which emphasizes simultaneous localization and mapping, 3D reconstruction focuses on generating a detailed three-dimensional model of an object or scene using images or sensor data. Its primary goal is to capture the geometry and appearance of the scene while excluding pose estimation. Due to these differences, the methodologies used for benchmarking 3D reconstruction differ from those of SLAM and are therefore reviewed separately. Following the benchmarking review, Section 2.3 identifies key research gaps, providing the motivation for developing a novel, holistic, and objective SLAM benchmarking approach.

2.1 Review of SLAM benchmarking

According to the RAWSEEDS project [88], a SLAM benchmarking system consists of two key components: the benchmark problem and the benchmark solution. The benchmark problem is defined by the sensory data and the evaluation metric, while the benchmark solution includes the SLAM algorithm, its output after processing the sensory data, and the performance rating assigned to the algorithm's output. This SLAM benchmarking framework can be visualized as a workflow, as depicted in Figure 2-1.

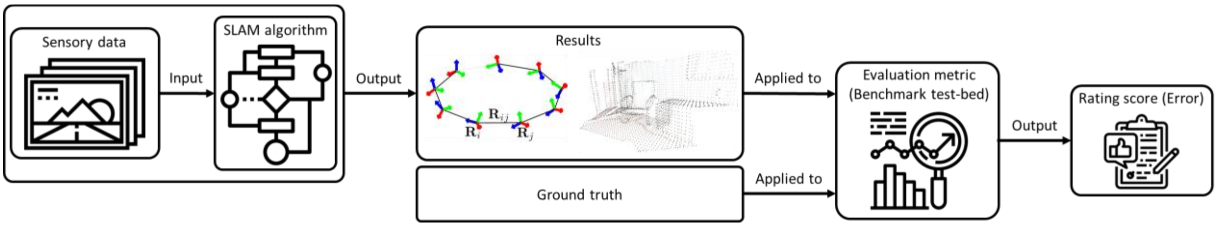


Figure 2-1. General workflow of the SLAM benchmark [89, 90].

As illustrated in the figure above, a typical SLAM benchmarking workflow begins with applying SLAM algorithms to sensory data to generate results. These results are then evaluated using specific metrics against the ground truth to assess the performance of the SLAM algorithms. Most existing

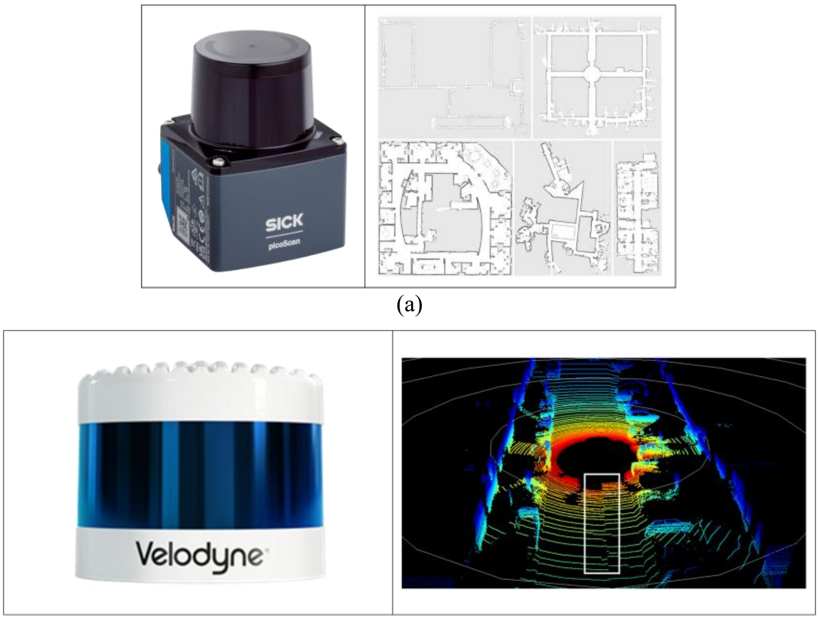
SLAM benchmarking studies primarily focus on defining the benchmark problem. The following are some widely recognized benchmarks commonly used in the field.

- TUM RGB-D [91]: Indoor, real RGB-D sequences with motion-capture ground truth; standard for dense RGB-D SLAM.
- EuRoc [92]: Indoor stereo+IMU from a micro-air-vehicle with aggressive motion and precise mocap/laser ground truth; VIO-oriented.
- New College [93]: Large outdoor campus traverses with many loop closures and appearance change; stresses place recognition.
- Malaga 2009 [94]: Outdoor urban driving (stereo); strong illumination/texture variation and long trajectories, with GPS/INS references.
- Ford Campus [95]: Outdoor campus driving with rich LiDAR + cameras + GPS/INS; suited to large-scale mapping.
- KITTI [96]: Outdoor city/highway driving with stereo/LiDAR and accurate GPS/INS ground truth; de-facto odometry benchmark.
- RAWSEEDS project [88]: Indoor/outdoor multi-sensor (stereo, IMU, encoders, laser) with careful calibration and reference trajectories.
- ICL-NUIM [97]: Synthetic photorealistic indoor RGB-D with perfect ground truth and controllable noise; ideal for ablation and simulation.

Existing SLAM benchmark problems primarily consist of three key components: sensory datasets, ground truth, and evaluation metrics. These three components are selected as the focus of this review because they are the fundamental building blocks that determine the validity and reliability of any SLAM benchmark. Sensory data defines the input and constraints of the problem, ground truth provides the necessary reference for objective performance evaluation, and evaluation metrics translate differences between estimation and reference into quantitative measures. Other aspects, such as algorithmic implementations or hardware platforms, are benchmark-specific and vary widely across studies, whereas these three components are common to all benchmarks and thus provide a consistent and comparable taxonomy for review. The following sections review SLAM benchmarks using this taxonomy, examining each component individually.

2.1.1 SLAM sensory data

SLAM techniques can be broadly categorized into LiDAR-based SLAM and vision-based SLAM. Accordingly, the sensory data input into SLAM algorithms can be either point clouds captured by LiDAR sensors or images recorded by cameras. LiDAR, which stands for Light Detection and Ranging, is a remote sensing technology that measures distances using Time-of-Flight (ToF) techniques. Due to its high precision in distance measurement and ability to capture fine details, LiDAR has been widely utilized in numerous studies [88, 96, 98-104]. For example, Kümmerle *et al.* [99] employed 2D LiDAR sensors to generate floor plans of indoor environments for testing their proposed SLAM benchmarking framework (Figure 2-2 (a)). In the widely recognized KITTI dataset [96], Geiger *et al.* used a two-axis, six-degree-of-freedom (6DoF) Velodyne 3D laser scanner to capture 3D point cloud maps of urban environments for object detection (Figure 2-2 (b)). Additionally, Wulf *et al.* [101] developed a 3D laser scanner by integrating multiple 2D laser scanners with servo drives to generate a sequence of 3D scans, which they used to test their proposed ground truth generation method.

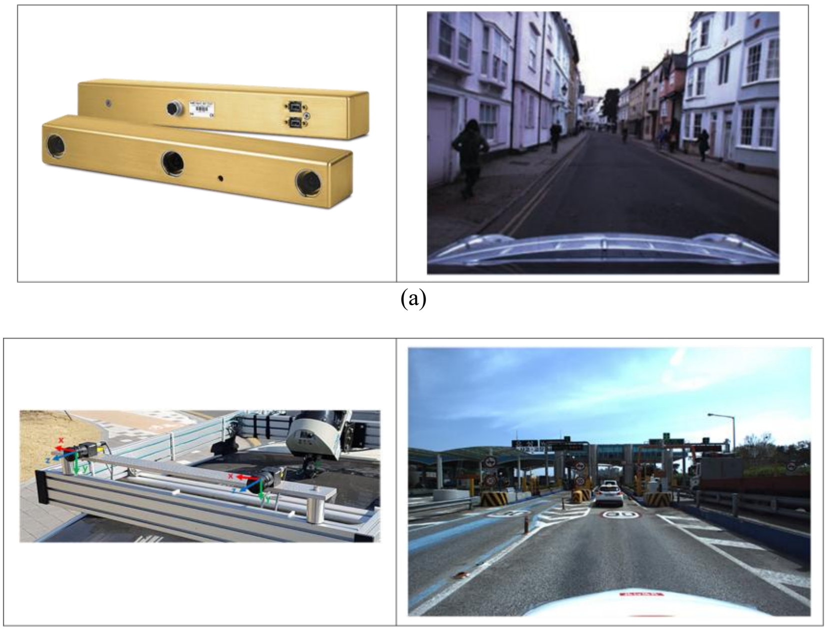


(a)

(b)

Figure 2-2. Examples of different LiDAR modalities used for data capture in SLAM benchmarks, along with their respective sensor outputs. (a) A 2D LiDAR sensor produced by SICK and its corresponding 2D map output [99, 105]. (b) Left: A Velodyne 3D LiDAR sensor used to collect the KITTI dataset [96]. Right: A single frame from the sequence of 3D point clouds captured by the Velodyne sensor in the KITTI dataset [106].

Although LiDAR provides superior accuracy and detailed environmental capture, its high cost—often thousands of dollars for multibeam LiDAR systems—limits its widespread use among researchers. In comparison, cameras are significantly less expensive and can provide richer visual information, despite typically offering lower measurement accuracy. With recent advancements in spatial computing technologies, cameras of various modalities (monocular, stereo, RGB-D, fisheye, etc.) have become increasingly popular. Stereo cameras, in particular, have been widely employed in numerous studies. For instance, the Oxford RobotCar dataset [102] by Maddern *et al.* utilized a Bumblebee XB3 wide-baseline stereo camera to capture imagery for visual odometry (Figure 2-3 (a)). Similarly, in the Complex Urban dataset [103], Jeong *et al.* formed a stereo camera setup using two FLIR monocular cameras to obtain visual data (Figure 2-3 (b)).



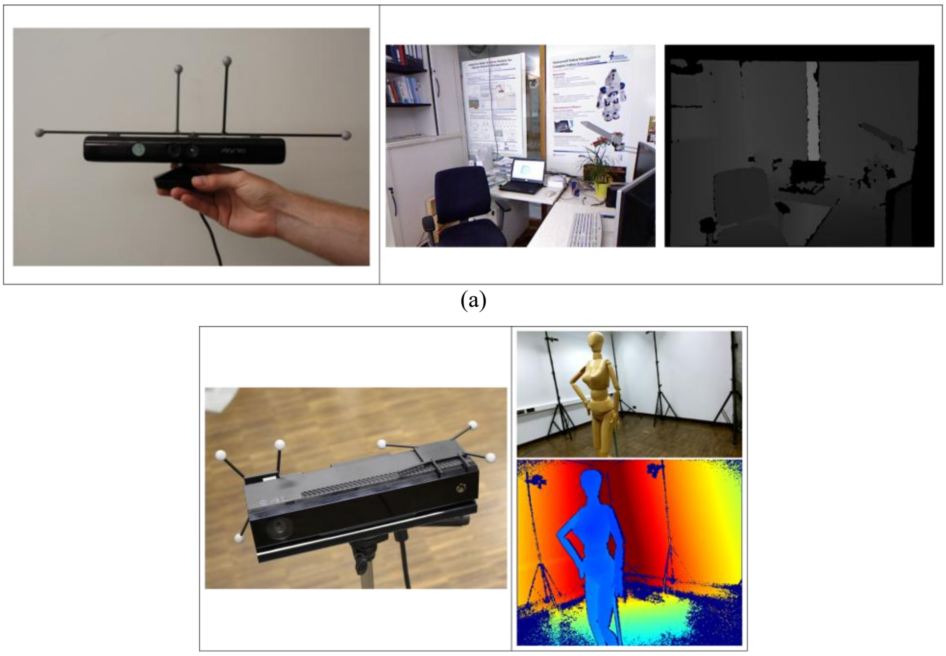
(a)

(b)

Figure 2-3. Examples of stereo cameras used for data acquisition in SLAM benchmarks, along with their respective sensor outputs. (a) A wide-baseline stereo camera manufactured by FLIR [107], and a sample stereo image from the Oxford RobotCar dataset [102]. (b) A stereo camera setup consisting of two FLIR monocular cameras, and its corresponding stereo image from the Complex Urban dataset [103].

RGB-D cameras are also widely used in many studies, with Microsoft's Kinect RGB-D camera being a particularly popular choice. In the TUM RGB-D SLAM dataset [91], Sturm *et al.* captured sequences of RGB and depth images using the Microsoft Kinect depth camera to evaluate RGB-D odometry (Figure 2-4 (a)). Similarly, in the CoRBS dataset [108], Wasenmüller *et al.* utilized the

Microsoft Kinect V2 depth camera to acquire RGB and depth image sequences for benchmarking RGB-D SLAM systems in terms of both odometry and mapping performance (Figure 2-4 (b)).



(a)

(b)

Figure 2-4. Examples of RGB-D cameras used for data collection in SLAM benchmarks, along with their respective sensor outputs. (a) Microsoft Kinect RGB-D camera with its corresponding RGB image and 8-bit grayscale depth image from the TUM RGB-D dataset [91]. (b) Microsoft Kinect v2 RGB-D camera and its corresponding RGB image and color-coded depth image from the CoRBS dataset [108].

In addition to using images captured by cameras in real-world environments as input sensory data, some studies also utilize synthetically generated images. For example, in the ICL-NUIM dataset [97], Handa *et al.* generated RGB-D camera sequences within synthetic environments using POVRay to benchmark RGB-D visual odometry, 3D reconstruction, and SLAM (Figure 2-5). Similarly, Funke *et al.* [109] developed a framework for evaluating visual SLAM systems using simulation-based approaches with rendered image sequences. Nardi *et al.* [110] introduced SLAMBench, a software framework based on the ICL-NUIM dataset [97], designed to quantitatively and comparably analyse trade-offs in performance, accuracy, and energy consumption in dense RGB-D SLAM systems.

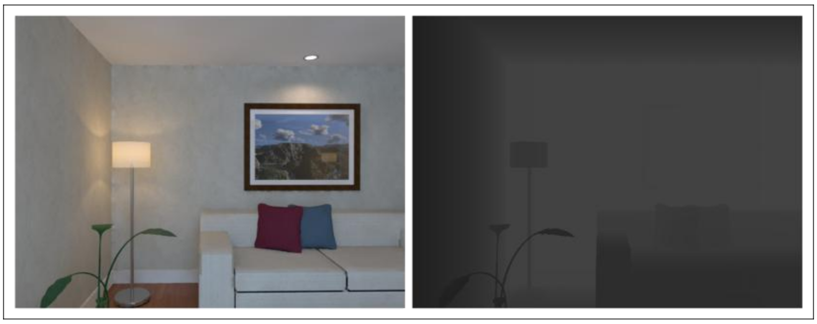


Figure 2-5. Synthetic scene generated using POVRay and its corresponding 8-bit grayscale depth image, taken from the ICL-NUIM dataset [97].

As mentioned earlier, cameras and LiDAR sensors each have distinct advantages and are complementary to one another. To leverage the strengths of both technologies, vision-LiDAR fusion has been widely adopted in various studies. Below is an example of a hardware setup used for vision-LiDAR fusion:

- New College dataset [93] (Figure 2-6)

In this setup, a wheeled robot serves as the mobile platform. It is equipped with a LadyBug panoramic camera mounted on top, a Bumblebee stereo camera at the front, and two SICK LMS 291-S14 2D LiDARs positioned on either side.

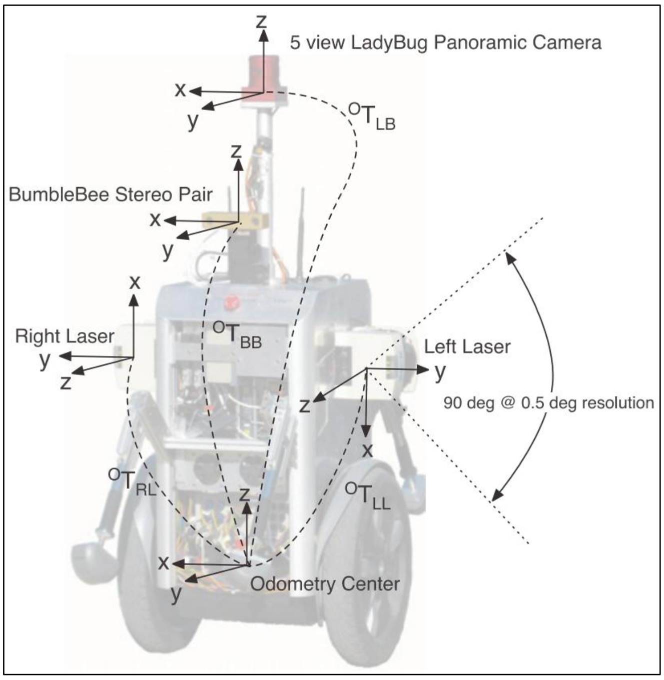


Figure 2-6. Hardware setup for vision-LiDAR fusion used in the New College dataset [93].

- Oxford RobotCar dataset [102] (Figure 2-7)

In this setup, the mobile platform is a Nissan LEAF electric car equipped with multiple vision and LiDAR sensors. A Point Grey Bumblebee XB3 (BBX3-13S2C-38) trinocular stereo camera is mounted at the front of the car roof, while two Point Grey Grasshopper2 (GS2-FW-14S5C-C) monocular cameras are positioned on either side at the rear of the roof. Another Point Grey Grasshopper2 (GS2-FW-14S5C-C) monocular camera is mounted at the back of the roof. For LiDAR sensing, a SICK LMS-151 2D LiDAR and a SICK LD-MRS 3D LiDAR are installed at the front of the car nose, with an additional SICK LMS-151 2D LiDAR positioned at the bottom of the trunk.

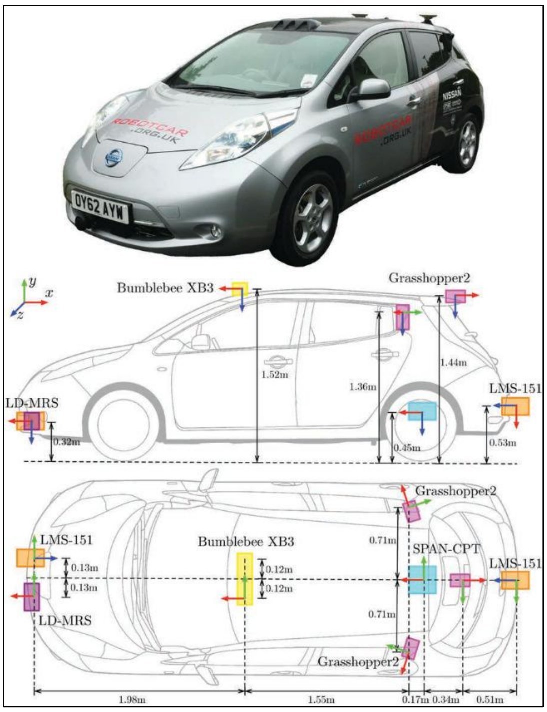


Figure 2-7. Hardware setup used for vision-LiDAR fusion in the Oxford RobotCar dataset [102].

- The RAWSEEDS project [88] (Figure 2-8)

In this setup, the mobile platform is a wheeled robot, similar to the one used in the New College dataset [93]. It is equipped with an omnidirectional camera mounted on top, a trinocular stereo camera, a HOKUYO laser range scanner, and a SICK laser range scanner at the front.

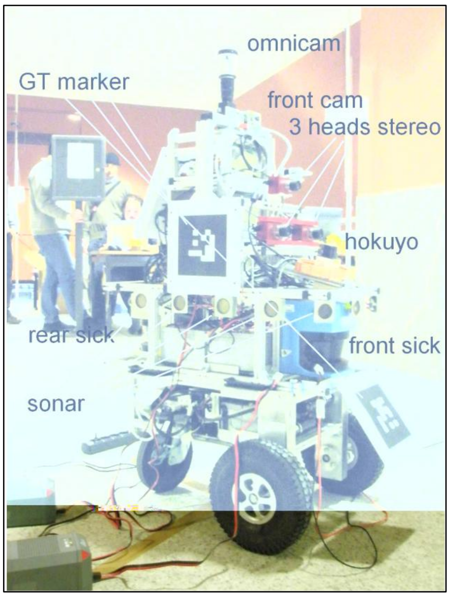


Figure 2-8. Hardware setup for vision-LiDAR fusion used in the RAWSEEDS project [88].

- Málaga dataset [94] (Figure 2-9)

In this setup, the mobile platform is a buggy-type electric vehicle equipped with multiple vision and LiDAR sensors. Two SICK LMS-220 laser rangefinders are mounted on either side at the rear of the vehicle's roof, while a SICK LMS-200 laser rangefinder is positioned at the front. Additionally, two AVT Marlin F-131C color cameras are installed at the front of the roof. A Hokuyo UTM-30LX scanning laser rangefinder is placed at the front, with another Hokuyo UTM-30LX scanning laser rangefinder mounted at the rear.

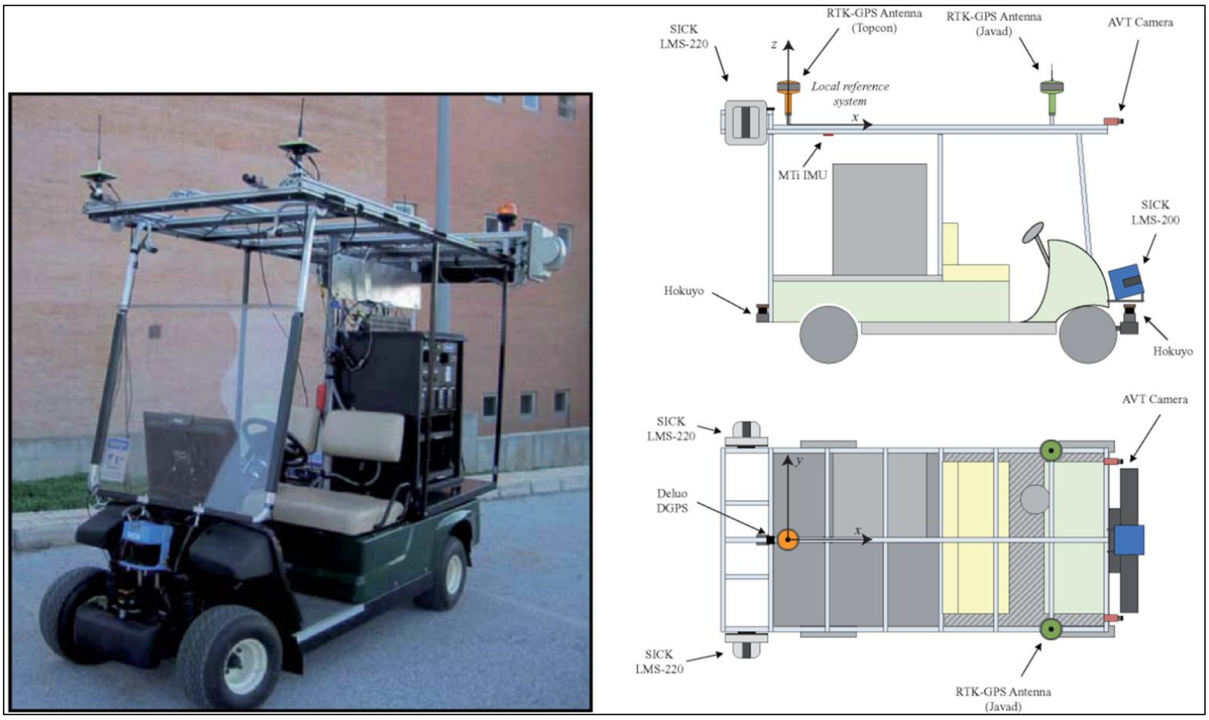


Figure 2-9. Hardware setup for vision-LiDAR fusion used in the Málaga dataset [94].

- Complex Urban dataset [103] (Figure 2-10)

In this setup, the mobile platform is a Toyota Prius equipped with multiple vision and LiDAR sensors. A stereo camera system, consisting of two FLIR FL3-U3-20E4C-C color cameras, is mounted at the front of the car roof. A SICK LMS-511 2D LiDAR is positioned in the middle of the roof, with another SICK LMS-511 2D LiDAR mounted at the rear. Additionally, two Velodyne VLP-16 3D LiDARs are installed on either side at the back of the roof.

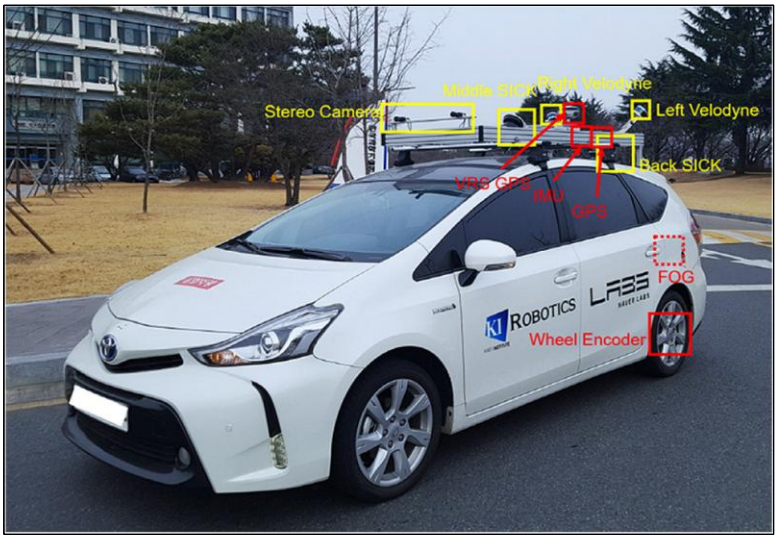


Figure 2-10. Hardware setup for vision-LiDAR fusion used in the Complex Urban dataset [103].

- KITTI dataset [111] (Figure 2-11)

In this setup, the mobile platform is a Volkswagen Passat station wagon equipped with multiple vision and LiDAR sensors. At the front of the car roof, it features two PointGrey Flea2 grayscale cameras (FL2-14S3MC) and two PointGrey Flea2 color cameras (FL2-14S3C-C). Additionally, a Velodyne HDL-64E 3D LiDAR is mounted in the center of the roof.

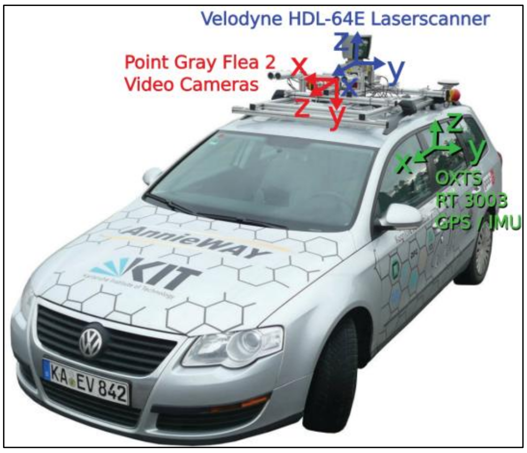


Figure 2-11. Hardware setup for vision-LiDAR fusion used in the KITTI dataset [111].

- Ford Campus dataset [95] (Figure 2-12)

In this setup, the mobile platform is a Ford F-250 pickup truck equipped with various LiDAR and vision sensors. Two Riegl LMS-Q120 2D laser scanners are mounted on either side at the front of the roof, while a Velodyne HDL-64E 3D LiDAR is positioned centrally at the front of the roof. Additionally, a Ladybug3 omnidirectional camera is installed in the middle of the roof.

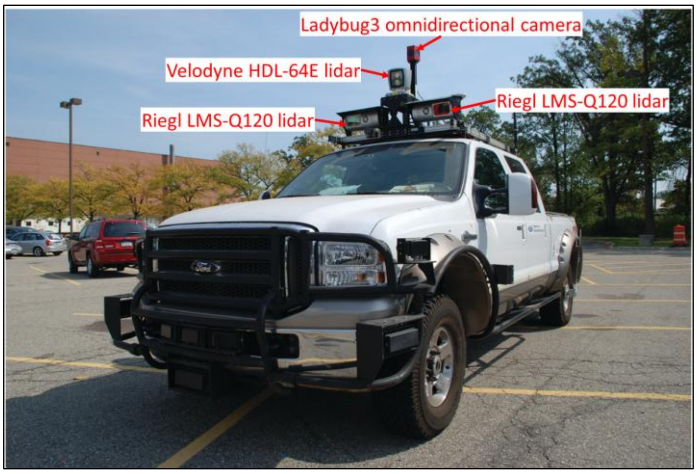


Figure 2-12. Hardware setup for vision-LiDAR fusion used in the Ford Campus dataset [95].

- Newer College dataset [104] (Figure 2-13)

This setup features a handheld vision-LiDAR fusion system, combining an Ouster OS1-64 3D LiDAR with an Intel RealSense D435i RGB-D camera for integrated sensing.

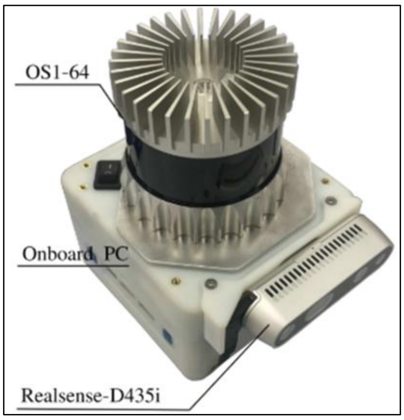


Figure 2-13. Hardware setup for vision-LiDAR fusion used in the Newer College dataset [104].

Camera-LiDAR fusion requires both spatial and temporal calibration to ensure accurate data alignment. Spatial calibration determines the Euclidean transformation between the coordinate frames of different cameras and LiDAR sensors. This process begins with an initial estimation of the Euclidean transformation, which is then refined using nonlinear optimization techniques. Spatial calibration can be performed using target-based, feature-based, or statistical approaches, with the target-based approach being the most commonly used. The target-based approach relies on fiducial objects that can be detected by both the camera and LiDAR to establish correspondences between their data, either manually or automatically. Among fiducial objects, planar checkerboards are the most widely used due to several advantages:

- High-contrast and repetitive pattern:

Checkerboards feature a distinct, repetitive pattern that allows algorithms to easily detect and match features across images. The corners of the black and white squares can be precisely located, aiding in accurate calibration.

- Known geometry:

The fixed, known dimensions of a checkerboard enable accurate estimation of transformation parameters.

- Planar surface:

The flat, two-dimensional nature of a checkerboard simplifies mathematical modeling, making it easier to map 2D image coordinates to 3D world coordinates.

- Consistency and repeatability:

Checkerboards provide a well-defined, unambiguous pattern, ensuring consistent and reproducible calibration results while minimizing errors due to feature misidentification.

- Robustness:

Checkerboards are highly resilient to variations in lighting and perspective, making them a reliable calibration tool across different environments and conditions.

Below are examples of using a checkerboard as a fiducial object for camera-LiDAR spatial calibration. Zhang *et al.* [112] proposed a calibration method that utilizes a checkerboard to estimate the extrinsic parameters of a camera relative to a 2D laser range finder. In this approach, the checkerboard is captured simultaneously by both the camera and the laser range finder at different poses, and the collected data is used to impose geometric constraints for estimating the Euclidean transformation.

The process begins with determining the camera pose relative to the checkerboard using Zhang *et al.*'s well-known camera calibration method [113]. Based on this camera pose, the calibration plane parameters are then estimated. These parameters are subsequently used to compute the Euclidean transformation between the camera and the 2D laser range finder using a linear least-squares approach. To further refine the transformation, the initial estimate serves as an input for nonlinear optimization, where the Levenberg-Marquardt method [114] is applied to minimize both the Euclidean distance from laser points to the checkerboard planes and the reprojection error.

Figure 2-14 (a) illustrates the camera-laser setup, while Figure 2-14 (b) shows the projection of laser range finder points onto the camera-captured image at two different poses.

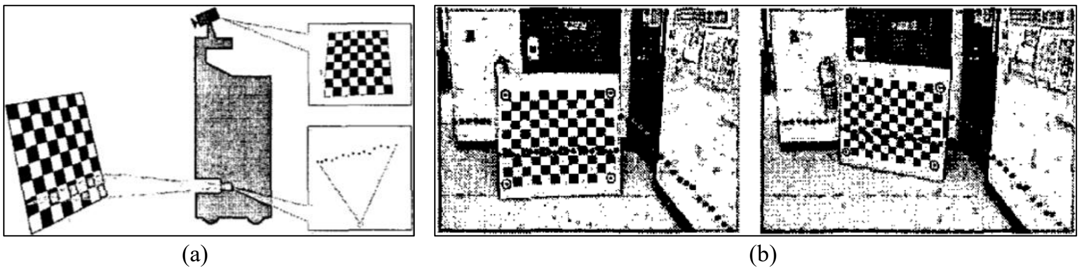
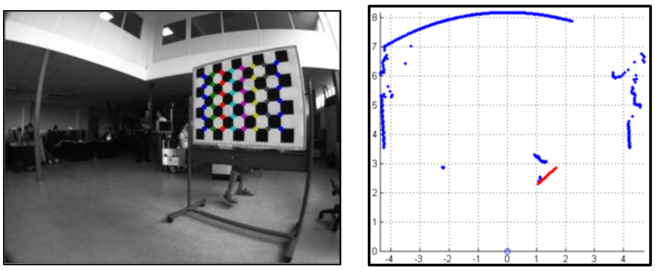


Figure 2-14. Example of using a checkerboard pattern for camera-LiDAR spatial calibration, following the method proposed by Zhang *et al.* [112]. (a) Illustration of the experimental setup consisting of a camera and a 2D laser range finder. (b) Projection of points acquired by the 2D laser range finder onto images captured by the camera at two different sensor poses.

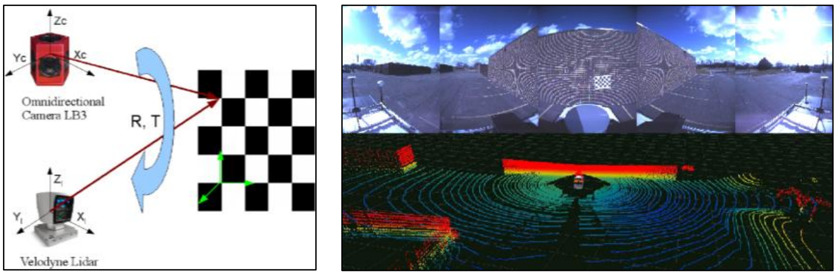
Kassir *et al.* [115] automated the manual calibration proposed by Zhang *et al.* [112] by introducing two algorithms for the automatic extraction of the checkerboard from both camera data and laser data respectively. These algorithms work jointly to enable fully automated camera-laser calibration. Figure 2-15 (a) illustrates the algorithm performance in detecting checkerboard corner features from a camera image, where the coloured dots represent the extracted corners. Figure 2-15 (b) demonstrates the algorithm's ability to extract the checkerboard from a 2D laser scan, with the red dotted line outlining the detected checkerboard.



(a) (b)

Figure 2-15. Example illustrating the use of a checkerboard pattern for camera-LiDAR spatial calibration, as presented by Kassir *et al.* [115]. (a) Demonstration of the algorithm extracting checkerboard corners from a camera image, with coloured dots representing the detected corners. (b) Demonstration of the algorithm extracting a straight line (red dotted line) from 2D laser scan data.

Pandey *et al.* [116] further extended the calibration method proposed by Zhang *et al.* [112] to support the calibration of a 3D laser scanner and an omnidirectional camera. The proposed method requires a minimum of three views in which the checkerboard is visible to both the omnidirectional camera and the 3D laser scanner. These observations establish constraints for nonlinear optimization, which is used to determine the extrinsic calibration parameters. Figure 2-16 (a) illustrates the omnidirectional camera and 3D laser scanner simultaneously observing a checkerboard corner. Figure 2-16 (b) displays the panoramic image captured by the omnidirectional camera alongside the 3D point cloud recorded by the 3D laser scanner.



(a) (b)

Figure 2-16. Example illustrating the use of a checkerboard for camera-LiDAR spatial calibration, as proposed by Pandey *et al.* [116]. (a) Illustration showing simultaneous observation of a checkerboard corner by an omnidirectional camera and a 3D laser scanner. (b) Panoramic image and corresponding 3D point cloud captured by the omnidirectional camera and the 3D laser scanner, respectively.

Geiger *et al.* [117] proposed an automatic calibration method for cameras and range sensors, including depth cameras and LiDAR. The method begins by generating two 3D point clouds of the checkerboards: one obtained through triangulation from stereo images and the other extracted from the range-scanned 3D point cloud. Next, global registration is performed between these two point clouds using the principal component analysis (PCA) technique to obtain an initial estimate of the calibration parameters. This estimate is then refined through fine registration using the well-known iterative closest point (ICP) algorithm [77]. Unlike other calibration methods, this approach does not require simultaneous captures of the fiducial object (checkerboards) by both the camera and range sensor at multiple poses. Instead, it only requires a single camera image and a range scan of multiple checkerboards placed at different positions. Figure 2-17 (a) illustrates two experimental setups for camera and range sensor calibration. The left side shows a 3D LiDAR paired with a trinocular camera, while the right side displays a depth camera paired with a binocular camera. Figure 2-17 (b) presents a camera image alongside a color-coded depth image of a checkerboard pattern captured by the depth camera. Figure 2-17 (c) shows an image of multiple checkerboards placed at different positions (top) and the corresponding LiDAR data, including a 3D point cloud and the final calibration result (bottom).

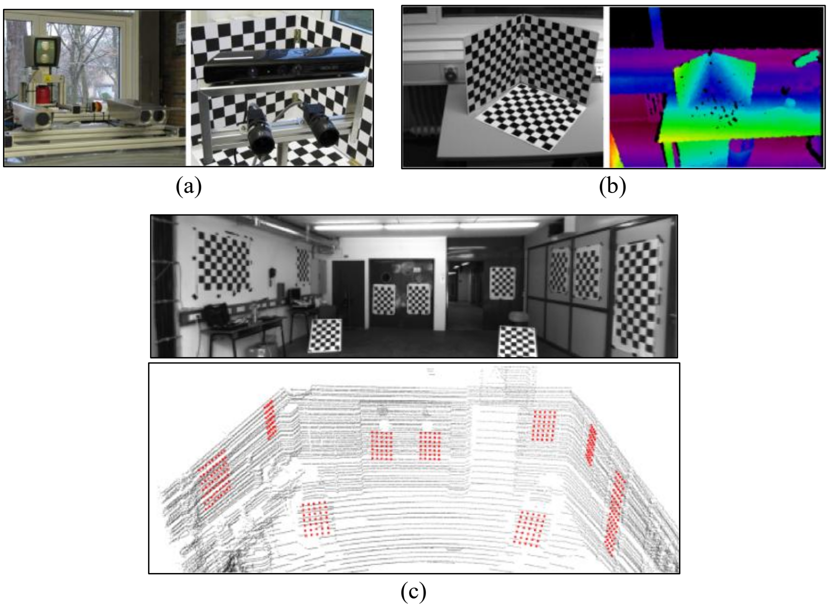
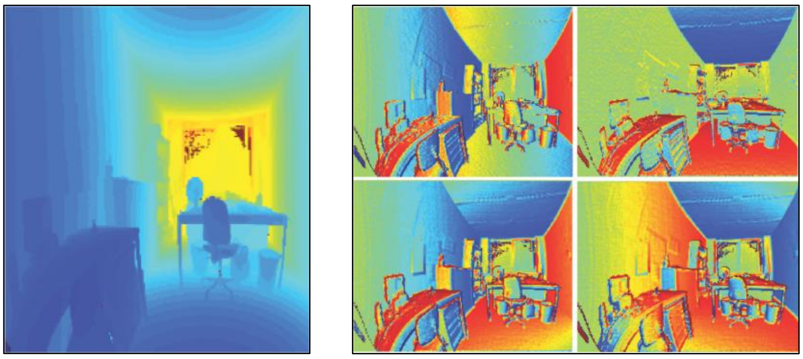


Figure 2-17. Example illustrating the use of checkerboard patterns for calibrating cameras and range sensors, as described by Geiger *et al.* [117]. (a) Two experimental setups demonstrating different camera and range sensor configurations: on the left, a Velodyne HDL-64E 3D LiDAR paired with a trinocular camera; on the right, a Microsoft Kinect depth camera paired with a binocular camera. (b) A camera image and the corresponding color-coded depth image of a checkerboard captured by the Microsoft Kinect depth camera. (c) Top: A camera image showing multiple checkerboards positioned at various locations; Bottom: Corresponding LiDAR data (3D point cloud) along with the resulting calibration.

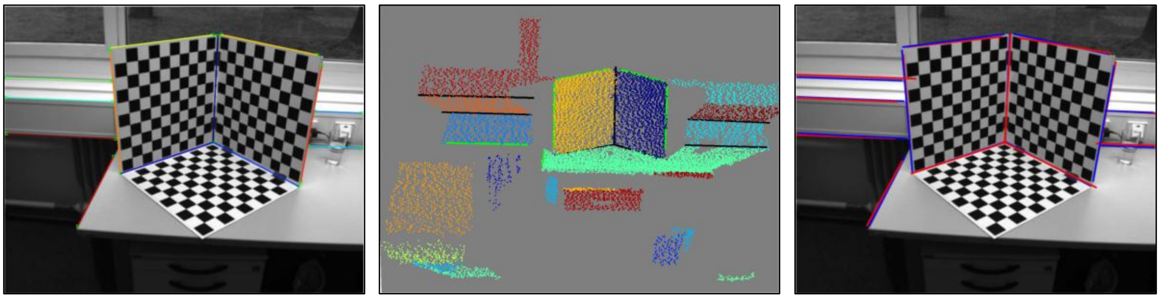
Target-based approaches require fiducial objects that can be simultaneously observed by both the camera and LiDAR. However, this requirement is difficult to meet in scenarios such as in-situ calibration. In such cases, feature-based approaches are used as an alternative. Unlike target-based methods, feature-based approaches rely on natural scene features to determine the Euclidean transformation between sensors, eliminating the need for a planar calibration pattern. Scaramuzza *et al.* [118] proposed an on-the-fly extrinsic calibration method for an omnidirectional camera and a 3D laser range finder, based on feature correspondences observed in natural scenes. To accomplish this, the authors introduced a novel technique that converts 3D range data into 2D bearing-angle images (BA images) serving as the 2D map. By matching features between the 2D map and the 2D camera image, the system effectively establishes correspondences between the original 3D range data and the 2D camera image. This transformation simplifies the challenge of aligning 3D and 2D sensor data by reducing it to a more manageable 2D-to-2D correspondence problem, significantly improving the establishment of camera-laser correspondences. Once the feature correspondences between the 2D camera image and 3D range data are identified, the well-known Perspective-n-Point (PnP) algorithm is applied to obtain an initial estimation of the Euclidean transformation between the omnidirectional camera and the 3D laser range finder. This estimated transformation is then refined by minimizing the reprojection error using a nonlinear optimization approach similar to that used by Zhang *et al.* [112]. Figure 2-18 (a) illustrates the 3D range information of a scene, presented as a 2D color-coded depth image. Figure 2-18 (b) displays the bearing angle (BA) images of the scene, computed from the 3D range data along four different angles.



(a) (b)

Figure 2-18. Example illustrating a feature-based approach for camera-LiDAR calibration, as presented by Scaramuzza *et al.* [118]. (a) Color-coded depth image representing the 3D range data of a scene. (b) Bearing-angle images computed from the 3D range data at four different viewing angles, represented as 2D maps.

Moghadam *et al.* [119] proposed an algorithm for the extrinsic calibration of a range sensor relative to an image sensor using linear features. The algorithm begins by extracting line segments from both the 3D point cloud captured by the range sensor and the 2D image captured by the camera. Next, correspondences between the 2D and 3D line segments are established and used to formulate line-to-line geometric constraints. Based on these constraints, the registration between the extracted 2D and 3D lines is performed by minimizing the reprojection error of the matched line pairs through a nonlinear optimization approach. This registration process ultimately yields the calibration parameters. Figure 2-19 (a) illustrates the line segments extracted from the 2D image using the Canny edge detector. Figure 2-19 (b) presents the line segments extracted from the 3D point cloud, including plane intersections and boundary lines, using a region-growing strategy. Figure 2-19 (c) displays the final registered line segments after the calibration process.



(a) (b) (c)

Figure 2-19. Example illustrating a feature-based approach for calibrating cameras and range sensors, as proposed by Moghadam *et al.* [119]. (a) Line segments extracted from the camera's 2D image using the Canny edge detector. (b) Line segments (including plane intersections and boundary lines) extracted from the 3D point cloud using a region-growing strategy. (c) Resulting aligned line segments after registration: The blue lines represent matched line segments from the 2D image, while the red lines indicate the projection of corresponding 3D point cloud line segments onto the 2D image.

There are also approaches utilizing the statistical dependence between image sensor data and range sensor data to determine the calibration parameters for both sensors. Boughorbal *et al.* [120] proposed a method that uses the χ^2-information metric to calibrate a camera and a range sensor. The χ^2-information metric is a statistical measure that quantifies the dependence between two probability distributions. Their approach determines the Euclidean transformation that maximizes the χ^2-similarity measure, which, in this context, represents the correlation between the projected reflectance image in the camera plane and the colour image.

Williams *et al.* [121] introduced a similar approach using the Chi-square statistic for camera-laser calibration. However, this method requires an initial guess of the calibration parameters before applying the optimization scheme. Another widely used statistical metric is mutual information (MI), which measures the mutual dependence between two variables. Alempijevic *et al.* [122] proposed an MI-based approach for automatic sensor registration and calibration. Their method utilizes feature-level MI to identify commonalities between different sensor spaces. By maximizing mutual information between multiple signal streams, the unknown sensor registration and calibration parameters can be estimated.

Taylor *et al.* [123] developed a method that applies normalized MI to register camera images and LiDAR scans in natural environments. The approach first projects the 3D LiDAR point cloud into a 2D LiDAR image using a camera model. Then, the normalized MI between the LiDAR image and the camera image is used as a metric to evaluate the alignment quality between the two sensors. The best alignment is achieved when the normalized MI reaches its global maximum.

Mastin *et al.* [124] introduced an MI-based calibration method for aligning aerial LiDAR scans with aerial optical imagery. In their method, MI is expressed as joint entropy (JE) between range data and image data, under the assumption that minimizing JE sufficiently approximates maximizing MI. However, Wang *et al.* [125] demonstrated that this method is not directly applicable to ground-level data (e.g., laser scans and images captured from a moving vehicle). In such cases, the entropy of a LiDAR image is not approximately constant under small perturbations, making JE minimization ineffective. Instead, Wang *et al.* proposed directly maximizing MI to achieve accurate camera-LiDAR registration.

As discussed in previous camera–LiDAR calibration methods, nonlinear optimization is commonly employed to refine the initial estimate of the calibration parameters, typically by minimizing the reprojection error. This error serves as a key metric for evaluating calibration accuracy. Figure 2-20 illustrates the concept of reprojection error. During calibration, a 3D point observed simultaneously by both the camera and the LiDAR appears as two measurements: one from the camera (shown in black) and one from the LiDAR (shown in red). The LiDAR point is then reprojected onto the image plane using the estimated Euclidean transformation between the LiDAR and camera coordinate frames. Ideally, this reprojected point should align with the camera observation. However, any deviation between the two indicates a reprojection error. The magnitude of this error reflects the accuracy of the estimated transformation: smaller reprojection errors correspond to more accurate calibrations. Therefore, minimizing reprojection error is essential for improving the overall precision of camera–LiDAR calibration.

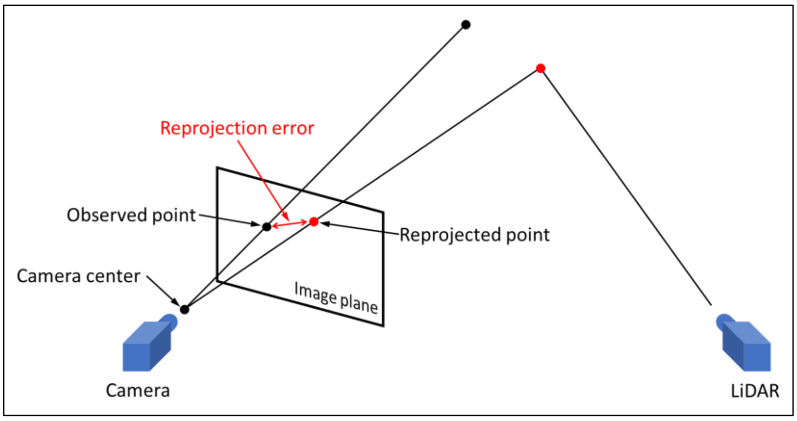


Figure 2-20. Illustration of the reprojection error concept used in camera-LiDAR calibration.

Beyond refining the initial estimation of calibration parameters, projecting LiDAR points onto camera images can also be used for qualitative analysis of spatial calibration accuracy through visual inspection. Figure 2-21 presents several examples of reprojecting 3D LiDAR points onto 2D camera images to assess calibration accuracy:

- Figure 2-21 (a): Scaramuzza *et al.* [118] projected edge points representing depth discontinuities onto a monocular image using the computed Euclidean transformation to qualitatively assess the accuracy of camera-LiDAR calibration.
- Figure 2-21 (b): In the KITTI dataset [111], depth-based color-coded 3D points captured by a Velodyne LiDAR were projected onto a monocular image using the computed Euclidean transformation to qualitatively assess the accuracy of camera-LiDAR calibration.
- Figure 2-21 (c): Pandey *et al.* [116] projected LiDAR range data, color-coded based on scene depth, onto a panoramic image captured by an omnidirectional camera using the computed Euclidean transformation to visually assess calibration accuracy with the computed transformation.
- Figure 2-21 (d): Pandey *et al.* [126] projected depth-based color-coded points captured by a time-of-flight 3D camera and a 2D LiDAR onto a monocular image using the computed Euclidean transformation to qualitatively evaluate the accuracy of mutual information (MI)-based calibration.

- Figure 2-21 (e): In the Oxford RobotCar dataset [102], 3D point clouds captured by a LiDAR were projected onto a monocular image using poses estimated by an Inertial Navigation System (INS) and visual odometry, respectively, to compare their accuracy. The comparison shows that in regions with weak GPS signals, the INS-based pose estimation suffers interruptions, resulting in inaccurate local 3D point clouds. Visual odometry, on the other hand, provides smoother local pose estimates but exhibits drift over longer distances.

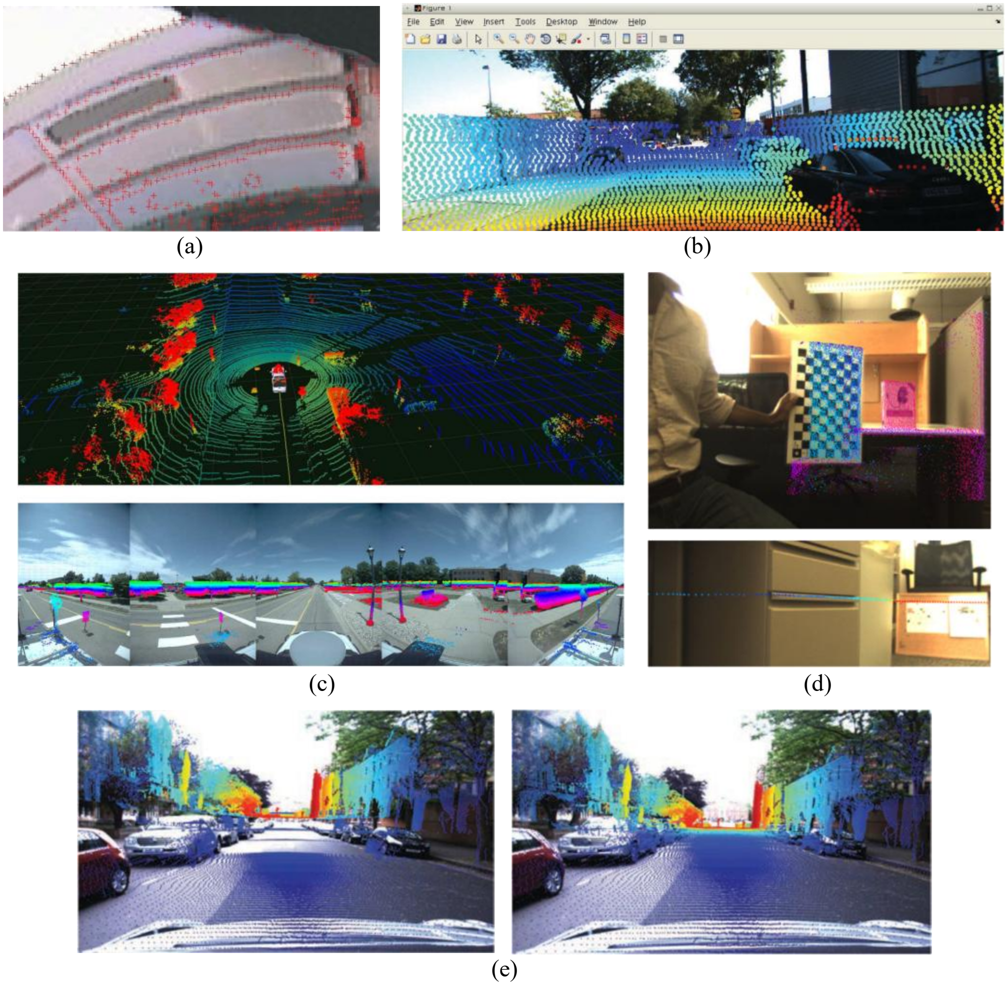


Figure 2-21. Examples illustrating qualitative evaluations of camera-LiDAR calibration accuracy by projecting LiDAR-captured points onto camera images. (a) Projection of edge points extracted from LiDAR data (shown in red), representing depth discontinuities, onto a monocular image using the computed Euclidean transformation (Scaramuzza *et al.* [118]). (b) Projection of depth-based, color-coded 3D points captured by a Velodyne LiDAR onto a monocular image using the computed Euclidean transformation (Geiger *et al.* [111]). (c) Top: A frame of 3D point cloud captured by the LiDAR; bottom: The corresponding projection of color-coded LiDAR points onto the panorama image using the computed Euclidean transformation (Pandey *et al.* [116]). (d) Projection of depth-based, color-coded points from a time-of-flight 3D camera and a 2D LiDAR onto a monocular image using the computed Euclidean transformation (Pandey *et al.* [126]). (e) Comparison between projections of a 3D LiDAR point cloud onto a monocular image using poses estimated by an Inertial Navigation System (INS) and visual odometry, respectively (Maddern *et al.* [102]).

Temporal calibration refers to the process of synchronizing data acquisition from multiple sensors, ensuring that the collected data is temporally aligned—meaning it corresponds to the same point in time. This is critical in applications where precise timing is essential, such as robotics, autonomous vehicles, and multi-sensor fusion systems.

The ideal approach for temporal calibration involves hardware-supported synchronization, including external triggers (commonly found in industrial cameras), GNSS timing, and Precision Time Protocol (PTP), also known as IEEE-1588 synchronization [127]. However, these solutions are often expensive and rarely available in consumer-grade products.

An alternative approach is to estimate the time offset caused by unsynchronized clocks or communication delays between different sensors [128, 129]. In this method, the time offset is modelled as a variable and determined through optimization techniques that either maximize cross-correlation or minimize a predefined error function between different sensor data streams.

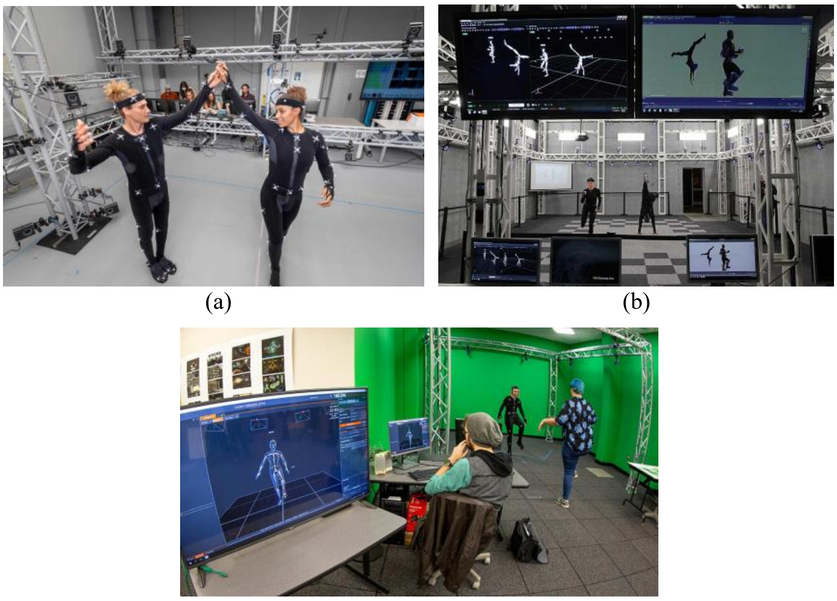
The review of sensory data concludes here. Following the taxonomy of the SLAM benchmark problem and the workflow outlined in Figure 2-1, the next section explores the ground truth data used in SLAM benchmarking.

2.1.2 Ground truth

The SLAM benchmark can be classified into two categories: localization benchmarking and mapping benchmarking. The review of ground truth data in SLAM benchmarking follows this classification. First, the discussion focuses on localization ground truth, specifically the ground truth trajectory of the mobile platform in real-world SLAM benchmarks. Next, it examines mapping ground truth, which refers to the ground truth map of the environment in real-world SLAM benchmarks. Finally, it reviews both localization and mapping ground truth used in SLAM benchmarks conducted in synthetic environments.

In general, the calibration process is a fundamental prerequisite to obtaining reliable ground truth in SLAM benchmarking. It typically involves several steps. (i) Data acquisition: raw measurements are collected from the sensors involved (e.g., cameras, LiDARs, IMUs) as well as from external ground truth devices such as motion capture systems or laser trackers. (ii) Intrinsic calibration: the internal parameters of each sensor are estimated, for example the focal length, principal point, and lens distortion for cameras, or systematic biases in IMU measurements. (iii) Extrinsic calibration: the relative position and orientation between multiple sensors, or between sensors and ground truth devices, are determined. This step usually involves solving for a rigid-body transformation matrix that aligns measurements into a common reference frame. (iv) Validation and refinement: calibration results are validated, often by projecting or aligning data across modalities (e.g., projecting LiDAR points onto camera images), and refined if misalignments are detected. (v) Application: the calibrated parameters are then applied during benchmarking, ensuring that estimated trajectories and maps can be consistently compared against ground truth data. Without such calibration, even high-precision ground truth cannot be meaningfully aligned with SLAM outputs, leading to biased or invalid evaluation results.

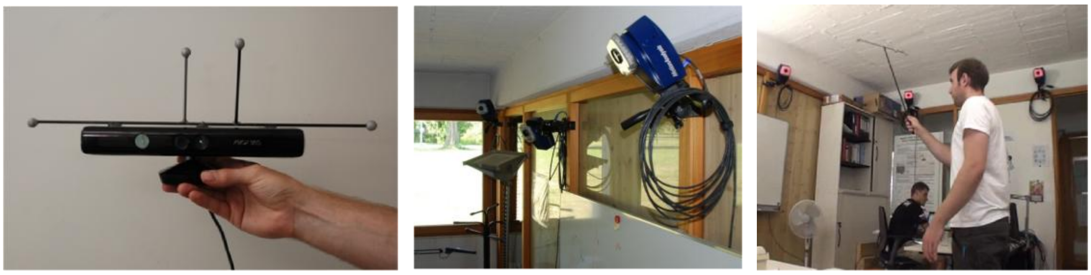
The acquisition of ground truth trajectories for mobile platforms in real-world environments primarily relies on high-precision positioning technologies, such as high-precision reference maps, RTK-GPS, motion capture (MoCap) systems, and laser trackers. Among these, MoCap systems are the most frequently used in indoor environments [91, 108, 130]. MoCap technology is designed to record and analyse the movement of objects or people by capturing motion data and converting it into digital models. It is widely applied in fields such as film, video games, sports analysis, and medical research. Figure 2-22 provides examples of MoCap system setups and their application scenarios. As shown in Figure 2-22 (a), multiple retroreflective markers are attached to key points on a subject’s body. Active optical cameras, mounted at elevated positions, emit infrared (IR) light, which illuminates these markers. The cameras then capture the reflected IR light, allowing the system to localize marker positions. This enables precise tracking and visualization of the subject’s movements, which can be displayed on specialized computer software, as illustrated in Figure 2-22 (b) and (c). Figure 2-23 and Figure 2-24 show two examples of MoCap hardware setups used in SLAM benchmarking to acquire the ground truth trajectory of a mobile platform. As depicted in Figure 2-23 (a) and Figure 2-24 (a), sensors are typically equipped with multiple markers, allowing their positions to be accurately tracked by the overhead-mounted cameras of the MoCap system.



(a) (b)

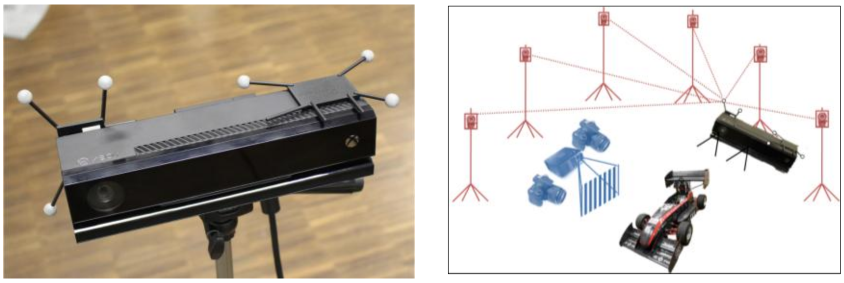
(c)

Figure 2-22. Examples of equipment setups and application scenarios for motion capture (MoCap) systems. (a) An example of a motion capture setup in which multiple cameras, positioned at different angles, capture human motion by tracking reflective markers placed at key points on the subject's body [131]. (b) An application scenario using the OptiTrack Studio MoCap system, where the subject's movements are visualized on-screen through specialized software [132]. (c) An application scenario using the Vicon MoCap system [133].



(a) (b) (c)

Figure 2-23. Example hardware setup of a MoCap system used in a SLAM benchmark to acquire the ground truth trajectory of a mobile platform (Sturm *et al.* [91]). (a) A sensor (Microsoft Kinect) equipped with multiple markers, allowing its ground truth trajectory to be tracked by MoCap cameras. (b) MoCap cameras mounted in elevated positions. (c) Calibration process for the MoCap camera setup.



(a) (b)

Figure 2-24. Example hardware setup of a MoCap system used in a SLAM benchmark to acquire the ground truth trajectory of a mobile platform (Wasenmüller *et al.* [108]). (a) A Microsoft Kinect V2 sensor is equipped with multiple markers, enabling MoCap cameras to accurately track its trajectory. (b) Overview of the experimental setup and operational principle of the proposed SLAM benchmark.

Although MoCap systems provide high-accuracy positional data at a high frame rate, they come with significant drawbacks, including high costs, complex hardware setups, and extensive calibration requirements. To address these challenges, cheaper and more user-friendly alternatives, such as fiducial markers [134], have been adopted in some studies. Fiducial markers are graphic patterns with known geometries printed on paper and placed within a scene to serve as reference points for imaging systems. They are commonly used for accurate alignment, calibration, and object tracking in fields such as computer vision, robotics, augmented reality (AR), and medical imaging. Figure 2-25 illustrates the fiducial marker setup used in the RAWSEEDS project [135]. In this setup, the robot platform is equipped with multiple planar fiducial markers, which are continuously monitored by a network of fixed video cameras. These cameras track the markers' positions relative to the camera reference frame. To determine the robot's pose in the world reference frame, a Perspective-n-Point (PnP) technique is applied. This process involves bridging the camera reference frame with the world reference frame and linking the marker reference frame to the robot's reference frame, enabling precise pose estimation.

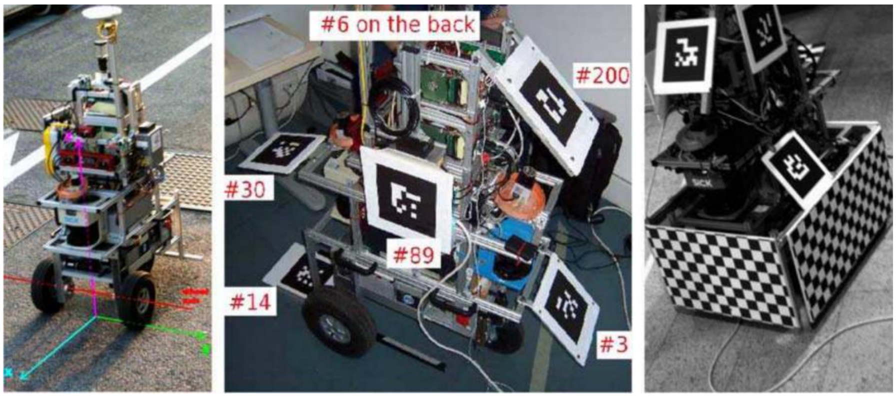


Figure 2-25. Example hardware setup illustrating the use of fiducial markers for acquiring the ground truth trajectory of a robotic platform (RAWSEEDS project [135]).

Beyond MoCap systems and their alternatives, several other methods are used to acquire ground truth trajectories in indoor SLAM benchmarks. In the RAWSEEDS project [135], in addition to the vision-based approach that utilizes fiducial markers for trajectory acquisition, Ceriani *et al.* proposed a laser-based approach for obtaining ground truth trajectories. This method first captures robot scans using a network of fixed laser scanners. Then, the robot model with a known pose is registered to these scans using the Iterative Closest Point (ICP) algorithm, allowing the robot's pose to be determined within the world frame. In the EuRoC MAV Dataset [92], Burri *et al.* employed a laser tracker to obtain the ground truth trajectory of a drone. The laser tracker determines the drone’s 3D coordinates by tracking a retroreflector target mounted on the drone. It measures the radial distance to the retroreflector using the Time-of-Flight (ToF) technique, while two angle encoders determine the polar and azimuth angles. These three measurements—radial distance, polar angle, and azimuth angle—are then combined to compute the drone’s 3D position within the coordinate frame of the laser tracker. Figure 2-26 provides a diagram illustrating the working principle of a laser tracker. The model shown in the figure is the FARO Vantage [136].

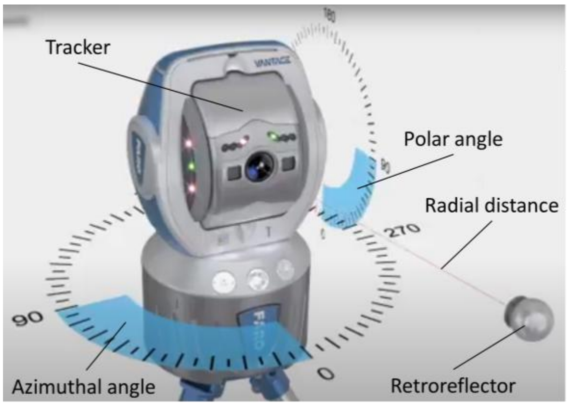


Figure 2-26. Illustration of the working principle of a laser tracker (FARO Vantage model) [136].

In outdoor environments, GNSS-based technologies are among the most reliable solutions for acquiring ground truth trajectories in SLAM benchmarks. GNSS (Global Navigation Satellite System) is a positioning system that consists of a constellation of satellites and a receiver. The satellites transmit positioning and timing data from space, which the receiver processes to determine its location anywhere on Earth. The most well-known GNSS systems include GPS (Global Positioning System), GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema), Galileo, and BeiDou, with GPS being the most widely used in SLAM research [96, 137, 138].

In practice, GNSS is often combined with Real-Time Kinematic (RTK) and Inertial Measurement Unit (IMU) technologies to enhance positioning accuracy.

- RTK (Real-Time Kinematic) correction is used alongside GNSS to reduce common errors and improve positioning accuracy. An RTK system consists of a base station and a rover. The base station, a fixed GNSS receiver at a known location, continuously receives GNSS signals and compares them with its actual position to compute correction data. The rover, a mobile GNSS receiver, moves through the environment, collecting satellite data. It receives correction data from the base station via a communication link (e.g., radio, cellular network, or internet) and integrates it with its GNSS data to compute high-accuracy positioning in real-time.

- IMU (Inertial Measurement Unit) is a sensor module consisting of an accelerometer, gyroscope, and sometimes a magnetometer. It provides high-frequency (e.g., 100 Hz or more) measurements of acceleration and angular velocity, enabling precise short-term motion tracking. This allows for reliable navigation in environments where GNSS signals are weak, intermittent, or obstructed.

Figure 2-27 shows an example scenario where GNSS signals in complex urban environments suffer from blockages due to high-rise buildings. As shown in the figure, yellow segments indicate areas with strong GNSS signals, red segments indicate areas with weak GNSS signals, and green segments represent areas where GNSS signals are entirely unavailable. As illustrated, GNSS signals are frequently disrupted, with areas of weak or no signal significantly outnumbering areas with strong signal reception. This demonstrates that in urban environments, GNSS performance is inevitably impacted by high-rise buildings. Consequently, an IMU is essential to ensure precise short-term motion tracking and reliable navigation during periods of GNSS signal loss.

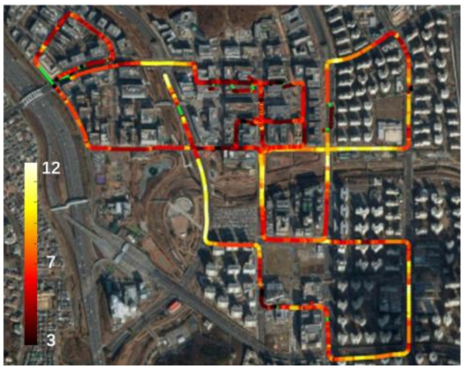
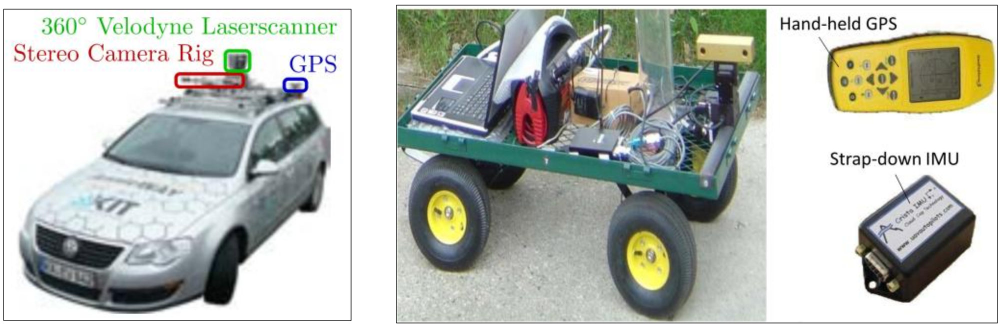


Figure 2-27. Scalar field visualization of GNSS signal intensity along the route used for data collection in a complex urban outdoor environment (Liu *et al.* [139]).

The fusion of GNSS, RTK, and IMU technologies for acquiring ground truth trajectories in SLAM benchmarks has been widely utilized in various studies. In the KITTI dataset [96], the ground truth trajectories of the mobile platform were obtained using the OXTS RT 3003 localization system, which integrates GPS, GLONASS, an IMU, and RTK correction signals. Figure 2-28 (a) illustrates the hardware setup used for data collection; however, while the GPS is labelled, other devices are not annotated. Similarly, Abdallah *et al.* [137] employed the hardware setup shown in Figure 2-28 (b) to collect data for a SLAM benchmark. In this setup, a handheld GPS and a strap-down IMU were used to acquire the ground truth trajectory of the mobile platform.



(a) (b)

Figure 2-28. Examples of hardware setups used for acquiring ground truth trajectories of mobile platforms in SLAM benchmarks. (a) Hardware configuration used for data collection in the KITTI dataset [96]. (b) Hardware configuration used by Abdallah *et al.* [137].

In outdoor environments, alongside GNSS-based technologies, Monte Carlo Localization (MCL) is frequently used in combination with other techniques for acquiring ground truth trajectories in SLAM benchmarks. Wulf *et al.* [101] proposed a method that integrates MCL with a highly accurate 2D reference map for trajectory estimation. In this approach, a laser-scanned 3D point cloud is matched against the 2D reference map to generate a virtual 2D scan, which is then used in MCL to determine the ground truth trajectory. A supervision step is later applied to manually validate the obtained trajectory. As illustrated in Figure 2-29, this method of ground truth trajectory acquisition, based on the 2D reference map and MCL, was implemented in the SLAM benchmark proposed by Wulf *et al.* [100] to assess localization accuracy. In the figure, the blue line represents the high-precision reference map, the red dotted line indicates the trajectory estimated by the SLAM algorithm, and the grey line shows the ground-truth trajectory obtained using Monte Carlo Localization (MCL) aligned with the high-precision reference map.

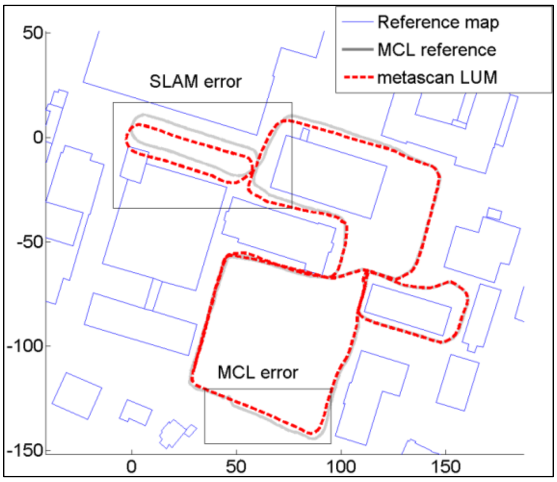


Figure 2-29. Evaluation of the localization performance of the SLAM algorithm using Monte Carlo Localization (MCL) and a reference map (Wulf *et al.* [100]).

Kümmerle *et al.* [99] proposed a method that utilizes Monte Carlo Localization (MCL) and aerial images to acquire the ground truth trajectory for SLAM benchmarking. In this approach, 2D aerial images provide prior information to estimate the likelihood of perceiving a 3D range scan from a given 2D pose within the aerial image. The likelihood is defined as a function of the distance between the reprojected 3D scan points (projected onto the 2D aerial image based on the robot's 2D pose) and their nearest corresponding points in the 2D aerial image. This computed likelihood is then incorporated into MCL to estimate the robot's trajectory. Additionally, the authors compared the MCL-based trajectory with the GPS-based trajectory. As illustrated in Figure 2-30, the trajectory obtained using MCL and aerial images (red line) demonstrates higher accuracy than the trajectory derived from GPS data (blue line).

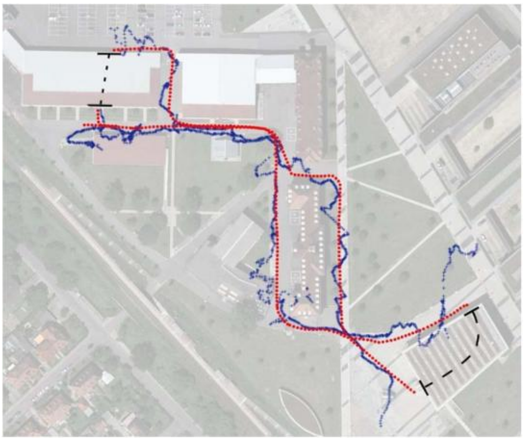


Figure 2-30. Comparison of trajectories obtained using GPS (blue line) and Monte Carlo Localization (MCL) combined with aerial imagery (red line) (Kümmerle *et al.* [99]).

The review of localization ground truth (the ground truth trajectory of the mobile platform) used in real-world SLAM benchmarks concludes here. The next section focuses on mapping ground truth (the ground truth map of the environment) in real-world SLAM benchmarks.

Compared to the number of SLAM benchmarking studies evaluating localization performance in real-world environments, significantly fewer studies focus on mapping performance due to the limited availability of ground truth maps. Acquiring ground truth maps for real-world environments often requires expensive and complex hardware setups, which are beyond the resources of many research institutions and individual researchers.

The following section provides a brief review of the hardware commonly used for ground truth map acquisition in real-world environments. These systems are typically 3D laser scanners or 3D structured-light scanners. Meister *et al.* [140] used a 3D structured-light scanner (Breuckmann smartSCAN-HE) to obtain ground truth geometries of several objects and a terrestrial laser scanner (Riegl VZ-400) to generate a ground truth map of an indoor scene. These reference geometries and maps were then compared against the dense 3D geometries produced by the SLAM algorithm KinectFusion to evaluate whether KinectFusion was sufficiently accurate for ground truth mapping in SLAM benchmarking.

Wasenmüller *et al.* [108] introduced the CoRBS dataset, a comprehensive benchmark for RGB-D SLAM systems. In this dataset, ground truth geometries of objects of varying sizes were obtained using a 3D structured-light scanner (3Digify). As illustrated in Figure 2-31, the structured-light scanner consists of a structured-light projector and two 18-megapixel cameras. The projector emits a fringe pattern onto the object’s surface, while the two cameras capture the distortion of the pattern. Using the triangulation method, the system reconstructs the object’s geometry with high precision. Figure 2-32 presents the objects included in the CoRBS dataset for mapping benchmark evaluation, alongside their corresponding reconstructed ground truth geometries, obtained using the structured-light scanner depicted in Figure 2-31.

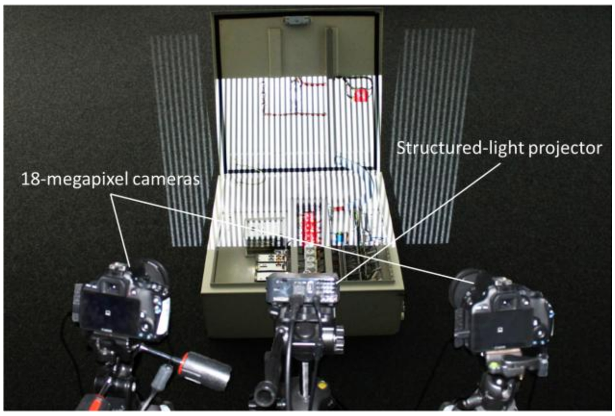


Figure 2-31. The 3D structured-light scanner used in the CoRBS dataset [108] performing a scan of an electrical cabinet.

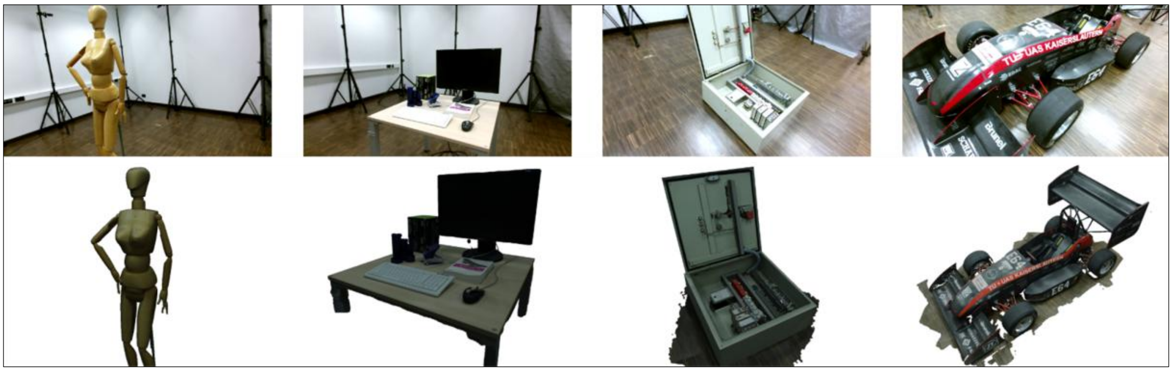


Figure 2-32. Objects and their reconstructed 3D geometries from the CoRBS dataset [108], used as a benchmark for evaluating mapping performance. Top: The objects used in the CoRBS dataset for mapping benchmark. Bottom: The corresponding reconstructed ground truth geometries of the objects shown in the top row.

The previous paragraphs have discussed the methods used in SLAM benchmarking to acquire localization ground truth (the ground truth trajectory of the mobile platform) and mapping ground truth (the ground truth map of the environment) in real-world environments. However, due to the presence of errors and uncertainties in real-world conditions, the so-called ground truth trajectories and maps are not perfectly accurate representations of reality.

Additionally, high-precision positioning and mapping technologies such as MoCap systems, RTK-GPS, and 3D scanners are often expensive and beyond the financial reach of many research institutions and individual researchers. To address these limitations, error-free synthetic environments generated using 3D modelling and rendering software have been widely adopted for SLAM benchmarking in various studies [97, 109, 141-143]. The following section reviews the methods used for acquiring ground truth trajectories and maps in synthetic environments.

Handa *et al.* [97] introduced the ICL-NUIM dataset, which was collected within synthetically generated environments for benchmarking RGB-D SLAM systems. As illustrated in Figure 2-33, the dataset's synthetic environments consist of 3D polygonal CAD models of indoor scenes with precisely defined geometries, constructed using 3D modelling and rendering software POVRay. These CAD models also serve as ground truth maps for mapping benchmarks.

The ground truth trajectories used in these synthetic environments originate from real-world SLAM experiments. The process begins by running the Kintinuous SLAM algorithm on a pre-collected dataset, producing an estimated trajectory. This estimated trajectory is then transformed and uniformly scaled to align properly with the synthetic environment generated in POVRay, ensuring a suitable trajectory for benchmarking. This trajectory acquisition method in synthetic environments was also applied by Handa *et al.* [144] to analyse the impact of high frame rates on camera motion tracking.

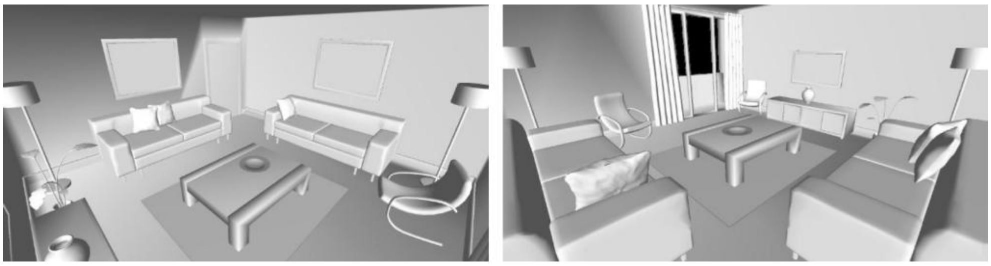


Figure 2-33. Interior geometries of the synthetic room scenes used in ICL-NUIM dataset [97].

Schofield *et al.* [145] proposed a semi-synthetic approach for creating visual-inertial odometry datasets for SLAM benchmarking. The method used for ground truth trajectory generation in this approach is similar to that of Handa *et al.* [97]. First, a visual-inertial system trajectory is recorded in a real-world environment using motion capture technology. This trajectory is then transformed and imported into the 3D modelling and rendering software Blender, where it serves as the ground truth trajectory to guide camera movement and image capture in the synthetic environment.

The review of ground truth acquisition methods used in SLAM benchmarking concludes here. Following the taxonomy of SLAM benchmarking problems and the SLAM benchmark workflow shown in Figure 2-1, the next section reviews evaluation metrics used in SLAM benchmarking. This review follows the same categorization framework used for ground truth acquisition methods. First, it examines the evaluation metrics used in localization benchmarks, followed by a review of evaluation metrics used in mapping benchmarks.

2.1.3 Evaluation metrics

Similar to the review of ground truth acquisition methods in Section 2.1.2 , which follows the categorization of SLAM benchmarks into localization benchmarking and mapping benchmarking, this section reviews the evaluation metrics used in SLAM benchmarking based on the same classification. First, the evaluation metrics for localization benchmarks are examined, followed by a review of the evaluation metrics for mapping benchmarks.

The most intuitive and commonly used approach for error evaluation in localization benchmarks is to directly compare the positions in the estimated trajectory with those in the ground truth trajectory. The most widely adopted evaluation metric is the Euclidean distance between corresponding positions in both trajectories. For example, as shown in Eq. (2.1), Wulf *et al.* [101] computed the 2D Euclidean distance e_i between each position $\left(x_i^{SLAM}, y_i^{SLAM}\right)$ in the SLAM-estimated trajectory and its corresponding position $\left(x_i^{REF}, y_i^{REF}\right)$ in the ground truth trajectory. Then, the root mean square (RMS) of all computed 2D Euclidean distance e_i is calculated and used as the metric to assess the localization accuracy of the SLAM system.

$$e_i = \sqrt{(x_i^{SLAM} - x_i^{REF})^2 + (y_i^{SLAM} - y_i^{REF})^2} \tag{2.1}$$

Both the ground truth and estimated trajectories consist of numerous poses, where each pose is defined by a position and an orientation. The Euclidean distance metric evaluates only positional discrepancies between the estimated and ground truth trajectories, neglecting orientation differences. To address this limitation, Sturm *et al.* [91] introduced the Absolute Trajectory Error (ATE) as an evaluation metric that directly compares poses in both trajectories, considering discrepancies in both position and orientation. This metric has been widely adopted in SLAM benchmark studies [97, 146] for localization evaluation and is formally defined in Eq. (2.2) and Eq. (2.3). As shown in Eq. (2.2), each estimated pose P_i is first aligned with the ground truth coordinate frame by multiplying its 3D homogeneous transformation matrix with a transformation matrix S. The error F_i between the transformed estimated pose SP_i and its corresponding ground truth pose Q_i is then computed as the product of the inverse of the ground truth pose matrix and the transformation matrix of the transformed estimated pose. Finally, as shown in Eq. (2.3), the Absolute Trajectory Error (ATE) is obtained by computing the root mean square (RMS) of the magnitudes of the translational components of all error matrices.

$$F_i := Q_i^{-1} S P_i \tag{2.2}$$

$$RMSE(F_{1:n}) := \left(\frac{1}{n}\sum_{i=1}^{n}\|trans(F_i)\|^2\right)^{1/2} \tag{2.3}$$

Although directly comparing elements in the estimated and ground truth trajectories is an intuitive approach for localization benchmarking, Burgard *et al.* [98] and Kümmerle *et al.* [99] argued that it is suboptimal, as it fails to objectively reflect the true accuracy of a SLAM system. To illustrate this, consider the scenario depicted in Figure 2-34, where a robot moves along a straight line. In the figure, the blue circles represent the robot's ground truth trajectory, while the red circles represent the SLAM-estimated trajectory. The correspondences between them are indicated by dashed lines: vertical dashed lines signify no error between the estimates and the ground truth, whereas diagonal dashed lines indicate the presence of errors. In the upper part of the figure, a local error occurs at the end of the trajectory, resulting in a relatively small global localization error. In contrast, in the lower part of the figure, the same amount of local error occurs at the beginning of the trajectory. As the error accumulates over time, it leads to a significantly larger global localization error. This example highlights that identical local errors can result in vastly different global localization errors, depending on their position within the trajectory. Consequently, direct element-wise comparison between the estimated and ground truth trajectories is inadequate as an evaluation metric for localization benchmarking.

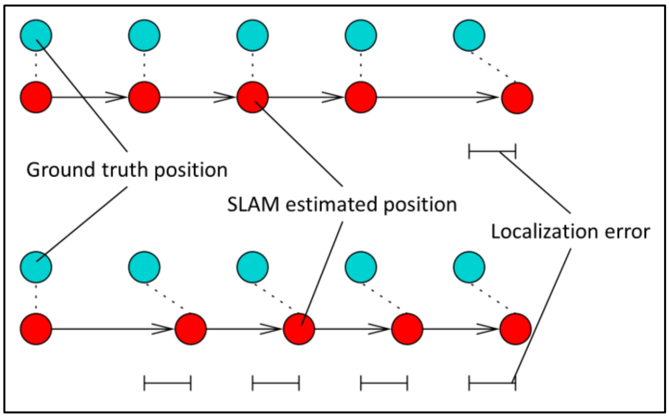


Figure 2-34. Illustration demonstrating why directly comparing elements of the estimated trajectory to the ground-truth trajectory is suboptimal for evaluating localization benchmarks (Kümmerle *et al.* [99]).

To address this issue, Burgard *et al.* [98] proposed an evaluation metric based on the graph of relative relations between poses. This metric is inspired by the concept of graph mapping introduced by Lu *et al.* [147] and the Normalized Estimation Error Squared (NEES) metric proposed by Bar-Shalom *et al.* [148]. It models poses as nodes and the connections between them as constraints, using the deformation energy required to align the estimated trajectory with the ground truth as the error measure. Rather than directly comparing individual poses in the estimated and ground truth trajectories, this metric evaluates the relative displacement between consecutive poses in the estimated trajectory and compares it with the corresponding relative displacements in the ground truth trajectory. The formal definition of this metric is given in Eq. (2.4).

$$\varepsilon(\delta) = \frac{1}{N}\sum_{i,j} trans\left(\delta_{i,j} \ominus \delta_{i,j}^{*}\right)^{2} + rot\left(\delta_{i,j} \ominus \delta_{i,j}^{*}\right)^{2} \tag{2.4}$$

In Eq. (2.4), $\delta_{i,j}$ represents the relative displacement between the i-th and j-th poses in the estimated trajectory, while $\delta_{i,j}^{*}$ represents the corresponding relative displacement in the ground truth trajectory. The deformation energy $\varepsilon(\delta)$ is defined as the sum of translational and rotational components, both of which are expressed as functions of the difference $\delta_{i,j} \ominus \delta_{i,j}^{*}$ between the two types of relative displacements. Burgard *et al.* [98] originally proposed this metric, which was later extended by Kümmerle *et al.* [99] with a more comprehensive explanation of the approach, a method for extracting relationships from aerial images, and an expanded experimental assessment.

The evaluation metric introduced by Burgard *et al.* [98] has since influenced and laid the foundation for many subsequent works [91, 96, 130]. Sturm *et al.* [130] proposed a similar metric, formulated in Eq. (2.5). As shown in Eq. (2.5), $\hat{x}$ denotes a pose in the ground truth trajectory, x denotes a pose in the estimated trajectory, and i and $i+\triangle$ represent time indices. The core concept of this metric is to compute the error by summing the distances between relative displacements of poses at consecutive time steps.

$$error = \sum_{i=1}^{n}[(\hat{x}_{i+\triangle} \ominus \hat{x}_i) \ominus (x_{i+\triangle} \ominus x_i)]^2 \tag{2.5}$$

Geiger *et al.* [96] further extended the evaluation metric proposed by Kümmerle *et al.* [99] in two key aspects. Instead of combining rotational and translational errors into a single metric, they assessed these errors separately. Additionally, the evaluation was conducted based on trajectory length and velocity, providing a more detailed analysis of localization accuracy. This evaluation metric is formally defined in Eq. (2.6) and Eq. (2.7). As shown in these equations, $\hat{p}$ represents a pose in the ground truth trajectory, p represents a pose in the estimated trajectory, and $\angle[\cdot]$ denotes the rotational angle.

$$E_{rot}(\mathcal{F}) = \frac{1}{|\mathcal{F}|}\sum_{(i,j)\in\mathcal{F}} \angle\left[\left(\hat{p}_j \ominus \hat{p}_i\right) \ominus \left(p_j \ominus p_i\right)\right] \tag{2.6}$$

$$E_{trans}(\mathcal{F}) = \frac{1}{|\mathcal{F}|}\sum_{(i,j)\in\mathcal{F}} \left\|\left(\hat{p}_j \ominus \hat{p}_i\right) \ominus \left(p_j \ominus p_i\right)\right\|_2 \tag{2.7}$$

Sturm *et al.* [91] introduced a variant of the metric proposed by Burgard *et al.* [98], known as the Relative Pose Error (RPE), which has been widely adopted in SLAM benchmark studies [108, 146]. The RPE is formally defined in Eq. (2.8), Eq. (2.9) and Eq. (2.10). As shown in Eq. (2.8), Eq. (2.9), the RPE builds upon the concept of relative displacement introduced by Burgard *et al.* [98]. For all time indices, the relative displacement matrix is first computed by multiplying the 3D homogeneous transformation matrices of poses at two time points separated by a fixed time interval △, for both the ground truth and estimated trajectories. Next, the distance matrix between the relative displacement matrices of the ground truth and the estimated trajectory is obtained through matrix multiplication. Finally, the root mean square (RMS) of the translational components of these distance matrices is computed across all time indices to quantify the error for the chosen time interval △. Furthermore, unlike the metric in Eq. (2.5), which calculates error for only a single fixed time interval, the RPE, as formulated in Eq. (2.10), considers errors for all possible time intervals and computes their average, providing a more comprehensive assessment of trajectory accuracy.

$$E_i := \left(Q_i^{-1} Q_{i+\triangle}\right)^{-1}\left(P_i^{-1} P_{i+\triangle}\right) \tag{2.8}$$

$$RMSE(E_{1:n}, \triangle) := \left(\frac{1}{m}\sum_{i=1}^{m} \|trans(E_i)\|^2\right)^{1/2} \tag{2.9}$$

$$RMSE(E_{1:n}) := \frac{1}{n}\sum_{\triangle=1}^{n} RMSE(E_{1:n}, \triangle) \tag{2.10}$$

The review of evaluation metrics used in localization benchmark concludes here. The next section discusses evaluation metrics for mapping benchmarks.

In mapping benchmarks, error evaluation is performed by directly comparing elements of the SLAM-estimated map (geometry) with those of the ground truth map. The most commonly used evaluation metric is the Euclidean distance between corresponding elements in the estimated and ground truth maps. Since maps (geometries) can be represented as either point clouds or meshes, the Euclidean distance metric is categorized based on the representation format into three types: the point-to-point distance metric [140], the point-to-mesh distance metric [108, 140], and the mesh-to-mesh distance metric.

The point-to-point distance metric first identifies the nearest neighbouring point in the ground truth point cloud for each point in the SLAM-estimated point cloud. The error between the estimated and ground truth maps is then computed as the root mean square (RMS) of pairwise distances between all nearest neighbouring points, similar to the Euclidean distance metric used in localization benchmarking.

For the point-to-mesh distance metric, since a mesh model consists of numerous triangular planes, the process is similar to the point-to-point distance metric. The nearest triangle or point in the ground truth map (geometry) is found for each point or triangle in the SLAM-estimated map. The error is then calculated as the RMS of the pairwise distances between all nearest neighbouring points and triangles in the two maps.

One important aspect to clarify is how the distance between a point and a triangle is defined. The distance between a point and a triangle is determined as the shortest distance from the point to any location on the triangle's surface. This relationship is formally expressed in Eq. (2.11), as defined by Aspert *et al.* [85].

$$d(p, S') = \min_{p' \in S'} \|p - p'\|_2 \tag{2.11}$$

As shown in Eq. (2.11), p represents the point, S' denotes the triangle, p' is a point on the triangle S', and $\|\cdot\|_2$ represents the Euclidean norm. To provide a more intuitive understanding, Figure 2-35 illustrates the geometric interpretation of Eq. (2.11), depicting three possible spatial relationships between the point and the triangle.

The point-to-triangle distance $d(p, S')$ can be categorized into three cases based on these spatial relationships:

- Projection inside the triangle:

If the projection of point p onto the plane containing triangle S' falls within the triangle's boundaries, the point-to-triangle distance is defined as the perpendicular distance from p to the plane.

- Projection outside the triangle:

  If the projection of p falls outside the triangle, the point-to-triangle distance is determined by either:

  - The shortest distance from p to the closest edge of S', or
  - The shortest distance from p to the nearest vertex of S'.

This classification provides a clear framework for computing point-triangle distances based on their geometric relationships.

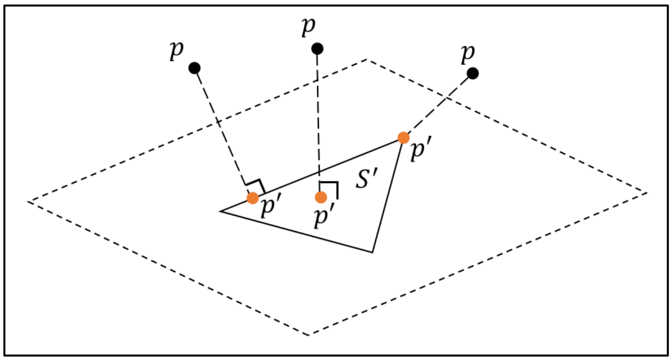


Figure 2-35. Illustration of the distance between a point and a triangle plane in a mesh model.

For the mesh-to-mesh distance metric, the process begins by identifying the nearest triangle plane in the ground truth mesh for each triangle plane in the SLAM-estimated mesh. The error between the estimated and ground truth maps is then computed as the root mean square (RMS) of the pairwise distances between all corresponding nearest neighbouring triangles in the two meshes. To ensure clarity, it is essential to define how the distance between two triangles is measured. Building upon the formula for the distance between a point and a triangle given in Eq. (2.11), the distance from one triangle S to another triangle S' is formally defined in Eq. (2.12), as proposed by Aspert *et al.* [85].

$$d(S,S') = \max_{p\in S} d(p,S') \tag{2.12}$$

It is important to note that this distance is not symmetrical, meaning that the distance from triangle S to triangle S' is not necessarily equal to the distance from S' to S, i.e., $d(S,S') \neq d(S',S)$. To address this asymmetry, it is convenient to define a symmetrical distance between two triangles, as formulated below (Aspert *et al.* [85]):

$$d_s(S,S') = \max[d(S,S'),d(S',S)] \tag{2.13}$$

Beyond the Euclidean distance metric, several other evaluation metrics are used in mapping benchmarks, including the feature-to-feature distance metric and the normal error metric, among others. However, since these metrics are less commonly used, only a brief introduction and some examples of their applications are provided below.

Funke *et al.* [109] employed the feature-to-feature distance metric in their visual-SLAM benchmarking framework. This metric first projects 3D features onto the camera frame using known camera poses to obtain ground truth 2D features. The mapping error is then computed as the mean Euclidean distance between the estimated 2D features and their corresponding ground truth features.

In addition to the point-to-point and point-to-mesh distance metrics, Meister *et al.* [140] utilized the normal error metric to assess the mapping performance of the KinectFusion system. Also referred to as the per-vertex angle error metric, this approach calculates the angular difference between the normal of each vertex in the KinectFusion-generated point cloud and the normal of its closest corresponding vertex in the ground truth point cloud. Since the normal error metric is particularly sensitive to corners and depth discontinuities, it is valuable for evaluating sections critical to certain image processing algorithms.

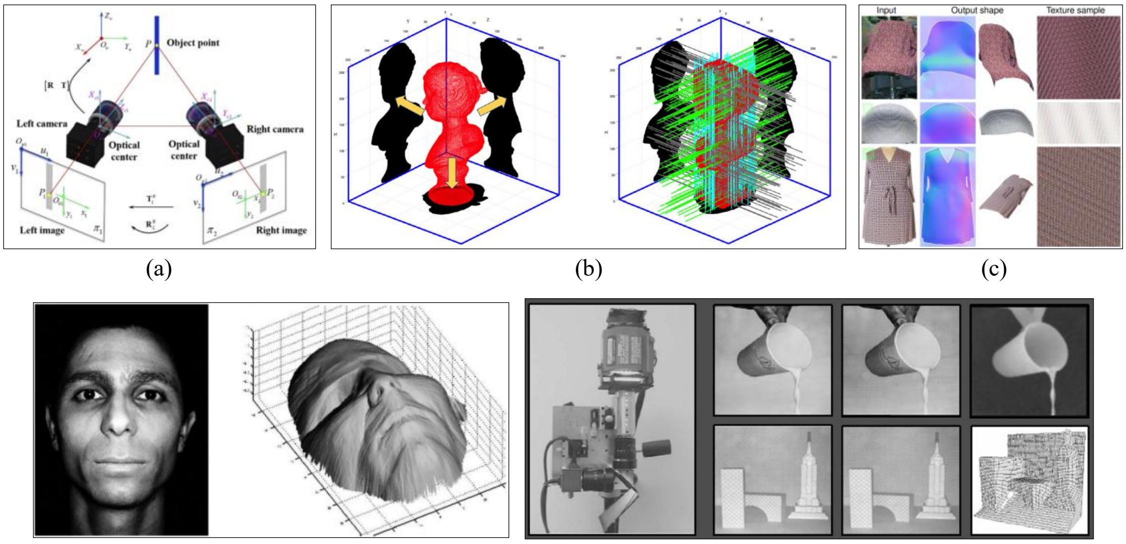
Apart from the quantitative evaluation metrics discussed above, qualitative evaluation metrics based on visual inspection have also been proposed [141, 149-154]. Instead of providing numerical analysis, these methods assess SLAM performance by visually examining and interpreting the output data. Such evaluations are particularly useful for analysing complex phenomena that cannot be easily quantified, often revealing unexpected insights or patterns that might inspire new approaches. However, qualitative assessments rely heavily on subjective interpretation, which introduces potential bias and reduces objectivity. Additionally, they are difficult to standardize across different cases, making comparisons less reliable.

2.2 Review of 3D reconstruction benchmarking

While SLAM benchmarking is primarily concerned with the joint evaluation of localization and mapping, its mapping component often overlaps conceptually with 3D reconstruction. In particular, the methodologies developed in 3D reconstruction benchmarking, although lacking temporal information, provide well-established practices for evaluating structural accuracy in 3D models. Many of the error metrics, datasets, and evaluation protocols from 3D reconstruction benchmarking can be transferred to SLAM mapping evaluation, especially in cases where reliable ground truth maps are difficult to obtain. Therefore, reviewing 3D reconstruction benchmarking here not only complements the discussion of SLAM benchmarking but also highlights transferable insights that can inform the development of holistic and objective SLAM benchmarks.

3D reconstruction is a key area within the field of computer vision, focusing on the process of acquiring the three-dimensional shapes of objects. It is also referred to as shape-from-X, where X represents the specific methodology used for shape acquisition. As exemplified in Figure 2-36, the most commonly employed methodologies for 3D reconstruction include the following:

- shape-from-stereo
- shape-from-silhouette
- shape-from-texture
- shape-from-shading
- shape-from-(de-)focus



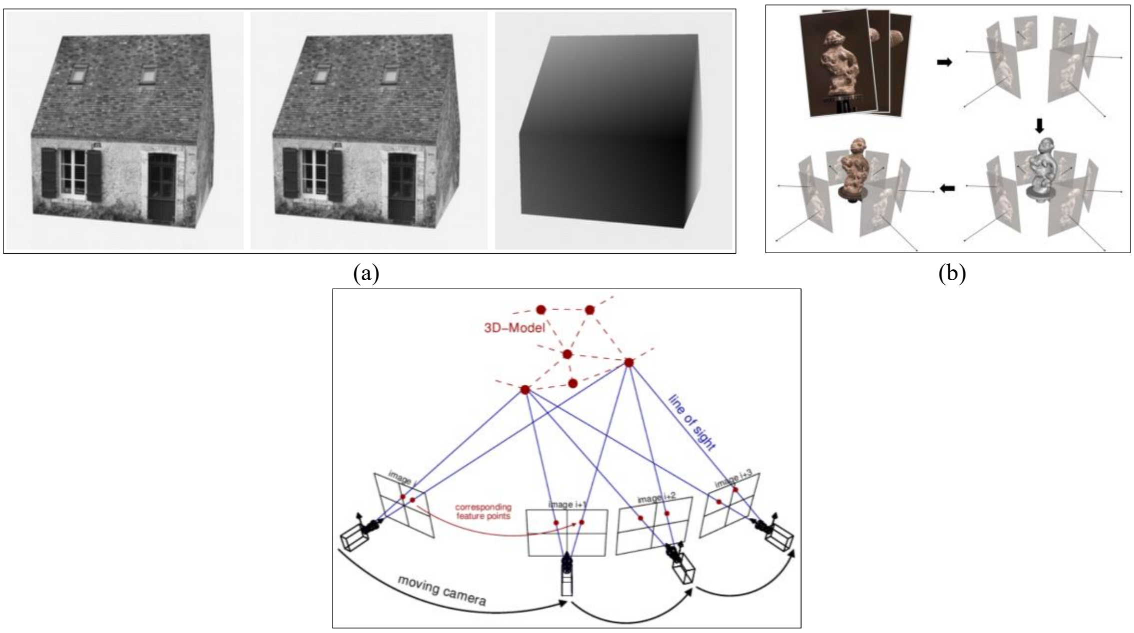
(a) (b) (c)

(d) (e)

Figure 2-36. Common methodologies employed for 3D reconstruction: (a) shape-from-stereo (Zhang *et al.* [155]); (b) shape-from-silhouette (Jang *et al.* [156]); (c) shape-from-texture (Verbin *et al.* [157]); (d) shape-from-shading (Paragios *et al.* [158]); and (e) shape-from-(de-)focus (Nayar *et al.* [159]).

Among these five methods, shape-from-stereo is used most widely. As exemplified in Figure 2-37, it can be further classified into the following three categories:

- Two-view stereo (binocular stereo)
- Multi-view stereo (MVS)
- Structure-from-Motion (SfM)



(a) (b)

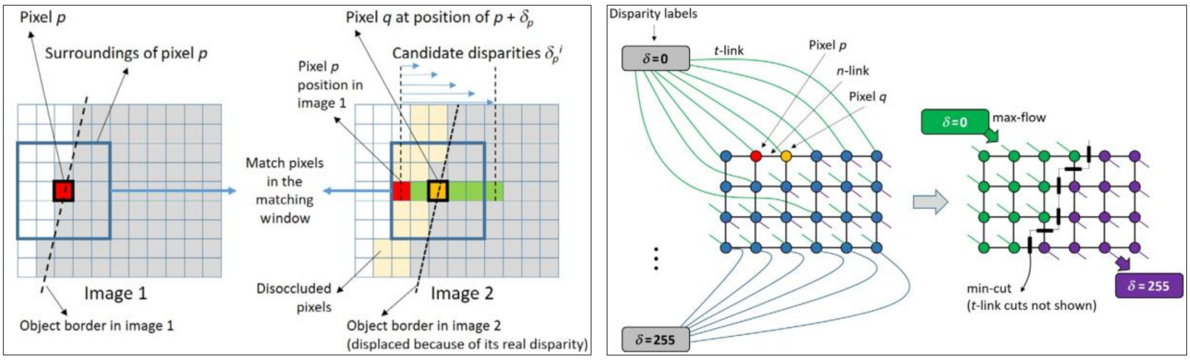
(c)

Figure 2-37. Overview of shape-from-stereo methods. (a) Example of the two-view stereo approach: The left and centre images form the stereoscopic pair, and the right image shows the computed disparity map (Sabater *et al.* [160]). (b) Example of a multi-view stereo pipeline (clockwise from top-left): Input images, posed images, reconstructed 3D geometry, and textured 3D geometry (Furukawa *et al.* [161]). (c) Illustration of Structure-from-Motion (SfM), highlighting its simultaneous estimation of camera poses and the 3D model (Hartley *et al.* [162]).

Over the past few decades, extensive research has been conducted on these three methods, leading to the development of various benchmarking approaches. The following sections review benchmarking efforts for shape-from-stereo methods, following the categorization shown in Figure 2-37. This review focuses on the three key components of benchmarking—imagery datasets, ground truth, and evaluation metrics—in a manner same as the SLAM benchmarking review presented in Section 2.1 .

2.2.1 Benchmark of two-view stereo techniques

Two-view stereo is a dense stereo technique that takes two images with known camera viewpoints as input and generates a single disparity map or depth map as output. The core of this technique is establishing correspondence between the two input images, and the methods used for this task can be categorized into local methods [163, 164] and global methods [165, 166]. The specific techniques employed in these methods are summarized in Figure 2-38.


Pixel p
Surroundings of pixel p
Image 1
Object border in image 1
Pixel q at position of $p + \delta_p$
Candidate disparities δ_p^i
Pixel p position in image 1
Match pixels in the matching window
Disoccluded pixels
Image 2
Object border in image 2 (displaced because of its real disparity)
Disparity labels
δ=0
t-link
Pixel p
n-link
Pixel q
δ=0
max-flow
δ=255
min-cut
(t-link cuts not shown)
δ=255

(a) (b)

Figure 2-38. Overview of the stereo correspondence methods (Chellappa *et al.* [167]). (a) Local method using a sliding-window approach. (b) Global method based on the graph-cut technique.

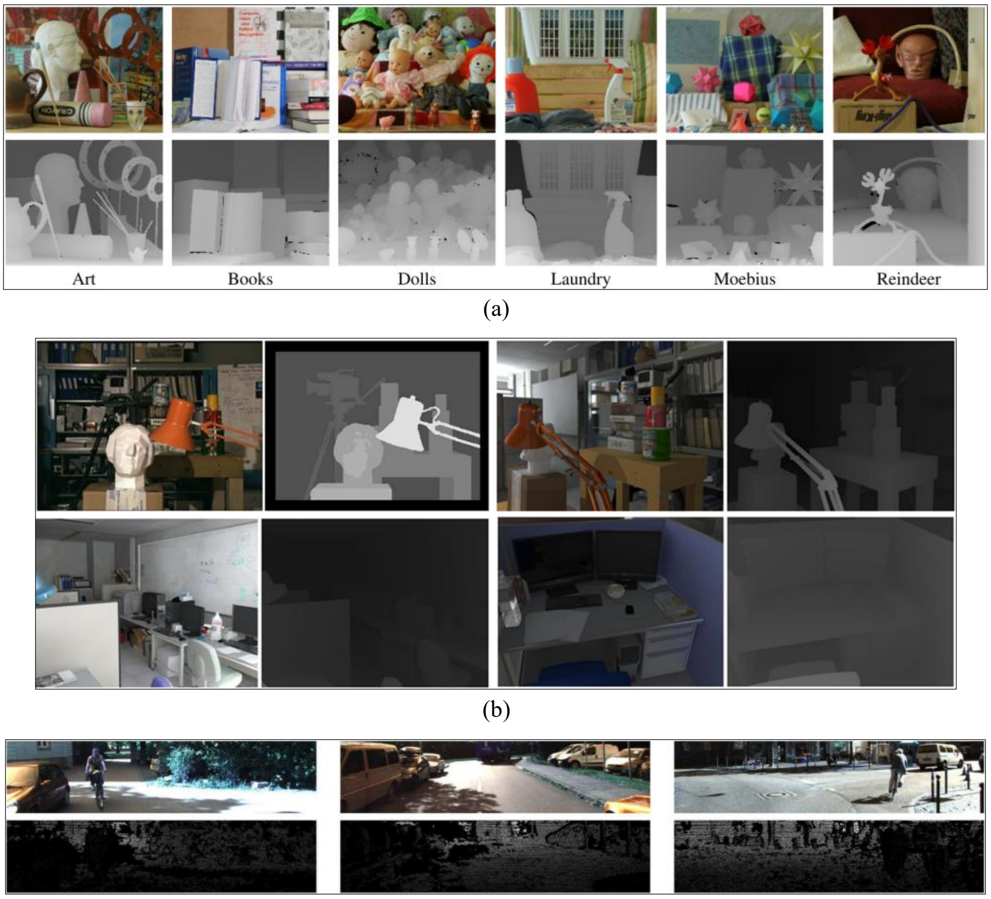
As illustrated in Figure 2-38 (a), local methods focus on individual pixels, determining correspondences in the other image using a window-based approach. In this approach, for each pixel p in image 1, its disparity is determined by comparing its intensity with N candidate pixels in image 2 within a sliding window. The candidate pixel in image 2 with the lowest matching cost is assigned as the corresponding match for pixel p. These methods are computationally efficient since the processing is confined within a local window. However, they struggle in ambiguous regions such as textureless areas, occlusions, and specular surfaces, making them prone to errors.

In contrast, global methods, shown in Figure 2-38 (b), adopt an optimization-based approach called graph-cut technique: A graph is constructed and partitioned into N subgraphs, where each pixel in a subgraph is assigned a disparity label following a specific pattern. This pattern is associated with a matching cost function, and the objective is to determine the disparity labels that minimize the total matching cost function across all subgraphs. Graph-cut technique considers all pixels collectively rather than processing them individually. It first establishes an energy function associated with the disparity map, which is then minimized to determine the optimal disparity values. Since global methods are less sensitive to local ambiguities than local methods, they are predominantly used in stereo matching tasks. A quantitative benchmark comparing several local [168, 169] and global [166] methods was proposed by Szeliski *et al.* [170].

Another approach to categorizing stereo correspondence methods is based on their processing pipeline. This classification breaks down the stereo matching workflow into distinct processing steps, grouping methods accordingly. Benchmarks following this taxonomy parameterize each step with multiple variables, systematically varying them to analyse their impact on performance [171].

The imagery dataset is a crucial component of the benchmarking workflow. It consists of stereo image pairs accompanied by ground truth disparity maps or depth maps. Using the provided stereo pairs, algorithms compute disparity or depth maps, which are then compared against ground truth data to quantify errors. As shown in Figure 2-39, numerous imagery datasets have been introduced over the years and below are some popular ones that are widely used:

- Middlebury [171-175]
- Tsukuba Imagery [176-178]
- KITTI [96, 179]



(a)

(b)

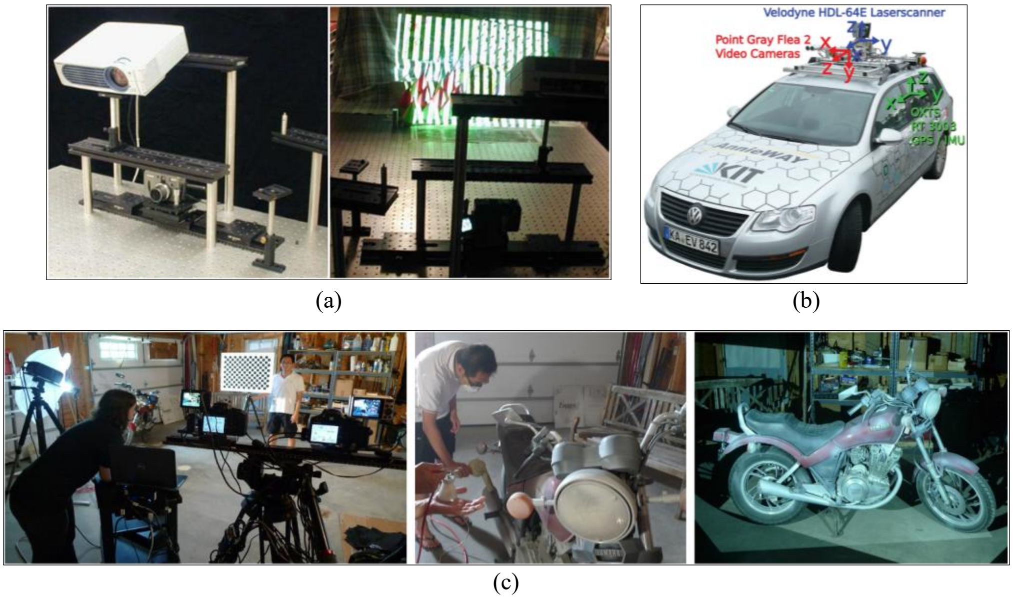
(c)

Figure 2-39. Examples of widely used imagery datasets for two-view stereo. (a) Middlebury dataset (Scharstein *et al.* [174]). Top: colour images. Bottom: ground truth depth images. (b) Tsukuba Imagery dataset (Synthetic CG images) [177, 178]. Colour and ground truth depth images. (c) KITTI dataset (Geiger *et al.* [96]). Top: colour images. Bottom: ground truth depth images.

Imagery datasets used for benchmarking can be broadly categorized into synthetic imagery datasets [177, 178] and real imagery datasets [96, 171-176, 179]. Synthetic imagery datasets are noise-free and provide perfect ground truth upon generation, making them ideal for controlled evaluations. However, their simplified geometries and textures often fail to accurately represent real-world environments, potentially leading to biased benchmark results [180].

On the other hand, real imagery datasets more accurately reflect complex real-world environments, but acquiring ground truth is significantly more challenging. Most real datasets are limited to indoor scenes [171-176], where precise ground truth is more feasible to obtain. This typically involves manual processing [170, 181] and sophisticated experimental setups [74, 172-175, 182, 183], as exemplified in Figure 2-40 (a) and (c). Figure 2-40 (a) shows the setup used by Scharstein *et al.* [172], where a digital camera mounted on a translating stage captured image pairs, and a structured-light projector positioned above uniquely labelled each pixel, enabling accurate correspondence estimation for ground truth. Figure 2-40 (c) illustrates the data collection setup for the Middlebury high-resolution stereo dataset [175], which used two DSLR cameras to capture high-resolution image pairs, while patterned-light projectors generated ground truth disparity maps. In some cases, disparity and depth values must be manually annotated pixel by pixel, making the process extremely time-consuming and labour-intensive [171].

To extend benchmarking to outdoor environments, some outdoor imagery datasets have been introduced [96, 184]. However, acquiring ground truth for these datasets is even more complex, requiring an experimental setup involving LiDAR and multiple cameras, as well as advanced calibration and registration techniques, as exemplified in Figure 2-40 (b). Figure 2-40 (b) shows the data collection platform for the KITTI dataset [111]. Designed for outdoor environments, this platform integrates LiDAR and multiple cameras with specialized calibration and registration techniques. While these datasets provide valuable real-world benchmarking, the demanding setup and high costs make them difficult to reproduce.


Velodyne HDL-64E Laserscanner
Point Gray Flea 2 Video Cameras
OXTS RT 3003 GPS / IMU

(a)

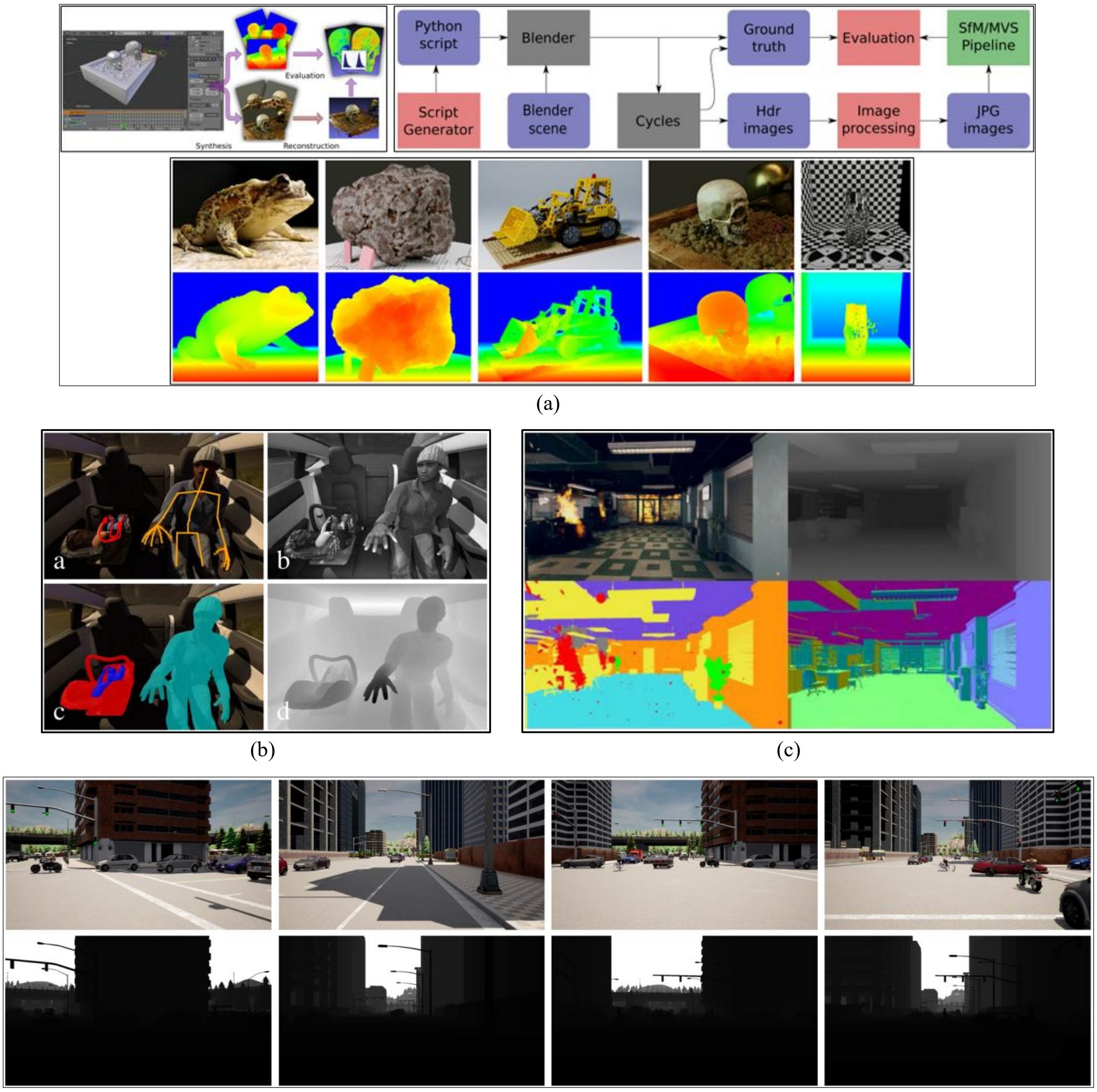
(b)

(c)

Figure 2-40. Equipment setups used for data collection in two-view stereo. (a) Setup from Scharstein *et al.* [172]. (b) The data collection platform for the KITTI dataset (Geiger *et al.* [111]). (c) The data collection setup for the Middlebury high-resolution stereo dataset (Scharstein *et al.* [175]).

To address the abovementioned challenge of acquiring ground truth in benchmarking, several methods [185, 186] have been proposed. Liu *et al.* [185] introduced an approach that approximates 3D geometry to generate estimated disparity ground truth for stereo sequences. Ley *et al.* [186], as illustrated in Figure 2-41 (a), proposed a middle ground between overly simplistic synthetic datasets and real datasets, which are often difficult to produce. Their framework generates synthetic photo-realistic images for benchmarking, providing both perfect ground truth and the ability to simulate real-world imperfections to some extent. This approach has since inspired further research into simulating real-world environments using synthetic data [187-194], with examples shown in Figure 2-41 (b), (c), and (d).

Beyond ground truth generation, other imagery properties also impact benchmarking, including image quality, robustness to noise, and radiometric conditions. The effects of these factors on benchmarking performance have been extensively studied in [173, 195-197].



(a)

(b)

(c)

(d)

Figure 2-41. Examples of synthetic datasets. (a) A synthetic dataset for 3D reconstruction benchmarking (Ley *et al.* [186]). Top: Framework and pipeline for generating the synthetic imagery dataset. Bottom: Rendered colour images and their corresponding ground truth depth images. (b) A synthetic dataset for passenger compartment scenes (Cruz *et al.* [189]). In a clockwise order: Colour image with pose estimation, grayscale infrared imitation, segmentation image and ground truth depth image. (c) A synthetic dataset for disaster response (Jeon *et al.* [191]). In a clockwise order: Colour image, ground truth depth image and segmentation images. (d) A synthetic collaborative perception dataset for autonomous driving (Li *et al.* [193]). Top: Colour images. Bottom: Ground truth depth images.

The final step in the benchmarking workflow is evaluating results using metrics. Various evaluation metrics exist, and they are commonly categorized into two groups which are metrics with ground truth and metrics without ground truth.

For metrics with ground truth, the most commonly used one is pixel-by-pixel comparison between the computed disparity map and the ground truth, followed by calculating the root mean square error

(RMSE) [170, 171]. Another commonly used metric is the Bad Matching Pixel (BMP) [171], which has been applied in numerous studies [198, 199]. This metric is formulated as Eq. (2.14):

$$B = \frac{1}{N}\sum_{(x,y)}(|d_C(x,y) - d_T(x,y)| > \delta_d) \tag{2.14}$$

In Eq. (2.14), $d_C(x,y)$ and $d_T(x,y)$ represent the intensity values at pixel (x,y) in the computed disparity map and the ground truth disparity map, respectively. N denotes the total number of pixels. A threshold δ_d is set, and any pixel where the intensity error $|d_C(x,y) - d_T(x,y)|$ exceeds this threshold is classified as a bad pixel. The Bad Matching Pixel (BMP) ratio, calculated as the proportion of bad pixels to the total number of pixels, serves as an indicator of disparity map quality—where a higher BMP ratio corresponds to lower quality.

However, BMP does not account for error magnitude, which can lead to misleading evaluations. For instance, two disparity maps with identical BMP indices may have vastly different quality levels. To address this issue, Cabezas *et al.* [200] introduced a BMP variant called BMPRE, which integrates BMP with Mean Relative Error (MRE)—the ratio of error magnitude to true disparity values. Experimental results demonstrate that incorporating MRE enables a more objective assessment of disparity maps.

The abovementioned metrics and their variants are fast and easy to implement; however, they do not align well with human visual perception. To overcome this limitation, Wang *et al.* [201] proposed the Structural Similarity Index (SSIM), which evaluates images based on three key attributes: luminance, contrast, and structure. In this system, the three attributes of the computed disparity map are compared against those extracted from the ground truth, and the results are combined to compute the SSIM index. Experimental studies show that SSIM provides a better correlation with human perception of image quality than other traditional metrics. Building on SSIM [201] and its variant MS-SSIM (Multi-scale Structural Similarity Index) [202], Malpica *et al.* [203] proposed R-SSIM (Range SSIM Index) as a quality metric for range images. Visual inspections suggest that R-SSIM more accurately reflects the subjective quality of range images, making it a more credible evaluation method.

Ground truth references are not always available in stereo benchmarks. In such cases, metrics that do not rely on ground truth are required. These metrics can be broadly categorized into two groups which are the prediction error [204] and the confidence measure [181, 205-209]. Prediction error metrics evaluate disparity maps by predicting the appearance of certain images in a dataset based on a subset of available images. The predicted images are then compared against the actual images using various error metrics. This approach has been widely adopted in several studies [170, 171, 183, 210, 211] as a viable evaluation method when ground truth data is unavailable.

Originally developed as cost functions in stereo matching to determine correspondences between images, confidence measures can also serve as error metrics. Over the years, several confidence measures have been introduced and applied in numerous studies [181, 184, 212-214], including:

- SVS (Single View Stereo): [181]
- LRC (Left Right Consistency): [215-220]
- MSM (Matching Score Measure): [221, 222]
- CUR (Curvature): [223-225]
- Entropy-like confidence measure: [226, 227]
- PKR (Peak Ratio): [215, 228]
- MLM (Maximum Likelihood Metric): [229]
- WMN (Winner Margin): [230]

Comprehensive evaluations of these confidence measures under various conditions, including occlusions, discontinuities, and textureless regions, have been conducted in [231-233].

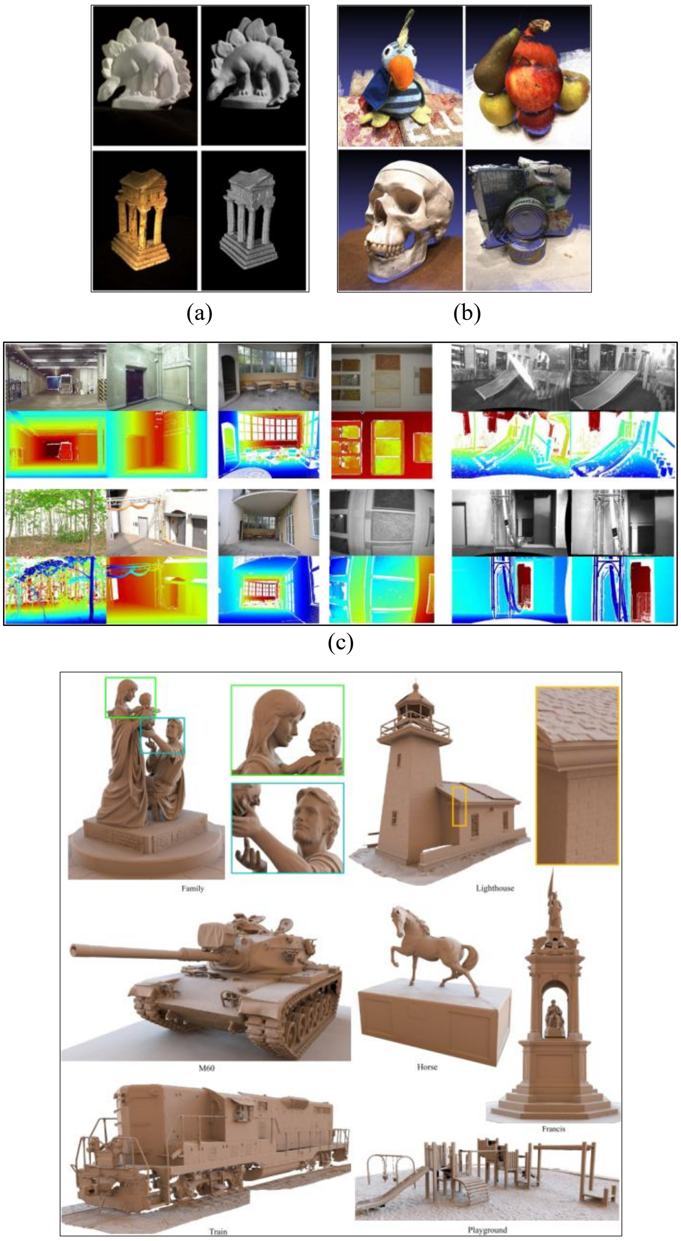
2.2.2 Benchmark of multi-view stereo (MVS) techniques

Similar to two-view stereo, multi-view stereo (MVS) is a technique that processes multiple images (more than two) with known camera viewpoints to generate 3D object models. MVS methods differ in various ways, and their classification follows a structure similar to the taxonomy and evaluation of two-view stereo methods proposed by Scharstein *et al.* [171]. A comprehensive taxonomy and evaluation of different MVS methods were later introduced by Seitz *et al.* [73].

Datasets also play a crucial role in MVS benchmarking. Some widely used MVS datasets, also exemplified in Figure 2-42, include:

- Middlebury multi-view dataset [73]

- EPFL multi-view evaluation dataset [234]
- DTU Robot MVS dataset [235]
- ETH3D multi-view stereo benchmark dataset [236]
- Tanks and Temples benchmark dataset [237]
- KITTI multi-view dataset [96]



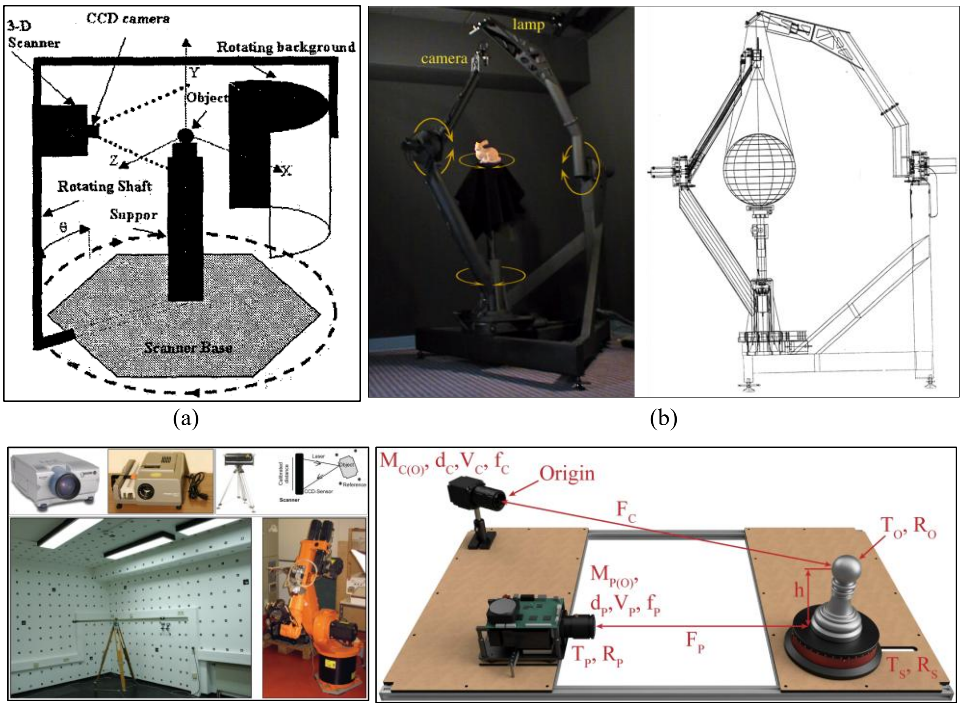
(a) (b)

(c)

(d)

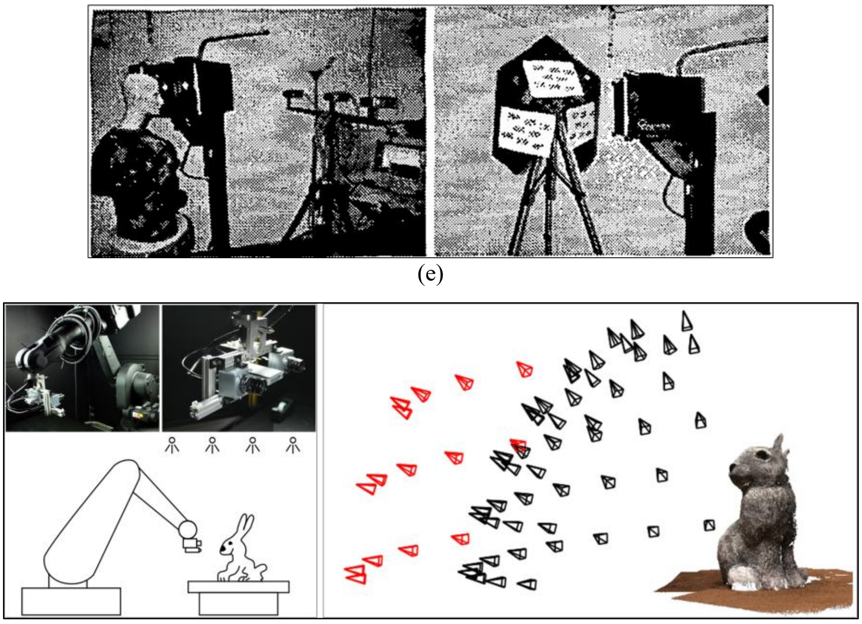
Figure 2-42. Examples of multi-view stereo datasets. (a) Middlebury multi-view dataset (Seitz *et al.* [73]). Two objects and their laser-scanned ground truth 3D models. (b) DTU Robot MVS dataset (Jensen *et al.* [235]). Point clouds of ground truth 3D models obtained by structure-light scanning. (c) ETH3D multi-view stereo benchmark dataset (Schops *et al.* [236]). A subset of laser-scanned ground truth 3D point cloud renderings and high-resolution images captured from different viewpoints in the dataset. (d) Tanks and Temples benchmark dataset (Knapitsch *et al.* [237]). A subset of laser-scanned ground truth 3D models in the dataset.

Datasets used for benchmarking multi-view stereo (MVS) consist of image sequences and ground truth 3D models of objects or scenes. The acquisition process varies depending on the scale and complexity of the objects or environments being reconstructed. For small-scale objects such as statuettes, figurines, and miniatures, image sequences are typically captured using high-precision motion platforms, including robotic arms [235] and gantries [73], as shown in Figure 2-43. These controlled setups ensure accurate and consistent viewpoints. For large-scale scenes and objects—such as architectures, statues, monuments, and building interiors—image sequences are often acquired through terrestrial or aerial photography [234, 237-239], which allows for extensive coverage of complex environments. Ground truth 3D models are obtained using high-precision measurement tools, such as laser scanners [73, 236, 237], LIDAR [234] or structured light scanners [235], depending on the dataset requirements. For large-scale scenes or objects, multiple individual scans are typically required to capture the full geometry, as illustrated in Figure 2-42 (c) and (d). Once acquired, individual scans can be used directly for benchmarking [238]. However, in most cases, the scans are aligned and merged into a single complete 3D model before evaluation. This fusion step ensures consistency and reduces errors caused by misalignment between separate scans.



(a) (b)

(c) (d)



(e)

(f)

Figure 2-43. Equipment setups used for data collection in multi-view stereo. (a) Setup from Eid *et al.* [240]: A scanner was used to obtain the ground truth 3D model of the object, while a camera captured images from different angles. Both the scanner and camera were mounted on a rotating platform with a rigid connection, ensuring that the relative positions between the captured images and the scanned 3D model were precisely known. (b) The Stanford Spherical Gantry (Levoy *et al.* [241]). A room-sized goniometer with two arms revolving concentrically around an object platform. Its working principle is similar to that of Figure 2-43 (a), simultaneously capturing images and ground truth 3D models with known relative positions. (c) Setup from Anke *et al.* [182]: A laser scanner was used to obtain the ground truth 3D model, while a projector encoded the images to derive ground truth disparity. A robotic arm was employed to capture images at predefined positions. (d) Setup from Kock *et al.* [192]: Coded patterns were projected onto the object using a projector to establish correspondences between camera and projector pixels. (e) Performance evaluation of 3D reconstruction for tele-presence [183]. A laser scanner was used to obtain the ground truth 3D model of a mannequin. (f) Setup from Jensen *et al.* [235]: Similar to Figure 2-43 (a), this setup employed a stereo camera and scanner mounted with a rigid link, capturing the object from different angles while maintaining known relative positions between them.

Beyond acquiring images and ground truth 3D models, camera calibration and image-to-model registration are also essential steps in MVS benchmarking. As illustrated in Figure 2-43 (a), (b), (d), and (f), when image sequences are captured using high-precision motion platforms (e.g., robotic arms [73] or gantries [235]), camera calibration is straightforward since the camera parameters are precisely controlled. If the camera and scanner are rigidly connected via fixed joints, registration is easily achieved through simple affine transformations [235]. However, if there is no fixed relationship between the camera and scanner, the scanned 3D model must be registered to the images using optimization-based approaches that minimize the alignment cost function between the images and the 3D model [73].

For large-scale scenes and objects, where images are captured via terrestrial or aerial photography, the process becomes more complex due to unconstrained camera movement. In such cases, camera calibration is typically obtained with Structure-from-Motion (SfM) techniques [236, 238, 239] or aid from the scanned ground truth 3D model [234, 237]. SfM estimates camera parameters directly from the images without requiring additional hardware. It first detects and matches corresponding features across multiple images, then estimates camera poses using triangulation and Perspective-n-Point (PnP) algorithms and finally refines the results using optimization techniques such as bundle adjustment. Over the years, SfM has undergone significant advancements [242] and is now embedded as a core function in many 3D reconstruction software packages, including COLMAP [242, 243], OpenMVG [244], and VisualSFM [245], as shown in Figure 2-44. These tools have significantly improved the accuracy and automation of the camera calibration process, making them widely used in multi-view stereo benchmarking.

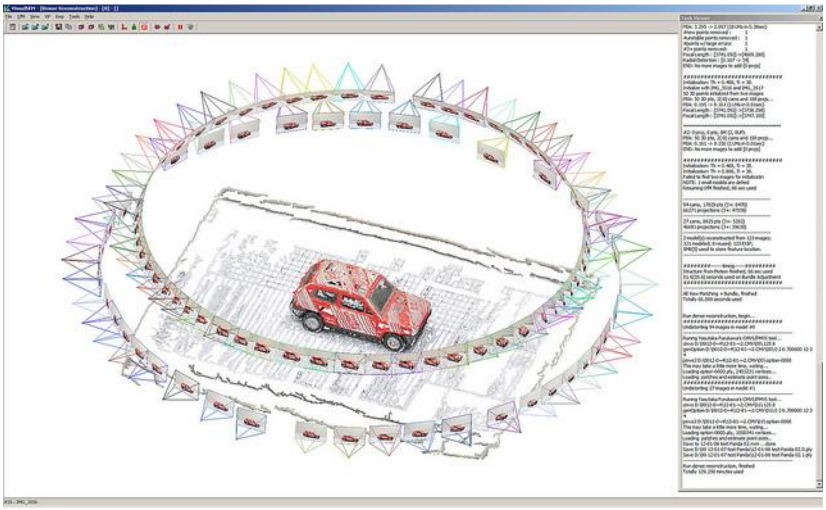


Figure 2-44. Interface of VisualSFM [246]. Camera poses are estimated from the imported image sequence.

In 3D reconstruction software like the ones just mentioned, camera calibrations are automatically computed upon image input, significantly streamlining the benchmarking workflow and improving efficiency. Numerous studies have evaluated and benchmarked different 3D reconstruction software [247-249], and interested readers can refer to these works for more details.

Beyond automated methods, scanned ground truth 3D models can also be used to obtain camera calibrations [234, 237]. As illustrated in Figure 2-45, in [234], 2D targets were manually placed within the scene in positions visible in both LiDAR data and images. The 3D coordinates of these targets, provided by the LiDAR scan, were then used to compute camera calibration parameters.

In addition to calibration, image-to-model registration can also be leveraged to obtain or refine camera parameters [236, 237]. In this process, the camera calibration and orientation of the 3D model are adjusted to minimize the projection error between the 3D model and the images. Several methods have been proposed to address this optimization problem [250-258], ensuring more accurate alignment between the reconstructed model and the input images.

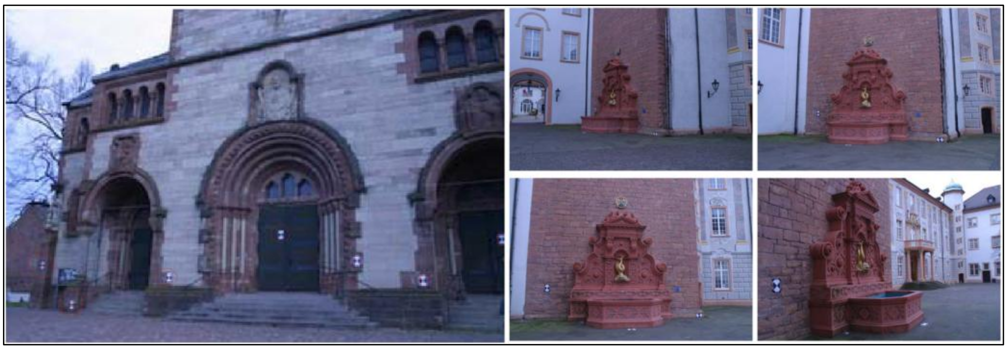


Figure 2-45. Benchmark setup in Strecha *et al.* [234]: Multiple 2D targets (highlighted with red circles) were placed in the scene, ensuring visibility to both cameras and LiDAR. This arrangement enables camera calibration using the scanned ground truth point cloud.

Evaluation metrics play a crucial role in assessing the quality of reconstructed 3D models. One of the most common and straightforward evaluation methods is computing the root mean square error (RMSE) between the ground truth 3D model (*G*) and the reconstructed model (*R*) using 3D processing software such as CloudCompare [87] and MeshLab [259]. This approach has been applied in several studies [238, 239]. However, a major drawback of RMSE is its sensitivity to outliers, which can lead to biased evaluations and misrepresent the true quality of the reconstruction.

To address this limitation, accuracy and completeness, two widely used evaluation metrics introduced by Seitz *et al.* [73], provide a more robust assessment. Accuracy measures how closely the reconstructed model *R* aligns with the ground truth model *G*. It is computed by identifying the nearest points in *G* from *R* and then determining a distance threshold d such that $X\%$ of the points in *R* lie within d of *G*. Completeness, on the other hand, evaluates how well the ground truth *G* has been reconstructed. Given the same distance threshold d, the points in G that fall within d of R are considered well-reconstructed, and their proportion relative to the entire *G* defines the completeness metric. These two metrics have been widely adopted in various benchmarks [234-237] and are primarily used for comparisons between ground truth and reconstructed 3D models. However, they can also be applied to depth map evaluations by comparing ground truth depth maps with estimated depth maps [234].

In addition to these metrics, the F-score, which combines accuracy and completeness, has also been used in some studies [236, 237]. The F-score provides a balanced evaluation by considering both aspects simultaneously and is formally defined in Eq. (2.15).

$$F(d) = \frac{2P(d)R(d)}{P(d)+R(d)} \tag{2.15}$$

In Eq. (2.15), for a given threshold d, precision $P(d)$ represents the accuracy, while recall $R(d)$ corresponds to completeness at that threshold. The F-score is advantageous because it computes the harmonic mean of precision and recall, rather than their arithmetic mean. This characteristic makes it a more reliable metric, as it approaches zero when either precision or recall is low, ensuring that a poor performance in one aspect is not masked by a high value in the other.

2.2.3 Benchmark of Structure-from-Motion (SfM) techniques

Structure-from-Motion (SfM) is a process that estimates camera poses and reconstructs the 3D structure of an environment from a sequence of images, without requiring any prior knowledge of the scene. The output is typically a sparse point cloud representation of the environment. SfM begins by detecting and matching features across images. Using these matched features, it then determines camera poses and 3D structures through epipolar geometry, triangulation, and Perspective-n-Point (PnP) algorithms. The software and pipelines discussed in Section 2.2.2 provide tools for performing SfM tasks, and benchmarks evaluating their performance have been proposed in [260, 261]. SfM benchmarking typically consists of evaluating camera pose estimation [262] and assessing 3D reconstruction quality [260-262]. The benchmarking methodology is similar to that used in Multi-View Stereo (MVS), where the estimated camera poses or reconstructed models are first aligned with ground truth data, followed by the computation of distance errors to assess accuracy. In addition to general performance evaluations, SfM benchmarks are widely used in various applications, including architecture, ecology, and other fields. A summary of these applications is presented in Table 2-1.

Table 2-1. Overview of SfM benchmark applications

Applications	References

Architecture	[263-265]
Ecology	[266, 267]
Medical imaging	[268]
Maritime science	[269-271]
Geoscience	[272-279]
Agriculture	[280]

2.3 Research gap and key innovations

2.3.1 Research gap

As discussed before, SLAM benchmarking is divided into localization benchmarking and mapping benchmarking. However, a fundamental requirement for any meaningful SLAM evaluation is that benchmarking must be both holistic and objective. Holism ensures that localization and mapping are assessed together as an integrated process, rather than as two independent tasks. Objectivity guarantees that the benchmark results truly reflect the intrinsic performance of the SLAM system, free from biases introduced by artificial manipulations such as manual alignment. Without holism and objectivity, benchmarking results can be misleading, providing an incomplete or distorted view of a system's real capabilities. Therefore, a comprehensive SLAM benchmark should consider both aspects; however, the literature review reveals that most existing SLAM benchmarks lack holism. They primarily focus on localization performance, neglecting the assessment of mapping accuracy due to the unavailability of ground truth maps.

Beyond completeness, objectivity is a critical requirement for SLAM benchmarking. The benchmark results must accurately reflect the true performance of a SLAM system. Since localization and mapping are inherently interconnected, evaluating them separately fails to capture their mutual influence. An objective benchmark not only assesses performance accurately but also preserves and highlights localization-mapping correlations, which can be leveraged for mapping accuracy improvement, a key focus of this work.

To ensure objectivity, the original pose relationships among estimated poses, ground truth poses, estimated maps, and ground truth maps must be preserved throughout the benchmarking process. These pose relationships reflect the true performance of the SLAM system and the interdependence between localization and mapping. Any perturbation to these relationships introduces bias, distorting the benchmark results and obscuring valuable insights.

In localization benchmarks, objectivity is maintained by transforming both the estimated and ground truth trajectories into a common coordinate frame, ensuring that the original pose relationships remain intact. The error is then computed based on the discrepancy between the two transformed trajectories using well-established evaluation metrics.

However, existing mapping benchmarks [71-75] often introduce bias. Instead of preserving the original pose relationships, they rely on manual alignment between the estimated and ground truth maps using various 3D registration techniques [76-83]. This alignment process modifies the relative poses, thereby distorting the correlations between localization and mapping and resulting in biased SLAM benchmark results. These observations clearly demonstrate that only a holistic and objective benchmarking framework can provide accurate, unbiased, and practically useful insights into SLAM performance, underscoring its central importance for advancing the field.

2.3.2 Key innovations of this work

To address these limitations, this work proposes novel methodologies for holistic and objective SLAM benchmarking and mapping accuracy improvement. The proposed SLAM benchmark:

- Evaluates localization and mapping as a unified process rather than treating them separately.
- Uses a global coordinate system, ensuring that original pose relationships among estimated trajectories, ground truth trajectories, estimated maps, and ground truth maps remain intact.
- Leverages benchmark results as feedback to enhance mapping accuracy, refining the estimated map based on localization-mapping correlations.

The mapping accuracy improvement is achieved by optimizing the estimated trajectory to minimize errors between the estimated and ground truth trajectories and then applying this optimization to the estimated map to improve its accuracy.

2.3.3 CPR-ICP: A novel registration method for mapping accuracy enhancement

To minimize errors and enhance mapping accuracy, this work introduces a novel ICP (Iterative Closest Point) variant, called CPR-ICP. Unlike classic ICP, CPR-ICP:

- Pre-aligns centroids and least-square planes of the point clouds before performing additional refinements.
- Applies classic ICP techniques for final alignment, leading to superior registration accuracy.

Experimental results demonstrate that CPR-ICP outperforms classic ICP in point cloud registration, significantly enhancing mapping accuracy in SLAM applications.

2.3.4 Implications for ground truth map acquisition in large-scale scenes

To the best of our knowledge, this is the first work that not only benchmarks both localization and mapping objectively but also refines the estimated map, enhancing its accuracy. This contribution provides new insights into the acquisition of ground truth maps for room-sized and large-scale environments, a major challenge in SLAM research.

Traditionally, obtaining ground truth maps requires high-precision 3D scanners, which are primarily designed for small-to-medium-sized objects. These scanners are not practical for large-scale environments, which are frequently encountered in SLAM applications. The mapping accuracy improvement method proposed in this work addresses this gap by utilizing SLAM benchmark results and localization-mapping correlations to generate accurate maps. These generated maps have the potential to serve as ground truth for large-scale environments, particularly for inaccessible locations such as nuclear facilities, industrial plants, and chemical pipelines.

By introducing a holistic and objective SLAM benchmarking framework and a novel mapping accuracy enhancement method, this work addresses critical gaps in existing benchmarks. The proposed approach not only ensures accurate SLAM performance evaluation but also facilitates ground truth map generation for large-scale scenes, expanding the applicability of SLAM in challenging environments.

Chapter 3 Mathematical Preliminaries

This chapter presents the mathematical preliminaries for the innovative methodologies for SLAM benchmarking and improving mapping accuracy discussed in the next two chapters. These preliminaries provide the foundation necessary for understanding the proposed novel methodologies and the subsequent development and implementation of them. The chapter begins with Section 3.1 , which introduces Euclidean transformation as the mathematical basis for representing rigid-body transformations in 3D space, including rotations and translations. This is critical for aligning and comparing trajectories and maps in the proposed SLAM benchmark. Next, Section 3.2 covers the Iterative Closest Point (ICP) algorithm, explaining its mathematical formulation and role in point cloud registration, map refinement, and error minimization. The discussion also highlights the limitations of classic ICP algorithm and its adaptation for improving mapping accuracy.

3.1 Euclidean transformation

Euclidean transformations, also known as rigid transformations, are fundamental in geometry, computer graphics, computer vision, and various other fields where manipulating geometric objects while preserving their properties is essential. These transformations provide a mathematical framework for understanding how objects can be moved or reoriented without altering their intrinsic geometric properties within Euclidean space.

Euclidean space is a geometric space that adheres to the principles of Euclidean geometry, incorporating fundamental notions such as distance and angles. Euclidean transformations encompass rotations, translations, and reflections, all of which preserve these fundamental properties. However, in this research, only rotations and translations are considered.

In three-dimensional Euclidean space, rotations are represented by 3×3 orthogonal matrices $\boldsymbol{R}$, where both rows and columns consist of mutually orthogonal unit vectors. These vectors correspond to the coordinates of the three standard basis vectors of one coordinate frame when expressed in another frame. Translations are represented by three-dimensional vectors $\boldsymbol{t}$, which define the position of the origin of one coordinate frame relative to another.

Conceptually, rotations and translations define the relationship between two coordinate frames. They allow for the transformation of point coordinates from one frame to another, making them indispensable for applications involving geometric transformations and spatial analysis.

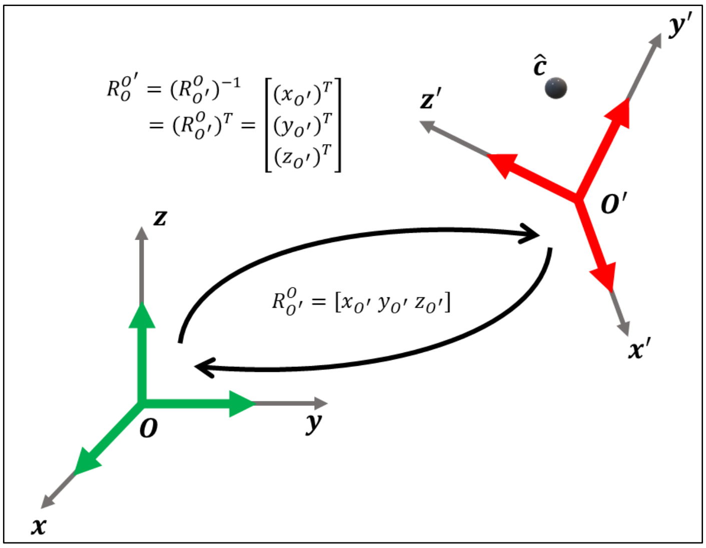

$R_{O}^{O'} = (R_{O'}^{O})^{-1} = (R_{O'}^{O})^{T} = \begin{bmatrix} (x_{O'})^T \\ (y_{O'})^T \\ (z_{O'})^T \end{bmatrix}$
y'
$\hat{c}$
z'
O'
z
$R_{O'}^{O} = [x_{O'}\ y_{O'}\ z_{O'}]$
x'
O
y
x

Figure 3-1. Transformation between two coordinate frames.

For instance, as illustrated in Figure 3-1, the three standard basis vectors of frame $\boldsymbol{O'}$ relative to frame $\boldsymbol{O}$ are denoted as $\boldsymbol{x_{O'}}$, $\boldsymbol{y_{O'}}$ and $\boldsymbol{z_{O'}}$, respectively. These vectors satisfy the following conditions:

$$(\boldsymbol{x_{O'}})^{\boldsymbol{T}}\boldsymbol{x_{O'}} = 1 \qquad (\boldsymbol{y_{O'}})^{\boldsymbol{T}}\boldsymbol{y_{O'}} = 1 \qquad (\boldsymbol{z_{O'}})^{\boldsymbol{T}}\boldsymbol{z_{O'}} = 1 \tag{3.1}$$

$$(\boldsymbol{x_{O'}})^{\boldsymbol{T}}\boldsymbol{y_{O'}} = 0 \qquad (\boldsymbol{x_{O'}})^{\boldsymbol{T}}\boldsymbol{z_{O'}} = 0 \qquad (\boldsymbol{y_{O'}})^{\boldsymbol{T}}\boldsymbol{z_{O'}} = 0 \tag{3.2}$$

The matrix $\boldsymbol{R_{O'}^{O}} = [\boldsymbol{x_{O'}}\ \boldsymbol{y_{O'}}\ \boldsymbol{z_{O'}}]$ they form is the rotation matrix that transfers points from coordinate frame $\boldsymbol{O'}$ to coordinate frame $\boldsymbol{O}$. According to Eq. (3.1) and Eq. (3.2), $\boldsymbol{R_{O'}^{O}}$ meets the following conditions:

$$(\boldsymbol{R_{O'}^{O}})^{\boldsymbol{T}}\boldsymbol{R_{O'}^{O}} = \boldsymbol{I} \tag{3.3}$$

$$\boldsymbol{R_{O'}^{O}}(\boldsymbol{R_{O'}^{O}})^{\boldsymbol{T}} = \boldsymbol{I} \tag{3.4}$$

$$(\boldsymbol{R_{O'}^{O}})^{\boldsymbol{T}} = (\boldsymbol{R_{O'}^{O}})^{-1} = \boldsymbol{R_{O}^{O'}} \tag{3.5}$$

Since $\boldsymbol{R_{O'}^{O}}$ is the rotation matrix that transfers points from coordinate frame $\boldsymbol{O'}$ to coordinate frame $\boldsymbol{O}$, its inverse matrix $(\boldsymbol{R_{O'}^{O}})^{-1}$ is the rotation matrix $\boldsymbol{R_{O}^{O'}}$, which transfers points from coordinate frame $\boldsymbol{O}$ to coordinate frame $\boldsymbol{O'}$; therefore, according to Eq. (3.5), the three columns of $(\boldsymbol{R_{O'}^{O}})^{-1}$ equal the three rows of $\boldsymbol{R_{O'}^{O}}$, which represent the coordinates of three standard basis vectors of frame $\boldsymbol{O}$ with respect to frame $\boldsymbol{O'}$. The coordinate of origin $\boldsymbol{O'}$ with respect to frame $\boldsymbol{O}$ is denoted as $\boldsymbol{t_{O'}^{O}}$, serving as the translation matrix that transfers points from coordinate frame $\boldsymbol{O'}$ to coordinate frame $\boldsymbol{O}$; vice versa, the coordinate of origin $\boldsymbol{O}$ with respect to frame $\boldsymbol{O'}$ is denoted as $\boldsymbol{t_{O}^{O'}}$, serving as the translation matrix that transfers points represented in coordinate frame $\boldsymbol{O}$ to coordinate frame $\boldsymbol{O'}$.

Given a point $\boldsymbol{\hat{c}}$, if its coordinate in frame $\boldsymbol{O'}$ is denoted as $\boldsymbol{\hat{c}_{O'}}$, its coordinate in frame $\boldsymbol{O}$ can be expressed using Eq. (3.6):

$$\boldsymbol{\hat{c}_{O}} = \boldsymbol{R_{O'}^{O}}\boldsymbol{\hat{c}_{O'}} + \boldsymbol{t_{O'}^{O}} \tag{3.6}$$

Vice versa, the transformation from frame $\boldsymbol{O}$ to frame $\boldsymbol{O'}$ can be represented as Eq. (3.7):

$$\boldsymbol{\hat{c}_{O'}} = \boldsymbol{R_{O}^{O'}}\boldsymbol{\hat{c}_{O}} + \boldsymbol{t_{O}^{O'}} \tag{3.7}$$

From Eq. (3.6) and Eq. (3.7), the following equations can be derived:

$$\boldsymbol{t_{O'}^{O}} = -\boldsymbol{R_{O'}^{O}}\boldsymbol{t_{O}^{O'}} \tag{3.8}$$

$$\boldsymbol{t_{O}^{O'}} = -\boldsymbol{R_{O}^{O'}}\boldsymbol{t_{O'}^{O}} \tag{3.9}$$

Eq. (3.8) and Eq. (3.9) align with the fundamental role of the rotation matrix $\boldsymbol{R}$, which is to transform the coordinates of vector $\boldsymbol{t}$ from one coordinate frame to another.

Eq. (3.6) and Eq. (3.7) incorporate both rotation and translation, and are therefore commonly expressed in homogeneous coordinates, as shown in Eq. (3.10) and Eq. (3.11):

$$\boldsymbol{T_{O'}^{O}} = \begin{bmatrix} \boldsymbol{R_{O'}^{O}} & \boldsymbol{t_{O'}^{O}} \\ \boldsymbol{0} & 1 \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{\hat{c}_{O}} \\ 1 \end{bmatrix} = \boldsymbol{T_{O'}^{O}} \begin{bmatrix} \boldsymbol{\hat{c}_{O'}} \\ 1 \end{bmatrix} \tag{3.10}$$

$$\boldsymbol{T_{O}^{O'}} = \begin{bmatrix} \boldsymbol{R_{O}^{O'}} & \boldsymbol{t_{O}^{O'}} \\ \boldsymbol{0} & 1 \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{\hat{c}_{O'}} \\ 1 \end{bmatrix} = \boldsymbol{T_{O}^{O'}} \begin{bmatrix} \boldsymbol{\hat{c}_{O}} \\ 1 \end{bmatrix} \tag{3.11}$$

In the homogeneous forms, transformations are represented as $\boldsymbol{T_{O'}^{O}}$ and $\boldsymbol{T_{O}^{O'}}$, incorporating both rotation and translation. Points are expressed in homogeneous coordinates, where an additional 1 is appended to the coordinate, converting it into a 4×1 matrix.

3.2 Iterative Closest Point (ICP)

The Iterative Closest Point (ICP) algorithm is a fundamental technique for aligning 3D point clouds, widely used in computer vision, robotics, and 3D reconstruction. Its primary goal is to iteratively refine the transformation that minimizes the distance between two point clouds, enabling accurate alignment. ICP is particularly useful in applications such as registering 3D scans (as shown in Figure 3-2), merging data from different sensors, and enhancing the spatial consistency of objects within a scene. Due to its versatility and effectiveness, ICP has become a standard approach for precise geometric alignment in various domains.

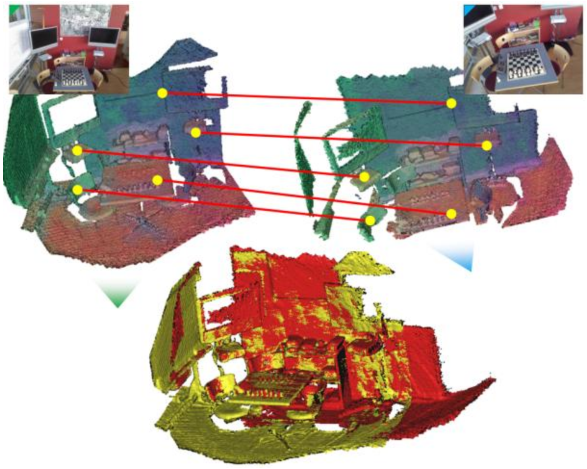


Figure 3-2. Example of registering two 3D scans [281]. The aim of registration is to align the scans by minimizing the distance difference between them.

The Iterative Closest Point (ICP) algorithm is commonly used to align two point clouds by iteratively refining their transformation until an optimal alignment is achieved. In this process, one point cloud is designated as the source point cloud, which remains fixed, while the other is defined as the target point cloud, which is transformed to minimize the distance between the two. ICP operates through a repetitive optimization process, alternating between two key steps which are establishing correspondences between points in the two datasets and updating the transformation parameters to minimize the alignment error.

To better understand how ICP refines the transformation in each iteration, consider a general case in an m-dimensional space, where correspondences between two datasets have been established. The datasets can be represented as two point set matrices $\boldsymbol{A} = [\boldsymbol{a_1} \dots \boldsymbol{a_N}] \in \mathbb{R}^{m \times N}$ and $\boldsymbol{B} = [\boldsymbol{b_1} \dots \boldsymbol{b_N}] \in \mathbb{R}^{m \times N}$. The discrepancy between $\boldsymbol{A}$ and $\boldsymbol{B}$ is quantified by the mean squared error of their distances, expressed as a cost function $f(\boldsymbol{R}, \boldsymbol{t})$, which is defined as Eq. (3.12):

$$f(\boldsymbol{R}, \boldsymbol{t}) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{b_i} - \boldsymbol{R}\boldsymbol{a_i} - \boldsymbol{t}\|_2 \tag{3.12}$$

The goal is to determine $\boldsymbol{R}$ and $\boldsymbol{t}$ such that Eq. (3.13) is satisfied, where $\boldsymbol{I_m}$ represents an $m \times m$ identity matrix.

$$< \boldsymbol{R}, \boldsymbol{t} >= min_{\boldsymbol{R},\boldsymbol{t}} f(\boldsymbol{R}, \boldsymbol{t}), s.t.\ \boldsymbol{R}\boldsymbol{R}^{\boldsymbol{T}} = \boldsymbol{I}_{\boldsymbol{m}} \tag{3.13}$$

In Eq. (3.12), $\|\cdot\|_2$ represents the l^2-norm of a vector. If $\boldsymbol{b}_{\boldsymbol{i}} - \boldsymbol{R}\boldsymbol{a}_{\boldsymbol{i}} - \boldsymbol{t} = [\boldsymbol{c}_{\boldsymbol{1}} \dots \boldsymbol{c}_{\boldsymbol{m}}]^{\boldsymbol{T}} \in \mathbb{R}^{m\times 1}$, then $\|\boldsymbol{b}_{\boldsymbol{i}} - \boldsymbol{R}\boldsymbol{a}_{\boldsymbol{i}} - \boldsymbol{t}\|_2 = \sqrt{\sum_{i=1}^{m} |\boldsymbol{c}_{\boldsymbol{i}}|^2}$. Since N is a constant, Eq. (3.12) can also be expressed as Eq. (3.14), in which $\boldsymbol{1} = [1, \dots, 1]^T$:

$$f(\boldsymbol{R}, \boldsymbol{t}) = \left\|\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A} - \boldsymbol{t}\boldsymbol{1}^{\boldsymbol{T}}\right\|_F^2 \tag{3.14}$$

$$= \left\|(\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A}) + (-\boldsymbol{t}\boldsymbol{1}^{\boldsymbol{T}})\right\|_F^2$$

In Eq. (3.14), the subscript F indicates the matrix Frobenius norm. The Frobenius norm of a matrix is the square root of the sum of the squares of all the matrix entries, which can be formulated as Eq. (3.15):

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \left|a_{ij}\right|^2} \tag{3.15}$$

The square of the Frobenius norm of the sum of two matrices can be expanded using Frobenius decomposition, as shown in Eq. (3.16):

$$\|\boldsymbol{A} + \boldsymbol{B}\|_F^2 = \|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2 + 2\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F \tag{3.16}$$

In Eq. (3.16), $\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F$ represents the Frobenius inner product, which is also the trace of the matrix product $\boldsymbol{A}\boldsymbol{B}^{\boldsymbol{T}}$ as shown in Eq. (3.17).

$$\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F = tr(\boldsymbol{A}\boldsymbol{B}^{\boldsymbol{T}}) \tag{3.17}$$

According to Eq. (3.16) and Eq. (3.17), Eq. (3.14) can be expressed as Eq. (3.18):

$$f(\boldsymbol{R}, \boldsymbol{t}) = \|\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A}\|_F^2 + \left\|-\boldsymbol{t}\boldsymbol{1}^{\boldsymbol{T}}\right\|_F^2 + 2\langle \boldsymbol{B} - \boldsymbol{R}\boldsymbol{A}, -\boldsymbol{t}\boldsymbol{1}^{\boldsymbol{T}}\rangle \tag{3.18}$$

$$= \|\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A}\|_F^2 + N\boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{t} - 2tr((\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A})\boldsymbol{1}\boldsymbol{t}^{\boldsymbol{T}})$$

As shown in Eq. (3.18), when $\boldsymbol{R}$ is constant, $f(\boldsymbol{R}, \boldsymbol{t})$ becomes a quadratic function of $\boldsymbol{t}$, and the minimum value of $f(\boldsymbol{R}, \boldsymbol{t})$ is obtained when its first-order derivative is equal to zero as shown in Eq. (3.19):

$$\frac{\partial f(\boldsymbol{R}, \boldsymbol{t})}{\partial \boldsymbol{t}} = 0 \tag{3.19}$$

In this case, $\boldsymbol{t}$ can be expressed as a function of $\boldsymbol{R}$ as shown in Eq. (3.20) and Eq. (3.21):

$$2N\boldsymbol{t} - 2(\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A})\boldsymbol{1} = 0 \tag{3.20}$$

$$\boldsymbol{t} = \frac{1}{N}(\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A})\boldsymbol{1} \tag{3.21}$$

Substituting $\boldsymbol{t}$ into $f(\boldsymbol{R},\boldsymbol{t})$ results in Eq. (3.22):

$$f(\boldsymbol{R},\boldsymbol{t}) = \|\boldsymbol{B}-\boldsymbol{RA}\|_F^2 + \frac{1}{N}\mathbf{1}^T(\boldsymbol{B}-\boldsymbol{RA})^T(\boldsymbol{B}-\boldsymbol{RA})\mathbf{1} - 2tr((\boldsymbol{B}-\boldsymbol{RA})\mathbf{1}\frac{1}{N}\mathbf{1}^T(\boldsymbol{B}-\boldsymbol{RA})^T) \tag{3.22}$$

In Eq. (3.22), the second term can be further simplified as Eq. (3.25), in which $\boldsymbol{\mu_a}$ is the mean vector of point set matrix $\boldsymbol{A} = [\boldsymbol{a_1} \ldots \boldsymbol{a_N}] \in \mathbb{R}^{m\times N}$ and $\boldsymbol{\mu_b}$ is the mean vector of point set matrix $\boldsymbol{B} = [\boldsymbol{b_1} \ldots \boldsymbol{b_N}] \in \mathbb{R}^{m\times N}$, as shown in Eq. (3.23) and Eq. (3.24):

$$\boldsymbol{\mu_a} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{a_i} \in \mathbb{R}^m \tag{3.23}$$

$$\boldsymbol{\mu_b} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{b_i} \in \mathbb{R}^m \tag{3.24}$$

$$\frac{1}{N}\mathbf{1}^T(\boldsymbol{B}-\boldsymbol{RA})^{\boldsymbol{T}}(\boldsymbol{B}-\boldsymbol{RA})\mathbf{1} \tag{3.25}$$

$$= \frac{1}{N}[1 \ldots 1]\begin{bmatrix}(\boldsymbol{b_1}-\boldsymbol{Ra_1})^{\boldsymbol{T}}\\ \vdots \\ (\boldsymbol{b_N}-\boldsymbol{Ra_N})^{\boldsymbol{T}}\end{bmatrix}[(\boldsymbol{b_1}-\boldsymbol{Ra_1}) \ldots (\boldsymbol{b_N}-\boldsymbol{Ra_N})]\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}$$

$$= \frac{1}{N}(((\boldsymbol{b_1}-\boldsymbol{Ra_1})^{\boldsymbol{T}} + \cdots + (\boldsymbol{b_N}-\boldsymbol{Ra_N})^{\boldsymbol{T}})((\boldsymbol{b_1}-\boldsymbol{Ra_1}) + \cdots + (\boldsymbol{b_N}-\boldsymbol{Ra_N})))$$

$$= \frac{1}{N}(\sum_{i=1}^{N}(\boldsymbol{b_i}-\boldsymbol{Ra_i})^{\boldsymbol{T}})(\sum_{i=1}^{N}(\boldsymbol{b_i}-\boldsymbol{Ra_i}))$$

$$= \frac{1}{N}N(\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})^{\boldsymbol{T}}N(\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})$$

$$= N\|\boldsymbol{\mu_b}-\boldsymbol{R\mu_a}\|^2$$

The third term in Eq. (3.22) can also be further simplified as Eq. (3.26):

$$2tr((\boldsymbol{B}-\boldsymbol{RA})\mathbf{1}\frac{1}{N}\mathbf{1}^T(\boldsymbol{B}-\boldsymbol{RA})^{\boldsymbol{T}}) \tag{3.26}$$

$$= 2\frac{1}{N}tr([(\boldsymbol{b_1}-\boldsymbol{Ra_1}) \ldots (\boldsymbol{b_N}-\boldsymbol{Ra_N})]\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}[1 \ldots 1]\begin{bmatrix}(\boldsymbol{b_1}-\boldsymbol{Ra_1})^{\boldsymbol{T}}\\ \vdots \\ (\boldsymbol{b_N}-\boldsymbol{Ra_N})^{\boldsymbol{T}}\end{bmatrix})$$

$$= 2\frac{1}{N}tr(((\boldsymbol{b_1}-\boldsymbol{Ra_1}) + \cdots + (\boldsymbol{b_N}-\boldsymbol{Ra_N}))((\boldsymbol{b_1}-\boldsymbol{Ra_1})^{\boldsymbol{T}} + \cdots + (\boldsymbol{b_N}-\boldsymbol{Ra_N})^{\boldsymbol{T}}))$$

$$= 2\frac{1}{N}tr((\sum_{i=1}^{N}(\boldsymbol{b_i}-\boldsymbol{Ra_i}))(\sum_{i=1}^{N}(\boldsymbol{b_i}-\boldsymbol{Ra_i})^{\boldsymbol{T}}))$$

$$= 2\frac{1}{N}tr(N(\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})N(\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})^{\boldsymbol{T}})$$

$$= 2Ntr((\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})(\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})^{\boldsymbol{T}})$$

$$= 2N\|\boldsymbol{\mu_b}-\boldsymbol{R\mu_a}\|^2$$

Substituting Eq. (3.25) and Eq. (3.26) back into Eq. (3.22) results in Eq. (3.27):

$$f(\boldsymbol{R},\boldsymbol{t}) = ||\boldsymbol{B}-\boldsymbol{RA}||_F^2 + N||\boldsymbol{\mu_b}-\boldsymbol{R\mu_a}||^2 - 2N||\boldsymbol{\mu_b}-\boldsymbol{R\mu_a}||^2 \tag{3.27}$$

$$= \sum_{i=1}^{N}||\boldsymbol{b_i}-\boldsymbol{Ra_i}||^2 + \sum_{i=1}^{N}||\boldsymbol{\mu_b}-\boldsymbol{R\mu_a}||^2 - 2\sum_{i=1}^{N}(\boldsymbol{b_i}-\boldsymbol{Ra_i})^{\boldsymbol{T}}(\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})$$

$$= \sum_{i=1}^{N}||(\boldsymbol{b_i}-\boldsymbol{Ra_i})-(\boldsymbol{\mu_b}-\boldsymbol{R\mu_a})||^2$$

$$= \sum_{i=1}^{N}||(\boldsymbol{b_i}-\boldsymbol{\mu_b})-\boldsymbol{R}(\boldsymbol{a_i}-\boldsymbol{\mu_a})||^2$$

$$= \sum_{i=1}^{N}||\boldsymbol{b_i'}-\boldsymbol{Ra_i'}||^2$$

$$= ||\boldsymbol{B'}-\boldsymbol{RA'}||_F^2$$

In Eq. (3.27):

$$\boldsymbol{a_i'} = \boldsymbol{a_i}-\boldsymbol{\mu_a},\ \ \boldsymbol{A'} = \boldsymbol{A}(\boldsymbol{I_N}-\frac{1}{N}\boldsymbol{11^T}) \tag{3.28}$$

$$\boldsymbol{b_i'} = \boldsymbol{b_i}-\boldsymbol{\mu_b},\ \ \boldsymbol{B'} = \boldsymbol{B}(\boldsymbol{I_N}-\frac{1}{N}\boldsymbol{11^T}) \tag{3.29}$$

Eq. (3.27) can be further expanded as Eq. (3.30):

$$f(\boldsymbol{R},\boldsymbol{t}) = ||\boldsymbol{B'}-\boldsymbol{RA'}||_F^2 \tag{3.30}$$

$$= ||\boldsymbol{B'}||_F^2 + ||\boldsymbol{RA'}||_F^2 - 2\langle\boldsymbol{B'},\boldsymbol{RA'}\rangle$$

$$= ||\boldsymbol{B'}||_F^2 + ||\boldsymbol{A'}||_F^2 - 2tr(\boldsymbol{RA'B'^T})$$

In Eq. (3.30), $\boldsymbol{A'B'^T}$ can be factorized using Singular Value Decomposition (SVD) as shown in Eq. (3.31), in which $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix:

$$\boldsymbol{A'B'^T} = \boldsymbol{U\Sigma V^T} \tag{3.31}$$

Substituting Eq. (3.31) into Eq. (3.30) results in Eq. (3.32):

$$f(\boldsymbol{R},\boldsymbol{t}) = ||\boldsymbol{B'}||_F^2 + ||\boldsymbol{A'}||_F^2 - 2tr(\boldsymbol{RU\Sigma V^T}) \tag{3.32}$$

Because $tr(\boldsymbol{ABCD}) = tr(\boldsymbol{BCDA}) = tr(\boldsymbol{CDBA}) = tr(\boldsymbol{DABC})$, Eq. (3.32) can also be expressed as Eq. (3.33):

$$f(\boldsymbol{R},\boldsymbol{t}) = ||\boldsymbol{B'}||_F^2 + ||\boldsymbol{A'}||_F^2 - 2tr(\boldsymbol{V^TRU\Sigma}) \tag{3.33}$$

$$= ||\boldsymbol{B'}||_F^2 + ||\boldsymbol{A'}||_F^2 - 2tr(\boldsymbol{T\Sigma})$$

$$= ||\boldsymbol{B'}||_F^2 + ||\boldsymbol{A'}||_F^2 - 2\sum_{i=1}^{m}T_{ii}\sigma_i$$

Since $||\boldsymbol{B'}||_F^2$ and $||\boldsymbol{A'}||_F^2$ are constants, and σ_i, the diagonal elements of $\boldsymbol{\Sigma}$ are also constant, the cost function $f(\boldsymbol{R},\boldsymbol{t})$ is minimized when $\sum_{i=1}^{m}T_{ii}\sigma_i$ reaches maximum. Thus, the problem is reformulated as finding $\boldsymbol{T}$ that satisfies Eq. (3.34):

$$< \boldsymbol{T} >= max_{\boldsymbol{T}} \sum_{i=1}^{m} T_{ii}\sigma_i \tag{3.34}$$

In Eq. (3.34), $\boldsymbol{T} = \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{U}$, which is an orthogonal matrix with elements satisfying $|T_{ii}| \leq 1$. Consequently, the cost function $f(\boldsymbol{R}, \boldsymbol{t})$ reaches its minimum when $\boldsymbol{T} = \boldsymbol{I_m}$, as shown in Eq. (3.35) and Eq. (3.36):

$$T_{ii} = 1,\ i.e.,\ \boldsymbol{T} = \boldsymbol{I_m} \tag{3.35}$$

$$\boldsymbol{T} = \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{U} = \boldsymbol{I_m} \tag{3.36}$$

Therefore, the desired $\boldsymbol{R}$ and $\boldsymbol{t}$ can be obtained with Eq. (3.37) and Eq. (3.38):

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^{\boldsymbol{T}} \tag{3.37}$$

$$\boldsymbol{t} = \frac{1}{N}(\boldsymbol{B} - \boldsymbol{R}\boldsymbol{A})\boldsymbol{1} \tag{3.38}$$

The computational process described above outlines the steps involved in each ICP iteration. The algorithm continues to iterate until the convergence criteria are met. These criteria include:

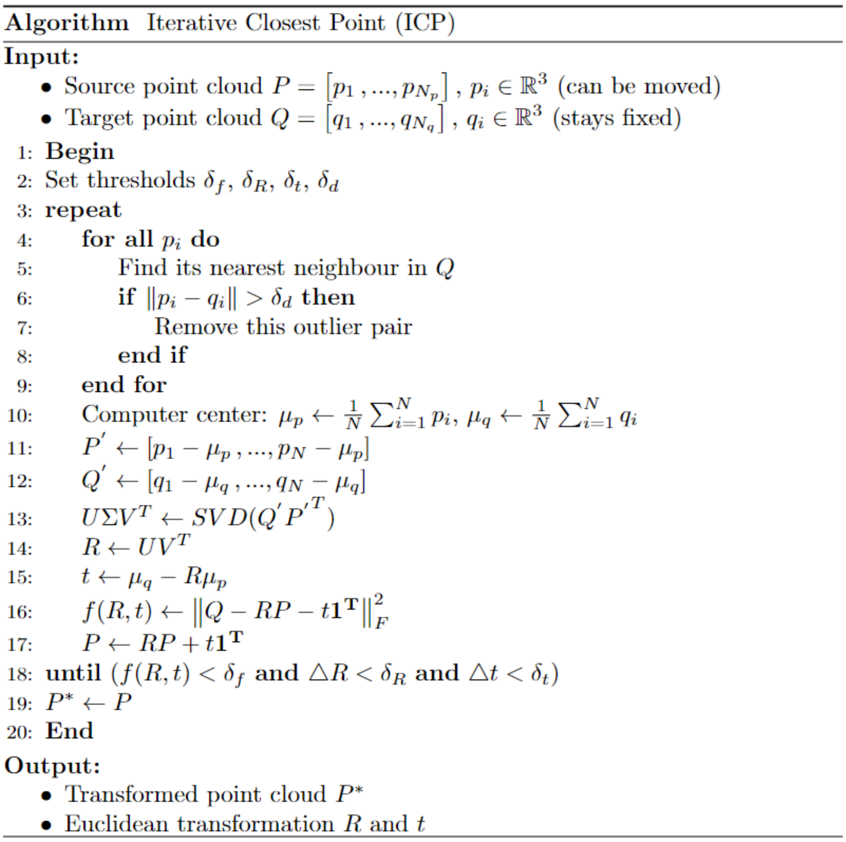
1) The cost function $f(\boldsymbol{R}, \boldsymbol{t})$ falling below a predefined threshold.
2) The changes in rotation ($\Delta\boldsymbol{R}$) and translation ($\Delta\boldsymbol{t}$) being smaller than a specific threshold.

Based on the mathematical derivation presented earlier, the pseudocode for the ICP algorithm pipeline is illustrated in Figure 3-3. To further demonstrate its effectiveness, Figure 3-4 showcases ICP applied to the well-known Stanford Bunny dataset.

As depicted in the figure, the initial alignment sets the source bunny point cloud (yellow) apart from the target bunny point cloud (blue), with a Root Mean Square Error (RMSE) of 10.07 cm. As the algorithm iterates, the source point cloud gradually moves closer to the target, with the RMSE decreasing progressively. After fourteen iterations, the source bunny achieves accurate alignment with the target bunny, reaching a final RMSE of 0.25 cm.

In this demonstration, ICP achieves ideal performance due to the favorable initial relative pose between the two point clouds, which provides a strong starting point for point matching. However, if the initial pose difference between the source and target point clouds is significant, poor initial point matching can lead to severe misalignment, as demonstrated in Section 4.4.1 .

To address this issue, a novel ICP variant, CPR-ICP, is proposed and discussed in detail in Section 4.4 . This improved method enhances alignment accuracy and mitigates the limitations of classic ICP in cases where initial point matching is poor.



```
Algorithm  Iterative Closest Point (ICP)
Input:
  • Source point cloud $P = [p_1 , ..., p_{N_p}]$ , $p_i \in \mathbb{R}^3$ (can be moved)
  • Target point cloud $Q = [q_1 , ..., q_{N_q}]$ , $q_i \in \mathbb{R}^3$ (stays fixed)
Begin
Set thresholds $\delta_f$, $\delta_R$, $\delta_t$, $\delta_d$
repeat
    for all $p_i$ do
        Find its nearest neighbour in $Q$
        if $\|p_i - q_i\| > \delta_d$ then
            Remove this outlier pair
        end if
    end for
    Computer center: $\mu_p \leftarrow \frac{1}{N}\sum_{i=1}^{N} p_i$, $\mu_q \leftarrow \frac{1}{N}\sum_{i=1}^{N} q_i$
    $P' \leftarrow [p_1 - \mu_p , ..., p_N - \mu_p]$
    $Q' \leftarrow [q_1 - \mu_q , ..., q_N - \mu_q]$
    $U\Sigma V^T \leftarrow SVD(Q' P'^T)$
    $R \leftarrow UV^T$
    $t \leftarrow \mu_q - R\mu_p$
    $f(R,t) \leftarrow \|Q - RP - t\mathbf{1}^\mathbf{T}\|_F^2$
    $P \leftarrow RP + t\mathbf{1}^\mathbf{T}$
until ($f(R,t) < \delta_f$ and $\triangle R < \delta_R$ and $\triangle t < \delta_t$)
$P^* \leftarrow P$
End
Output:
  • Transformed point cloud $P^*$
  • Euclidean transformation $R$ and $t$
```

Figure 3-3. Pseudocode of the ICP algorithm pipeline.

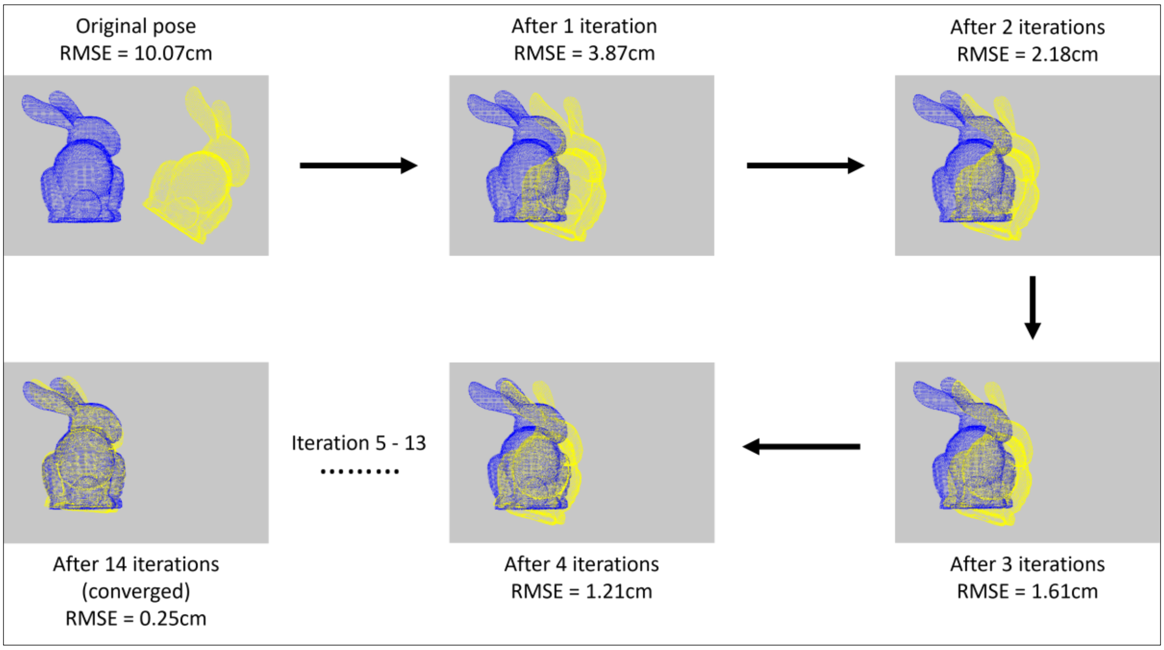


Figure 3-4. Performance demonstration of the Iterative Closest Point (ICP) algorithm. The distance (error) between two point clouds decreases with each iteration, converging at the fourteenth iteration.

Chapter 4 Innovative Unified Framework for Holistic and Objective SLAM Benchmarking and Mapping-Accuracy Enhancement

The previous chapter established the mathematical foundation underlying the proposed novel methodologies and concepts. Building upon this foundation, this chapter presents an innovative unified pipeline that connects holistic and objective SLAM benchmarking with mapping-accuracy enhancement. First, Section 4.1 briefly discusses the limitations of existing SLAM benchmarking methods, highlighting the motivation for developing a novel SLAM benchmark. Following this, Section 4.2 introduces the proposed benchmarking framework in detail, covering key aspects such as equipment setup and the Euclidean transformations between different coordinate frames. Then, Section 4.3 develops the concept of using benchmarking results as feedback to improve mapping accuracy through a rigid transformation. Finally, Section 4.4 provides the point cloud registration approach that yields this transformation, completing the pipeline from evaluation to accuracy enhancement.

4.1 Limitations of existing benchmarking methodologies

Current SLAM benchmarking methods have the limitation of lacking holism and objectivity. A holistic and objective SLAM benchmark should assess the combined performance of localization and mapping on a global scale, rather than evaluating them in isolation. However, many current SLAM benchmarks only gauge localization performance, as mapping assessment is hindered by the absence of ground truth for the map. This challenge is particularly significant in large-scale environments, where obtaining an accurate ground truth map is difficult and impractical. Therefore, most existing mapping benchmark studies predominantly focus on small to medium-sized objects, whose ground truth geometries (maps) can be relatively easy to obtain. A prevalent benchmarking approach employed in these studies involves first aligning the estimated map with the ground truth map through a 3D shape registration method, followed by error computation based on predefined error metrics. This approach is actually designed for 3D reconstruction benchmarking, where the objects' estimated geometries are treated as standalone entities; however, it is not well-suited for SLAM benchmarking.

The fundamental flaw in this method is that an objective mapping benchmark should preserve the original pose relationship between the estimated and ground truth maps to ensure that the computed error accurately reflects the unbiased performance of the SLAM system. However, the manual alignment process used in this method alters the original error distribution, leading to biased evaluation results and misrepresenting the actual performance of the SLAM system.

4.2 Innovative approach to holistic and objective SLAM benchmarking

Since existing SLAM mapping benchmarks face two major limitations: they struggle to provide unbiased evaluation results and are unsuitable for large-scale environments commonly encountered in SLAM applications, this section introduces a holistic and objective SLAM benchmark designed to evaluate both localization and mapping accuracy within a unified global coordinate system. The illustration of the proposed benchmark is shown in Figure 4-1. As shown in the figure, the blue squared box represents the scene where the SLAM camera operates. As the camera moves within the scene, the SLAM algorithm simultaneously performs localization and mapping. The SLAM camera frame, denoted as $\boldsymbol{Sc}$ (SLAM camera frame), serves as the camera's reference frame and moves continuously along with it. The trajectory traced by the origin of $\boldsymbol{Sc}$ is referred to as the SLAM camera trajectory. The SLAM algorithm produces two key outputs: the estimated SLAM camera trajectory, represented as $\boldsymbol{P}$, which consists of a sequence of estimated poses of the SLAM camera frame over time, and the estimated map (geometry), denoted as $\boldsymbol{Q}$, representing the reconstructed environment.

To ensure a holistic and objective assessment on the accuracy of these two estimates, the ground truth SLAM camera trajectory $\boldsymbol{P}^*$ and the ground truth map (geometry) $\boldsymbol{Q}^*$ must be obtained. Moreover, as illustrated in Figure 4-1, a global coordinate frame $\boldsymbol{G}$ needs to be established to serve as a common reference frame, enabling accurate transformation and comparison of all estimated and ground truth data ($\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{P}^*$, $\boldsymbol{Q}^*$). By integrating localization and mapping into a single evaluation framework, this benchmark ensures that both aspects are assessed together rather than in isolation, providing a more comprehensive and objective assessment of SLAM performance.

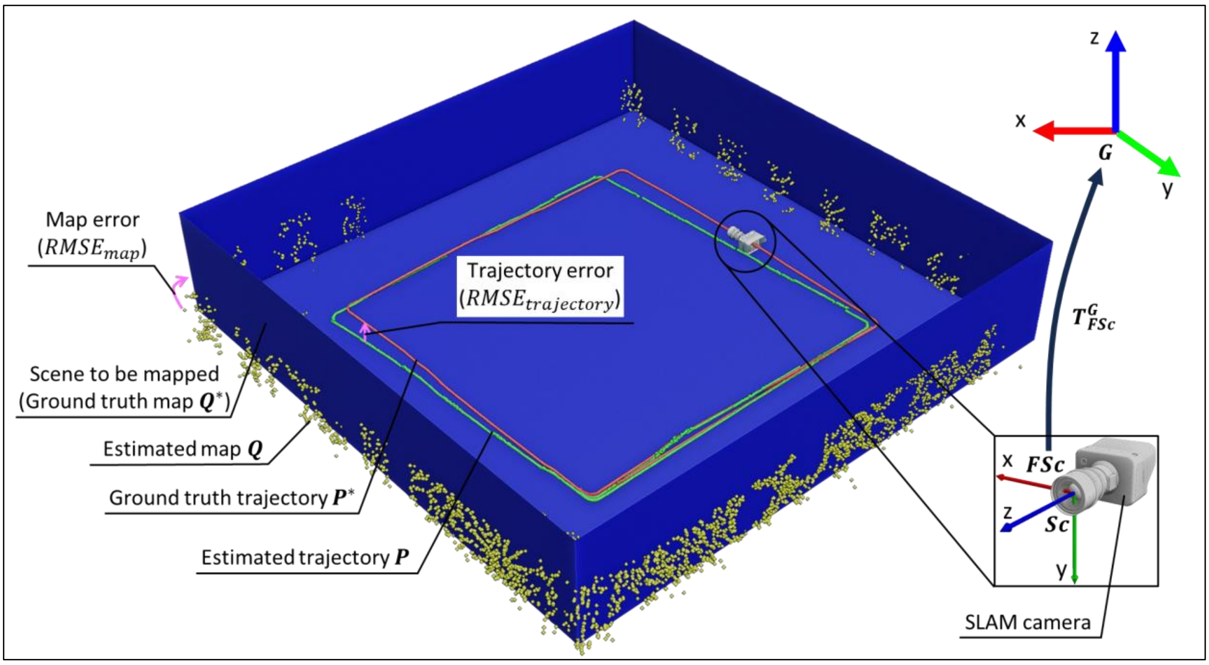

Map error
($RMSE_{map}$)
Trajectory error
($RMSE_{trajectory}$)
Scene to be mapped
(Ground truth map $\boldsymbol{Q}^*$)
Estimated map $\boldsymbol{Q}$
Ground truth trajectory $\boldsymbol{P}^*$
Estimated trajectory $\boldsymbol{P}$
z
x
$\boldsymbol{G}$
y
$\boldsymbol{T}_{FSc}^{G}$
x
$\boldsymbol{FSc}$
z
$\boldsymbol{Sc}$
y
SLAM camera

Figure 4-1. Illustration of the benchmark methodology.

As previously mentioned, the SLAM camera trajectory is defined as a sequence of positions corresponding to the origin of the SLAM camera frame $\boldsymbol{Sc}$. In simulations, obtaining the ground truth positions of the $\boldsymbol{Sc}$ origin is straightforward, as they can be directly extracted from the simulation software. This process is detailed in the simulation-based experiments presented in Chapter 5. However, in real-world experiments, acquiring the ground truth positions of the $\boldsymbol{Sc}$ origin is significantly more challenging and requires high-precision measurement tools, such as the VICON motion capture system. Unlike in simulations, these positions cannot be obtained directly, as the $\boldsymbol{Sc}$ origin is located on the lens surface of the SLAM camera, making it physically inaccessible for direct measurement. To overcome this limitation, the ground truth positions of the $\boldsymbol{Sc}$ origin must be determined indirectly by introducing auxiliary reference frames and applying Euclidean transformations. This approach enables an accurate estimation of the $\boldsymbol{Sc}$ origin despite its inaccessibility. The detailed methodology for this process is presented in the real-world experiments discussed in Chapter 6.

As for the acquisition of the ground truth map (geometry), in real-world experiments, the ground truth map (geometry) is represented by a CAD model, which is created based on manual measurements of key dimensions of the scene. This CAD model serves as the ground truth geometry $\boldsymbol{Q}^*$. However, due to inevitable measurement inaccuracies, discrepancies exist between the geometries and dimensions of the drawn CAD model and those of the actual scene. Consequently, the mapping error reported by the SLAM benchmark is expected to be larger than the true mapping error. This implies that the actual mapping performance of the SLAM system is likely better than what is reflected in the SLAM benchmark. A detailed discussion of this effect is provided in the real-world experiments presented in Chapter 6. In simulations, however, the SLAM benchmark accurately reflects the true mapping performance. This is because the drawn CAD model is used directly as both the scene and the ground truth geometry (map), ensuring that there are no external measurement errors. The details of this process are elaborated in the simulation-based experiments presented in Chapter 5.

As previously discussed, ensuring benchmark objectivity requires the establishment of a global reference frame $\boldsymbol{G}$, allowing all ground truth and estimated data to be transformed into a common coordinate system for evaluation. In both simulation-based and real-world experiments, the estimated trajectory $\boldsymbol{P}$ and estimated map $\boldsymbol{Q}$ are originally captured within the SLAM camera frame $\boldsymbol{Sc}$ of the first image in the captured sequence, which is denoted as $\boldsymbol{FSc}$ in Figure 4-1. The ground truth trajectory $\boldsymbol{P}^*$ and ground truth map $\boldsymbol{Q}^*$, however, are obtained differently in simulations and real-world experiments. In simulations, these data are directly captured within the reference frame of the simulation software, while in real-world experiments, they are recorded using a high-precision measuring tool. In simulations, each ground truth position $\boldsymbol{p}_i^* \in \boldsymbol{P}^*$ is recorded alongside its corresponding orientation of the SLAM camera frame $\boldsymbol{Sc}$. The first position $\boldsymbol{p}_1^* \in \boldsymbol{P}^*$ and its associated orientation define the pose of $\boldsymbol{Sc}$ at the first captured image relative to the simulation software's world reference frame. Therefore, the world frame of the simulation software serves as the global reference frame $\boldsymbol{G}$, and the pose of $\boldsymbol{FSc}$ represents the Euclidean transformation required to align all ground truth and estimated data ($\boldsymbol{P}$, $\boldsymbol{P}^*$, $\boldsymbol{Q}$, $\boldsymbol{Q}^*$) within $\boldsymbol{G}$ for evaluation. A similar approach is applied in real-world experiments, where each ground truth position $\boldsymbol{p}_i^* \in \boldsymbol{P}^*$ is captured using auxiliary reference frames and Euclidean transformations from a high-precision measuring tool, along with the corresponding orientation of $\boldsymbol{Sc}$. Consequently, the reference frame of the high-precision measuring tool is designated as the global reference frame $\boldsymbol{G}$. As illustrated in Figure 4-1, the Euclidean transformation that aligns all ground truth and estimated data ($\boldsymbol{P}$, $\boldsymbol{P}^*$, $\boldsymbol{Q}$, $\boldsymbol{Q}^*$) within the global reference frame $\boldsymbol{G}$ is denoted as $\boldsymbol{T}_{\boldsymbol{FSc}}^{\boldsymbol{G}}$.

As illustrated in Figure 4-1, once all estimated and ground truth data for both the trajectory and map ($\boldsymbol{P}$, $\boldsymbol{P}^*$, $\boldsymbol{Q}$, $\boldsymbol{Q}^*$) are transformed into the global reference frame $\boldsymbol{G}$, the trajectory error ($RMSE_{trajectory}$) between $\boldsymbol{P}$ and $\boldsymbol{P}^*$, as well as the map error ($RMSE_{map}$) between $\boldsymbol{Q}$ and $\boldsymbol{Q}^*$, are computed using specific evaluation metrics. For trajectory error computation, the evaluation metric used is the root mean square error (RMSE) of the nearest-neighbour distances. As expressed in Eq. (4.1), within the global frame $\boldsymbol{G}$, each point $\boldsymbol{p_1}$, … , $\boldsymbol{p_n}$ from the estimated SLAM camera trajectory $\boldsymbol{P}$ ($\boldsymbol{p_1}$, … , $\boldsymbol{p_n} \in \boldsymbol{P}$) is matched to its nearest neighbouring point in the ground truth trajectory $\boldsymbol{P}^*$, denoted as $\boldsymbol{p}^*_{\boldsymbol{k_1}}$, … , $\boldsymbol{p}^*_{\boldsymbol{k_n}}$ ($\boldsymbol{p}^*_{\boldsymbol{k_1}}$, … , $\boldsymbol{p}^*_{\boldsymbol{k_n}} \in \boldsymbol{P}^*$). The RMSE of all nearest-neighbour distances is then calculated to quantify the trajectory error.

$$RMSE_{trajectory} = \left(\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{p_i} - \boldsymbol{p}^*_{\boldsymbol{k_i}}\right\|_2\right)^{1/2} \tag{4.1}$$

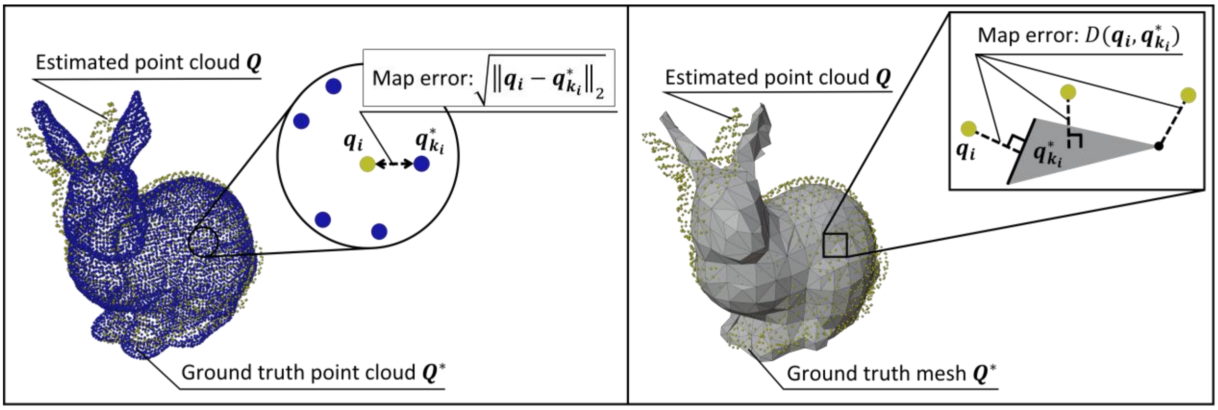
For the computation of map error, since the estimated map is represented as a point cloud, the choice of evaluation metric depends on the representation format of the ground truth map (geometry), which can be either a point cloud or a mesh. As illustrated in Figure 4-2 (a), when the ground truth map is also represented as a point cloud, the map error metric ($RMSE_{map\_pointcloud}$) is the same as the one used for trajectory error. This metric is computed as the root mean square of the pairwise distances between all nearest neighbouring points ($\boldsymbol{q_i}, \boldsymbol{q}^*_{\boldsymbol{k_i}}$), as defined in Eq. (4.2). For cases where the ground truth map is represented as a mesh, as shown in Figure 4-2 (b), the evaluation metric ($RMSE_{map\_mesh}$) is computed as the root mean square of all distances $D(\boldsymbol{q_i}, \boldsymbol{q}^*_{\boldsymbol{k_i}})$ between each point $\boldsymbol{q_i}$ in the estimated map point cloud $\boldsymbol{Q}$ and its nearest neighbouring triangle $\boldsymbol{q}^*_{\boldsymbol{k_i}}$ in the ground truth mesh $\boldsymbol{Q}^*$, as defined in Eq. (4.3). When calculating the point-to-triangle distance $D(\boldsymbol{q_i}, \boldsymbol{q}^*_{\boldsymbol{k_i}})$, the relative position of the point $\boldsymbol{q_i}$ with respect to the triangle $\boldsymbol{q}^*_{\boldsymbol{k_i}}$ must be considered. Figure 4-2 (b) illustrates the three possible positional relationships:

- If the projection of $\boldsymbol{q_i}$ onto the triangle's plane lies inside the triangle $\boldsymbol{q}^*_{\boldsymbol{k_i}}$, the point-triangle distance $D(\boldsymbol{q_i}, \boldsymbol{q}^*_{\boldsymbol{k_i}})$ is the length of the normal vector from $\boldsymbol{q_i}$ to the plane.

- If the projection of $\boldsymbol{q_i}$ onto the triangle's plane lies outside the triangle $\boldsymbol{q}^*_{\boldsymbol{k_i}}$, the point-triangle distance $D(\boldsymbol{q_i}, \boldsymbol{q}^*_{\boldsymbol{k_i}})$ is defined as the shortest distance from $\boldsymbol{q_i}$ to either the closest triangle edge or vertex.

$$RMSE_{map\_pointcloud} = \left(\frac{1}{m}\sum_{i=1}^{m}\left\|\boldsymbol{q_i} - \boldsymbol{q}^*_{\boldsymbol{k_i}}\right\|_2\right)^{1/2} \tag{4.2}$$

$$RMSE_{map\_mesh} = \left(\frac{1}{m}\sum_{i=1}^{m} D^2(\boldsymbol{q_i}, \boldsymbol{q}^*_{\boldsymbol{k_i}})\right)^{1/2} \tag{4.3}$$


Estimated point cloud $\boldsymbol{Q}$
Map error: $\sqrt{\left\|\boldsymbol{q_i} - \boldsymbol{q}^*_{\boldsymbol{k_i}}\right\|_2}$
$\boldsymbol{q_i}$
$\boldsymbol{q}^*_{\boldsymbol{k_i}}$
Ground truth point cloud $\boldsymbol{Q}^*$
Map error: $D(\boldsymbol{q_i}, \boldsymbol{q}^*_{\boldsymbol{k_i}})$
Estimated point cloud $\boldsymbol{Q}$
$\boldsymbol{q_i}$
$\boldsymbol{q}^*_{\boldsymbol{k_i}}$
Ground truth mesh $\boldsymbol{Q}^*$

(a) (b)

Figure 4-2. Error metrics for evaluating mapping performance. (a) When the ground truth map is a point cloud, the error is computed as the RMSE of nearest-neighbour point distances. (b) When the ground truth map is a mesh, the error is computed as the RMSE of point-to-mesh distances.

This concludes the introduction of the methodology for holistic and objective SLAM benchmarking. Building on this foundation, the next section delves into the methodology specifically designed to enhance mapping accuracy.

4.3 Innovative concept of improving mapping accuracy

As illustrated in Figure 1-1, Simultaneous Localization and Mapping (SLAM) is an iterative process in which the estimated poses and map are continuously updated based on each other. This interdependence creates a strong correlation between localization and mapping, where localization errors directly influence mapping errors. The concept of mapping accuracy improvement leverages this correlation to enhance mapping precision by predicting the numerical optimization for reducing mapping errors based on the numerical optimization of localization errors, which is derived from the SLAM benchmark. The primary motivation for this approach is to improve the accuracy of the estimated map, enabling the generation of high-precision geometries that can possibly serve as ground truth maps in SLAM benchmarking. The acquisition of ground truth maps is a fundamental challenge in SLAM benchmarking. Unlike existing methods that rely on high-precision measuring tools [108] or impose strict accuracy requirements on SLAM algorithms [140], the proposed approach utilizes results obtained from the SLAM benchmarking proposed in Section 4.2 as feedback to minimize mapping errors and generate accurate, high-quality maps. This method has the potential to be applied across a wide range of SLAM scenarios for obtaining ground truth maps of objects and environments of varying scales. It is particularly useful for large-scale structures where blueprints are no longer available due to age, such as legacy nuclear facilities, historical heritage sites, and abandoned chemical plants. Additionally, it offers valuable insights into robot navigation, where map accuracy plays a crucial role in path planning, obstacle avoidance, and autonomous decision-making.

The objective of mapping accuracy improvement is to first understand the correlation between localization and mapping and then systematically exploit it. Since this correlation is reflected in the relationship between localization and mapping errors, obtaining unbiased and reliable localization and mapping errors that accurately represent the true performance of the SLAM system is essential. The holistic and objective SLAM benchmark introduced in Section 4.2 evaluates localization and mapping as a unified process, rather than treating them separately. By transforming all key elements—including the estimated trajectory $\boldsymbol{P}$, the ground truth trajectory $\boldsymbol{P}^*$, the estimated map $\boldsymbol{Q}$, and the ground truth map $\boldsymbol{Q}^*$—into a global reference frame, the benchmark preserves the original pose relationships between these elements. As a result, it maintains the correlation between localization and mapping, preventing distortions that could introduce bias into the evaluation, and yielding unbiased localization and mapping errors. The relevance of localization to mapping is evident in the positive correlation between their accuracies: in qualitative terms, mapping errors tend to increase as localization errors increase. By harnessing this relationship, the proposed approach systematically reduces mapping errors, ultimately improving SLAM-generated maps and enabling their use as ground truth references in real-world applications.

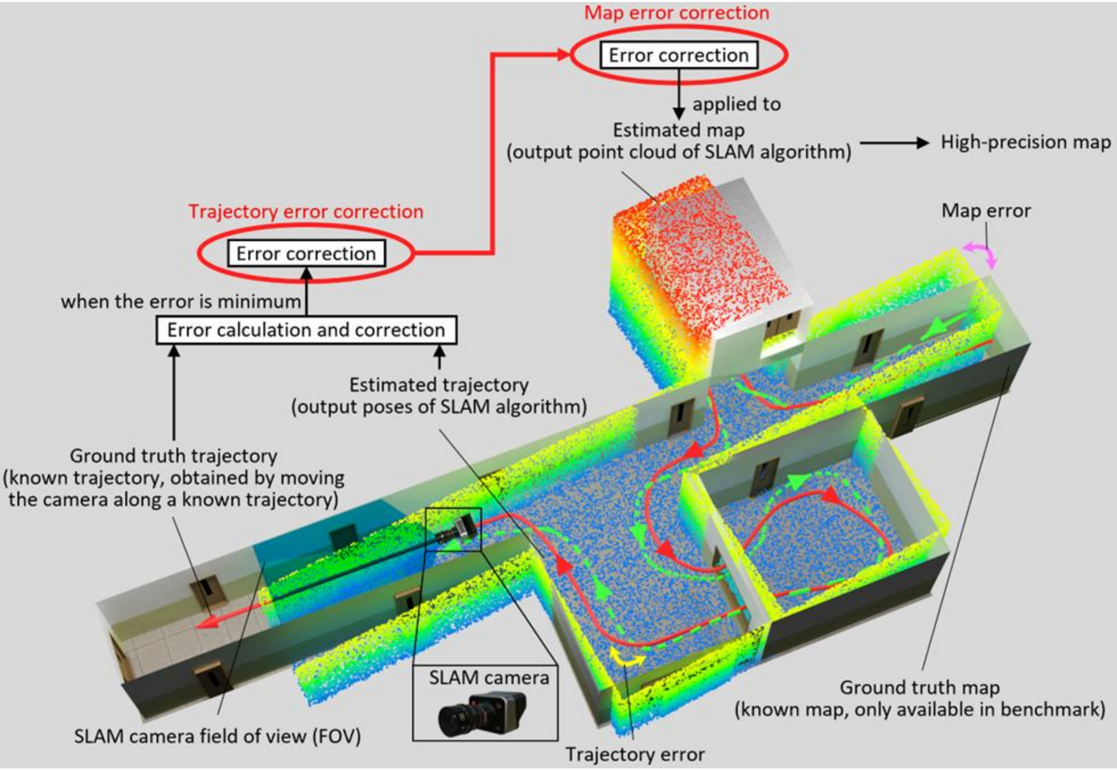

Map error correction
Error correction
applied to
Estimated map
(output point cloud of SLAM algorithm)
High-precision map
Trajectory error correction
Error correction
Map error
when the error is minimum
Error calculation and correction
Estimated trajectory
(output poses of SLAM algorithm)
Ground truth trajectory
(known trajectory, obtained by moving the camera along a known trajectory)
SLAM camera
Ground truth map
(known map, only available in benchmark)
SLAM camera field of view (FOV)
Trajectory error

Figure 4-3. Concept of mapping accuracy enhancement.

Once the correlation between localization and mapping is well understood, it can be leveraged to enhance mapping accuracy. The working mechanism of this method is illustrated in Figure 4-3, which depicts a typical SLAM process where a camera is used to map a corridor. As shown in the figure, the ground truth trajectory of the camera sensor is represented by a solid red arrowed line, while the estimated trajectory is depicted as a dashed green arrowed line. The trajectory error, which represents the deviation between these two trajectories, is indicated by a yellow two-way curved arrow. For mapping, the ground truth map (geometry) of the corridor is represented by a CAD model, whereas the estimated map (geometry) is visualized as a colorized point cloud. The map error, representing the difference between the estimated and ground truth maps, is marked by a pink two-way curved arrow.

The proposed mapping accuracy improvement method formulates error reduction for both localization and mapping as a numerical optimization problem. The process consists of two key steps: first, obtaining a numerical optimization that minimizes the trajectory error between the estimated trajectory and the ground truth trajectory; second, applying this optimized transformation directly to the estimated map to improve its accuracy. The reason behind this practice is that since localization and mapping are inherently linked and there exists a positive correlation between their errors, the numerical optimization minimizing localization errors is also effective for reducing mapping error.

The numerical optimization in this context refers to point cloud registration, which applies Euclidean transformation to minimize discrepancies between point clouds. The most widely used technique for this task is the Iterative Closest Point (ICP) algorithm, which iteratively refines the alignment between two point clouds by minimizing the distance between corresponding points. However, the classic ICP method has the limitation of being highly sensitive to the initial alignment, meaning that poor initial positioning can lead to local minima or incorrect convergence. Additionally, ICP assumes that corresponding points exist in both point clouds, making it prone to errors in cases where outliers or partial overlaps occur. These limitations reduce the robustness and accuracy of point cloud registration, particularly in complex mapping scenarios where misalignments or structural inconsistencies are present.

To address these challenges, an enhanced ICP variant called Centre Point Registration-ICP (CPR-ICP) is proposed. This method improves registration by pre-aligning point clouds based on their centroids and least-square planes before applying classic ICP optimization. The pre-alignment step significantly reduces initial misalignment errors, improving convergence stability and accuracy.

The following sections first demonstrate the limitation of the classic ICP method with and example and then introduce CPR-ICP, detailing its implementation and its role in mapping accuracy enhancement.

4.4 Innovative approach to point cloud registration

4.4.1 Limitation of the classic ICP algorithm in point cloud registration

Figure 3-4 in Section 3.2 illustrates a scenario where the ICP algorithm performs optimally. However, as shown in Figure 4-4 (a), if the source point cloud (yellow bunny) is intentionally placed in a significantly different pose from the target point cloud (grey bunny), the classic ICP method struggles to achieve proper alignment due to the lack of an adequate initial feature match. In contrast, as shown in Figure 4-4 (b), the proposed CPR-ICP method successfully achieves precise alignment by first performing pre-alignment (CPR), which establishes a reliable feature match between the two point clouds. The details of the innovative CPR-ICP method are elaborated in the next section.

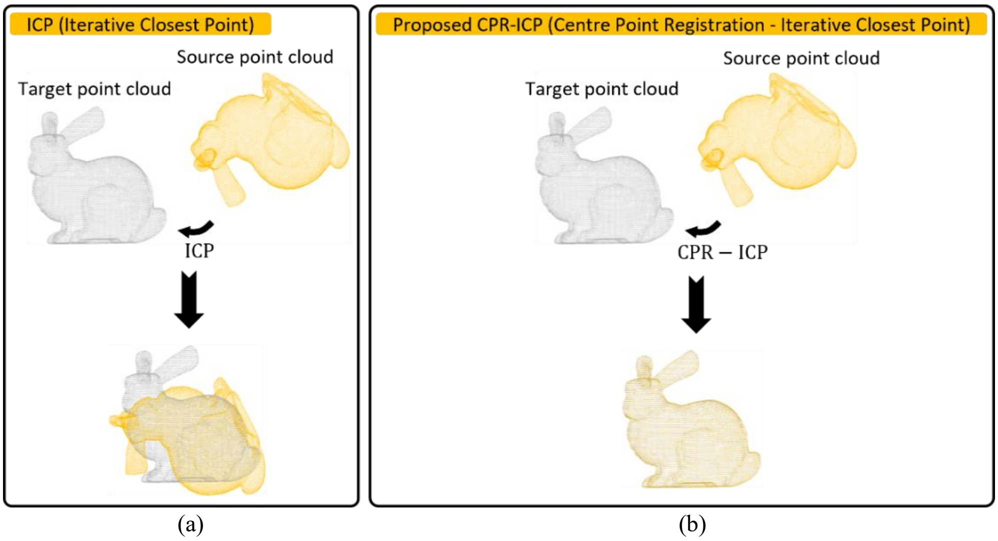

ICP (Iterative Closest Point)
Source point cloud
Target point cloud
ICP
Proposed CPR-ICP (Centre Point Registration - Iterative Closest Point)
Source point cloud
Target point cloud
CPR – ICP
(a)
(b)

Figure 4-4. Performance comparison between ICP and CPR-ICP under the circumstance of a decent initial feature matching being absent. a) Performance of the classic ICP. b) Performance of the proposed CPR-ICP.

4.4.2 CPR-ICP: An innovative ICP variant

As discussed in the previous section, the proposed mapping accuracy improvement method consists of two key steps. The first step involves computing a numerical optimization that minimizes the error between the estimated trajectory $\boldsymbol{P}$ and the ground truth trajectory $\boldsymbol{P}^*$. The second step applies this optimized transformation to the estimated map $\boldsymbol{Q}$ to enhance its accuracy. Thus, the workflow of mapping accuracy improvement is structured into two primary phases:

1) Trajectory Error Minimization (Step 1) – Aligning the estimated trajectory with the ground truth trajectory using point cloud registration.
2) Map (Geometry) Error Reduction (Step 2) – Applying the optimized transformation obtained in Step 1 to the estimated map for refinement.

Minimizing trajectory error involves registering the point cloud of the estimated trajectory with the point cloud of the ground truth trajectory. Since this registration process is fundamentally a Euclidean transformation, it is typically performed using the Iterative Closest Point (ICP) algorithm [77, 282-284], whose mathematical foundation is detailed in Section 3.2 . As demonstrated in Figure 3-4, classic ICP can achieve accurate registration when there is a strong initial point correspondence between the two point clouds. However, as discussed in Section 4.4.1 , when the initial point matching is poor, ICP struggles to converge correctly, leading to misalignment and high residual errors. To address this issue, a novel ICP variant called Centre Point Registration-ICP (CPR-ICP) is proposed. The core principle of CPR-ICP is to improve the initial alignment between two point clouds, thereby establishing a strong initial point correspondence before executing the ICP refinement process. This pre-alignment process significantly enhances the accuracy and robustness of point cloud registration. The Centre Point Registration (CPR) step is designed to bring corresponding points closer together before ICP is applied. This is achieved by aligning three reference planes for each point cloud. These three reference planes are determined using least-squares fitting and are designed to ensure a robust pre-alignment:

- The first reference plane is the least-squares plane fitted to the point cloud, minimizing the sum of squared distances between each point and the plane.

  This plane is unique for non-degenerate point clouds, meaning the points are not collinear or coincident. Since the experimental point clouds are collected from realistic robot motion and environment scanning, they naturally satisfy this non-degeneracy condition. The formal mathematical justification for this uniqueness is presented in the subsequent mathematical derivation.

- The second reference plane is defined by two constraints: 1) It is perpendicular to the first reference plane; 2) Under the first constraint, it minimizes the sum of squared distances between each point and the plane.

  This plane is also unique when, in addition to the point cloud being non-degenerate (i.e., not restricted to a single straight line), its distribution in the plane orthogonal to the first reference plane is not rotationally symmetric. This condition is satisfied by all experimental point clouds in this work, so the second reference plane is unique, and the formal proof is provided later in the mathematical derivation.

- The third reference plane is also defined by two constraints: 1) It is perpendicular to the first and second reference planes; 2) Under the first constraint, it minimizes the sum of squared distances between each point and the plane.

Once the first and second reference planes are uniquely determined, the third reference plane is also uniquely defined, because it is fixed by their orthogonality. The mathematical reasoning is given in the subsequent mathematical derivation.

A plane in three-dimensional space is defined by a normal vector and a point through which it passes. If a plane passes through a point $\hat{\boldsymbol{c}}$ and is perpendicular to a vector $\hat{\boldsymbol{u}}$, then the vector between $\hat{\boldsymbol{c}}$ and any other point on the plane is orthogonal to $\hat{\boldsymbol{u}}$. Thus, a plane can be mathematically described by a normal vector $\hat{\boldsymbol{u}} = [A\ B\ C]^T$ which is perpendicular to the plane, and a point $\hat{\boldsymbol{c}} = [\hat{x}\ \hat{y}\ \hat{z}]$ on the plane. Using this formulation, the equation of the plane is given by Eq. (4.4):

$$A(x - \hat{x}) + B(y - \hat{y}) + C(z - \hat{z}) = 0 \tag{4.4}$$

If $-A\hat{x} - B\hat{y} - C\hat{z} = D$, then Eq. (4.4) can be rewritten in the form of Eq. (4.5):

$$Ax + By + Cz + D = 0 \tag{4.5}$$

Eq. (4.5) indicates that a plane in three-dimensional space is fully defined by the four coefficients A, B, C and D.

Since the first reference plane is defined as the least-squares plane fitted to the point cloud, determining this plane can be formulated as a mathematical optimization problem: given a point cloud $\boldsymbol{P} = \{\boldsymbol{p_i} \mid \boldsymbol{p_i} = [x_i\ y_i\ z_i]^T, i \in \mathbf{N}\ \cap\ i \in [1, n]\}$, for a plane $A_1x + B_1y + C_1z + D_1 = 0$ with the unit normal vector $\hat{\boldsymbol{u}}_{\boldsymbol{1}} = [A_1\ B_1\ C_1]^T$, the sum of the squared distances between each point $\boldsymbol{p_i}$ in the point cloud $\boldsymbol{P}$ and the plane $A_1x + B_1y + C_1z + D_1 = 0$ can be expressed as a cost function shown in Eq. (4.6):

$$f(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1) = \sum_{i=1}^{n}\left(\hat{\boldsymbol{u}}_{\boldsymbol{1}}{}^T\boldsymbol{p_i} + D_1\right)^2 \tag{4.6}$$

The goal is to determine the plane equation that minimizes the cost function $f(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1)$ defined in Eq. (4.6). Specifically, this requires finding the unit normal vector $\hat{\boldsymbol{u}}_{\boldsymbol{1}}$ and variable D_1 that minimize $f(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1)$. This optimization can be formally expressed as Eq. (4.7):

$$< \hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1 > = min_{\hat{\mathbf{u}}_{\mathbf{1}}, D_1} f(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1) \tag{4.7}$$

When the cost function $f(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1)$ reaches its minimum, its partial derivative with respect to the variable D_1 must be zero. This condition can be mathematically expressed as Eq. (4.8) and Eq. (4.9):

$$f_D'(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1) = 2\sum_{i=1}^{n}(\hat{\boldsymbol{u}}_{\boldsymbol{1}}{}^T\boldsymbol{p_i} + D_1) \tag{4.8}$$

$$= 0$$

$$D_1 = -\hat{\boldsymbol{u}}_{\boldsymbol{1}}{}^T \overline{\boldsymbol{p}} \tag{4.9}$$

In Eq. (4.9), the term $\overline{\boldsymbol{p}} = \frac{\sum_{i=1}^{n} \boldsymbol{p}_i}{n}$ represents the centroid of the point cloud $\boldsymbol{P}$. Substituting Eq. (4.9) into the plane equation $A_1x + B_1y + C_1z + D_1 = 0$ for the first reference plane yields Eq. (4.10):

$$A_1x + B_1y + C_1z - \hat{\boldsymbol{u}}_{\boldsymbol{1}}{}^T \overline{\boldsymbol{p}} = 0 \tag{4.10}$$

$$A_1x + B_1y + C_1z - (A_1\bar{x} + B_1\bar{y} + C_1\bar{z}) = 0$$

$$A_1(x - \bar{x}) + B_1(y - \bar{y}) + C_1(z - \bar{z}) = 0$$

In Eq. (4.10), the term $[\bar{x}\ \bar{y}\ \bar{z}]$ represents the coordinates of the point cloud centroid $\overline{\boldsymbol{p}}$. This implies that, regardless of the unit normal vector $\hat{\boldsymbol{u}}_{\boldsymbol{1}}$ of the first reference plane, the plane must always pass through the centroid $\overline{\boldsymbol{p}}$ of the point cloud. As illustrated in Figure 4-5, this property allows for a simplification of the problem of finding the first reference plane. Instead of directly computing the plane, the point cloud $\boldsymbol{P}$ can be normalized by subtracting its centroid $\overline{\boldsymbol{p}}$ from each point $\boldsymbol{p}_{\boldsymbol{i}}$ in the point cloud. This transformation yields a normalized point cloud $\boldsymbol{P'}$ whose centroid $\overline{\boldsymbol{p'}}$ is at the origin $[0\ 0\ 0]^T$, making the computation of the reference plane more straightforward.

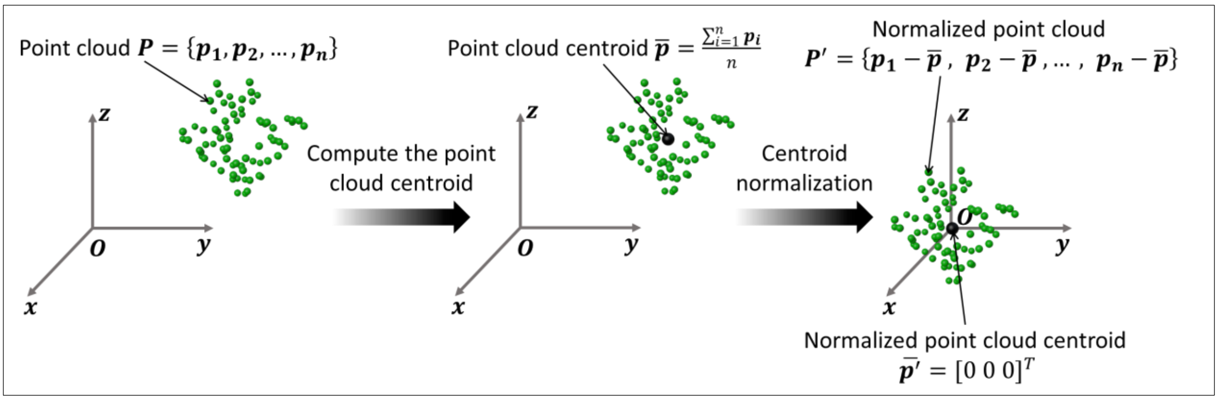

Point cloud $\boldsymbol{P} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \dots, \boldsymbol{p_n}\}$
z
O
y
x
Compute the point cloud centroid
Point cloud centroid $\overline{\boldsymbol{p}} = \frac{\sum_{i=1}^{n} \boldsymbol{p}_i}{n}$
z
O
y
x
Centroid normalization
Normalized point cloud
$\boldsymbol{P'} = \{\boldsymbol{p_1} - \overline{\boldsymbol{p}},\ \boldsymbol{p_2} - \overline{\boldsymbol{p}}, \dots,\ \boldsymbol{p_n} - \overline{\boldsymbol{p}}\}$
z
O
y
x
Normalized point cloud centroid
$\overline{\boldsymbol{p'}} = [0\ 0\ 0]^T$

Figure 4-5. Centroid normalization of the point cloud $\boldsymbol{P}$.

By applying centroid normalization, the least-squares plane of the shifted point cloud is forced to pass through the origin, which results in $D_1 = 0$. Consequently, the plane equation simplifies from $A_1x + B_1y + C_1z + D_1 = 0$ to $A_1x + B_1y + C_1z = 0$. Since the unit normal vector of the fitted plane remains unchanged, this transformation reduces the number of unknowns in the cost function $f(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1)$ from two variables $(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, D_1)$ to one variable $(\hat{\boldsymbol{u}}_{\boldsymbol{1}})$, simplifying the optimization process. Thus, the problem of determining the first reference plane can be reformulated as: given a normalized point cloud

$\boldsymbol{P'} = \left\{\boldsymbol{p_i}' \mid \boldsymbol{p_i}' = \boldsymbol{p_i} - \overline{\boldsymbol{p}}\,,\ \boldsymbol{p_i} = [x_i\ y_i\ z_i]^T,\ \overline{\boldsymbol{p}} = \frac{\sum_{i=1}^{n} \boldsymbol{p_i}}{n},\ i \in \mathbf{N}\ \cap\ i \in [1,n]\right\}$ and a plane equation $A_1x + B_1y + C_1z = 0$ with a unit normal vector $\widehat{\boldsymbol{u}}_{\boldsymbol{1}} = [A_1\ B_1\ C_1]^T$, the sum of squared distances between each point $\boldsymbol{p_i}'$ in the normalized point cloud $\boldsymbol{P'}$ and the plane can be expressed as Eq. (4.11):

$$f(\widehat{\boldsymbol{u}}_{\boldsymbol{1}}) = \sum_{i=1}^{n}\left(\widehat{\boldsymbol{u}}_{\boldsymbol{1}}{}^{T}\boldsymbol{p_i}'\right)^2 \tag{4.11}$$

The objective is to find the unit normal vector $\widehat{\boldsymbol{u}}_{\boldsymbol{1}}$ that minimizes $f(\widehat{\boldsymbol{u}}_{\boldsymbol{1}})$, which can be formulated as Eq. (4.12):

$$< \widehat{\boldsymbol{u}}_{\boldsymbol{1}} > = min_{\widehat{\mathbf{u}}_1} f(\widehat{\boldsymbol{u}}_{\boldsymbol{1}}) \tag{4.12}$$

The point clouds $\boldsymbol{P}$ and $\boldsymbol{P'}$ can be expressed in the matrix form as Eq. (4.13) and Eq. (4.14):

$$\boldsymbol{P} = [\boldsymbol{p_1}, \boldsymbol{p_2}, \dots, \boldsymbol{p_n}] \tag{4.13}$$

$$\boldsymbol{P'} = [\boldsymbol{p_1} - \overline{\boldsymbol{p}}\,,\ \boldsymbol{p_2} - \overline{\boldsymbol{p}}\,, \dots,\ \boldsymbol{p_n} - \overline{\boldsymbol{p}}] \tag{4.14}$$

Using this matrix notation, Eq. (4.11) and Eq. (4.12) can be rewritten as:

$$f(\widehat{\boldsymbol{u}}_{\boldsymbol{1}}) = \widehat{\boldsymbol{u}}_{\boldsymbol{1}}{}^{T}(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)^T\widehat{\boldsymbol{u}}_{\boldsymbol{1}} \tag{4.15}$$

$$< \widehat{\boldsymbol{u}}_{\boldsymbol{1}} > = min_{\widehat{\mathbf{u}}_1}\widehat{\boldsymbol{u}}_{\boldsymbol{1}}{}^{T}(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)^T\widehat{\boldsymbol{u}}_{\boldsymbol{1}} \tag{4.16}$$

where

$$\boldsymbol{1} = [1,1,\dots,1]_{\boldsymbol{n}\times\boldsymbol{1}}{}^{T} \tag{4.17}$$

In Eq. (4.15) and Eq. (4.16), the matrix $(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)^T$ is symmetric, meaning it can be factorized into the form of $\boldsymbol{U\Sigma U}^T$ using SVD (Singular Value Decomposition) [285]; therefore, Eq. (4.16) can then be rewritten as Eq. (4.18):

$$< \widehat{\boldsymbol{u}}_{\boldsymbol{1}} > = min_{\widehat{\mathbf{u}}_1}(\widehat{\boldsymbol{u}}_{\boldsymbol{1}})^T\boldsymbol{U}\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}\boldsymbol{U}^{\boldsymbol{T}}\widehat{\boldsymbol{u}}_{\boldsymbol{1}} \tag{4.18}$$

In Eq. (4.18), $\boldsymbol{U}$ is an orthogonal matrix, meaning that its column vectors are the eigenvectors of the symmetric matrix $(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)^T$. The eigenvalues of this matrix are denoted as λ_1, λ_2, λ_3, and they are distinct when the point cloud is fully distributed in three dimensions, rather than being constrained to a line or a plane. Specifically, this requires that the points are neither collinear nor coplanar, and that their spatial distribution is not rotationally symmetric within any principal plane. In this work, all experimental point clouds satisfy these non-degeneracy conditions, so the three eigenvalues are guaranteed to be different. Since they are distinct, we can, without loss of generality, order them as $\lambda_1 > \lambda_2 > \lambda_3$. Consequently, the three corresponding reference planes derived from these eigenvalues are also uniquely defined. As discussed in Section 3.1 , an orthogonal matrix is also a rotation matrix, meaning that both its columns and rows are unit orthogonal vectors. Consequently, $\boldsymbol{U^T}$ is also a rotation matrix, and multiplying it by a unit vector $\widehat{\boldsymbol{u}}_{\mathbf{1}}$ results in another unit vector $\widehat{\boldsymbol{u}}_{\mathbf{1}}{}' = [A_1{}'\ B_1{}'\ C_1{}']^T$. Using this notation, Eq. (4.15) and Eq. (4.16) can be rewritten as Eq. (4.19), Eq. (4.20) and Eq. (4.21):

$$(A_1{}')^2 + (B_1{}')^2 + (C_1{}')^2 = 1 \tag{4.19}$$

$$f(A_1{}', B_1{}', C_1{}') = \lambda_1 (A_1{}')^2 + \lambda_2 (B_1{}')^2 + \lambda_3 (C_1{}')^2 \tag{4.20}$$

$$< A_1{}', B_1{}', C_1{}' > = min_{A_1{}', B_1{}', C_1{}'} f(A_1{}', B_1{}', C_1{}') \tag{4.21}$$

Since $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and based on Eq. (4.19), the cost function $f(A_1{}', B_1{}', C_1{}')$ in Eq. (4.20) reaches its minimum when Eq. (4.22) holds:

$$\begin{aligned} A_1{}' &= 0 \\ B_1{}' &= 0 \\ C_1{}' &= 1 \end{aligned} \tag{4.22}$$

Eq. (4.22) implies that $\widehat{\boldsymbol{u}}_{\mathbf{1}}{}' = [0\ 0\ 1]^T$, leading to Eq. (4.23) and Eq. (4.24):

$$\boldsymbol{U}^T \widehat{\boldsymbol{u}}_{\mathbf{1}} = [0\ 0\ 1]^T \tag{4.23}$$

$$\widehat{\boldsymbol{u}}_{\mathbf{1}} = \boldsymbol{U}[0\ 0\ 1]^T \tag{4.24}$$

Eq. (4.24) shows that when $\widehat{\boldsymbol{u}}_{\mathbf{1}}$ is equal to the last column vector of $\boldsymbol{U}$, the cost function $f(\widehat{\boldsymbol{u}}_{\mathbf{1}}, D)$ reaches its minimum, satisfying Eq. (4.7). Therefore, the unit normal vector of the first reference plane is given by the last column vector of $\boldsymbol{U}$. With the unit normal vector determined, the plane equation of the first reference plane is expressed as $A_1 x + B_1 y + C_1 z + D_1 = 0$. Since the first reference plane passes through the centroid of the point cloud, substituting the centroid coordinates into this equation allows for the determination of D_1. In this way, the complete equation of the first reference plane is obtained. The workflow for acquiring the first reference plane is illustrated in Figure 4-6.

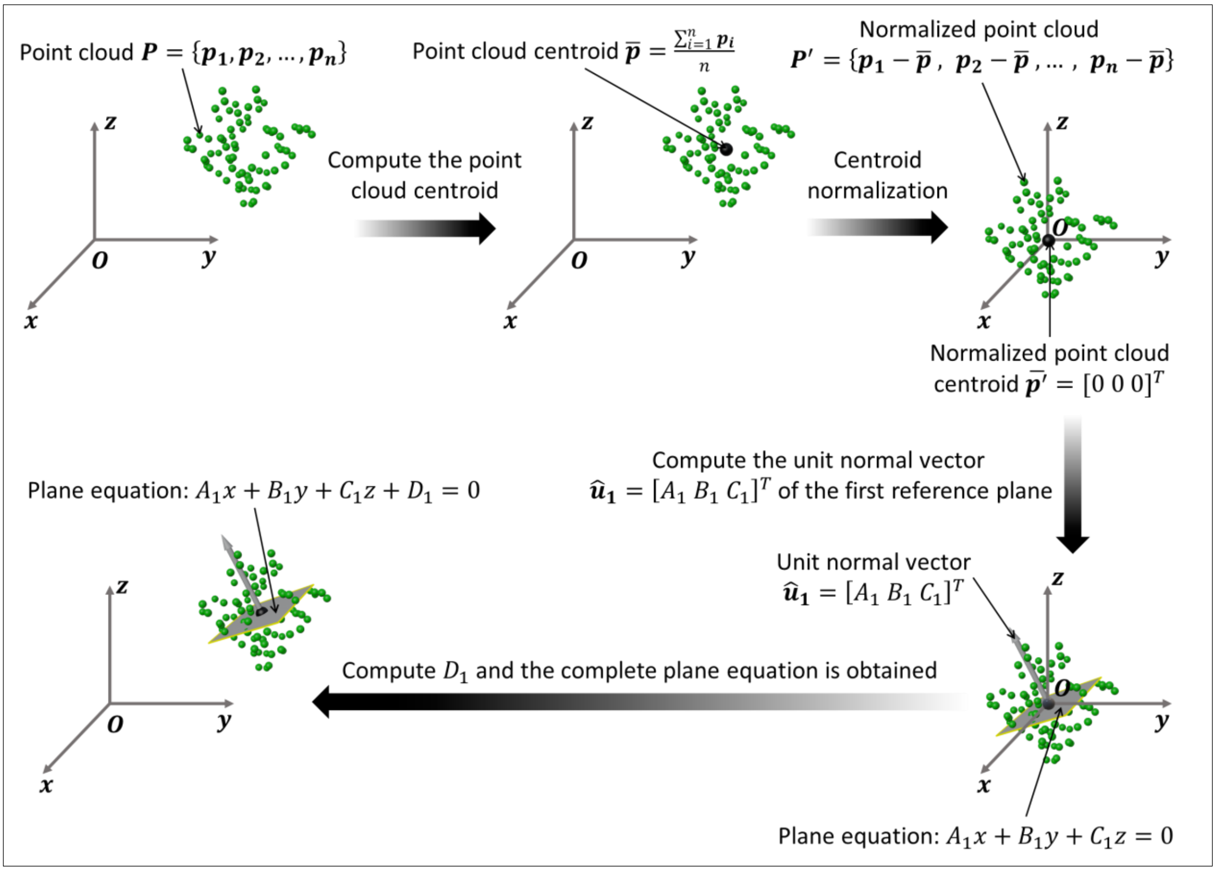

Point cloud $\boldsymbol{P} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \dots, \boldsymbol{p_n}\}$
Point cloud centroid $\overline{\boldsymbol{p}} = \frac{\sum_{i=1}^{n} \boldsymbol{p_i}}{n}$
Normalized point cloud
$\boldsymbol{P'} = \{\boldsymbol{p_1} - \overline{\boldsymbol{p}}, \boldsymbol{p_2} - \overline{\boldsymbol{p}}, \dots, \boldsymbol{p_n} - \overline{\boldsymbol{p}}\}$
Compute the point cloud centroid
Centroid normalization
Normalized point cloud centroid $\overline{\boldsymbol{p}'} = [0\ 0\ 0]^T$
Compute the unit normal vector $\boldsymbol{\hat{u}_1} = [A_1\ B_1\ C_1]^T$ of the first reference plane
Plane equation: $A_1x + B_1y + C_1z + D_1 = 0$
Unit normal vector $\boldsymbol{\hat{u}_1} = [A_1\ B_1\ C_1]^T$
Compute D_1 and the complete plane equation is obtained
Plane equation: $A_1x + B_1y + C_1z = 0$
z
y
x
O

Figure 4-6. Workflow for acquiring the first reference plane of the point cloud $\boldsymbol{P}$.

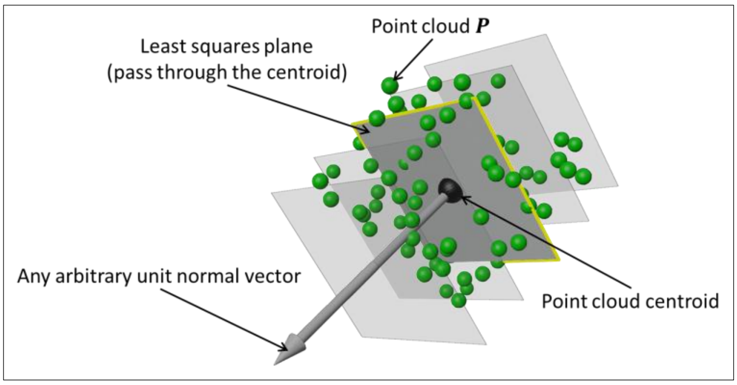

Point cloud $\boldsymbol{P}$
Least squares plane (pass through the centroid)
Any arbitrary unit normal vector
Point cloud centroid

Figure 4-7. Illustration showing that the least squares plane corresponding to any arbitrary unit normal vector must pass through the centroid of the point cloud. Any parallel planes (shown as translucent grey squares) that do not pass through the centroid fail to achieve the minimum sum of squared distances.

Eq. (4.8), Eq. (4.9) and Eq. (4.10) demonstrate that for any arbitrary unit normal vector, the corresponding least-squares plane that minimizes the cost function in Eq. (4.6) must pass through the point cloud centroid. The concept is visually illustrated in Figure 4-7. Therefore, Since the second and third reference planes are also least-squares planes, they must also pass through the centroid of the point cloud as well. The equation of the second reference plane is expressed as $A_2x + B_2y + C_2z + D_2 = 0$.

Since it is perpendicular to the first reference plane, its unit normal vector $\hat{\boldsymbol{u}}_{\mathbf{2}} = [A_2\ B_2\ C_2]^T$ is orthogonal to the unit normal vector $\hat{\boldsymbol{u}}_{\mathbf{1}}$ of the first reference plane. This condition can be expressed as Eq. (4.25):

$$(\hat{\boldsymbol{u}}_{\mathbf{2}})^T \hat{\boldsymbol{u}}_{\mathbf{1}} = 0 \tag{4.25}$$

Following the same notation used for $\boldsymbol{U}^T\hat{\boldsymbol{u}}_{\mathbf{1}}$, the unit vector $\boldsymbol{U}^T\hat{\boldsymbol{u}}_{\mathbf{2}}$ can be expressed as $\hat{\boldsymbol{u}}_{\mathbf{2}}{}' = [A_2{}'\ B_2{}'\ C_2{}']^T$. Since $\hat{\boldsymbol{u}}_{\mathbf{2}}{}'$ must also be orthogonal to $\hat{\boldsymbol{u}}_{\mathbf{1}}{}'$, this condition can be rewritten as Eq. (4.26):

$$(\hat{\boldsymbol{u}}_{\mathbf{2}}{}')^T \hat{\boldsymbol{u}}_{\mathbf{1}}{}' = 0 \tag{4.26}$$

$$A_1{}'\,A_2{}' + B_1{}'B_2{}' + C_1{}'C_2{}' = 0$$

Since it has already been established that $\hat{\boldsymbol{u}}_{\mathbf{1}}{}' = [0\ 0\ 1]^T$, it follows from Eq. (4.26) that $C_2{}' = 0$. Thus, $\hat{\boldsymbol{u}}_{\mathbf{2}}{}'$ takes the form in Eq. (4.27):

$$\hat{\boldsymbol{u}}_{\mathbf{2}}{}' = [A_2{}'\ B_2{}'\ 0]^T \tag{4.27}$$

Moreover, since $\hat{\boldsymbol{u}}_{\mathbf{2}}{}'$ is a unit vector, it must satisfy the normalization constraint shown in Eq. (4.28):

$$(A_2{}')^2 + (B_2{}')^2 = 1 \tag{4.28}$$

As with the first reference plane, the second reference plane also minimizes the sum of squared distances from each point in the point cloud. Using the cost function formulation from Eq. (4.15) and substituting $\hat{\boldsymbol{u}}_{\mathbf{2}}$ in place of $\hat{\boldsymbol{u}}_{\mathbf{1}}$, Eq. (4.29) can be obtained:

$$\begin{aligned} f(\hat{\boldsymbol{u}}_{\mathbf{2}}) &= (\hat{\boldsymbol{u}}_{\mathbf{2}})^T(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)(\boldsymbol{P} - \overline{\boldsymbol{p}}\boldsymbol{1}^T)^T\hat{\boldsymbol{u}}_{\mathbf{2}} \\ &= (\hat{\boldsymbol{u}}_{\mathbf{2}})^T\boldsymbol{U}\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}\boldsymbol{U}^T\hat{\boldsymbol{u}}_{\mathbf{2}} \\ &= (\hat{\boldsymbol{u}}_{\mathbf{2}}{}')^T\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}\hat{\boldsymbol{u}}_{\mathbf{2}}{}' \end{aligned} \tag{4.29}$$

The aim is to find the $\hat{\boldsymbol{u}}_{\mathbf{2}}{}'$ that minimizes the cost function $f(\hat{\boldsymbol{u}}_{\mathbf{2}})$. This can be expressed as Eq. (4.30):

$$< \hat{\boldsymbol{u}}_{\mathbf{2}}{}' > = min_{\hat{\boldsymbol{u}}_{\mathbf{2}}{}'}\hat{\boldsymbol{u}}_{\mathbf{2}}{}'^T\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}\hat{\boldsymbol{u}}_{\mathbf{2}}{}' \tag{4.30}$$

As shown in Eq. (4.27), $\hat{\boldsymbol{u}}_{\mathbf{2}}{}'$ equals $[A_2{}'\ B_2{}'\ 0]^T$; therefore, Eq. (4.29) and Eq. (4.30) can be rewritten as Eq. (4.31) and Eq. (4.32):

$$f(A_2{}', B_2{}', 0) = \lambda_1 A_2{}'^2 + \lambda_2 B_2{}'^2 \tag{4.31}$$

$$< A_2', B_2' > = min_{A_2', B_2'} f(A_2', B_2') \tag{4.32}$$

Since $\lambda_1 \geq \lambda_2$ and based on Eq. (4.28), the cost function $f(A_2', B_2', 0)$ shown in Eq. (4.31) reaches its minimum when Eq. (4.33) holds:

$$A_2' = 0 \tag{4.33}$$

$$B_2' = 1$$

Eq. (4.33) implies $\boldsymbol{\hat{u}_2}' = [0\ 1\ 0]^T$. Therefore, substituting this into the orthogonal transformation equation gives Eq. (4.34) and Eq. (4.35):

$$\boldsymbol{U}^T \boldsymbol{\hat{u}_2} = [0\ 1\ 0]^T \tag{4.34}$$

$$\boldsymbol{\hat{u}_2} = \boldsymbol{U}[0\ 1\ 0]^T \tag{4.35}$$

Eq. (4.35) means when $\boldsymbol{\hat{u}_2}$ corresponds to the second-to-last column vector of $\boldsymbol{U}$, the cost function $f(\boldsymbol{\hat{u}_2})$ is minimized, satisfying Eq. (4.30). Therefore, the second to last column vector of $\boldsymbol{U}$ represents the unit normal vector of the second reference plane. With the normal vector $\boldsymbol{\hat{u}_2}$ determined, the equation of the second reference plane can be expressed as $A_2x + B_2y + C_2z + D_2 = 0$. Since the second reference plane also passes through the point cloud centroid, the centroid coordinates can be substituted into the plane equation to solve for D_2. Hence, the complete equation of the second reference plane is obtained. The process of acquiring the second reference plane is illustrated in Figure 4-8.

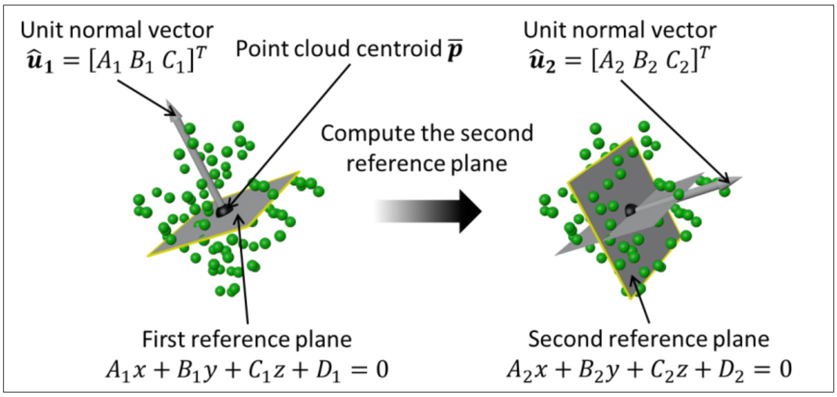

Unit normal vector $\boldsymbol{\hat{u}_1} = [A_1\ B_1\ C_1]^T$
Point cloud centroid $\boldsymbol{\bar{p}}$
Unit normal vector $\boldsymbol{\hat{u}_2} = [A_2\ B_2\ C_2]^T$
Compute the second reference plane
First reference plane $A_1x + B_1y + C_1z + D_1 = 0$
Second reference plane $A_2x + B_2y + C_2z + D_2 = 0$

Figure 4-8. Illustration of the process of acquiring the second reference plane based the first reference plane.

The third reference plane is defined by the equation $A_3x + B_3y + C_3z + D_3 = 0$. Since it is perpendicular to both the first and second reference planes, its unit normal vector $\boldsymbol{\hat{u}_3} = [A_3\ B_3\ C_3]^T$ must be orthogonal to the unit normal vectors $\boldsymbol{\hat{u}_1}$ and $\boldsymbol{\hat{u}_2}$ of the first and second reference planes, respectively. The normal vector $\hat{\boldsymbol{u}}_{\boldsymbol{3}}$ can therefore be computed as the cross product of $\hat{\boldsymbol{u}}_{\boldsymbol{1}}$ and $\hat{\boldsymbol{u}}_{\boldsymbol{2}}$ as shown in Eq. (4.36):

$$\hat{\boldsymbol{u}}_{\boldsymbol{3}} = \hat{\boldsymbol{u}}_{\boldsymbol{1}} \times \hat{\boldsymbol{u}}_{\boldsymbol{2}} \tag{4.36}$$

As previously established, $\boldsymbol{U}$ is an orthogonal matrix, meaning that its column vectors are mutually orthogonal. Therefore, since $\hat{\boldsymbol{u}}_{\boldsymbol{1}}$ and $\hat{\boldsymbol{u}}_{\boldsymbol{2}}$ correspond to the last and second-to-last columns of the matrix $\boldsymbol{U}$, it follows that $\hat{\boldsymbol{u}}_{\boldsymbol{3}}$ corresponds to the first column of the matrix $\boldsymbol{U}$. This can be expressed as Eq. (4.37):

$$\hat{\boldsymbol{u}}_{\boldsymbol{3}} = \boldsymbol{U}[1\ 0\ 0]^T \tag{4.37}$$

Using this result, the coefficients A_3, B_3 and C_3 of the plane equation $A_3x + B_3y + C_3z + D_3 = 0$ are determined. Since the third reference plane must also pass through the centroid of the point cloud, the centroid coordinates can be substituted into the plane equation to solve for D_3. Thus, the full equation of the third reference plane is obtained. The process of acquiring the third reference plane is illustrated in Figure 4-9, while Figure 4-10 provides an overview of the workflow for obtaining all three reference planes.

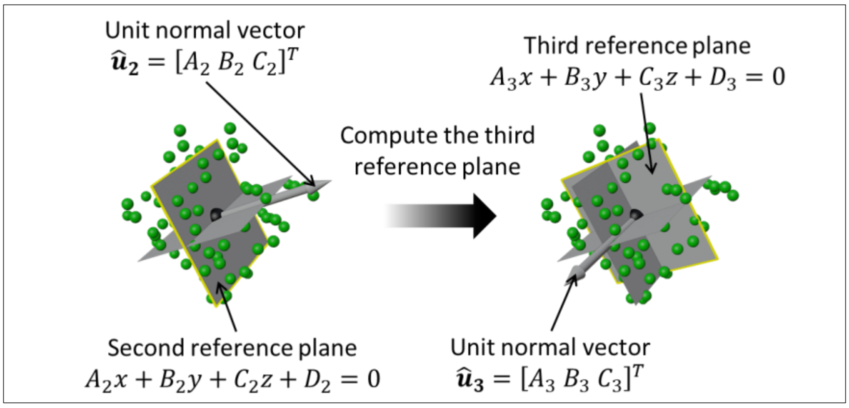

Unit normal vector
$\hat{\boldsymbol{u}}_{\boldsymbol{2}} = [A_2\ B_2\ C_2]^T$
Third reference plane
$A_3x + B_3y + C_3z + D_3 = 0$
Compute the third reference plane
Second reference plane
$A_2x + B_2y + C_2z + D_2 = 0$
Unit normal vector
$\hat{\boldsymbol{u}}_{\boldsymbol{3}} = [A_3\ B_3\ C_3]^T$

Figure 4-9. Illustration of the process of acquiring the third reference plane based on the first and second reference planes.

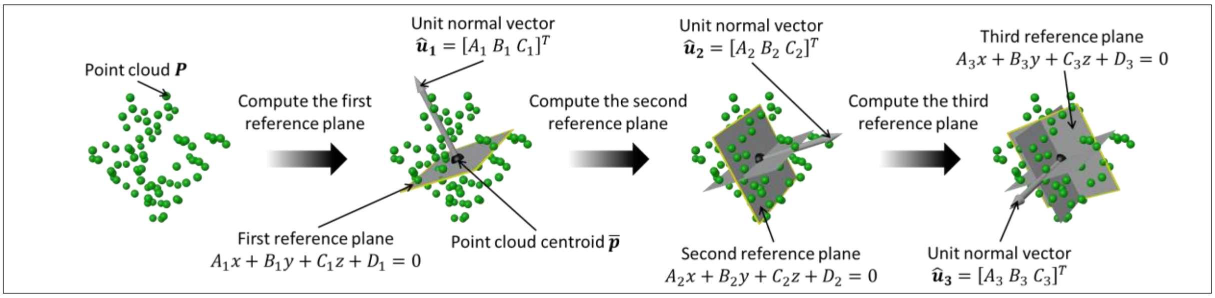

Point cloud $\boldsymbol{P}$
Compute the first reference plane
Unit normal vector
$\hat{\boldsymbol{u}}_{\boldsymbol{1}} = [A_1\ B_1\ C_1]^T$
First reference plane
$A_1x + B_1y + C_1z + D_1 = 0$
Point cloud centroid $\overline{\boldsymbol{p}}$
Compute the second reference plane
Unit normal vector
$\hat{\boldsymbol{u}}_{\boldsymbol{2}} = [A_2\ B_2\ C_2]^T$
Second reference plane
$A_2x + B_2y + C_2z + D_2 = 0$
Compute the third reference plane
Third reference plane
$A_3x + B_3y + C_3z + D_3 = 0$
Unit normal vector
$\hat{\boldsymbol{u}}_{\boldsymbol{3}} = [A_3\ B_3\ C_3]^T$

Figure 4-10. Workflow for acquiring the three mutually orthogonal reference planes.

Once the three reference planes have been determined for both point clouds, the final step of Centre Point Registration (CPR) is to align the two point clouds using these reference planes. The schematic diagram illustrating this alignment process is shown in Figure 4-11. As depicted in the figure, the two point clouds consist of a source point cloud and a target point cloud. During the alignment process, the source point cloud is transformed, while the target point cloud remains fixed. The alignment procedure consists of two sequential transformations:

1) Translation – The source point cloud is translated to the target point cloud so that their centroids coincide.
2) Rotation – The translated source point cloud is rotated around its centroid until its three reference planes align with the corresponding reference planes of the target point cloud.

This transformation ensures that the source point cloud is properly aligned with the target point cloud, forming a strong initial correspondence for further ICP refinement.

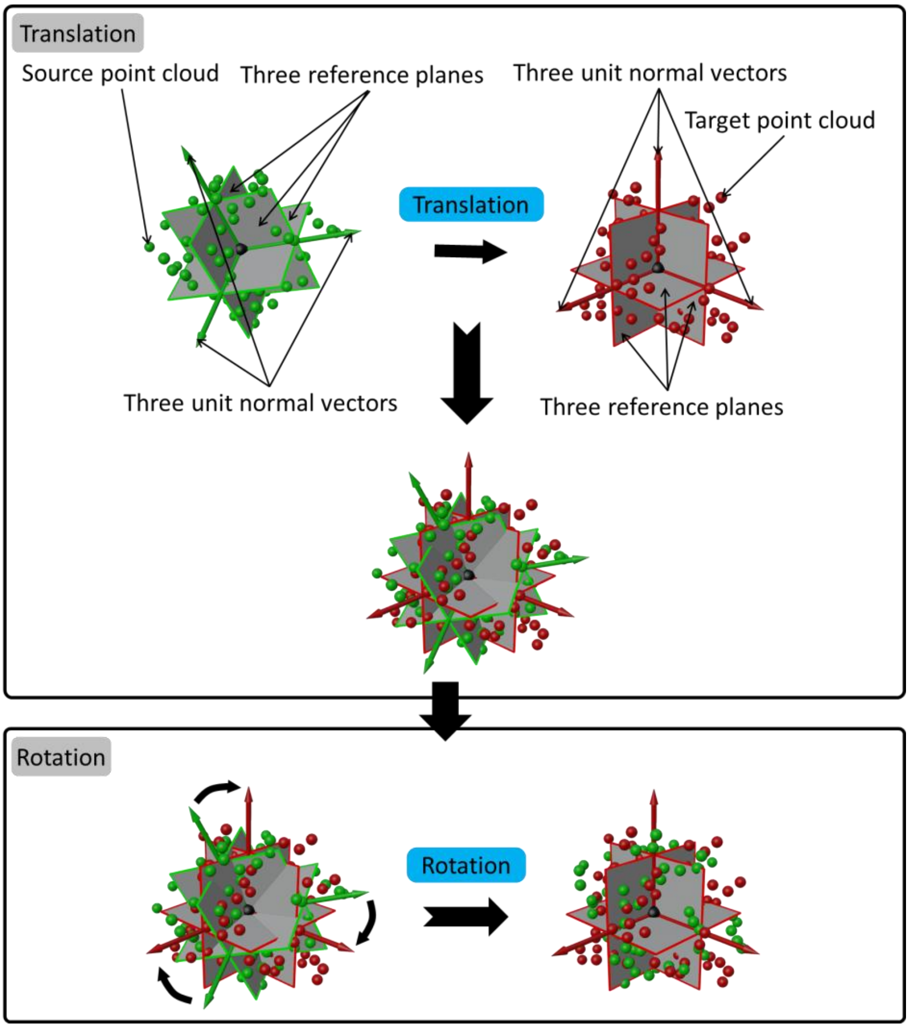


Figure 4-11. Alignment of the source point cloud to the target point cloud using their three reference planes.

After the Centre Point Registration (CPR) operation, which pre-aligns the two point clouds and establishes a strong initial point correspondence, the classic Iterative Closest Point (ICP) algorithm is applied to further refine the alignment. As illustrated in Figure 4-12, ICP is performed on the source point cloud, iteratively minimizing the error between it and the target point cloud, ensuring a more precise and accurate registration.

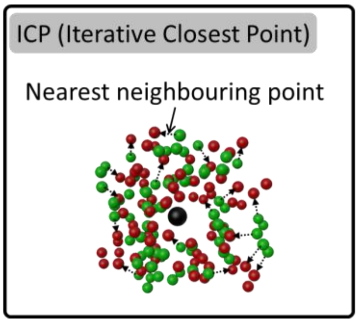


Figure 4-12. Application of the Iterative Closest Point (ICP) algorithm to the source point cloud.

As previously discussed, the CPR-ICP (Centre Point Registration-Iterative Closest Point) algorithm is used to minimize the trajectory error between the estimated trajectory $\boldsymbol{P}$ and the ground truth trajectory $\boldsymbol{P}^*$. This process constitutes the first step in the mapping accuracy improvement workflow, producing a numerical optimization that will later be applied to the estimated map $\boldsymbol{Q}$ in the next step—map error reduction—to enhance its accuracy. The trajectory error minimization process consists of six key steps:

1) Compute the centroid of each trajectory point cloud – The centroid is obtained by calculating the mean of all point coordinates in the trajectory point cloud. Figure 4-13 illustrates this step.

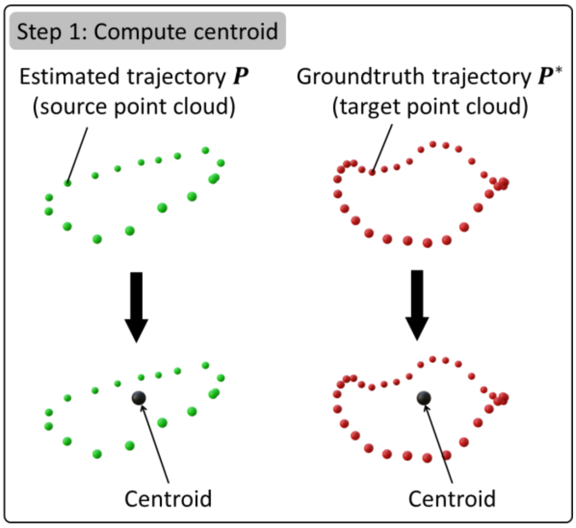


Figure 4-13. First step of trajectory error minimization: computing the centroid of each trajectory point cloud.

2) Normalize each trajectory point cloud – The centroid coordinate is subtracted from each point in the trajectory point cloud, ensuring that the normalized point cloud is cantered at the origin. Figure 4-14 illustrates this transformation.

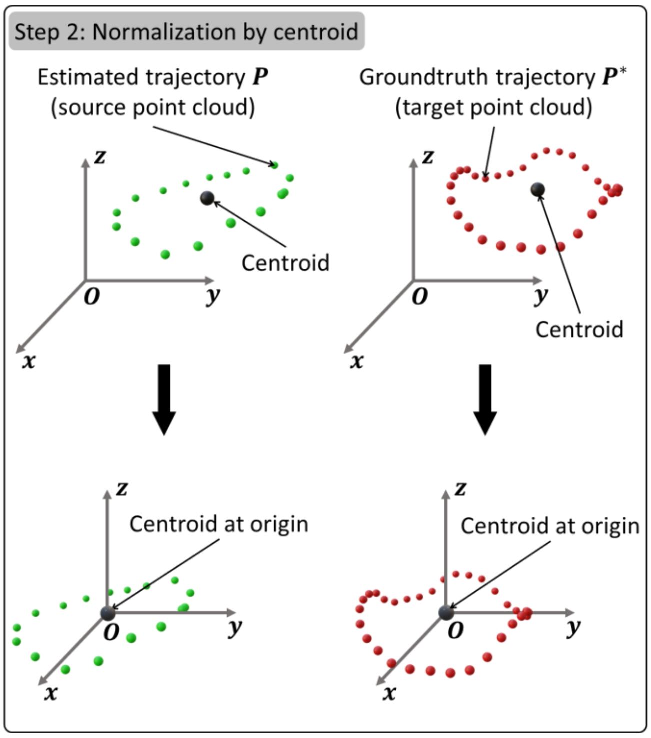

Step 2: Normalization by centroid
Estimated trajectory $\boldsymbol{P}$
(source point cloud)
Groundtruth trajectory $\boldsymbol{P}^*$
(target point cloud)
z
O
y
x
Centroid
z
O
y
x
Centroid
z
Centroid at origin
O
y
x
z
Centroid at origin
O
y
x

Figure 4-14. Second step of trajectory error minimization: normalizing each trajectory point cloud by its centroid.

3) Compute the three reference planes for each trajectory point cloud – This is achieved by applying Singular Value Decomposition (SVD) to the normalized trajectory point cloud, extracting three mutually orthogonal reference planes. Figure 4-15 provides a visual representation of this step.

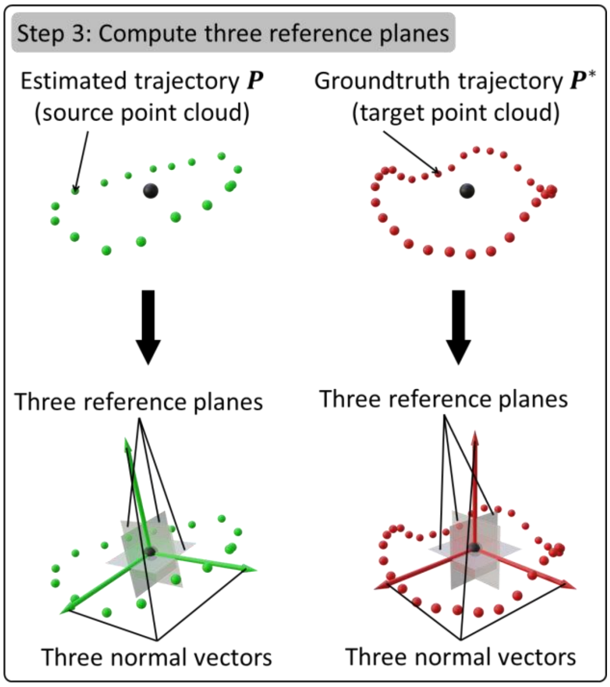

Step 3: Compute three reference planes
Estimated trajectory $\boldsymbol{P}$ (source point cloud)
Groundtruth trajectory $\boldsymbol{P}^*$ (target point cloud)
Three reference planes
Three reference planes
Three normal vectors
Three normal vectors

Figure 4-15. Third step of trajectory error minimization: computing the three reference planes of each trajectory point cloud.

4) Translate the estimated trajectory point cloud $\boldsymbol{P}$ to align with the centroid of the ground truth trajectory $\boldsymbol{P}^*$– This step ensures that both trajectories share the same centroid. The corresponding translation matrix $\boldsymbol{t}$ is obtained, which will later be used in map error reduction. Figure 4-16 illustrates this process.

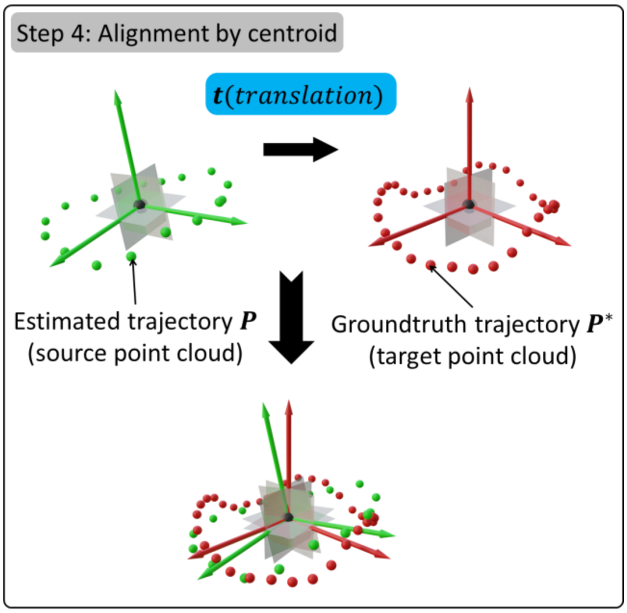

Step 4: Alignment by centroid
$\boldsymbol{t}(translation)$
Estimated trajectory $\boldsymbol{P}$ (source point cloud)
Groundtruth trajectory $\boldsymbol{P}^*$ (target point cloud)

Figure 4-16. Fourth step of trajectory error minimization: aligning the estimated trajectory point cloud $\boldsymbol{P}$ with the ground truth trajectory point cloud $\boldsymbol{P}^*$ by their centroids.

5) Rotate the translated estimated trajectory point cloud $\boldsymbol{P}$ around its centroid – The point cloud is rotated until its three reference planes align with the corresponding reference planes of the ground truth trajectory $\boldsymbol{P}^*$. The corresponding rotation matrix $\boldsymbol{R}$ is computed, which will also be used in map error reduction. Figure 4-17 provides an illustration of this step.

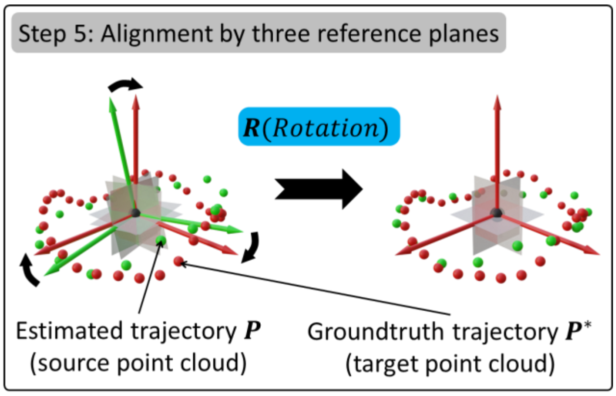

Step 5: Alignment by three reference planes
$\boldsymbol{R}(Rotation)$
Estimated trajectory $\boldsymbol{P}$
(source point cloud)
Groundtruth trajectory $\boldsymbol{P}^*$
(target point cloud)

Figure 4-17. Fifth step of trajectory error minimization: rotating the translated point cloud of the estimated trajectory $\boldsymbol{P}$ around its centroid to align its three reference planes with the corresponding reference planes of the ground truth trajectory point cloud $\boldsymbol{P}^*$.

6) Apply the classic ICP (Iterative Closest Point) algorithm – ICP is used to further reduce the error between the estimated trajectory $\boldsymbol{P}$ and the ground truth trajectory $\boldsymbol{P}^*$. The transformation matrix $\boldsymbol{T}$ output by the ICP algorithm is then acquired and will be applied in the next step—map error reduction. Figure 4-18 illustrates this final step.

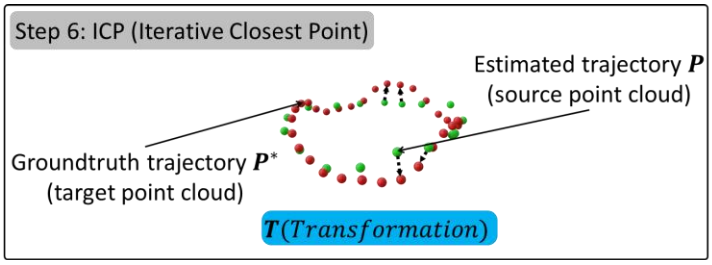

Step 6: ICP (Iterative Closest Point)
Estimated trajectory $\boldsymbol{P}$
(source point cloud)
Groundtruth trajectory $\boldsymbol{P}^*$
(target point cloud)
$\boldsymbol{T}(Transformation)$

Figure 4-18. Sixth step of trajectory error minimization: applying the classic ICP (Iterative Closest Point) algorithm to the point cloud of the estimated trajectory $\boldsymbol{P}$ to further reduce the error between it and the point cloud of the ground truth trajectory $\boldsymbol{P}^*$.

Following the trajectory error minimization step, the next phase in the mapping accuracy improvement workflow is map error reduction. In this step, the numerical optimization obtained from trajectory error minimization is applied to the estimated map $\boldsymbol{Q}$ to enhance its accuracy. The numerical optimization is represented as a Euclidean transformation matrix $\boldsymbol{T}^*$, which is constructed using three transformation components obtained in the previous step:

- The translation matrix $\boldsymbol{t}$ from Step 4 (illustrated in Figure 4-16).
- The rotation matrix $\boldsymbol{R}$ from Step 5 (illustrated in Figure 4-17).
- The transformation matrix $\boldsymbol{T}$ from Step 6 (illustrated in Figure 4-18).

Thus, the Euclidean transformation matrix $\boldsymbol{T}^*$ can be expressed as Eq. (4.38):

$$\boldsymbol{T}^* = \boldsymbol{T} \cdot \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{4.38}$$

As illustrated in Figure 4-19, applying the transformation $\boldsymbol{T}^*$ to the estimated map $\boldsymbol{Q}$ produces a refined estimated map $\widehat{\boldsymbol{Q}}$ with improved accuracy. Mathematically, this transformation is represented as Eq. (4.39):

$$\widehat{\boldsymbol{Q}} = \boldsymbol{T}^* \cdot \boldsymbol{Q} \tag{4.39}$$

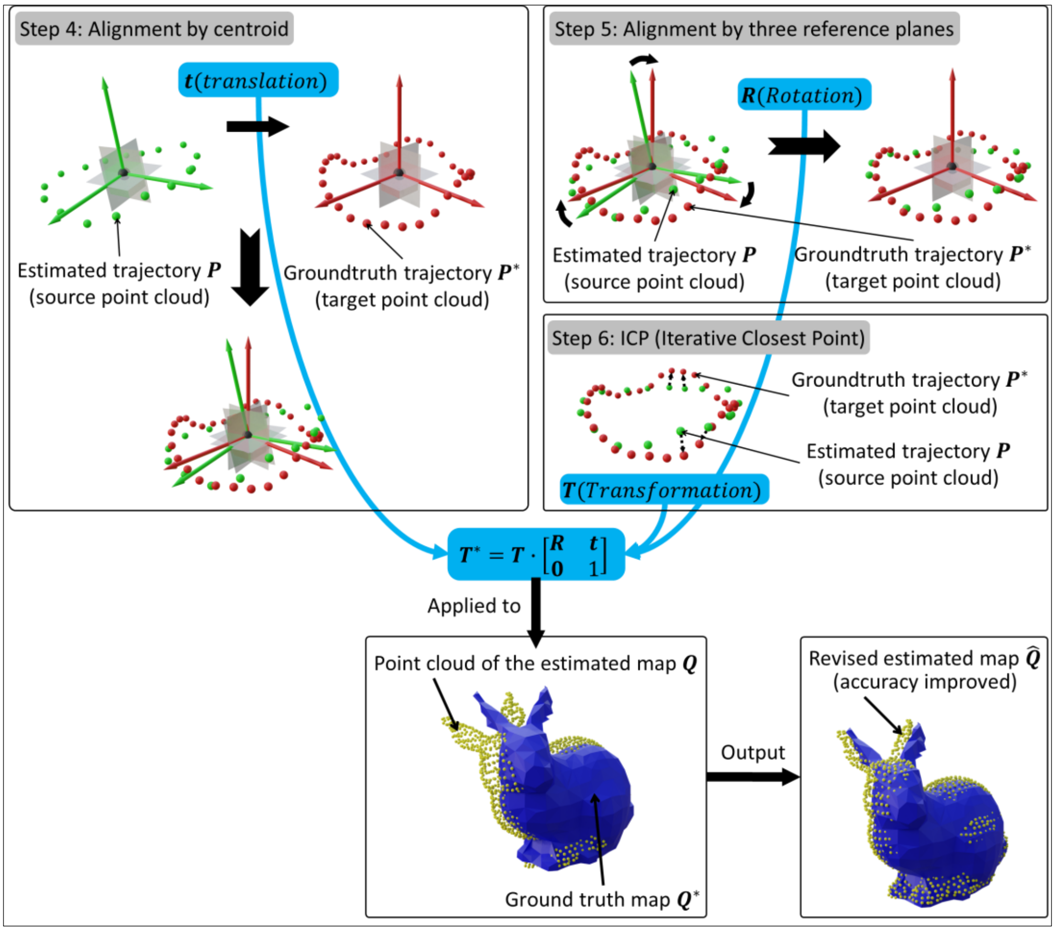

Step 4: Alignment by centroid
$\boldsymbol{t}(translation)$
Estimated trajectory $\boldsymbol{P}$
(source point cloud)
Groundtruth trajectory $\boldsymbol{P}^*$
(target point cloud)
Step 5: Alignment by three reference planes
$\boldsymbol{R}(Rotation)$
Estimated trajectory $\boldsymbol{P}$
(source point cloud)
Groundtruth trajectory $\boldsymbol{P}^*$
(target point cloud)
Step 6: ICP (Iterative Closest Point)
Groundtruth trajectory $\boldsymbol{P}^*$
(target point cloud)
Estimated trajectory $\boldsymbol{P}$
(source point cloud)
$\boldsymbol{T}(Transformation)$
$\boldsymbol{T}^* = \boldsymbol{T} \cdot \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$
Applied to
Point cloud of the estimated map $\boldsymbol{Q}$
Ground truth map $\boldsymbol{Q}^*$
Output
Revised estimated map $\widehat{\boldsymbol{Q}}$
(accuracy improved)

Figure 4-19. Final step: map error reduction step.

The steps outlined above constitute the complete workflow of the mapping accuracy improvement concept. When integrated, these substeps form a systematic process for refining mapping accuracy. The full workflow is visually represented in Figure 4-20, illustrating how each step contributes to the overall accuracy enhancement of the estimated map.

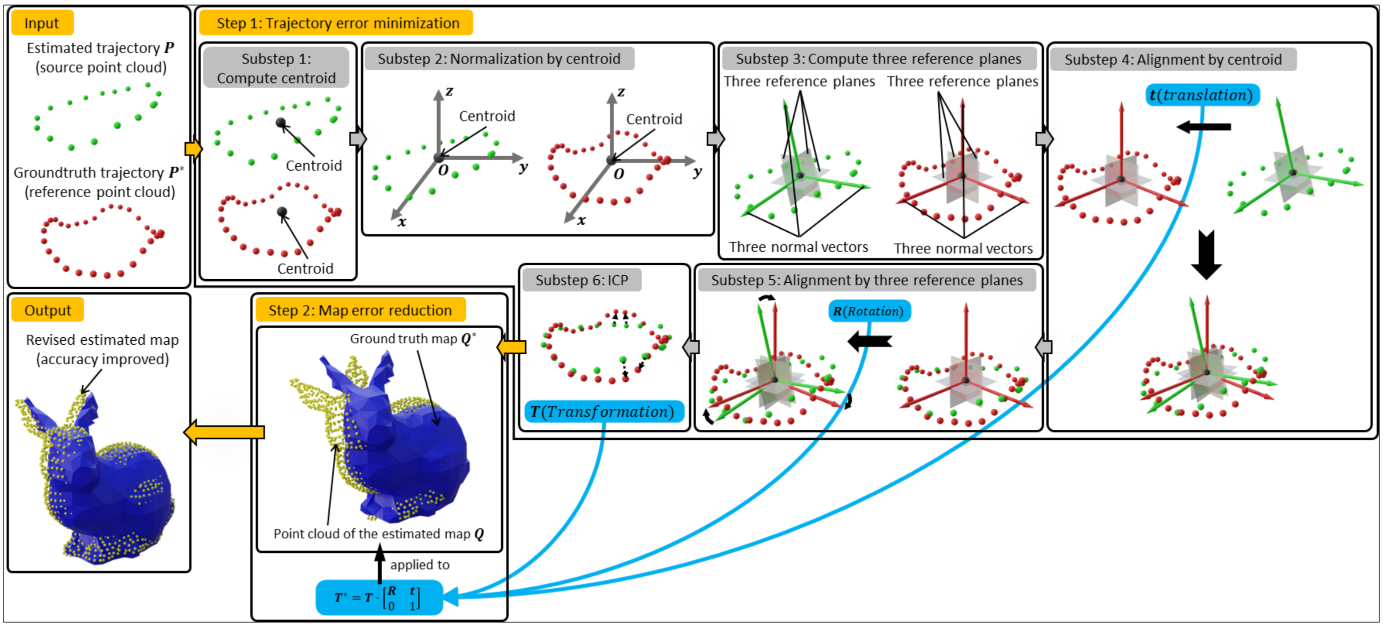

Input
Estimated trajectory P (source point cloud)
Groundtruth trajectory P^* (reference point cloud)
Step 1: Trajectory error minimization
Substep 1: Compute centroid
Centroid
Centroid
Substep 2: Normalization by centroid
Centroid
Centroid
z
y
x
O
Substep 3: Compute three reference planes
Three reference planes
Three reference planes
Three normal vectors
Three normal vectors
Substep 4: Alignment by centroid
$t(translation)$
Substep 5: Alignment by three reference planes
$R(Rotation)$
Substep 6: ICP
$T(Transformation)$
Step 2: Map error reduction
Ground truth map Q^*
Point cloud of the estimated map Q
applied to
$T^* = T \cdot \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$
Output
Revised estimated map (accuracy improved)

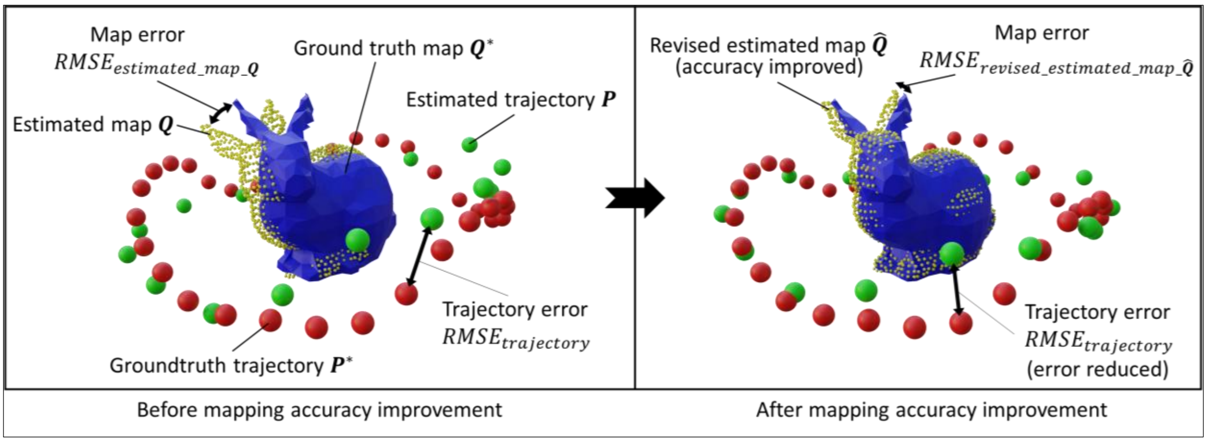
Figure 4-20. Complete workflow for mapping accuracy enhancement.

Similar to Figure 4-3, which provides a conceptual overview of mapping accuracy enhancement, the effect of this enhancement can also be visualized in Figure 4-21. This figure illustrates how trajectory error ($RMSE_{trajectory}$) and map error are both reduced after applying the mapping accuracy enhancement process, resulting in a refined estimated map denoted as $\widehat{\boldsymbol{Q}}$ with enhanced accuracy.

To quantitatively evaluate the effectiveness of mapping accuracy enhancement, the map error before and after refinement are compared. The original map error, corresponding to the estimated map $\boldsymbol{Q}$, is denoted as $RMSE_{estimated\_map\_\boldsymbol{Q}}$, while the map error after refinement, corresponding to the revised estimated map $\widehat{\boldsymbol{Q}}$, is denoted as $RMSE_{revised\_estimated\_map\_\widehat{\boldsymbol{Q}}}$. The performance of mapping accuracy enhancement is assessed using the Relative Reduction to Original (rRtO) metric, which measures the proportion of the reduced map error relative to the original map error. This metric is defined as Eq. (4.40):

$$rRtO = \frac{RMSE_{estimated\_map\_\boldsymbol{Q}} - RMSE_{revised\_estimated\_map\_\widehat{\boldsymbol{Q}}}}{RMSE_{estimated\_map\_\boldsymbol{Q}}} \tag{4.40}$$

This formulation provides a normalized measure of error reduction, allowing for a clear and objective evaluation of the effectiveness of the mapping accuracy enhancement process.


Map error
$RMSE_{estimated\_map\_Q}$
Ground truth map $\boldsymbol{Q}^*$
Estimated trajectory $\boldsymbol{P}$
Estimated map $\boldsymbol{Q}$
Trajectory error
$RMSE_{trajectory}$
Groundtruth trajectory $\boldsymbol{P}^*$
Revised estimated map $\widehat{\boldsymbol{Q}}$
(accuracy improved)
Map error
$RMSE_{revised\_estimated\_map\_\widehat{Q}}$
Trajectory error
$RMSE_{trajectory}$
(error reduced)

Before mapping accuracy improvement

After mapping accuracy improvement

Figure 4-21. Visualization of the effect of mapping accuracy enhancement.

This concludes the introduction of the proposed concept and methodology for enhancing mapping accuracy. The following two chapters provide validation for this approach—alongside the holistic and objective SLAM benchmarking method introduced earlier—through a series of simulation-based and real-world experiments.

Chapter 5 Experimental Validation in Simulated Environments

Previous chapters introduced the proposed innovative methodologies and concept for holistic and objective SLAM benchmarking and mapping accuracy enhancement. This chapter, along with the next, validates these methodologies through simulation-based and real-world experiments, respectively. This chapter begins with Section 5.1 , which details the setup and configuration of the simulation-based experiments. Section 5.2 then presents the experimental results and analysis across four distinct scene types, followed by a data aggregation and summary. Finally, Section 5.3 provides a concluding discussion on the findings of this chapter.

5.1 Experimental setup and configuration

The simulation-based experiments were conducted using ROS (Robot Operating System), Gazebo, RViz, and ORB-SLAM2. Each software tool played a distinct role in the simulation:

- ROS was used for developing robot applications and facilitating communication between different modules.
- Gazebo was used for constructing simulation environments and running the simulations.
- RViz was used for data visualization.
- ORB-SLAM2 was selected as the SLAM algorithm due to its widespread adoption and recognition as one of the most reliable feature-based monocular and RGB-D SLAM systems. It provides real-time operation, loop-closure detection, and map reuse capabilities, making it a strong candidate for evaluating benchmarking methodologies. Moreover, its open-source implementation and modular design ensure reproducibility and comparability across different experimental setups. At the same time, ORB-SLAM2 has known characteristics that may influence the results. For instance, its reliance on ORB feature extraction makes it highly dependent on textured environments, and its performance can degrade in low-texture or repetitive-pattern scenes. Furthermore, while ORB-SLAM2 achieves robust localization, its map representations are sparse and not metrically consistent in large-scale environments, which can affect mapping accuracy evaluation. These characteristics

are important to acknowledge, as they provide context for interpreting the benchmarking results in this chapter.

As illustrated in Figure 5-1, the simulation setup consists of the experimental scene model, sensor model, and robot model. (The two front walls appear invisible because the exported walls are zero-thickness, single-sided meshes whose normals face outward; Gazebo's renderer performs back-face culling, so their inward faces are not drawn when viewed from inside. This is a visualization artifact only and does not affect simulation and SLAM.) The scene model is a cuboid box with thin walls, painted with rich image textures to facilitate feature extraction for the SLAM algorithm. The robot model is a two-wheeled mobile robot equipped with a Kinect depth camera mounted on top. The depth camera has an effective depth range of $0.1m{\sim}20m$, and its parameters, including focal length and coordinate offsets, are specified in its configuration file. The simulation includes several reference frames, among which the global frame $\boldsymbol{G}$ and SLAM camera frame $\boldsymbol{Sc}$ are the only two frames that need to be considered. As mentioned in Section 4.2 , the global frame $\boldsymbol{G}$ serves as the world frame in the simulation software. During the SLAM process, the world frame $\boldsymbol{G}$ stays fixed, and the camera frame $\boldsymbol{Sc}$ moves constantly with the camera as the robot navigates the scene. The camera pose is considered as its relative pose with respect to the world frame $\boldsymbol{G}$. Ground truth camera poses can be obtained directly from ROS, while the estimated camera poses are initially defined in the first camera frame of the captured image sequence. To align them with the global frame, the estimated poses must be transformed using the first captured ground truth camera pose, as shown in Figure 4-1. Similarly, for the map, its ground truth can be obtained since its relative dimensions to the world frame are known. However, like the estimated camera poses, the estimated map is initially referenced in the first camera frame, requiring a transformation to the global frame, following the same approach used for camera pose transformation.

Figure 5-2 and Figure 5-3 showcase the data visualization interfaces used in the simulation. As illustrated in Figure 5-2, while the robot moves within the scene, the RGB image (Figure 5-2 (a)) and depth image (Figure 5-2 (b)) within the field of view (Figure 5-2 (c)) of the Kinect camera are constantly captured and fed into the SLAM algorithm. As shown in Figure 5-3, the SLAM algorithm processes the greyed-out RGB image, extracting key features and generating sparse point clouds while simultaneously estimating the robot's trajectory.

Simulation-based experiments offer multiple advantages over real-world experiments, including:

- Eliminating the need for expensive and complex hardware setups and calibrations.
- Providing a controlled environment for evaluating SLAM performance.
- Allowing repeatability and consistency in experiments.

However, the most significant advantage is that, in simulation, the camera frame can be arbitrarily defined, and its ground truth poses relative to the global frame are directly available. In contrast, real-world experiments require additional reference frames and complex geometric transformations to determine ground truth camera poses within the global frame, as discussed in Section 6.1 .

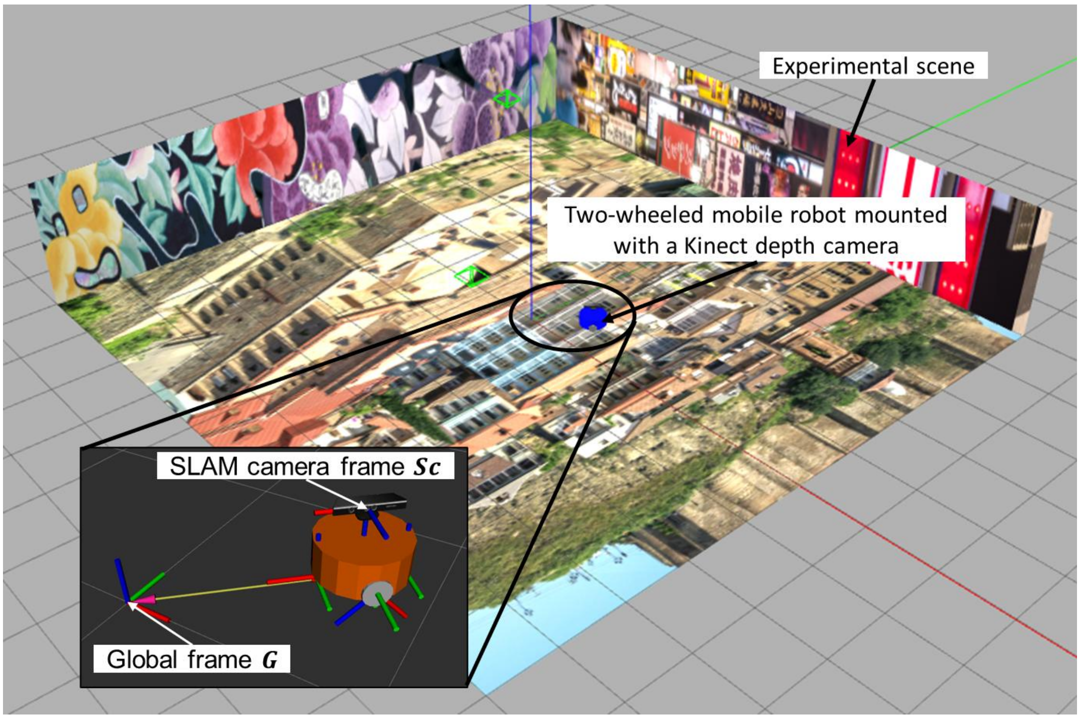


Figure 5-1. Simulation setup in Gazebo.

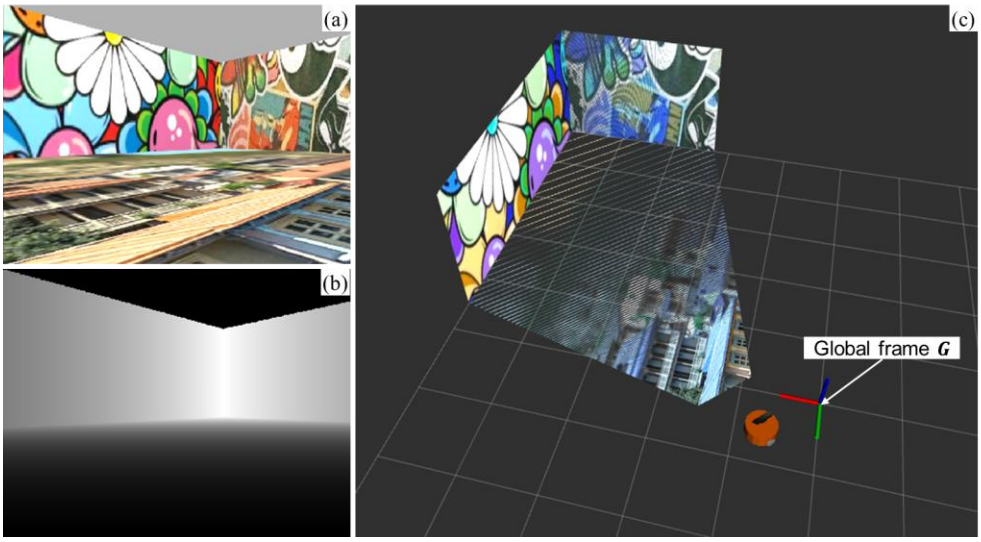
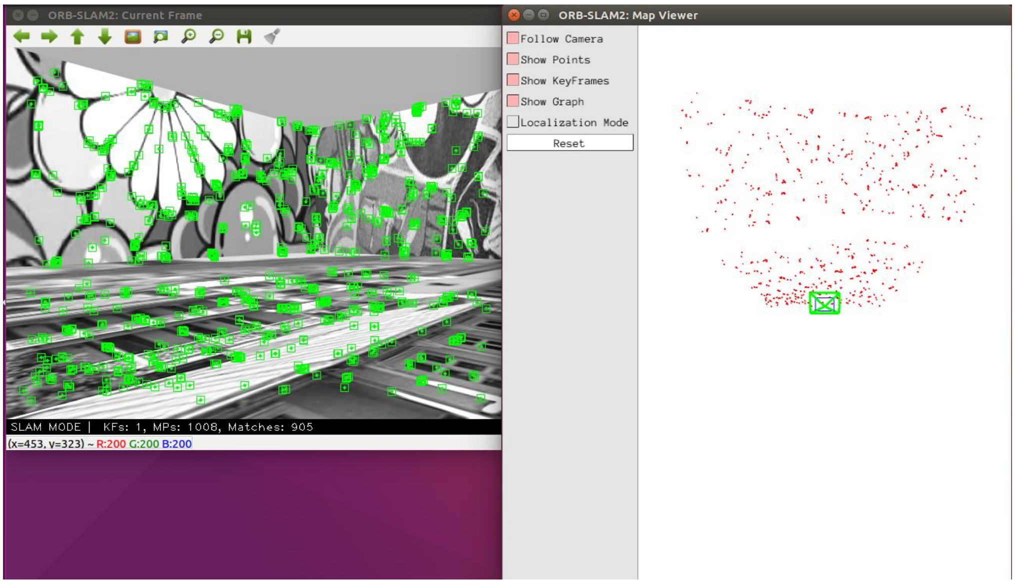


Figure 5-2. Data visualization interface in RViz. (a) RGB image captured by the Kinect camera. (b) Corresponding grayscale depth image captured by the Kinect camera. (c) Robot operating in the environment. As the robot moves through the scene, a point cloud of its field of view is generated and displayed.



☑ Follow Camera
☑ Show Points
☑ Show KeyFrames
☑ Show Graph
□ Localization Mode

```
SLAM MODE |  KFs: 1, MPs: 1008, Matches: 905
(x=453, y=323) ~ R:200 G:200 B:200
```

Figure 5-3. User interface of the ORB-SLAM2 algorithm. The left panel displays a grayscale RGB image captured by the Kinect camera at a fixed frame rate of 30 FPS during the simulation. Green markers indicate features detected by the algorithm. The right panel visualizes the corresponding 3D map points as a point cloud.

5.2 Experiments in synthetic environments

Although more complex synthetic environments could have been used, this study deliberately selected cuboid-shaped synthetic scenes with varying sizes and trajectories. The rationale is twofold. First, simpler cuboid environments enable controlled and systematic evaluation of the proposed holistic SLAM benchmarking and mapping accuracy improvement framework. By isolating variables such as scene dimensions and trajectory geometry, the influence of these factors on benchmarking results can be clearly identified without interference from unnecessary environmental complexity. Second, the aim of the synthetic experiments is not to replicate real-world complexity, but rather to establish a reliable baseline for testing the proposed methodology under controlled conditions. More complex environments are subsequently addressed in Chapter 6 through real-world experiments, ensuring that the methodology is validated across both simplified and realistic settings.

Figure 5-4 presents a bird's-eye view of the four cuboid scenes used in the simulation-based experiments, with dimensions $5m \times 5m \times 2m$, $10m \times 10m \times 2m$, $15m \times 15m \times 2m$ and $20m \times 20m \times 2m$. Within each scene, multiple trials were conducted with robot trajectories of varying geometries, as illustrated in Figure 5-5 (marked in red outlines). The trials were structured as follows:

- For the $5m \times 5m \times 2m$ and $10m \times 10m \times 2m$ scenes (Figure 5-5 (a) and Figure 5-5 (b)), thirty trials were conducted in each scene. These trials were divided into six groups, with each group containing five trials following the same robot trajectory.
- For the $15m \times 15m \times 2m$ scene (Figure 5-5 (c)), forty-five trials were conducted, divided into nine groups, with each group containing five trials for the same robot trajectory.
- For the $20m \times 20m \times 2m$ scene (Figure 5-5 (d)), sixty trials were conducted, divided into twelve groups, with each group consisting of five trials following the same robot trajectory.

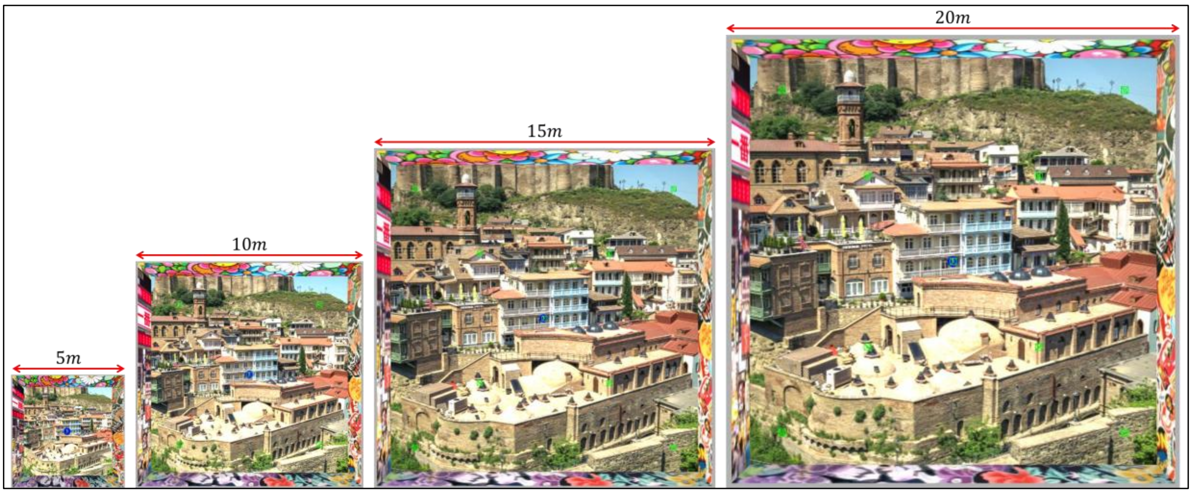
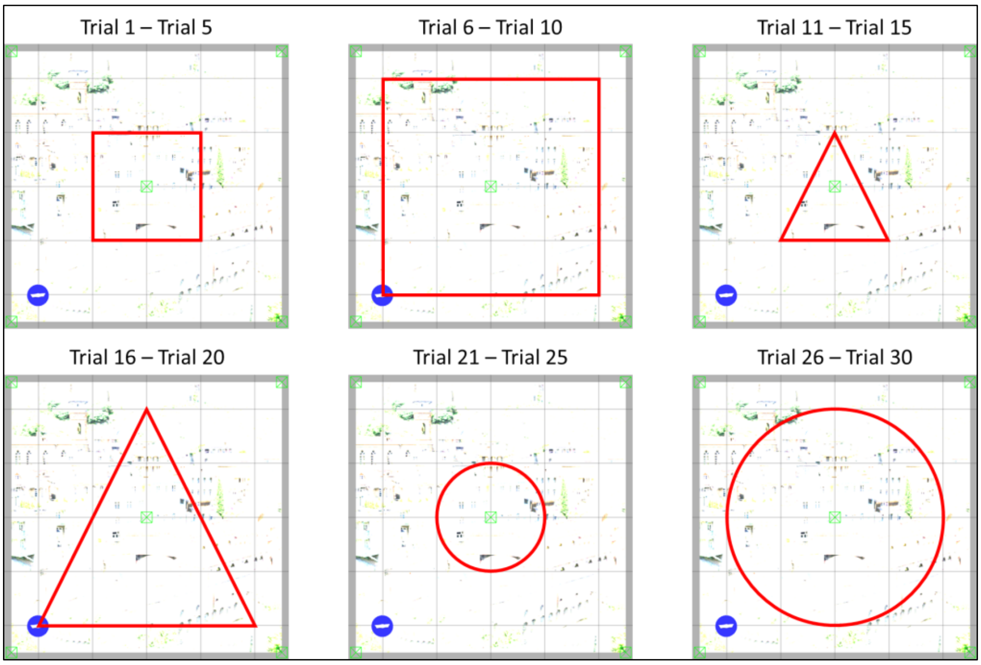
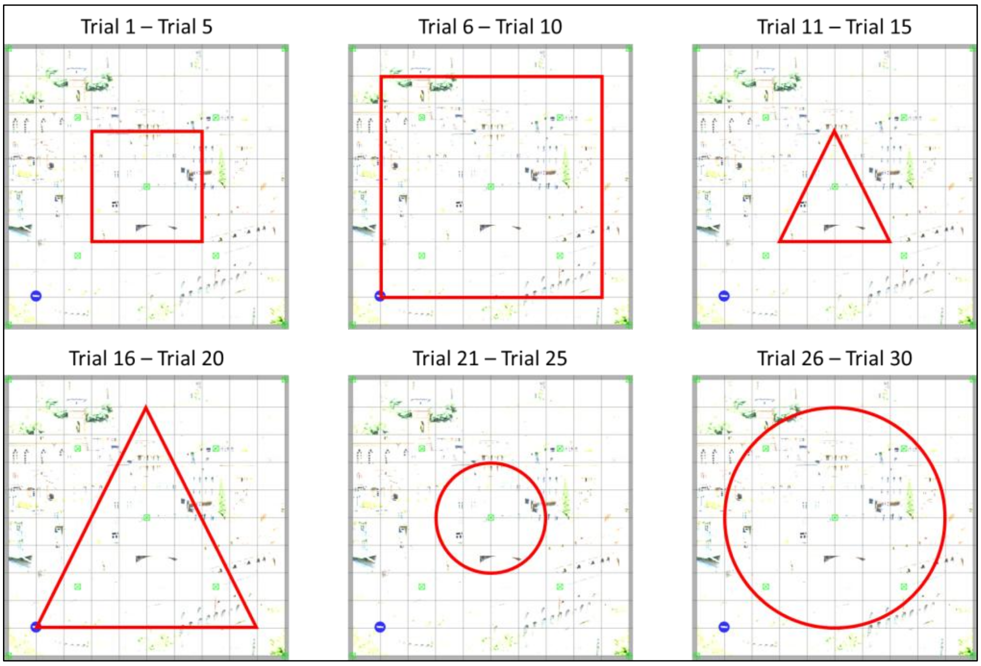


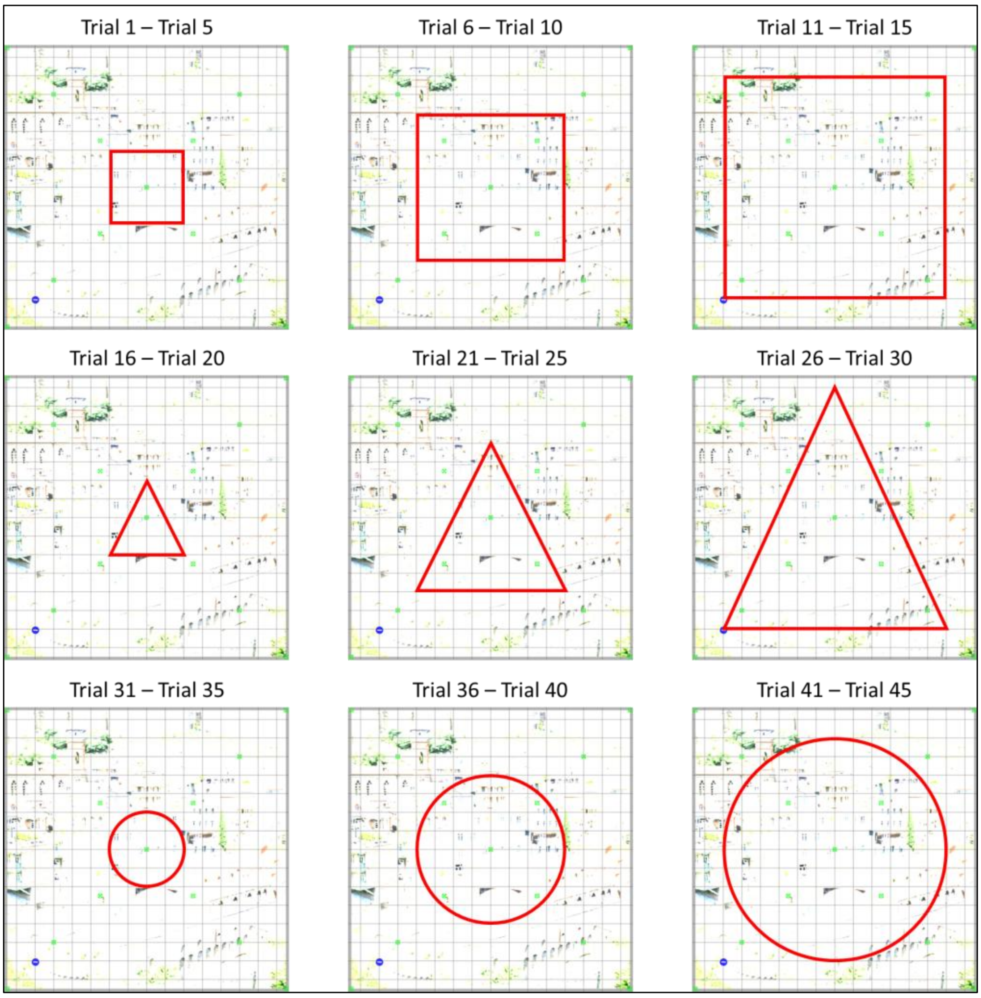
Figure 5-4. Four cuboid scenes of different sizes used in simulation-based experiments.


Trial 1 – Trial 5
Trial 6 – Trial 10
Trial 11 – Trial 15
Trial 16 – Trial 20
Trial 21 – Trial 25
Trial 26 – Trial 30

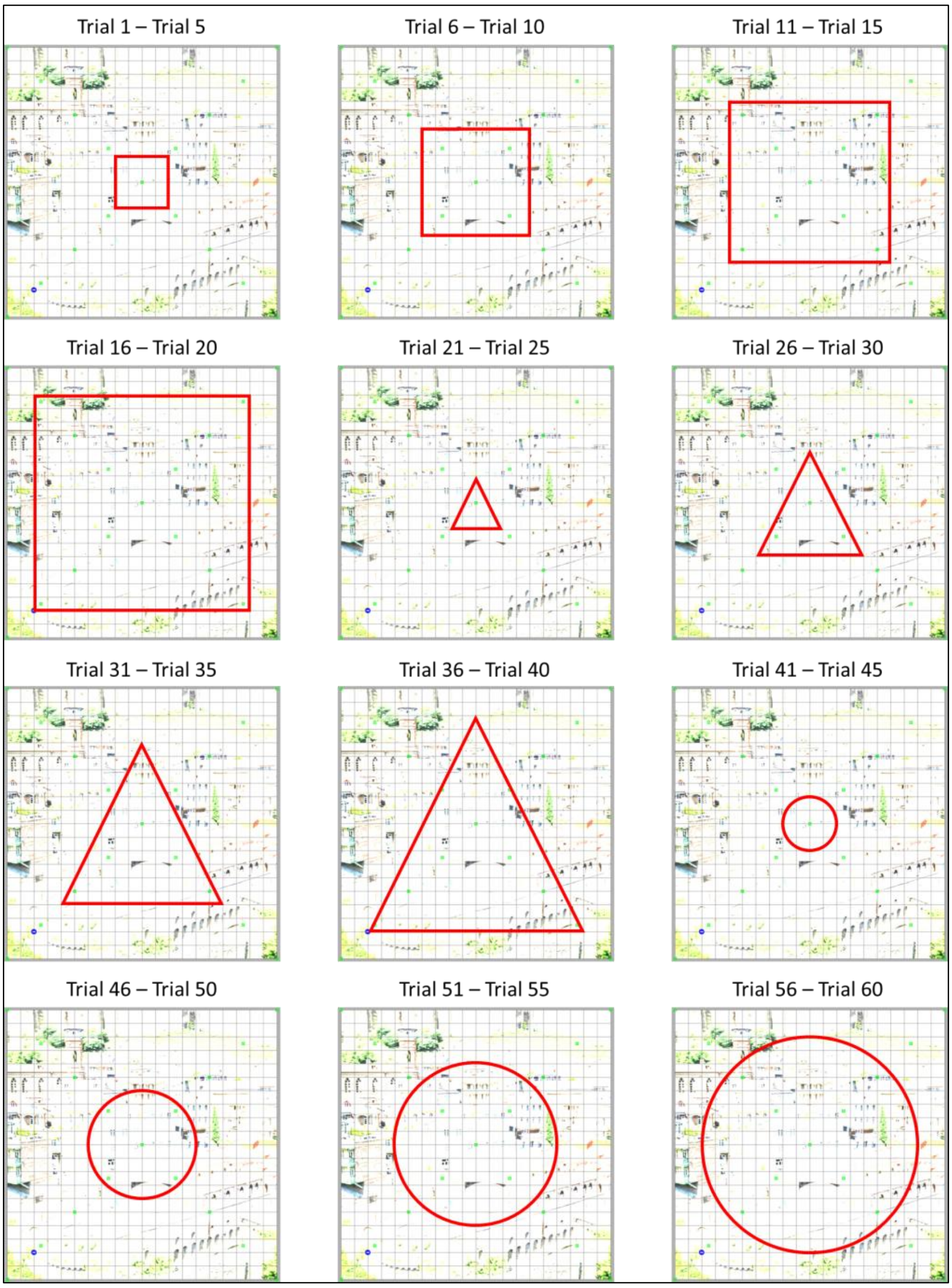
(a)


Trial 1 – Trial 5
Trial 6 – Trial 10
Trial 11 – Trial 15
Trial 16 – Trial 20
Trial 21 – Trial 25
Trial 26 – Trial 30

(b)


Trial 1 – Trial 5
Trial 6 – Trial 10
Trial 11 – Trial 15
Trial 16 – Trial 20
Trial 21 – Trial 25
Trial 26 – Trial 30
Trial 31 – Trial 35
Trial 36 – Trial 40
Trial 41 – Trial 45

(c)



(d)

Figure 5-5. Robot trajectory patterns used in each type of scene. (a) Trajectory patterns used in the $5m \times 5m \times 2m$ scene. (b) Trajectory patterns used in the $10m \times 10m \times 2m$ scene. (c) Trajectory patterns used in the $15m \times 15m \times 2m$ scene. (d) Trajectory patterns used in the $20m \times 20m \times 2m$ scene.

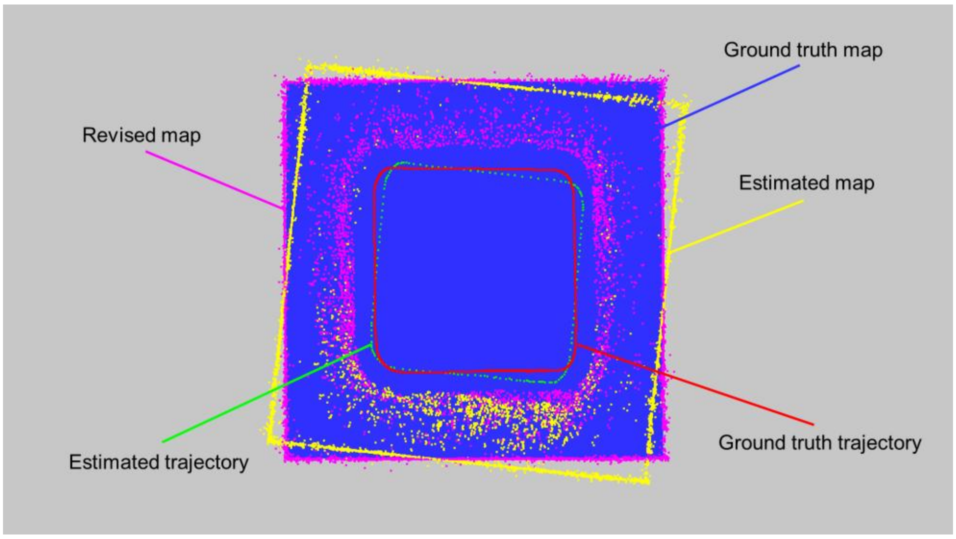


Figure 5-6. Outputs of each trial and the ground truth map.

The use of different scene sizes and varied robot trajectories aims to examine the impact of scene size and trajectory geometry on localization accuracy and mapping accuracy in SLAM benchmarking. Additionally, this setup allows for an in-depth analysis of how the proposed mapping accuracy improvement method is affected by these factors. For a given scene and trajectory, multiple repeated trials were conducted to increase the reliability and accuracy of the results, reduce the influence of random errors, and ensure consistency and reproducibility in the findings.

As shown in Figure 5-6, each trial generates an estimated trajectory, a ground truth trajectory, an estimated map and its corresponding revised map after error reduction. The revised version of the estimated map was generated using the concept of mapping accuracy enhancement shown in Section 4.3 and the approaches used for accuracy improvement (error reduction) were the classic ICP method and the proposed methodology of mapping accuracy enhancement shown in Section 4.4 . The ground truth map is originally captured within the global frame $\boldsymbol{G}$ and is readily available for benchmarking. Both trajectories were represented as point clouds; however, the ground truth trajectory had a significantly higher point density than the estimated trajectory. This disparity results from the ground truth poses being recorded at a high sampling rate (30 samples per second or more), whereas the estimated trajectory consisted only of keyframe poses. To ensure consistency, the ground truth trajectory was down-sampled using a nearest-neighbour approach relative to the estimated trajectory, aligning the number of points before applying ICP or CPR-ICP. This down-sampling step inevitably introduces a small loss of spatial detail in the ground truth trajectory, as intermediate poses between the sampled points are omitted. However, this effect is negligible for two main reasons. First, the original ground truth trajectory has an extremely high temporal resolution (typically 30 Hz or higher), meaning that adjacent ground truth poses are very close spatially; hence, removing some poses does not significantly alter the overall trajectory geometry. Second, the purpose of the down-sampling is to ensure one-to-one correspondence between the estimated and ground truth poses during ICP-based alignment, thereby enabling consistent and unbiased error computation. Retaining the original dense ground truth trajectory without down-sampling would lead to non-uniform correspondence and potentially biased error estimation. Therefore, the adopted down-sampling approach balances computational efficiency and accuracy, ensuring that precision loss is minimal and the resulting error calculation remains reliable. The primary goal of the experiment is to compute and compare trajectory error and map error before and after applying different error reduction techniques. These comparisons enable an evaluation of the effectiveness of each technique and an investigation of their correlation with scene size and trajectory geometry. The detailed results and analysis for experiments conducted in each cuboid scene are presented in the next four sections (Section 5.2.1 to 5.2.4). A comprehensive comparison and overall conclusions are provided in Section 5.2.5 .

5.2.1 Experimental results and analysis for the $5m \times 5m \times 2m$ cuboid scene

For the experiments conducted in the $5m \times 5m \times 2m$ cuboid scene, as shown in Figure 5-7, the robot trajectories used were categorized as small square, medium square, small triangle, medium triangle, small circle and medium circle. For each type of trajectory geometry, five repeated trials were conducted, totalling thirty trials.

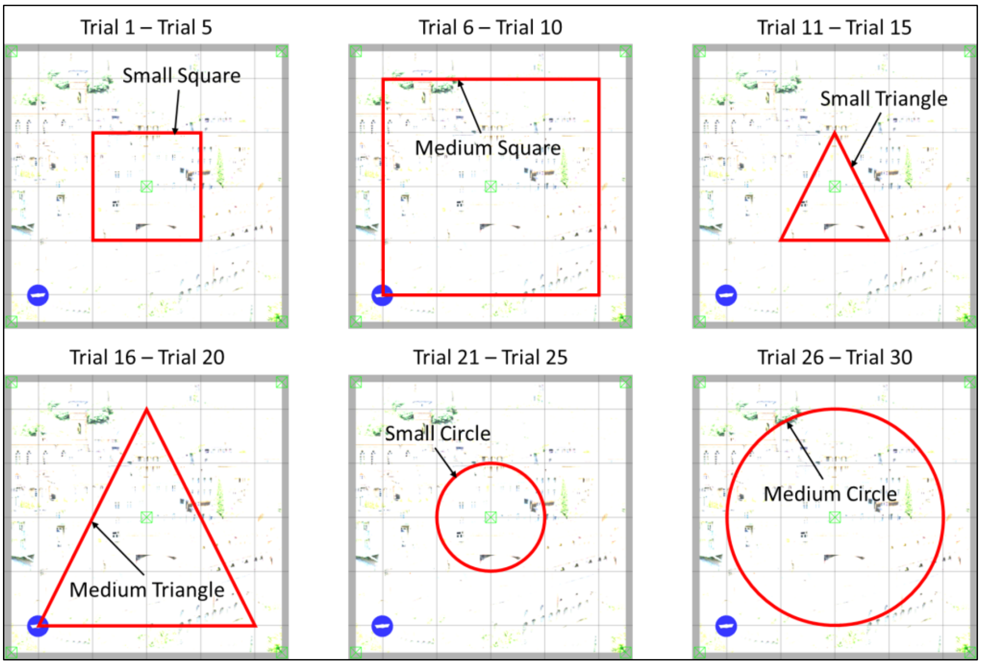


Figure 5-7. Robot trajectories of six different geometry types used in trials conducted in the $5m \times 5m \times 2m$ cuboid scene.

The goal was to compute and compare the map error before and after applying the two different error reduction techniques (classic ICP vs. CPR-ICP). Additionally, the effectiveness of these techniques was analysed in relation to scene size and trajectory geometry. Firstly, the original map error was computed. Since the estimated map was in the form of point cloud and the ground truth map was in the form of mesh and already available, the original map error between them was computed with Eq. (4.3) using the proposed SLAM benchmarking method shown in Section 4.2 . Then, the error reduction transformations based on ICP and CPR-ICP were derived from the estimated trajectory and ground truth trajectory, respectively. These transformations were applied to the estimated map to obtain a revised version. Finally, the map errors after ICP and CPR-ICP reductions were computed. For clarity and brevity, only a subset of the results is presented in Table 5-1, while the remaining data has been omitted. In the table, the following notations are used:

- RMSE_map – Original map error.
- RMSE_map_ar_ICP – Map error after ICP-based reduction.
- RMSE_map_ar_CPR_ICP – Map error after CPR-ICP-based reduction.

All error values are presented in millimetres. The results provide quantitative insights into the effectiveness of each error reduction technique and their correlation with scene size and trajectory geometry.

Table 5-1. RMSE of map before and after ICP and CPR-ICP error reductions for trials in the $5m \times 5m \times 2m$ cuboid scene

Trial No.	Trajectory shape	RMSE_map/mm	RMSE_map_ar_ICP/mm	RMSE_map_ar_CPR_ICP/mm
1	small square	15.96	14.15	14.01
⋮				
6	medium square	121.85	82.71	66.90
⋮				
11	small triangle	14.47	13.48	13.35
⋮				
16	medium triangle	45.34	26.62	26.68
⋮				
21	small circle	17.93	16.80	16.81
⋮				
26	medium circle	43.58	40.40	40.02
⋮				
30	medium circle	37.96	32.15	31.41

Using the error data from Table 5-1, the map error reduction rates for ICP and CPR-ICP were first calculated for each trial using Eq. (5.1) and Eq. (5.2). Subsequently, the median map error reduction rates for both methods across different trajectory types were computed and summarized in Table 5-2. The median was chosen over the mean as it is less influenced by outliers or extreme values, making it a more robust measure of overall error reduction effectiveness.

$$Map\_error\_reduction\_rate\_ICP = \frac{RMSE\_map - RMSE\_map\_ar\_ICP}{RMSE\_map} \tag{5.1}$$

$$Map\_error\_reduction\_rate\_CPR\_ICP = \frac{RMSE\_map - RMSE\_map\_ar\_CPR\_ICP}{RMSE\_map} \tag{5.2}$$

Table 5-2. Median map error reduction rates for ICP and CPR-ICP across different trajectory types in the $5m \times 5m \times 2m$ cuboid scene

Trajectory shape	Median map error reduction rate	
	ICP	CPR-ICP
small square	11.35%	12.20%
medium square	30.53%	30.58%
small triangle	9.34%	10.27%
medium triangle	40.68%	41.16%
small circle	5.78%	5.78%
medium circle	15.30%	17.25%

In Table 5-2, all median map error reduction rates for both CPR-ICP and ICP are positive, validating the efficacy of the proposed concept for improving mapping accuracy. The full map error data reveals that the proposed CPR-ICP method shows a higher map error reduction rate in fifteen trials, an equal rate in one trial, and a lower rate in fourteen trials compared to the classic ICP method. These results demonstrate the overall superiority of CPR-ICP in improving mapping accuracy over the classic ICP method. To further compare the performances of these two methods and explore their correlation with scene sizes and trajectory geometries, the median map error reduction rates from Table 5-2 are visualized as a histogram chart shown in Figure 5-8. As shown in the histograms, except for the scenario of small circle where ICP and CPR-ICP achieve the same median map error reduction rate, CPR-ICP achieves a higher rate for the other five trajectory geometries. This further demonstrates the superiority of the proposed CPR-ICP method over the classic ICP method. Additionally, for both methods, the median map error reduction rate increases with increasing trajectory sizes, regardless of the shape of trajectories. This indicates a positive correlation between the map error reduction rate and trajectory size for a given scene.

When do the methods tie in the $5m \times 5m \times 2m$ scene:

Table 5-2 and Figure 5-8 show one category (small circle) where ICP and CPR-ICP are equal. This configuration combines a short, closed loop with high point overlap and nearly isotropic local geometry, and the pre-reduction misalignment is already small. Under such favourable conditions, the additional cross-pose regularization in CPR-ICP brings little extra benefit over the geometric alignment that ICP already achieves, hence the tie.

Error-bar convention for the histograms (applicable to Figure 5-8, Figure 5-10, Figure 5-12, and Figure 5-14):

Bars report the median map-error reduction across the five repeated trials for each trajectory category. Error bars denote the interquartile range (IQR) computed from the same five values, with asymmetric whiskers equal to (median − Q1) and (Q3 − median), where Q1 and Q3 are the 25th and 75th percentiles respectively. Using IQR aligns with the median and reduces the influence of outliers. All reduction rates follow Eq. (5.1) and Eq. (5.2) and are expressed in percentage points.

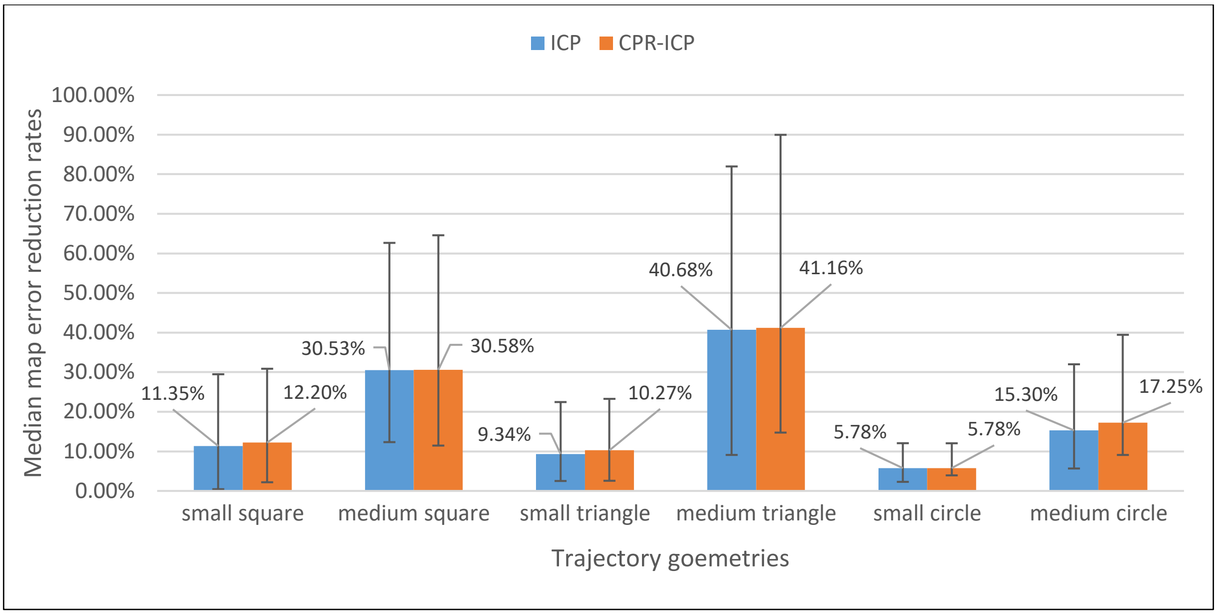


Figure 5-8. Median map error reduction rates of ICP and CPR-ICP operations for each type of trajectory geometry used in $5m \times 5m \times 2m$ cuboid scene.

5.2.2 Experimental results and analysis for the $10m \times 10m \times 2m$ cuboid scene

Following the experimental procedure used for the $5m \times 5m \times 2m$ cuboid scene, a similar set of experiments was conducted in the $10m \times 10m \times 2m$ cuboid scene. As shown in Figure 5-9, the robot trajectories used in this scene were categorized as: medium square, large square, medium triangle, large triangle, medium circle, and large circle. For each type of trajectory geometry, five repeated trials were conducted, totalling thirty trials.

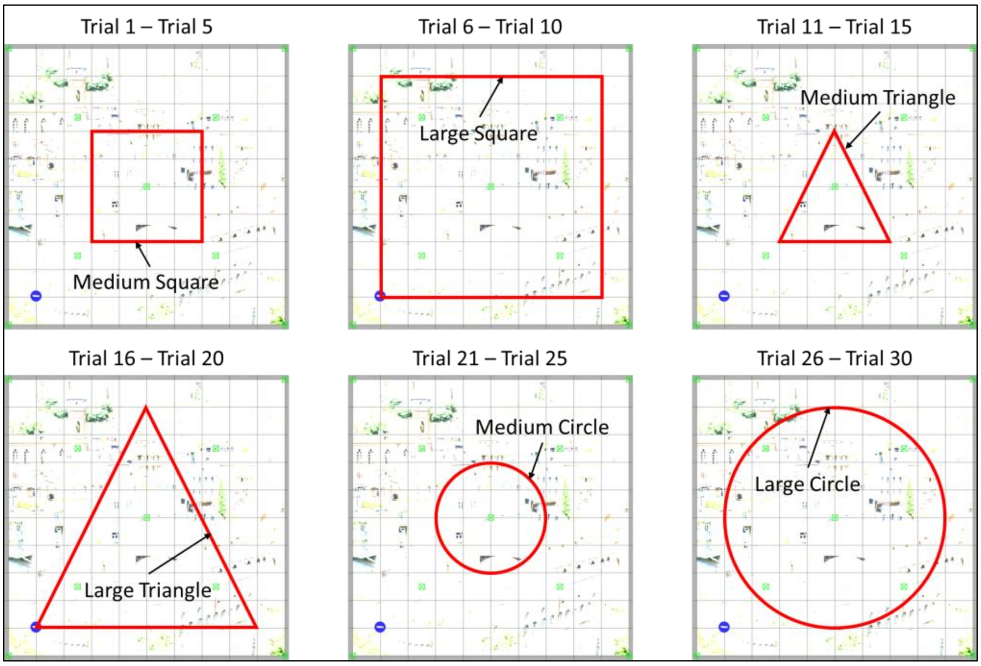


Figure 5-9. Robot trajectories of six different geometry types used in trials conducted in the $10m \times 10m \times 2m$ cuboid scene.

The partial map error data are presented in Table 5-3, while the median map error reduction rates for ICP and CPR-ICP are presented in Table 5-4. The symbols and notation used in Table 5-3 and Table 5-4 are consistent with those used in Table 5-1 and Table 5-2 for the $5m \times 5m \times 2m$ cuboid scene, maintaining the same physical meaning for ease of comparison.

Table 5-3. RMSE of map before and after ICP and CPR-ICP error reductions for trials in the $10m \times 10m \times 2m$ cuboid scene

Trial No.	Trajectory shape	RMSE_map/mm	RMSE_map_ar_ICP/mm	RMSE_map_ar_CPR_ICP/mm
1	medium square	38.33	37.77	37.77
⋮				
6	large square	98.22	74.53	74.53
⋮				
11	medium triangle	39.36	38.64	38.64
⋮				
16	large triangle	50.68	34.27	34.28
⋮				
21	medium circle	33.00	32.82	32.80
⋮				
26	large circle	37.91	34.34	34.34
⋮				
30	large circle	43.07	32.33	32.33

Table 5-4. Median map error reduction rates for ICP and CPR-ICP across different trajectory types in the $10m \times 10m \times 2m$ cuboid scene

Trajectory shape	Median map error reduction rate	
	ICP	CPR-ICP
medium square	1.89%	1.44%
large square	24.12%	24.12%
medium triangle	1.54%	1.82%
large triangle	24.08%	24.08%
medium circle	1.65%	1.65%
large circle	24.92%	24.92%

The median map error reduction rates for both CPR-ICP and ICP, as presented in Table 5-4, remain consistently positive, reinforcing the effectiveness of the proposed concept in enhancing mapping accuracy. The full map error data shows that CPR-ICP achieves a higher map error reduction rate in sixteen trials, while ICP performs better in eleven trials, with both methods producing identical results in three trials. Despite some variations, CPR-ICP demonstrates an overall superior performance in reducing map error compared to the classic ICP method. To further examine the differences between

ICP and CPR-ICP and analyse their correlation with scene sizes and trajectory geometries, the median map error reduction rates from Table 5-4 are visualized in a histogram chart in Figure 5-10, similar to the visualization in Figure 5-10 from the previous section. The histogram reveals that for one trajectory geometry, each method outperforms the other, while for the remaining four geometries, both methods yield identical median reduction rates. Although ICP performs better in some cases, CPR-ICP achieves a higher reduction rate in a greater number of trials, reaffirming its advantage in improving mapping accuracy. Another notable trend observed in the histogram is that for both methods, the median map error reduction rate increases with trajectory size, regardless of the shape of the trajectory. This suggests a positive correlation between trajectory size and map error reduction, indicating that larger trajectories tend to yield greater improvements in mapping accuracy for a given scene.

Why do several categories tie and one favour ICP in the $\mathbf{10}\boldsymbol{m} \times \mathbf{10}\boldsymbol{m} \times \mathbf{2}\boldsymbol{m}$ scene:

In Figure 5-10 and Table 5-4, large square/triangle/circle yield results that are indistinguishable between the two methods, while the medium-square category shows a slight ICP advantage. These cases correspond to near-rigid residual misalignment with high, well-distributed overlap and effective loop closure. Under such conditions ICP converges to the same solution as CPR-ICP, and in the medium-square category the purely geometric fit is marginally better than the trajectory-regularized one.

Note on error bars:

The same histogram convention as in Section 5.2.1 applies here: bar height is the median across five trials and error bars indicate the IQR (Q1–Q3) around that median.

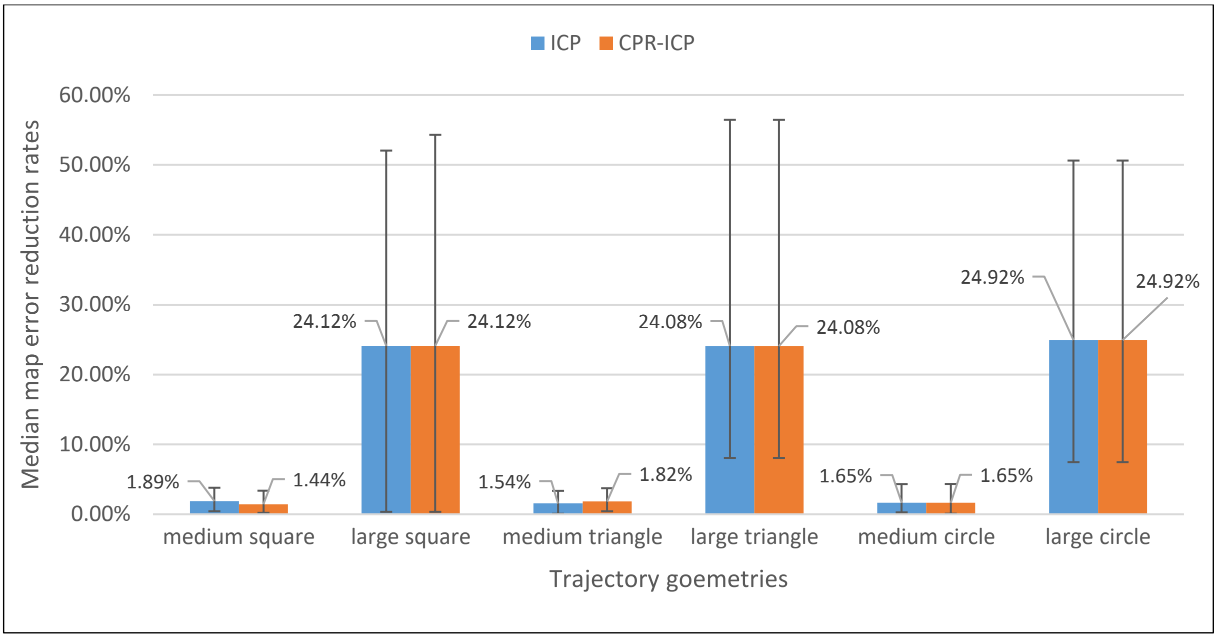


Figure 5-10. Median map error reduction rates of ICP and CPR-ICP operations for each type of trajectory geometry used in $10m \times 10m \times 2m$ cuboid scene.

5.2.3 Experimental results and analysis for the $15m \times 15m \times 2m$ cuboid scene

Following the methodology used in the $5m \times 5m \times 2m$ and $10m \times 10m \times 2m$ cuboid scenes, experiments in the $15m \times 15m \times 2m$ scene were conducted using a set of predefined robot trajectories. As illustrated in Figure 5-11, the trajectories were categorized into medium square, large square, XL square, medium triangle, large triangle, XL triangle, medium circle, large circle, and XL circle. Each trajectory type was tested across five repeated trials, leading to a total of forty-five trials.

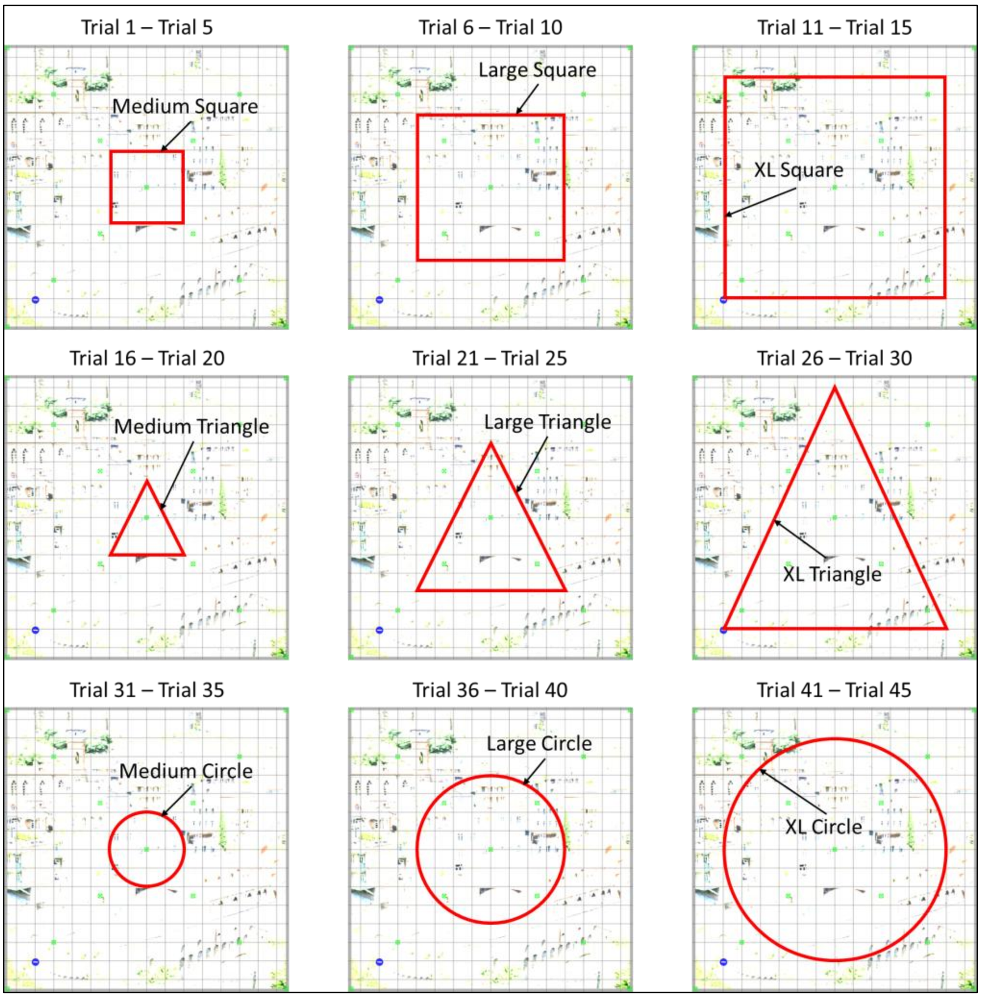


Figure 5-11. Robot trajectories of nine different geometry types used in trials conducted in the $15m \times 15m \times 2m$ cuboid scene.

The partial map error data collected from these trials are presented in Table 5-5, while the corresponding median map error reduction rates for ICP and CPR-ICP are provided in Table 5-6. The notation and symbols used in these tables remain consistent with those used for the $5m \times 5m \times 2m$ and $10m \times 10m \times 2m$ cuboid scenes, maintaining the same physical meaning to ensure comparability across different scene sizes.

Table 5-5. RMSE of map before and after ICP and CPR-ICP error reductions for trials in the $15m \times 15m \times 2m$ cuboid scene

Trial No.	Trajectory shape	RMSE_map/mm	RMSE_map_ar_ICP/mm	RMSE_map_ar_CPR_ICP/mm
1	medium square	65.16	64.77	64.73
⋮				
6	large square	37.78	31.86	31.17
⋮				
11	XL square	489.49	223.39	201.00
⋮				
16	medium triangle	102.99	102.82	102.82

⋮				
21	large triangle	45.48	43.61	43.61
⋮				
26	XL triangle	119.43	69.33	46.54
⋮				
31	medium circle	90.50	89.99	90.06
⋮				
36	large circle	36.80	34.58	35.16
⋮				
41	XL circle	43.05	27.50	27.53
⋮				
45	XL circle	85.30	61.45	61.45

Table 5-6. Median map error reduction rates for ICP and CPR-ICP across different trajectory types in the $15m \times 15m \times 2m$ cuboid scene

Trajectory shape	Median map error reduction rate	
	ICP	CPR-ICP
medium square	0.59%	0.65%
large square	15.68%	17.50%
XL square	30.26%	30.39%
medium triangle	0.69%	0.71%
large triangle	4.10%	4.10%
XL triangle	13.58%	13.58%
medium circle	1.13%	1.61%
large circle	9.72%	11.19%
XL circle	27.96%	27.96%

The map error reduction rates presented in Table 5-6 remain consistently positive, reaffirming the effectiveness of the proposed approach in enhancing mapping accuracy. A comparative assessment from the full map error data reveals that CPR-ICP outperforms ICP in sixteen trials, achieves equivalent results in three trials, and falls short in eleven trials, demonstrating its overall advantage over the classic ICP method. To further analyse the differences between the two methods and examine their relationship with scene sizes and trajectory geometries, the median map error reduction rates from Table 5-6 are visualized in Figure 5-12 using a histogram chart, consistent with the approach taken in earlier sections. The histogram illustrates that CPR-ICP achieves superior error reduction in six of the nine trajectory geometries, while in the remaining three cases, both methods yield identical results. These findings further support the superior performance of CPR-ICP in reducing map errors compared to ICP. Moreover, the data reveal a clear trend for both methods, where the median map error reduction rate increases as trajectory size increases, irrespective of trajectory shape. This suggests a direct correlation between trajectory size and mapping accuracy improvements, indicating that larger trajectories generally lead to more substantial reductions in map errors within a given scene.

Equal-performance regimes in the $15m \times 15m \times 2m$ scene:

As summarized in Table 5-6 and Figure 5-12, most categories favour CPR-ICP, but three (large triangle, XL triangle, XL circle) are effectively equal. These categories share two properties: a balanced, feature-rich geometry that yields robust correspondences, and a small residual misalignment after loop closure. Under these conditions, both methods recover essentially the same alignment within the variability of repeated trials.

Note on error bars:

Same as in Section 5.2.1 , error bars denote the IQR (25th–75th percentile) across five trials.

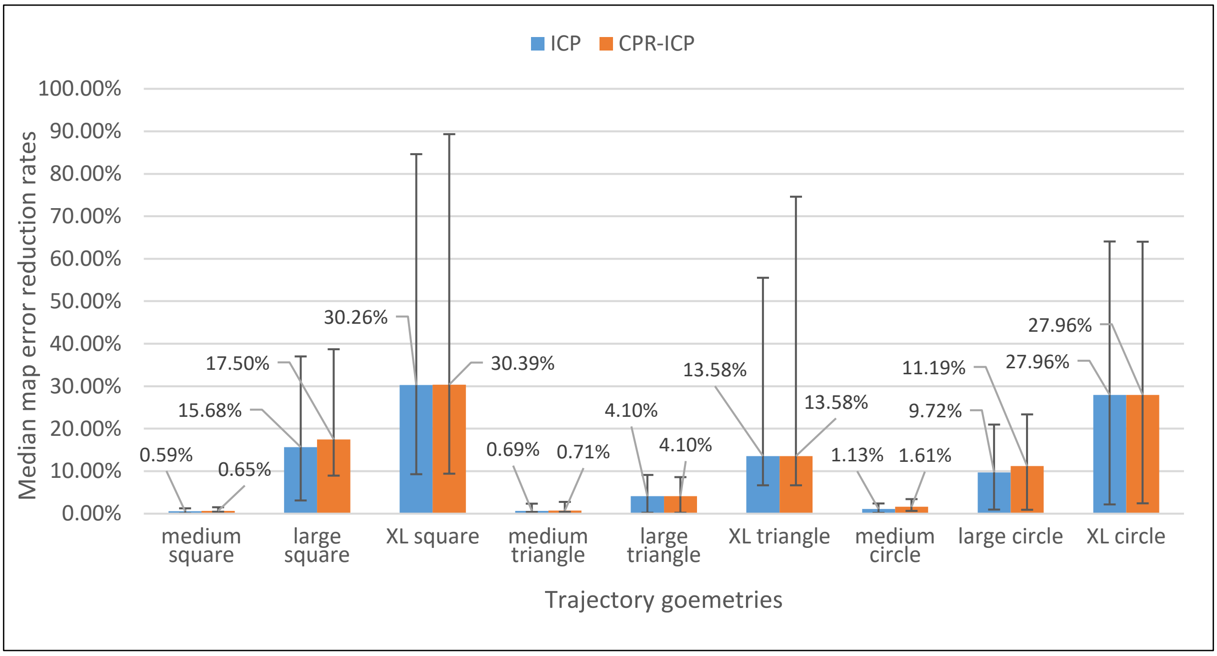


Figure 5-12. Median map error reduction rates of ICP and CPR-ICP operations for each type of trajectory geometry used in $15m \times 15m \times 2m$ cuboid scene.

5.2.4 Experimental results and analysis for the $20m \times 20m \times 2m$ cuboid scene

Building on the methodology used in the previous three sections, the experiments conducted in the $20m \times 20m \times 2m$ cuboid scene followed a similar approach. As illustrated in Figure 5-13, the robot trajectories were categorized into medium, large, XL, and XXL variations of square, triangle, and circle trajectories, resulting in a total of twelve trajectory types. For each trajectory geometry, five repeated trials were performed, leading to a total of sixty trials.

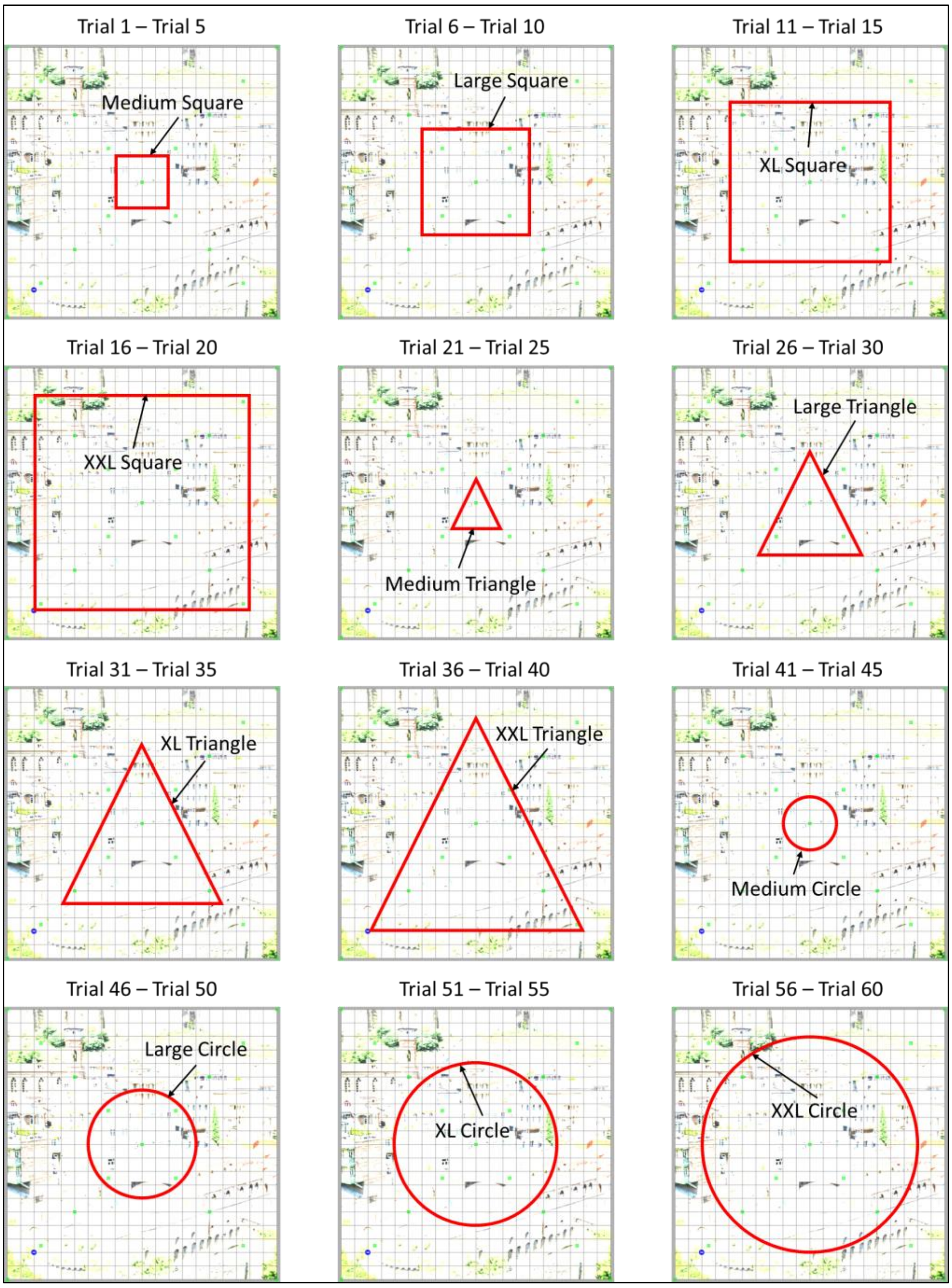


Figure 5-13. Robot trajectories of twelve different geometry types used in trials conducted in the $20m \times 20m \times 2m$ cuboid scene.

The partial map error data collected from these trials are presented in Table 5-7, while the corresponding median map error reduction rates for ICP and CPR-ICP are detailed in Table 5-8. The notation and symbols used in these tables are consistent with those from the $5m \times 5m \times 2m$,

$10m \times 10m \times 2m$, and $15m \times 15m \times 2m$ cuboid scene experiments, ensuring continuity in data interpretation and facilitating direct comparisons across different scene sizes.

Table 5-7. RMSE of map before and after ICP and CPR-ICP error reductions for trials in the $20m \times 20m \times 2m$ cuboid scene

Trial No.	Trajectory shape	RMSE_map/mm	RMSE_map_ar_ICP/mm	RMSE_map_ar_CPR_ICP/mm
1	medium square	124.34	124.06	124.05
⋮				
6	large square	256.70	145.59	148.22
⋮				
11	XL square	43.40	41.56	41.15
⋮				
16	XXL square	128.16	66.41	66.11
⋮				
21	medium triangle	142.29	141.99	141.99
⋮				
26	large triangle	82.25	81.57	81.48
⋮				
31	XL triangle	62.65	60.92	60.89
⋮				
36	XXL triangle	408.83	46.80	47.62
⋮				
41	medium circle	108.06	107.31	107.31
⋮				
46	large circle	57.37	57.18	57.18
⋮				
51	XL circle	52.22	45.48	43.85
⋮				
56	XXL circle	110.27	75.43	76.39
⋮				
60	XXL circle	58.71	43.34	43.34

Table 5-8. Median map error reduction rates for ICP and CPR-ICP across different trajectory types in the $20m \times 20m \times 2m$ cuboid scene

Trajectory shape	Median map error reduction rate	
	ICP	CPR-ICP
medium square	0.23%	0.23%
large square	1.20%	0.94%
XL square	20.99%	20.60%
XXL square	24.65%	24.65%
medium triangle	0.19%	0.21%
large triangle	0.82%	0.82%
XL triangle	2.77%	2.82%
XXL triangle	51.56%	51.56%

medium circle	0.99%	1.04%
large circle	4.55%	4.55%
XL circle	14.02%	16.01%
XXL circle	25.64%	25.28%

The median map error reduction rates for both CPR-ICP and ICP, as detailed in Table 5-8, remain consistently positive, reaffirming the effectiveness of the proposed approach in enhancing mapping accuracy. Full map error data indicates that CPR-ICP outperforms ICP in twenty-eight trials, achieves identical reduction rates in six trials, and underperforms in twenty-six trials, demonstrating its overall advantage over the classic ICP method. To further analyse the differences between the two methods and explore their relationship with scene sizes and trajectory geometries, the median map error reduction rates from Table 5-8 are visualized in Figure 5-14 using a histogram chart, following the approach used in previous sections. The histogram reveals that CPR-ICP achieves higher median map error reduction in six of the twelve trajectory geometries, while ICP outperforms CPR-ICP in three cases, and both methods yield identical results in the remaining three cases. These findings reinforce the superior performance of CPR-ICP in reducing map errors. A noticeable trend observed across both methods is that the median map error reduction rate increases with trajectory size, regardless of trajectory shape. This pattern suggests a positive correlation between trajectory size and mapping accuracy improvements, indicating that larger trajectories lead to more substantial reductions in map errors within a given scene.

Why can ICP be marginally higher in the $20m \times 20m \times 2m$ scene:

Table 5-8 and Figure 5-14 include several ties (e.g., medium square, XXL square, large triangle, XXL triangle, large circle) and a few categories where ICP is marginally higher (e.g., large square, XL square, XXL circle). These regimes feature high overlap and largely isotropic structure, with residual misalignment that is already small and close to rigid. In such cases ICP fully exploits local geometric cues, while CPR-ICP mildly regularizes the solution using cross-pose relations; when those relations carry small sampling or synchronization bias, the regularization can slightly reduce the marginal gain, leading to equality or a very small ICP advantage.

Note on error bars:

Same as in Section 5.2.1 , error bars denote the IQR (25th–75th percentile) across five trials.

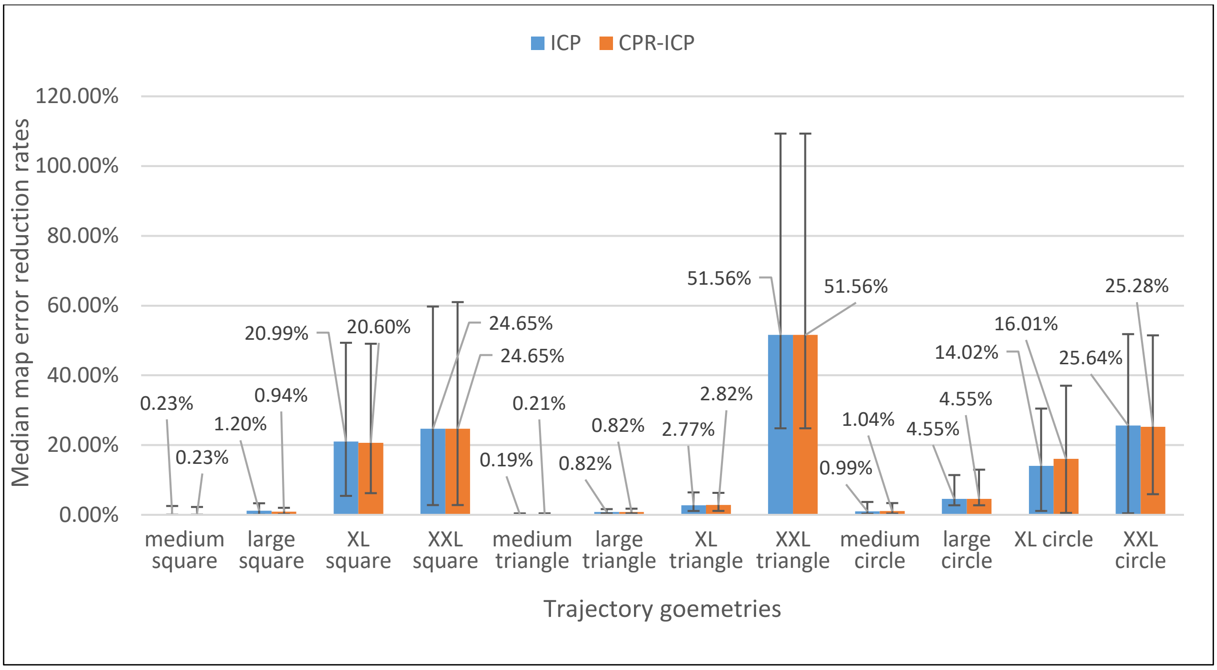


Figure 5-14. Median map error reduction rates of ICP and CPR-ICP operations for each type of trajectory geometry used in $20m \times 20m \times 2m$ cuboid scene.

5.2.5 Experimental data aggregation and analysis

To provide a comprehensive overview of the map error reduction rates discussed in the previous four sections, and to identify patterns, trends, and insights regarding the performance of ICP and CPR-ICP in improving mapping accuracy, the median map error reduction rates from Table 5-2, Table 5-4, Table 5-6, and Table 5-8 are consolidated in Table 5-9 and Table 5-10. These tables present the median map error reduction rates for both the classic ICP method and the proposed CPR-ICP method across different trajectory geometries and scene sizes. To visualize these trends more effectively, the data from Table 5-9 and Table 5-10 are represented as line charts in Figure 5-15 to Figure 5-20. Figure 5-15, Figure 5-16, and Figure 5-17 illustrate the median map error reduction rates of ICP for medium-sized, large-sized, and XL-sized trajectories, respectively, while Figure 5-18, Figure 5-19, and Figure 5-20 present the corresponding rates for CPR-ICP. From these line charts, a consistent trend emerges: the median map error reduction rates for both ICP and CPR-ICP decrease as scene size increases, aligning with the trends previously observed in the histograms from Figure 5-8, Figure 5-10, Figure 5-12, and Figure 5-14. However, despite this decline, the charts also confirm that the median map error reduction rate increases with larger trajectory sizes, reinforcing the observation that wider spacing between trajectories and scene walls enhances error reduction effectiveness.

Synthesis across scenes:

The ties and occasional small ICP advantages reported in Table 5-4, Table 5-6, Table 5-8 and Figure 5-10, Figure 5-12, Figure 5-14 occur under favourable data conditions—low initial misalignment, strong and well-distributed overlap, and effective loop closure—where the geometric data term alone already identifies the correct rigid transform. Conversely, CPR-ICP provides larger reductions as the problem becomes harder (coarser initial alignment or less uniform overlap), because the cross-pose relations supply complementary constraints that stabilize correspondence search and suppress residual drift. This explains the few equal or ICP-slightly-better outcomes without altering the overall conclusion that CPR-ICP is the more effective choice in typical benchmarking settings.

Table 5-9. The median map error reduction rate of ICP for each type of trajectory geometry and scene size

Scene sizes / Trajectory geometries	5m×5m	10m×10m	15m×15m	20m× 20m
Small square	11.350%	-	-	-
Small triangle	9.343%	-	-	-
Small circle	5.775%	-	-	-
Medium square	30.532%	1.892%	0.591%	0.226%
Medium triangle	40.684%	1.540%	0.686%	0.193%
Medium circle	15.302%	1.649%	1.130%	0.989%
Large square	-	24.116%	15.675%	1.200%
Large triangle	-	24.078%	4.099%	0.816%
Large circle	-	24.922%	9.722%	4.546%
XL square	-	-	30.259%	20.989%
XL triangle	-	-	13.576%	2.770%
XL circle	-	-	27.956%	14.016%
XXL square	-	-	-	24.652%
XXL triangle	-	-	-	51.559%
XXL circle	-	-	-	25.636%

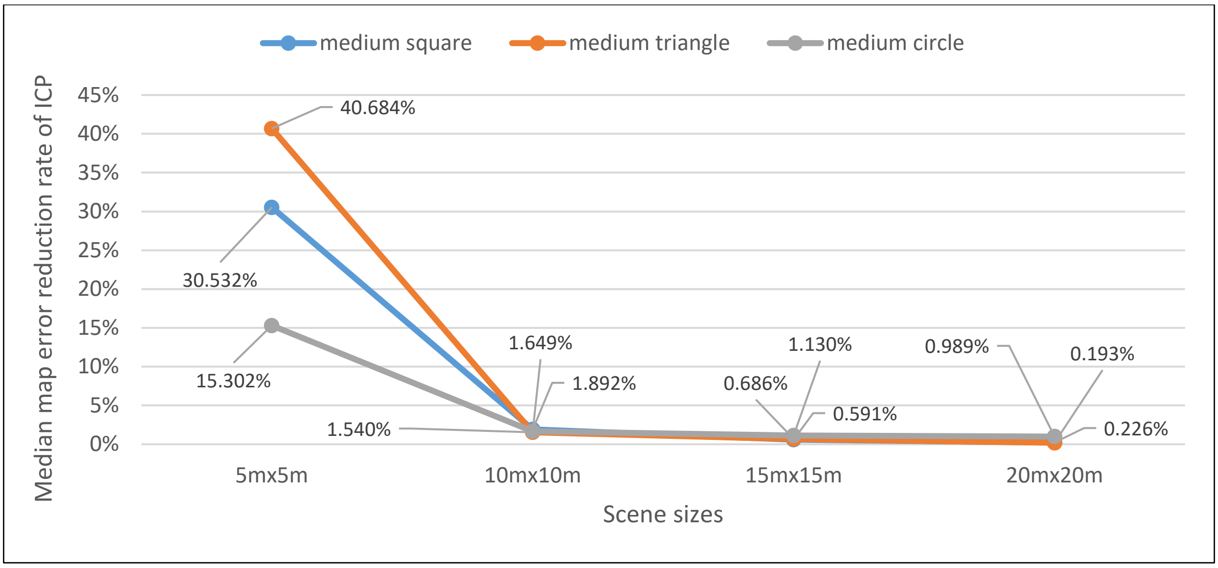


Figure 5-15. Median map error reduction rates of ICP for medium-sized trajectories (square, triangle, circle) across all four different scene sizes ($5m \times 5m$, $10m \times 10m$, $15m \times 15m$, $20m \times 20m$).

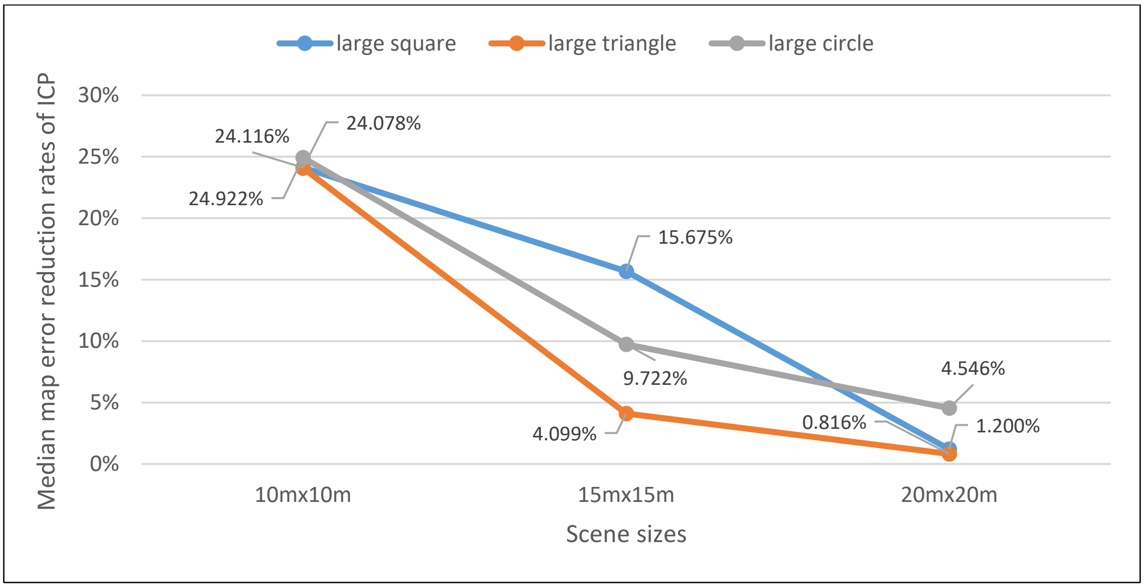


Figure 5-16. Median map error reduction rates of ICP for large-sized trajectories (square, triangle, circle) across three different scene sizes ($10m \times 10m$, $15m \times 15m$, $20m \times 20m$).

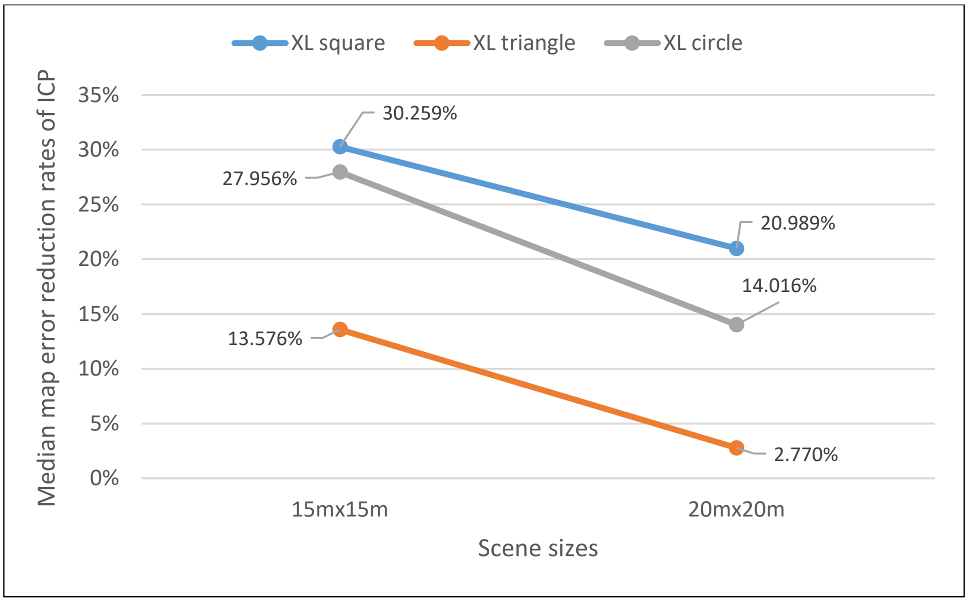


Figure 5-17. Median map error reduction rates of ICP for XL-sized trajectories (square, triangle, circle) across two different scene sizes ($15m \times 15m$, $20m \times 20m$).

Table 5-10. The median map error reduction rate of CPR-ICP for each type of trajectory geometry and scene size

Scene sizes / Trajectory geometries	5m×5m	10m×10m	15m×15m	20m× 20m
Small square	12.198%	-	-	-
Small triangle	10.272%	-	-	-
Small circle	5.775%	-	-	-
Medium square	30.576%	1.444%	0.655%	0.230%
Medium triangle	41.161%	1.822%	0.709%	0.207%
Medium circle	17.247%	1.649%	1.610%	1.037%
Large square	-	24.116%	17.500%	0.939%
Large triangle	-	24.078%	4.099%	0.821%
Large circle	-	24.922%	11.191%	4.546%
XL square	-	-	30.385%	20.600%
XL triangle	-	-	13.576%	2.821%
XL circle	-	-	27.956%	16.014%
XXL square	-	-	-	24.652%
XXL triangle	-	-	-	51.559%
XXL circle	-	-	-	25.280%

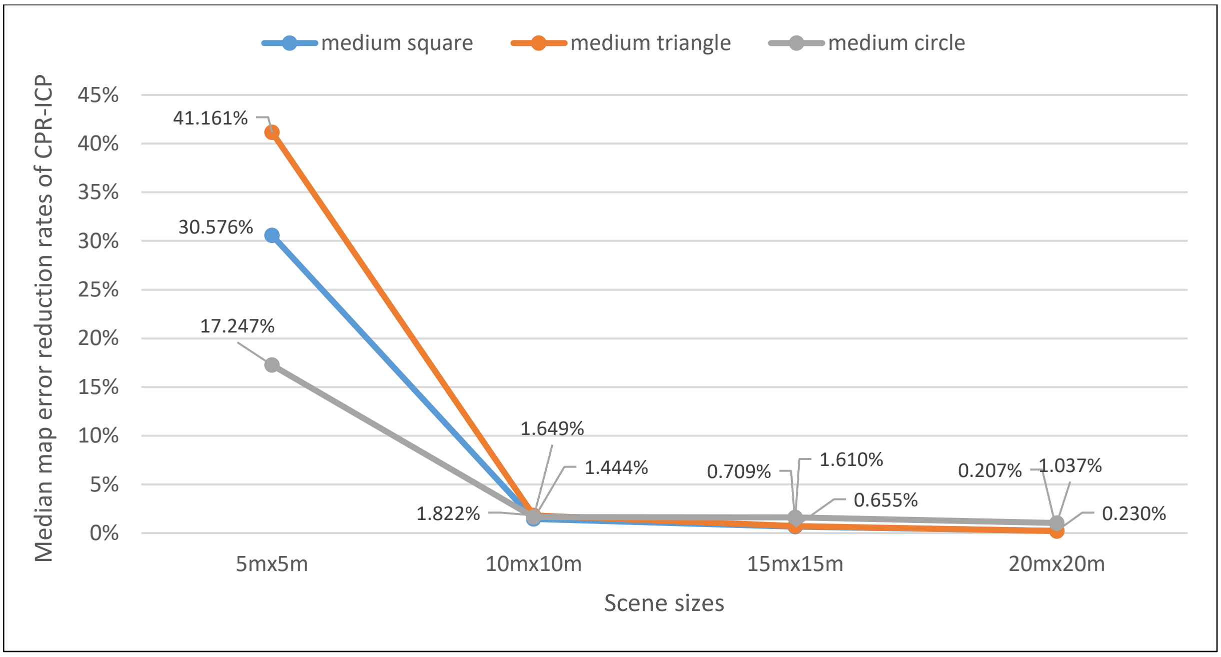


Figure 5-18. Median map error reduction rates of CPR-ICP for medium-sized trajectories (square, triangle, circle) across all four different scene sizes ($5m \times 5m$, $10m \times 10m$, $15m \times 15m$, $20m \times 20m$).

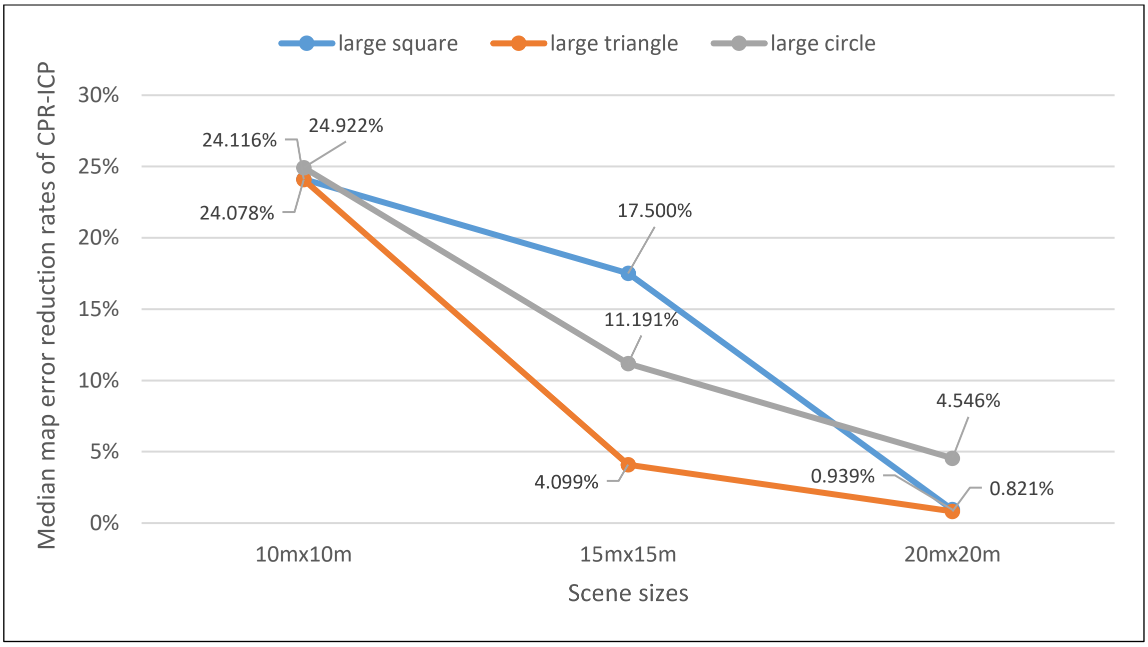


Figure 5-19. Median map error reduction rates of CPR-ICP for large-sized trajectories (square, triangle, circle) across three different scene sizes ($10m \times 10m$, $15m \times 15m$, $20m \times 20m$).

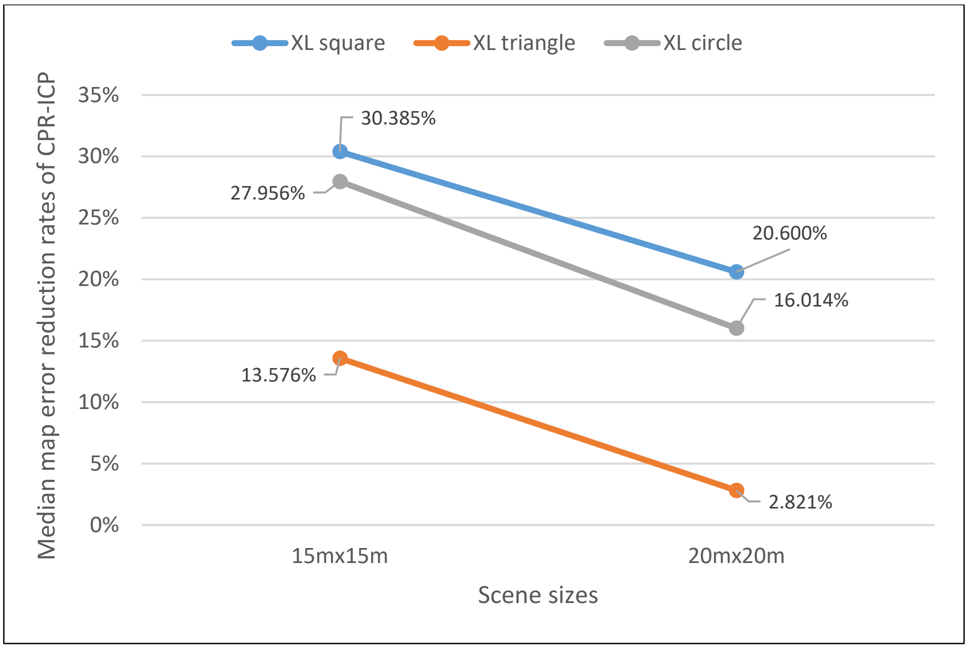


Figure 5-20. Median map error reduction rates of CPR-ICP for XL-sized trajectories (square, triangle, circle) across two different scene sizes ($15m \times 15m$, $20m \times 20m$).

5.3 Conclusions

The simulation-based experiments demonstrate that the proposed approach effectively enhances mapping accuracy. Across scenes and trajectory geometries, CPR-ICP generally achieves larger reductions in map error than classic ICP. A small number of ties and marginal ICP advantages observed in Section 5.2 are consistent with favourable alignment and overlap conditions—namely, small pre-reduction misalignment, high and well-distributed map/mesh overlap, and effective loop closure—under which both methods converge to similar rigid transforms and the geometric data term alone suffices. Under more demanding conditions (coarser initial alignment or less uniform overlap), CPR-ICP provides clearer gains. Taken together, these findings indicate that CPR-ICP is preferable for typical benchmarking scenarios. To further validate the proposed concept and methodologies under real-world conditions, the next chapter presents their evaluation through real-world experiments.

Chapter 6 Experimental Validation in Real-world Environments

Building on the simulation-based validation presented in the previous chapter, which demonstrated the effectiveness of the proposed concept and the superiority of CPR-ICP over the classic ICP method in reducing map errors, this chapter further reinforces these findings through real-world experiments. The chapter begins with Section 6.1 , which outlines the general experimental setup and configuration. The subsequent three sections (Section 6.2 , Section 6.3 , and Section 6.4) provide detailed descriptions of the specific setups for three different scene types, along with Section 6.5 presenting their corresponding experimental results and analysis. Finally, Section 6.6 presents a concluding discussion on the findings from the real-world experiments.

6.1 General experimental setup and configuration

The general experimental setup for real-world environments is illustrated in Figure 6-1. As shown in Figure 6-1 (a), a VICON motion capture system is used to obtain the ground truth of the SLAM camera trajectory and map. As discussed in Section 4.2 , the coordinate system of the motion capture system serves as the global coordinate frame $\boldsymbol{G}$. The system tracks object motion by detecting markers attached to the object and is positioned at an elevated location to ensure uninterrupted visibility of all markers.

For the ground truth map, the markers attached to scene elements allow it to be obtained directly within the global frame $\boldsymbol{G}$ without requiring pose transformations. However, as shown in Figure 6-1 (b), the ground truth trajectory of the SLAM camera cannot be acquired directly, since the origin of the SLAM camera frame $\boldsymbol{Sc}$ is located on the lens surface of the camera. As outlined in Section 4.2 , determining the ground truth positions of the SLAM camera origin requires the use of additional auxiliary reference frames and Euclidean transformations. To achieve this, a 3D-printed attachment is mounted on the back of the SLAM camera, holding three tracking markers. The reference frame of these markers is denoted as $\boldsymbol{Sca}$ (SLAM Camera Attachment Frame), with its origin at the centroid of the triangle formed by the three markers. Its x and y axes align with the attachment's length and width, respectively. The pose transformation from $\boldsymbol{Sca}$ to the global frame $\boldsymbol{G}$, denoted as $\boldsymbol{T}_{\boldsymbol{Sca}}^{\boldsymbol{G}}$, is continuously tracked by the motion capture system. Since the relative position of the SLAM camera origin within $\boldsymbol{Sca}$ is known, the ground truth trajectory of the SLAM camera can be determined within the global frame $\boldsymbol{G}$ using this known transformation.

The tracking markers on the SLAM camera attachment are placed at three corners instead of four to form an asymmetric pattern. A symmetric pattern, such as a rectangle with four markers, can cause issues with the motion capture system's pattern recognition, leading to frame flipping errors during operation. In contrast, the asymmetric triangular pattern improves the stability of the SLAM camera attachment frame $\boldsymbol{Sca}$, reducing the likelihood of tracking errors and ensuring robust and reliable motion capture. This asymmetric three-marker rigid body was kept throughout to guarantee a fixed $\boldsymbol{Sca}$ definition and avoid additional occlusions, mass/centroid changes, and repeated $\boldsymbol{T}_{\boldsymbol{Sc}}^{\boldsymbol{Sca}}$ calibrations; additional or alternate patterns would not yield meaningful pose-accuracy gains under the Vicon setup.

As for the estimated trajectory and map, as discussed in Section 4.2 , they are initially recorded within the SLAM camera frame $\boldsymbol{Sc}$ of the first captured image. Therefore, they can be transformed into the global frame $\boldsymbol{G}$ using the known position of $\boldsymbol{Sc}$ within $\boldsymbol{Sca}$ and the first captured transformation $\boldsymbol{T}_{\boldsymbol{Sca}}^{\boldsymbol{G}}$.

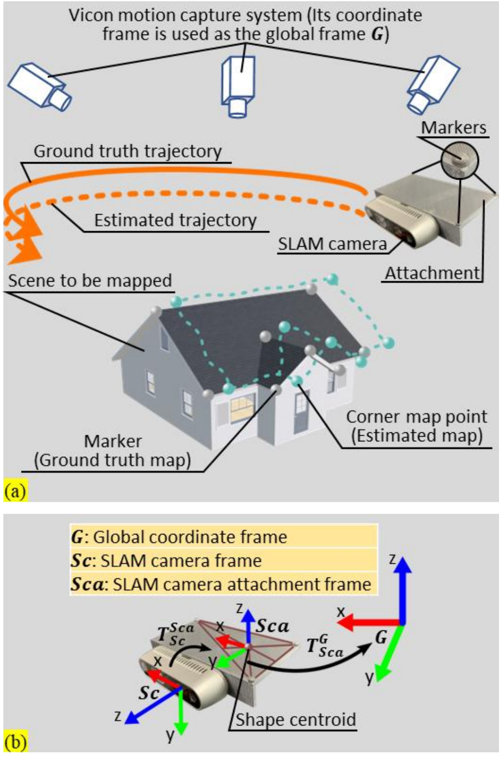

Vicon motion capture system (Its coordinate frame is used as the global frame $\boldsymbol{G}$)
Markers
Ground truth trajectory
Estimated trajectory
SLAM camera
Attachment
Scene to be mapped
Corner map point (Estimated map)
Marker (Ground truth map)
(a)
$\boldsymbol{G}$: Global coordinate frame
$\boldsymbol{Sc}$: SLAM camera frame
$\boldsymbol{Sca}$: SLAM camera attachment frame
T_{Sc}^{Sca}
Sca
T_{Sca}^{G}
G
Sc
x
y
z
Shape centroid
(b)

Figure 6-1. Illustration of the general experimental setup and configuration for validating the proposed methodologies in real-world environments. (a) Experimental setup. (b) Pose transformations between different frames.

The specific experimental setups, configurations, results, and analysis for three different scene types are presented in the following sections. These experiments aim to further evaluate and compare the trajectory and map errors before and after applying two distinct error reduction techniques—ICP and CPR-ICP. Additionally, they provide deeper insights into the performance of these techniques and their relationship with scene size, allowing for a more comprehensive assessment of their effectiveness in real-world environments.

6.2 Experiments in small-scale scene

6.2.1 Experimental setup and configuration

The proposed methodologies were initially tested in a small-scale scene to evaluate their performance in a controlled environment. The experimental setup and configuration are illustrated in Figure 6-2. As shown in Figure 6-2 (a), the hardware setup includes a Vicon motion capture system, a laptop, and an RGB-D camera. The Vicon motion capture system is used to acquire the ground truth trajectory and establish the global coordinate frame $\boldsymbol{G}$. The laptop (SLAM laptop), equipped with an Intel Core i9-10980HK CPU (2.40GHz, 8 cores, 16 logical processors) and 32GB of memory, runs the ORB-SLAM2 algorithm [286]. The RGB-D camera, an Intel RealSense D435i, is mounted on top of the SLAM laptop and serves as the SLAM camera for capturing depth and RGB images.

As shown in Figure 6-2 (b), the scene consists of three reference points, which correspond to the centroids of three rectangular regions on a $75 \times 75cm$ table surface. The table surface is divided into four equal rectangular regions using coloured tape, and twelve markers are positioned at the corners of three of these regions. This setup allows the centroid coordinates of the three selected regions (blue points in the figure) to be directly recorded within the global frame by the motion capture system, serving as the ground truth map.

During the experiment, the operator moves around the table while holding the laptop and SLAM camera, which continuously scans the table surface to generate its point cloud representation. Within the generated point cloud, the four corner points of each rectangular region with markers are manually selected, and their mean point is computed to form the estimated map. Since the estimated map is initially captured in the SLAM camera frame, it must be transformed into the global frame for further analysis. The processing of the ground truth and estimated trajectories follows the methodology detailed in the previous section and will not be reiterated here.

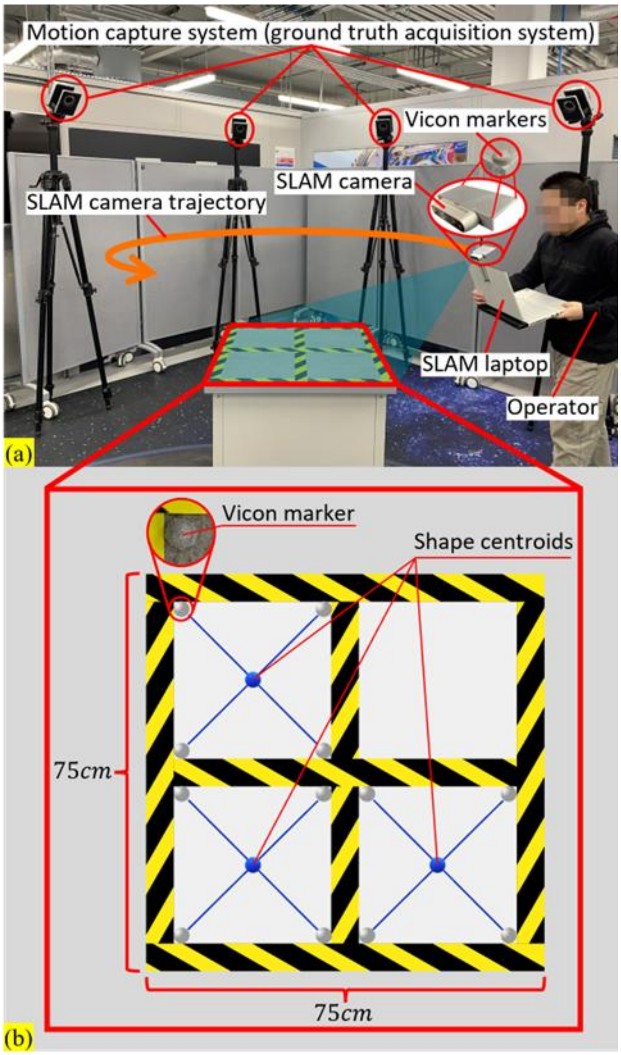


Figure 6-2. Experimental setup and configuration for the small-scale scene. (a) SLAM system and scanning process. (b) Layout of the map setup on the table surface.

6.2.2 Experimental results and analysis

A total of nine trials were conducted in the small-scale scene to improve the reliability and accuracy of results, minimize random errors, and ensure consistency and reproducibility. As illustrated in Figure 6-3, each trial generated an estimated trajectory, a ground truth trajectory, an estimated map, and a revised map after error reduction. The ground truth map was pre-captured using the motion capture system and was readily available for benchmarking. Since both the estimated and ground truth map were represented as point clouds, the original map errors were computed using Eq. (4.2), following the SLAM benchmark methodology outlined in Section 4.2 . To maintain consistency with the simulation-based experiments, the ground truth trajectory was first down-sampled to match the number of points in the estimated trajectory. Next, error reduction transformations were derived from the estimated and ground truth trajectories using both ICP and CPR-ICP. These transformations were applied to refine the estimated trajectory and map, and the revised trajectory and map errors were then computed after implementing ICP and CPR-ICP. All error measurements are summarized in Table 6-1, using the same notation as in Table 5-1 for consistency:

- RMSE_map – Original map error.
- RMSE_map_ar_ICP – Map error after ICP-based reduction.
- RMSE_map_ar_CPR_ICP – Map error after CPR-ICP-based reduction.

All values are reported in millimetres to ensure precise and accurate comparisons.

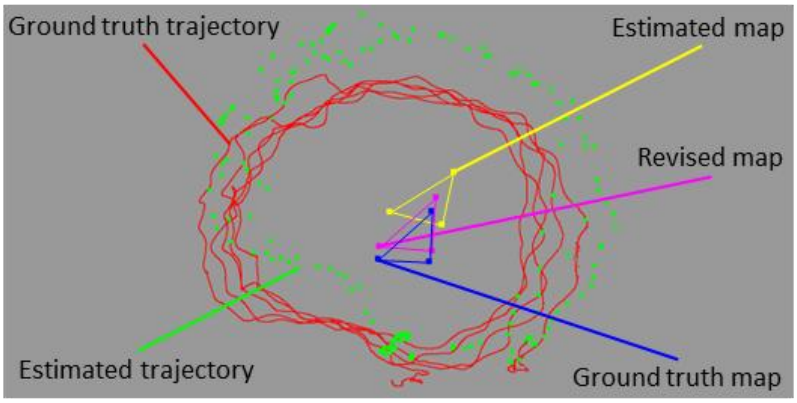


Figure 6-3. Trial outputs and ground truth map for small-scale scene experiments.

Table 6-1. RMSE of map before and after ICP and CPR-ICP error reductions for trials in the real-world small-scale scene

Trial No.	RMSE_map/mm	RMSE_map_ar_ICP/mm	RMSE_map_ar_CPR_ICP/mm
1	59.54	51.20	51.20
2	67.26	54.22	54.22
3	72.14	52.80	52.80
4	68.73	60.12	60.06
5	127.73	121.78	121.78
6	72.33	38.80	38.80
7	279.18	89.36	89.24
8	139.88	112.44	112.44
9	65.77	55.21	55.21

The map error reduction rates for both ICP and CPR-ICP, derived from the error data in Table 6-1, are presented in Table 6-2. In Table 6-2, the term “Map_error_reduction_rate_ICP” represents the map error reduction rate obtained using the classic ICP method, while

“Map_error_reduction_rate_CPR_ICP” refers to the rate achieved by the proposed CPR-ICP method. These values were computed using Eq. (5.1) and Eq. (5.2), as detailed in Section 5.2.1 . The term “mean” indicates the average map error reduction rate across nine repeated trials, providing a reliable measure of error reduction performance.

Table 6-2. Map error reduction rates for ICP and CPR-ICP operations in trials conducted in the real-world small-scale scene

Trial No.	Map_error_reduction_rate_ICP	Mean	Map_error_reduction_rate_CPR_ICP	Mean
1	14.01%	25.27%	14.01%	25.28%
2	19.38%		19.38%	
3	26.81%		26.81%	
4	12.54%		12.62%	
5	4.66%		4.66%	
6	46.36%		46.36%	
7	67.99%		68.04%	
8	19.62%		19.62%	
9	16.07%		16.07%	

The map error reduction rates for both CPR-ICP and ICP in Table 6-2 remain consistently positive, further validating the effectiveness of the proposed concept in enhancing mapping accuracy. Notably, CPR-ICP outperforms ICP in two trials, achieves identical reduction rates in the remaining trials, and attains a higher mean error reduction rate, reinforcing its superiority in improving mapping accuracy.

6.3 Experiments in medium-scale scene

6.3.1 Experimental setup and configuration

The proposed methodologies were further tested in a medium-scale scene using the same hardware setup as in the small-scale experiments. As shown in Figure 6-4 (a), the experimental scene consisted of a room measuring $243\,cm \times 259\,cm \times 414\,cm$. In Figure 6-4 (b), the global frame $\boldsymbol{G}$ was established at a corner where two walls meet, using Vicon markers for reference. The relative dimensions and geometry of the room with respect to $\boldsymbol{G}$ were manually measured to create a CAD mesh model representing the ground truth map. The CAD ground-truth mesh inherits small geometric inaccuracies from manual measurements. This uncertainty acts as a common reference bias: both ICP and CPR-ICP are evaluated against the same CAD mesh in the same global frame using the same point-to-mesh metric. As a result, the bias affects the absolute RMSE magnitudes but does not alter the relative ranking or the error-reduction rates reported for ICP versus CPR-ICP. In other words, the conclusions about which method achieves larger reductions remain valid, while absolute values should be interpreted with this background tolerance in mind.

During the experiment, the operator moved in looped trajectories, holding a laptop and SLAM camera, while continuously scanning the walls and floor to generate a point cloud representation of the room. As discussed previously, the estimated map was initially captured in the SLAM camera frame rather than the global frame and therefore required transformation into the global frame for further processing. The handling of ground truth and estimated trajectories follows the procedure outlined earlier and will not be reiterated here.

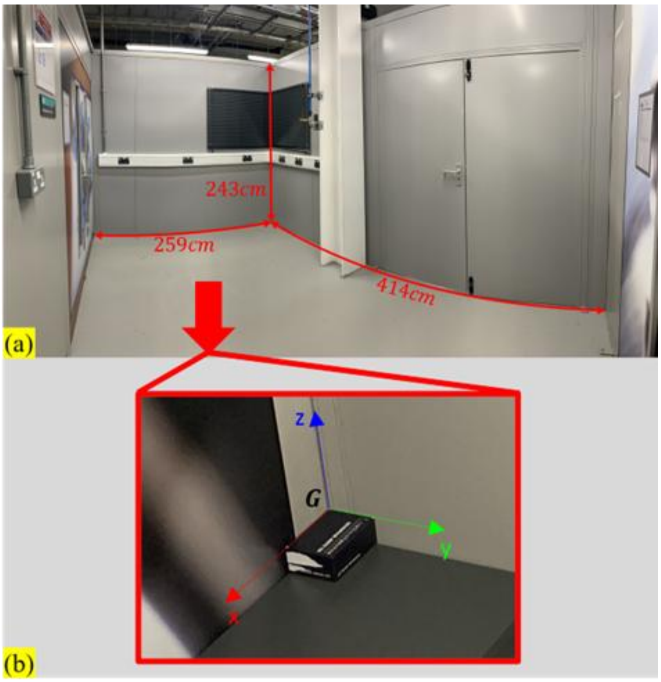


Figure 6-4. Experimental setup and configuration for the medium-scale scene. (a) Panoramic view of the medium-scale room. (b) Global coordinate frame and ground truth map.

6.3.2 Experimental results and analysis

A total of ten trials were conducted for the medium-scale scene, with the trial outputs and ground truth map displayed in Figure 6-5. This figure presents the point cloud representations of the estimated trajectory, ground truth trajectory, estimated map, and revised map following accuracy enhancements. The ground truth map, a pre-constructed CAD mesh model, was readily available for benchmarking. The experimental results are summarized in Table 6-3 and Table 6-4. Table 6-3 lists the original map errors along with their corresponding errors before and after applying ICP and CPR-ICP reductions.

Table 6-4 presents the map error reduction rates for ICP and CPR-ICP, respectively. The symbols and notation used in Table 6-3 and Table 6-4 remain consistent with those in Table 6-1 and Table 6-2 from the small-scale scene experiments, ensuring comparability and consistency across different scene sizes.

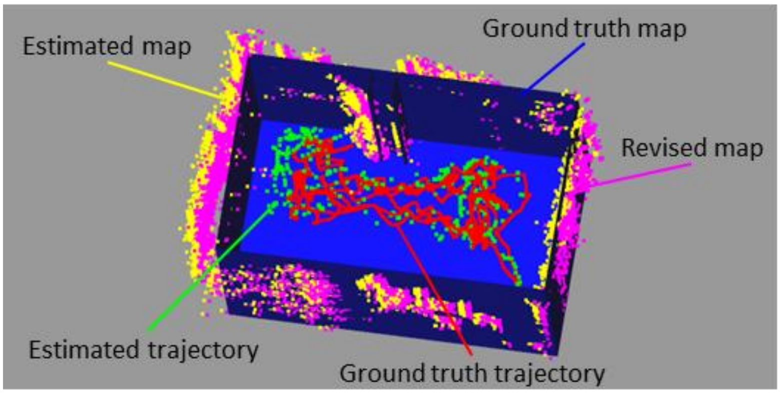


Figure 6-5. Trial outputs and ground truth map for medium-scale scene experiments.

Table 6-3. RMSE of map before and after ICP and CPR-ICP error reductions for trials in the real-world medium-scale scene

Trial No.	RMSE_map/mm	RMSE_map_ar_ICP/mm	RMSE_map_ar_CPR_ICP/mm
1	216.90	212.28	211.83
2	224.86	219.50	219.50
3	214.65	213.87	213.87
4	193.88	140.60	140.41
5	154.71	100.24	100.26
6	104.01	86.27	86.22
7	117.12	93.95	93.95
8	140.93	104.93	104.94
9	118.06	100.09	100.09
10	141.49	115.99	115.99

Table 6-4. Map error reduction rates for ICP and CPR-ICP operations in trials conducted in the real-world medium-scale scene

Trial No.	Map_error_reduction_rate_ICP	Mean	Map_error_reduction_rate_CPR_ICP	Mean
1	2.13%	16.32%	2.337%	16.35%
2	2.38%		2.382%	
3	0.36%		0.362%	
4	27.48%		27.579%	
5	35.21%		35.195%	
6	17.06%		17.105%	
7	19.78%		19.783%	
8	25.54%		25.534%	
9	15.22%		15.224%	
10	18.02%		18.023%	

The map error reduction rates for both CPR-ICP and ICP, as presented in Table 6-4, remain consistently positive, further validating the effectiveness of the proposed approach in improving mapping accuracy. Notably, CPR-ICP outperforms ICP in four trials, achieves identical reduction rates in another four, and falls slightly behind in two trials. Furthermore, CPR-ICP achieves a higher mean error reduction rate, reinforcing its advantage over ICP in enhancing mapping precision.

6.4 Experiments in large-scale scene

6.4.1 Experimental setup and configuration

The proposed methodologies were ultimately tested in a large-scale environment, using the same hardware setup as in the previous experiments. As illustrated in Figure 6-6 (a), the scene consisted of a room measuring $243\ cm\ \times\ 659\ cm\ \times\ 692\ cm$. The experimental configuration, shown in Figure 6-6 (b), followed the same setup as the medium-scale scene, with the global frame $\boldsymbol{G}$ established at a corner where two walls intersect, marked by Vicon markers. The room's dimensions and layout relative to $\boldsymbol{G}$ were manually measured to create the CAD mesh model used as the ground truth map.

Since the ground truth map was generated from a manually drawn CAD model, minor inaccuracies may be present. Same as Section 6.3.1 : the CAD reference adds a small common bias, shifting absolute RMSEs slightly but not the ICP–CPR-ICP comparison.

Compared with smaller rooms, the large-scale scene introduces several challenges: (i) longer path length leads to odometric/IMU drift accumulation and less frequent loop closures, so initial misalignment between sub-maps becomes larger; (ii) long planar surfaces and repetitive panels reduce distinctive features and cause perceptual aliasing and geometric degeneracy (low parallax); (iii) lighting/reflectance variations across the room (e.g., metallic doors, uneven illumination) make photometric tracking less stable; (iv) increased camera-to-wall range (up to ≈7 m) lowers point density and degrades depth/triangulation accuracy; (v) longer coverage induces faster motions and motion blur/rolling-shutter as well as more occlusions. These factors yield coarser pre-reduction alignment and less uniform overlap, making the large-scale case more demanding than the small/medium scenes.

During testing, the operator moved in looped trajectories, carrying a laptop and SLAM camera, while continuously scanning the walls and floor to generate a point cloud representation of the scene. As previously noted, the estimated map was initially captured in the SLAM camera frame, requiring transformation into the global frame for further analysis. The processing of ground truth and estimated trajectories follows the same methodology outlined earlier and will not be reiterated here.

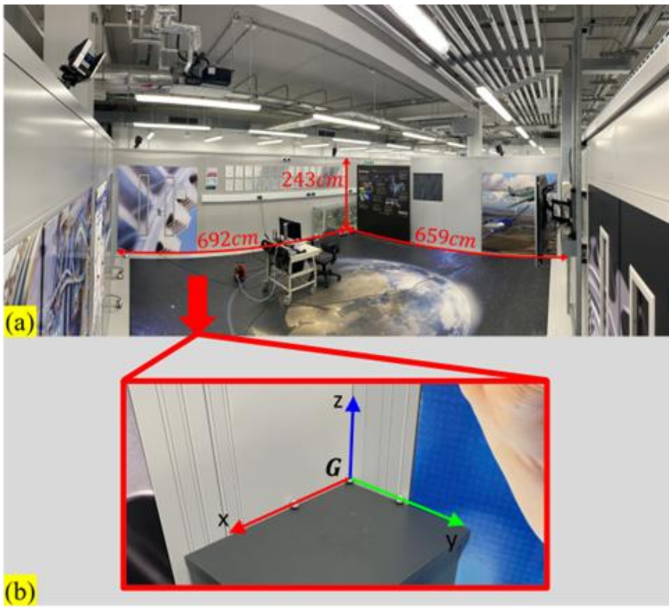


Figure 6-6. Experimental setup and configuration for the large-scale scene. (a) Panoramic view of the large-scale room. (b) Global coordinate frame and ground truth map.

6.4.2 Experimental results and analysis

A total of twenty trials were conducted in the large-scale scene, with their outputs and the ground truth map displayed in Figure 6-7. This figure presents the point cloud representations of the estimated trajectory, ground truth trajectory, estimated map, and the refined map following accuracy enhancements. The ground truth map, a pre-constructed CAD mesh model, was prepared in advance for benchmarking purposes. The experimental results are summarized in Table 6-5 and Table 6-6. Table 6-5 lists the initial map errors along with their corresponding values before and after applying ICP and CPR-ICP corrections. Table 6-6 presents the map error reduction rates for ICP and CPR-ICP. The notations and terminology used in Table 6-5 and Table 6-6 remain consistent with those from the small-scale and medium-scale scene experiments, ensuring comparability and maintaining the same physical significance across different scene sizes.

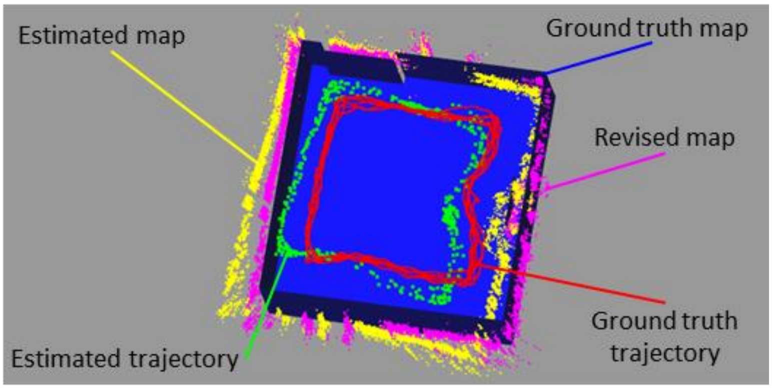


Figure 6-7. Trial outputs and ground truth map for large-scale scene experiments.

Table 6-5. RMSE of map before and after ICP and CPR-ICP error reductions for trials in the real-world large-scale scene

Trial No.	RMSE_map/mm	RMSE_map_ar_ICP/mm	RMSE_map_ar_CPR_ICP/mm
1	428.94	326.33	322.87
2	1039.40	558.71	507.42
3	274.51	251.02	251.23
4	460.69	354.82	233.10
5	334.94	312.60	312.60
6	187.98	178.77	178.77
7	463.55	298.94	298.95
8	469.03	308.64	308.64
9	495.08	263.74	264.09
10	483.76	278.14	279.97
11	257.76	253.14	253.31
12	297.18	218.74	218.74
13	312.10	222.27	222.25
14	352.08	281.90	279.36
15	434.42	264.44	264.44
16	344.99	240.92	240.92
17	409.83	321.17	321.00
18	289.85	232.04	231.91
19	486.89	329.49	328.41
20	313.41	250.56	251.20

Table 6-6. Map error reduction rates for ICP and CPR-ICP operations in trials conducted in the real-world large-scale scene

Trial No.	Map_error_reduction_rate_ICP	Mean	Map_error_reduction_rate_CPR_ICP	Mean
1	23.92%	25.62%	24.73%	27.24%
2	46.25%		51.18%	
3	8.56%		8.48%	
4	22.98%		49.40%	
5	6.67%		6.67%	
6	4.90%		4.90%	

7	35.51%		35.51%	
8	34.20%		34.20%	
9	46.73%		46.66%	
10	42.50%		42.13%	
11	1.79%		1.73%	
12	26.40%		26.40%	
13	28.78%		28.79%	
14	19.93%		20.66%	
15	39.13%		39.13%	
16	30.17%		30.17%	
17	21.63%		21.67%	
18	19.94%		19.99%	
19	32.33%		32.55%	
20	20.06%		19.85%	

The map error reduction rates for both CPR-ICP and ICP in Table 6-6 remain consistently positive, further validating the effectiveness of the proposed approach in enhancing mapping accuracy. Notably, CPR-ICP outperforms ICP in nine trials, produces identical results in five trials, and falls behind in six. Furthermore, CPR-ICP achieves a higher mean error reduction rate, further demonstrating its superior capability in refining mapping precision compared to the conventional ICP method.

6.5 Experimental data aggregation and analysis

To consolidate the map error reduction rates from the previous three sections and identify patterns, trends, and insights into how ICP and CPR-ICP contribute to mapping accuracy improvements, Table 6-7 compiles the mean map error reduction rates from Table 6-2, Table 6-4 and Table 6-6. This table presents a comparison of the classic ICP method and the proposed CPR-ICP method, along with the differences in their mean error reduction rates across the three scene types. These differences serve as a quantitative measure of CPR-ICP's advantage over ICP in reducing map errors.

To better illustrate how this difference evolves with increasing scene size, Figure 6-8 visualizes the data from Table 6-7 in a line chart. The chart reveals a clear trend: as the scene size increases, the gap between CPR-ICP and ICP's mean map error reduction rates widens. This trend indicates that CPR-ICP's advantage in improving mapping accuracy becomes more pronounced in larger environments, demonstrating its greater effectiveness in handling complex and expansive mapping scenarios.

The increased difference between CPR-ICP and ICP with scene size is not only a consequence of handling larger initial misalignments. In larger real-world spaces, several effects accumulate: (i) longer trajectories cause greater drift and less frequent/less reliable loop closures, which yields coarser pre-alignment between sub-maps; (ii) overlap becomes patchy and anisotropic (long walls, large empty areas), making nearest-neighbour matches in ICP unstable or locally ambiguous; (iii) greater camera-to-surface range reduces point density and increases depth/triangulation noise; (iv) repetitive/planar structure produces geometric degeneracy and perceptual aliasing; (v) lighting and motion-blur variations further degrade correspondence quality.

CPR-ICP leverages cross-pose relations to enforce multi-pose consistency of correspondences and the resulting rigid transform, which stabilises alignment under coarse pre-alignment, non-uniform overlap, and noisier point sets. Consequently, the absolute reduction achieved by CPR-ICP grows faster than that of ICP as the environment size increases, leading to the larger numerical gap observed in Figure 6-8 (e.g., 1.62 % in the large-scale scene versus ⩽0.03 % in the smaller scenes).

Table 6-7. Mean map error reduction rates of ICP and CPR-ICP and their differences across different real-world scenes

Real-world scene type	Mean map error reduction rate of ICP	Mean map error reduction rate of CPR-ICP	Difference between the mean map error reduction rates of CPR-ICP and ICP
Small-scale scene	25.27%	25.28%	0.01%
Medium-scale scene	16.32%	16.35%	0.03%
Large-scale scene	25.62%	27.24%	1.62%

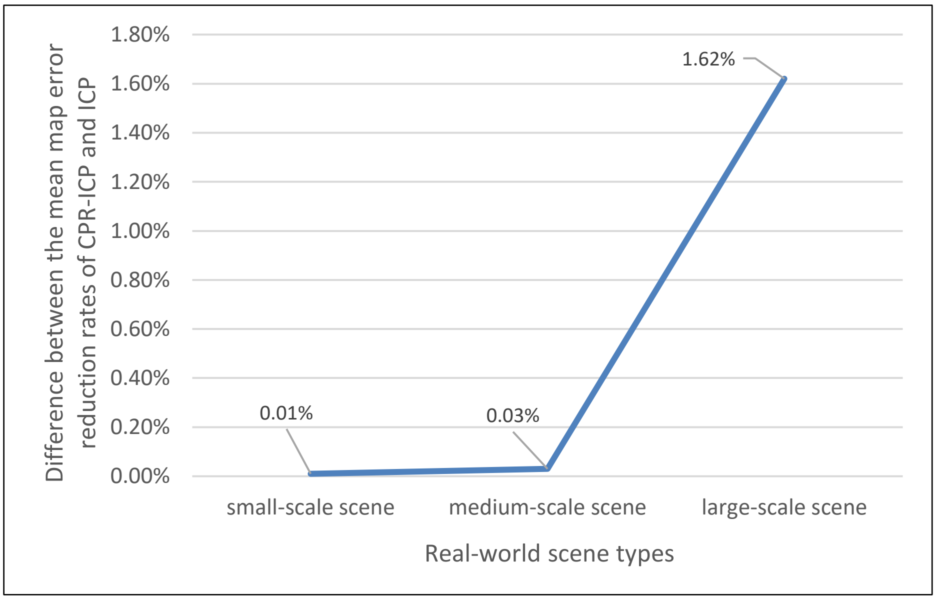


Figure 6-8. Difference between the mean map error reduction rates of CPR-ICP and ICP for each type of scene.

6.6 Conclusions

The real-world experiments presented in this chapter further validate the effectiveness of the proposed concept in enhancing mapping accuracy while demonstrating the clear advantage of CPR-ICP over the classic ICP method in reducing map errors. Conducted across diverse environments, these trials confirm the robustness and reliability of the proposed framework while revealing a compelling trend: CPR-ICP's superiority becomes more pronounced as scene size increases.

This pattern suggests that CPR-ICP not only delivers higher accuracy overall but also scales more effectively in larger environments, making it a more adaptable and efficient solution for complex mapping scenarios. Compared to ICP, CPR-ICP exhibits enhanced performance in handling the challenges posed by expansive spaces, offering a practical advantage in real-world applications. Through rigorous testing, this chapter establishes CPR-ICP as a more capable and scalable tool for improving mapping accuracy, reinforcing its superiority over conventional ICP methods.

Chapter 7 Conclusions and Future Works

7.1 Discussion on the research challenges and research objectives

7.1.1 Research challenges

Current SLAM benchmarks have two major challenges. The first challenge is ensuring a holistic and objective assessment on SLAM system performance. Many existing benchmarks either prioritize localization assessment while overlooking mapping—due to challenges in securing a dependable ground truth map—or treat localization and mapping as separate tasks rather than an interconnected process, undermining a comprehensive evaluation. Typically, mapping benchmark studies focus on 3D reconstruction of small to medium-sized objects, using available CAD models as ground truth. In these cases, estimated maps are evaluated independently by aligning them with the ground truth map via 3D shape registration techniques, followed by error measurement using specific metrics. However, this approach has a significant drawback: localization and mapping are inherently interdependent in the SLAM process, and an effective SLAM benchmark must therefore capture this relationship to truly assess system performance. This means preserving the original pose relationship between the estimated and ground truth maps during evaluation, reflecting the interplay between localization and mapping. Such an approach yields an unbiased, objective measure of SLAM performance. Yet, as previously noted, most studies isolate mapping, treating the estimated map as a standalone entity and relying on manual alignment, which obscures these vital correlations and produces biased results.

The second challenge is the acquisition of high-precision maps for SLAM benchmarking and various other robotic applications. Creating these maps—whether for use as reliable ground truth references or for purposes like navigation—is a slow and resource-heavy task. This difficulty has fuelled interest in devising a more efficient, less demanding method to produce quality maps that can serve as ground truth for SLAM benchmarking or support other robotic functions. Realizing this objective relies on reducing mapping errors to ensure estimated maps align more accurately with ground truth, emphasizing the value of holistic and objective SLAM benchmarks. Only these benchmarks can provide unbiased error measurements that enhance mapping precision and may enable the creation of high-quality maps. By contrast, traditional benchmarks generate biased error assessments that prove insufficient for this task.

7.1.2 Research objectives

To demonstrate the attainment of each research objective, the following section maps them to the corresponding chapters, sections, and key results. This provides a clear link between the conceptual developments and their experimental validation.

1. Identify the limitations of existing SLAM benchmarks
    - Critically analyse prior work and show the lack of holism/objectivity and the difficulty of obtaining ground-truth maps.

      Demonstrated in: Chapter 2 (Section 2.1 – Section 2.3) and Chapter 4 (Section 4.1).
2. Develop a holistic and objective SLAM benchmarking framework
    - Formulate a method that evaluates localization and mapping jointly in a common global frame and preserves their correlation.

      Demonstrated in: Chapter 4 (Section 4.2).
3. Leverage benchmark results as feedback to improve mapping accuracy
    - Establish and use the principle that unbiased localization error from the benchmark predicts a transformation that reduces map error.

      Demonstrated in: Chapter 4 (Section 4.3) and applied in the experiments of Chapter 5 (Section 5.2 , Section 5.2.5) and Chapter 6 (Section 6.2 – Section 6.5).
4. Propose a robust point-cloud registration technique (CPR-ICP)
    - Introduce a centroid/plane pre-alignment followed by ICP to stabilise convergence and enable the feedback step.

      Demonstrated in: Chapter 4 (Section 4.4 ; Section 4.4.1 – Section 4.4.2), with performance evidenced in Chapter 5 (Section 5.2 , Section 5.2.5) and Chapter 6 (Section 6.2 – Section 6.5).
5. Validate the unified methodology in simulation and real-world environments
    - Show consistent error reductions and trends across scene sizes and geometries.

Demonstrated in: Chapter 5 (Section 5.1 – Section 5.3) and Chapter 6 (Section 6.1 – Section 6.6), with aggregated analyses in Section 5.2.5 and Section 6.5 .

7.2 Academic findings and engineering validations

7.2.1 Innovative methodology for SLAM benchmarking

A novel SLAM benchmarking method is proposed, offering a holistic and objective evaluation of a SLAM system's overall performance. Unlike existing approaches that assess localization and mapping as separate, unrelated tasks, this method evaluates them as an integrated whole. By preserving the interdependence between these processes, it ensures objective, unbiased performance measurements, providing a more accurate representation of a SLAM system's true capabilities.

7.2.2 Innovative concept of mapping accuracy enhancement

A novel approach to improving mapping accuracy is introduced. The goal is to obtain maps with sufficient precision to act as ground-truth references for SLAM benchmarking or to support applications such as robot navigation, path planning, and obstacle avoidance. Instead of acquiring additional metrology data or performing manual alignments, the method reuses outputs of the unified benchmark as feedback: a single CPR-ICP registration is computed on the trajectory point clouds, and the resulting Euclidean transform is applied to the map. This reduces procedural overhead because no extra sensing hardware, scene annotation, or repeated data collection is required. Quantitative measurements of computational time or memory are not reported in this thesis; the discussion of efficiency is therefore qualitative and refers to the elimination of those additional data-acquisition and preparation steps. A systematic computational profiling is left for future work.

7.2.3 Innovative method for point cloud registration

A novel ICP variant CPR-ICP is proposed. It is designed to better minimize differences between point clouds by improving their alignment. The classic ICP method often fail when aligning point clouds without dependable initial feature matching. To overcome this, CPR-ICP starts by pre-aligning the point clouds using their centroids and least-squares planes, establishing a more precise starting point. It then builds on this foundation by applying the standard ICP algorithm to fine-tune the alignment and further reduce discrepancies.

7.2.4 Engineering validations for the proposed methodologies and concept

To validate the proposed methodologies and concepts for enhancing SLAM benchmarking and mapping accuracy, a series of experiments are conducted, categorized into two types: simulation-based and real-world tests. For both experiment types, the setup includes configuring the SLAM algorithm, Euclidean transformation algorithms to align coordinate frames, ICP and CPR-ICP algorithms, and evaluation metrics to measure errors. Simulation-based experiments require additional configuration of simulation software, whereas real-world tests involve defining the mapping environment and selecting instruments to establish accurate ground truth references. Multiple trials are performed across diverse environmental settings in both categories, showcasing the effectiveness of the proposed SLAM benchmark and the concept of improving mapping accuracy. By leveraging the impartial results from this benchmark, the CPR-ICP method is shown to outperform the classic ICP method in enhancing mapping accuracy. Furthermore, the experiments investigate how the performance of both methods in improving mapping accuracy relates to the geometries of the scenes and trajectories. The key correlations emerging from these investigations are: in simulation, for a fixed scene, enlarging the trajectory (larger loops/wider coverage) increases the map-error reduction rate; in simulation, for a fixed trajectory geometry, enlarging the scene decreases the reduction rate; in real-world experiments, the difference between the reduction rates of CPR-ICP and ICP widens as scene size increases.

These regularities are clearer in simulation due to tightly controlled conditions, whereas real-world trials involve additional, less controllable factors (e.g., lighting variability, dynamic elements, texture scarcity, sensor-range limits, calibration drift) that can attenuate or mask the patterns. Despite this variability, all experiments confirm that the mapping-accuracy-improvement procedure reduces map error, and CPR-ICP outperforms classic ICP in most cases. A more granular, factor-controlled comparison is left for future work. Building on these findings, several implications for future research are:

- Benchmarking and reporting: Treat scene size and trajectory scale/anisotropy as controlled factors; report initial misalignment, overlap/coverage, and loop-closure strength alongside error reductions.
- Trajectory planning and datasets: Collect larger, anisotropic loops to expose method differences; curate datasets stratified by scene size and overlap patterns.
- Algorithm choice: Prefer CPR-ICP in large or sparsely overlapped scenes or when pre-alignment is coarse; classic ICP suffices for small scenes with dense, uniform overlap.

7.3 Future works

In this PhD project, the proposed concept and methodologies were only tested on one SLAM algorithm using limited number of environmental settings. To extend their applicability to broader applications, they will be further tested on a wide range of SLAM systems, including laser-based and other Visual SLAM systems, across more diverse environmental conditions. Additionally, computational cost will be assessed, and robustness will be examined under extreme conditions, such as occlusions and low illumination, where point clouds may be incomplete or noisy.

Furthermore, the method for mapping accuracy enhancement can be improved in two aspects:

- **Optimization of trajectory alignment:**

  The first step in enhancing mapping accuracy involves optimizing the alignment between the ground truth and estimated trajectories to minimize error. Both trajectories are represented as point clouds, but the ground truth trajectory contains significantly more points than the estimated one. This discrepancy arises because ground truth poses are captured at a high sampling rate (30 samples per second or higher), whereas the estimated trajectory consists only of keyframe poses.

  To address this, the ground truth trajectory is down-sampled using a spatial nearest-neighbour technique relative to the estimated trajectory before applying ICP or CPR-ICP. However, this approach has a limitation: it relies on spatial distance to match corresponding points between the two trajectories, whereas using time difference would be a more logical criterion. Ideally, if conditions permit, synchronization techniques should be employed to capture points for both the ground truth and estimated trajectories simultaneously, ensuring that corresponding points are those recorded at the same time.

- **Leveraging machine learning for mapping accuracy improvement:**

  In the current approach, Euclidean transformations that minimize localization errors are directly applied to the estimated map to enhance its accuracy. Experimental results have confirmed the effectiveness of this method, reinforcing the inherent link between localization and mapping and suggesting a positive correlation between their errors. This insight presents an opportunity to integrate machine learning techniques for further improvement.

  The proposed machine learning approach would involve conducting a large number of experimental trials, similar to those presented in Chapter 5 and Chapter 6. For each trial, the respective Euclidean transformations for minimizing localization and mapping errors would be recorded. A neural network would then be trained to learn the relationship between localization and mapping error minimization transformations. As more trials are conducted, the network would become increasingly robust. Once trained, this network model could be applied to enhance mapping accuracy in various SLAM scenarios.

References

[1] Hoppe, H., et al. *Surface reconstruction from unorganized points*. in *Proceedings of the 19th annual conference on computer graphics and interactive techniques*. 1992.

[2] Weinmann, M., B. Jutzi, and C. Mallet, *Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014. **2**: p. 181-188.

[3] Song, S. and J. Xiao. *Sliding shapes for 3d object detection in depth images*. in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. 2014. Springer.

[4] Brodu, N. and D. Lague, *3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology.* ISPRS journal of photogrammetry and remote sensing, 2012. **68**: p. 121-134.

[5] Mallet, C., U. Soergel, and F. Bretar. *Analysis of full-waveform lidar data for classification of urban areas*. in *ISPRS Congress 2008*. 2008.

[6] Hackel, T., J.D. Wegner, and K. Schindler, *Fast semantic segmentation of 3D point clouds with strongly varying density.* ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, 2016. **3**: p. 177-184.

[7] Qi, C.R., et al. *Pointnet: Deep learning on point sets for 3d classification and segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[8] Qi, C.R., et al., *Pointnet++: Deep hierarchical feature learning on point sets in a metric space.* Advances in neural information processing systems, 2017. **30**.

[9] Zhou, Y. and O. Tuzel. *Voxelnet: End-to-end learning for point cloud based 3d object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[10] Wang, Y., et al., *Dynamic graph cnn for learning on point clouds.* ACM Transactions on Graphics (tog), 2019. **38**(5): p. 1-12.

[11] Zhao, H., et al. *Point transformer*. in *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.

[12] Harris, C. and M. Stephens. *A combined corner and edge detector*. in *Alvey vision conference*. 1988. Citeseer.

[13] Lowe, D.G., *Distinctive image features from scale-invariant keypoints.* International journal of computer vision, 2004. **60**: p. 91-110.

[14] Bay, H., T. Tuytelaars, and L. Van Gool. *Surf: Speeded up robust features*. in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. 2006. Springer.

[15] Rublee, E., et al. *ORB: An efficient alternative to SIFT or SURF*. in *2011 International conference on computer vision*. 2011. Ieee.

[16] Calonder, M., et al. *Brief: Binary robust independent elementary features*. in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. 2010. Springer.

[17] Engel, J., T. Schöps, and D. Cremers. *LSD-SLAM: Large-scale direct monocular SLAM*. in *European conference on computer vision*. 2014. Springer.

[18] Lucas, B.D. and T. Kanade. *An iterative image registration technique with an application to stereo vision*. in *IJCAI'81: 7th international joint conference on Artificial intelligence*. 1981.

[19] DeTone, D., T. Malisiewicz, and A. Rabinovich. *Superpoint: Self-supervised interest point detection and description*. in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.

[20] Ono, Y., et al., *LF-Net: Learning local features from images.* Advances in neural information processing systems, 2018. **31**.

[21] Dusmanu, M., et al. *D2-net: A trainable cnn for joint description and detection of local features*. in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2019.

[22] Cover, T. and P. Hart, *Nearest neighbor pattern classification.* IEEE transactions on information theory, 1967. **13**(1): p. 21-27.

[23] Fix, E., *Discriminatory analysis: nonparametric discrimination, consistency properties*. Vol. 1. 1985: USAF school of Aviation Medicine.

[24] Thrun, S., *Probabilistic robotics.* Communications of the ACM, 2002. **45**(3): p. 52-57.

[25] Neira, J. and J.D. Tardós, *Data association in stochastic mapping using the joint compatibility test.* IEEE Transactions on robotics and automation, 2001. **17**(6): p. 890-897.

[26] Cox, I.J. and S.L. Hingorani, *An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking.* IEEE Transactions on pattern analysis and machine intelligence, 1996. **18**(2): p. 138-150.

[27] Kim, G. and A. Kim. *Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map*. in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018. IEEE.

[28] Cummins, M. and P. Newman, *FAB-MAP: Probabilistic localization and mapping in the space of appearance.* The International journal of robotics research, 2008. **27**(6): p. 647-665.

[29] Cadena, C., et al., *Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age.* IEEE Transactions on robotics, 2016. **32**(6): p. 1309-1332.

[30] Rusu, R.B. and S. Cousins. *3d is here: Point cloud library (pcl)*. in *2011 IEEE international conference on robotics and automation*. 2011. IEEE.

[31] Elfes, A., *Using occupancy grids for mobile robot perception and navigation.* Computer, 1989. **22**(6): p. 46-57.

[32] Moravec, H. and A. Elfes. *High resolution maps from wide angle sonar*. in *Proceedings. 1985 IEEE international conference on robotics and automation*. 1985. IEEE.

[33] Agarwal, S., et al., *Building rome in a day.* Communications of the ACM, 2011. **54**(10): p. 105-112.

[34] Huang, A.S., et al. *Visual odometry and mapping for autonomous flight using an RGB-D camera*. in *Robotics Research: The 15th International Symposium ISRR*. 2017. Springer.

[35] Li, R., et al., *Initial results of rover localization and topographic mapping for the 2003 Mars Exploration Rover mission.* Photogrammetric Engineering & Remote Sensing, 2005. **71**(10): p. 1129-1142.

[36] Cheng, Y., M.W. Maimone, and L. Matthies, *Visual odometry on the Mars exploration rovers-a tool to ensure accurate driving and science imaging.* IEEE Robotics & Automation Magazine, 2006. **13**(2): p. 54-62.

[37] Di, K., et al., *Photogrammetric processing of rover imagery of the 2003 Mars Exploration Rover mission.* ISPRS Journal of Photogrammetry and Remote Sensing, 2008. **63**(2): p. 181-201.

[38] Martin-Mur, T.J., et al. *Mars science laboratory navigation results*. in *23rd international symposium on space flight dynamics*. 2012.

[39] Liu, B., et al., *Descending and landing trajectory recovery of Chang'e-3 lander using descent images.* Journal of Remote Sensing, 2014. **18**(5): p. 981-987.

[40] Wan, W., et al., *A cross-site visual localization method for Yutu rover.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014. **40**: p. 279-284.

[41] Wan, W., et al., *Descent image matching based position evaluation for Chang'e-3 landing point.* Spacecraft Engineering, 2014. **23**(4): p. 5-12.

[42] Liu, Z., et al., *High precision landing site mapping and rover localization for Chang'e-3 mission.* Science China Physics, Mechanics & Astronomy, 2015. **58**: p. 1-11.

[43] Levinson, J.S., *Automatic laser calibration, mapping, and localization for autonomous vehicles*. 2011: Stanford University.

[44] Cheng, J., et al., *A review of visual SLAM methods for autonomous driving vehicles.* Engineering Applications of Artificial Intelligence, 2022. **114**: p. 104992.

[45] Hidalgo, F. and T. Bräunl. *Review of underwater SLAM techniques*. in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*. 2015. IEEE.

[46] Zhao, W., et al. *Review of slam techniques for autonomous underwater vehicles*. in *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*. 2019.

[47] Mejias, L., et al. *COLIBRI: A vision-guided UAV for surveillance and visual inspection*. in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007. IEEE.

[48] Krajník, T., et al. *A simple visual navigation system for an UAV*. in *International Multi-Conference on Systems, Signals & Devices*. 2012. IEEE.

[49] Jensfelt, P., et al. *Integrating slam and object detection for service robot tasks*. in *IROS 2005 Workshop on Mobile Manipulators: Basic Techniques, New Trends and Applications*. 2005.

[50] Jung, M.-J., et al. *Ambiguity resolving in structured light 2D range finder for SLAM operation for home robot applications*. in *IEEE Workshop on Advanced Robotics and its Social Impacts, 2005*. 2005. IEEE.

[51] Jensfelt, P., et al. *Augmenting slam with object detection in a service robot framework*. in *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*. 2006. IEEE.

[52] Schlegel, C. and S. Hochdorfer. *Bearing-Only SLAM with an Omnicam: An Experimental Evaluation for Service Robotics Applications*. in *Autonome Mobile Systeme 2005: 19. Fachgespräch Stuttgart, 8./9. Dezember 2005*. 2006. Springer.

[53] Ekvall, S., D. Kragic, and P. Jensfelt, *Object detection and mapping for service robot tasks.* Robotica, 2007. **25**(2): p. 175-187.

[54] Tribelhorn, B. and Z. Dodds. *Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education*. in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007. IEEE.

[55] Tribelhorn, B. and Z. Dodds. *Envisioning the Roomba as AI Resource: A Classroom and Laboratory Evaluation*. in *AAAI Spring Symposium: Semantic Scientific Knowledge Integration*. 2007.

[56] Hochdorfer, S. and C. Schlegel. *Landmark rating and selection according to localization coverage: Addressing the challenge of lifelong operation of SLAM in service robots*. in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009. IEEE.

[57] Vallivaara, I., et al. *Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task*. in *2011 15th international conference on advanced robotics (ICAR)*. 2011. IEEE.

[58] Zhang, J., et al. *An approach to restaurant service robot SLAM*. in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2016. IEEE.

[59] Lee, T.-j., C.-h. Kim, and D.-i.D. Cho, *A monocular vision sensor-based efficient SLAM method for indoor service robots.* IEEE Transactions on Industrial Electronics, 2018. **66**(1): p. 318-328.

[60] Veerannapeta, V., *Low Power Indoor Robotic Vacuum Cleaner Using Sensors and SLAM.* International Research Journal of Innovations in Engineering and Technology, 2019. **3**(4): p. 51.

[61] Ouyang, M., et al. *A collaborative visual SLAM framework for service robots*. in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021. IEEE.

[62] Lee, S. and S. Lee, *Embedded visual SLAM: Applications for low-cost consumer robots.* IEEE Robotics & Automation Magazine, 2013. **20**(4): p. 83-95.

[63] Winterhalter, W., et al. *Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans*. in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015. IEEE.

[64] Garon, M., et al. *Real-time high resolution 3D data on the HoloLens*. in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 2016. IEEE.

[65] Evans, G., et al. *Evaluating the Microsoft HoloLens through an augmented reality assembly application*. in *Degraded environments: sensing, processing, and display 2017*. 2017. SPIE.

[66] Froehlich, M., S. Azhar, and M. Vanture. *An investigation of Google Tango® tablet for low cost 3D scanning*. in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*. 2017. IAARC Publications.

[67] Lee, J. *4‐1: Invited Paper: Mobile AR in Your Pocket with Google Tango*. in *SID Symposium Digest of Technical Papers*. 2017. Wiley Online Library.

[68] Nguyen, K.A. and Z. Luo. *On assessing the positioning accuracy of Google Tango in challenging indoor environments*. in *2017 international conference on indoor positioning and indoor navigation (IPIN)*. 2017. IEEE.

[69] Ranganathan, A. *The Oculus Insight positional tracking system*. 2022; Available from: https://www.aiacceleratorinstitute.com/the-oculus-insight-positional-tracking-system-2/.

[70] Vom Hofe, N., et al. *Robotics meets Augmented Reality: Real-Time Mapping with Boston Dynamics Spot and Microsoft HoloLens 2*. in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2023. IEEE.

[71] Klette, R., et al. *Evaluation of surface reconstruction methods*. in *New Zealand Image and Vision Computing Workshop*. 1995. Citeseer.

[72] Eid, A. and A. Farag. *A unified framework for performance evaluation of 3-D reconstruction techniques*. in *2004 Conference on Computer Vision and Pattern Recognition Workshop*. 2004. IEEE.

[73] Seitz, S.M., et al. *A comparison and evaluation of multi-view stereo reconstruction algorithms*. in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. 2006. IEEE.

[74] Bellmann, A., et al. *A benchmarking dataset for performance evaluation of automatic surface reconstruction algorithms*. in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007. IEEE.

[75] Krolla, B. and D. Stricker, *Heterogeneous dataset acquisition for a continuously expandable benchmark (CEB)*. 2015.

[76] Chen, Y. and G. Medioni, *Object modeling by registration of multiple range images, 1991*. S.

[77] Besl, P.J. and N.D. McKay. *Method for registration of 3-D shapes*. in *Sensor fusion IV: control paradigms and data structures*. 1992. Spie.

[78] Li, X. and I. Guskov. *Multiscale Features for Approximate Alignment of Point-based Surfaces*. in *Symposium on geometry processing*. 2005. Citeseer.

[79] Aiger, D., N.J. Mitra, and D. Cohen-Or, *4-points congruent sets for robust pairwise surface registration*, in *ACM SIGGRAPH 2008 papers*. 2008. p. 1-10.

[80] Fischler, M.A. and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.* Communications of the ACM, 1981. **24**(6): p. 381-395.

[81] Rusinkiewicz, S. and M. Levoy. *Efficient variants of the ICP algorithm*. in *Proceedings third international conference on 3-D digital imaging and modeling*. 2001. IEEE.

[82] Low, K.-L., *Linear least-squares optimization for point-to-plane icp surface registration.* Chapel Hill, University of North Carolina, 2004. **4**(10): p. 1-3.

[83] Censi, A. *An ICP variant using a point-to-line metric*. in *2008 IEEE International Conference on Robotics and Automation*. 2008. Ieee.

[84] Cignoni, P., C. Rocchini, and R. Scopigno. *Metro: measuring error on simplified surfaces*. in *Computer graphics forum*. 1998. Wiley Online Library.

[85] Aspert, N., D. Santa-Cruz, and T. Ebrahimi. *Mesh: Measuring errors between surfaces using the hausdorff distance*. in *Proceedings. IEEE international conference on multimedia and expo*. 2002. IEEE.

[86] Girardeau-Montaut, D., *Détection de changement sur des données géométriques tridimensionnelles*. 2006, Télécom ParisTech.

[87] Girardeau-Montaut, D., *CloudCompare.* France: EDF R&D Telecom ParisTech, 2016. **11**.

[88] Fontana, G., M. Matteucci, and D.G. Sorrenti. *The RAWSEED proposal for representation-independent benchmarking of SLAM*. in *Workshop on experimental methodology and benchmarking in robotics research (RSS 2008)*. 2008.

[89] Carlone, L., et al. *Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization*. in *2015 IEEE international conference on robotics and automation (ICRA)*. 2015. IEEE.

[90] Zhou, Q.-Y., J. Park, and V. Koltun, *Open3D: A modern library for 3D data processing.* arXiv preprint arXiv:1801.09847, 2018.

[91] Sturm, J., et al. *A benchmark for the evaluation of RGB-D SLAM systems*. in *2012 IEEE/RSJ international conference on intelligent robots and systems*. 2012. IEEE.

[92] Burri, M., et al., *The EuRoC micro aerial vehicle datasets.* The International Journal of Robotics Research, 2016. **35**(10): p. 1157-1163.

[93] Smith, M., et al., *The new college vision and laser data set.* The International Journal of Robotics Research, 2009. **28**(5): p. 595-599.

[94] Blanco, J.-L., F.-A. Moreno, and J. Gonzalez, *A collection of outdoor robotic datasets with centimeter-accuracy ground truth.* Autonomous Robots, 2009. **27**: p. 327-351.

[95] Pandey, G., J.R. McBride, and R.M. Eustice, *Ford campus vision and lidar data set.* The International Journal of Robotics Research, 2011. **30**(13): p. 1543-1552.

[96] Geiger, A., P. Lenz, and R. Urtasun. *Are we ready for autonomous driving? the kitti vision benchmark suite*. in *2012 IEEE conference on computer vision and pattern recognition*. 2012. IEEE.

[97] Handa, A., et al. *A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM*. in *2014 IEEE international conference on Robotics and automation (ICRA)*. 2014. IEEE.

[98] Burgard, W., et al. *A comparison of SLAM algorithms based on a graph of relations*. in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009. IEEE.

[99] Kümmerle, R., et al., *On measuring the accuracy of SLAM algorithms.* Autonomous Robots, 2009. **27**: p. 387-407.

[100] Wulf, O., et al., *Benchmarking urban six ‐degree ‐of ‐freedom simultaneous localization and mapping.* Journal of Field Robotics, 2008. **25**(3): p. 148-163.

[101] Wulf, O., et al. *Ground truth evaluation of large urban 6D SLAM*. in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007. IEEE.

[102] Maddern, W., et al., *1 year, 1000 km: The oxford robotcar dataset.* The International Journal of Robotics Research, 2017. **36**(1): p. 3-15.

[103] Jeong, J., et al., *Complex urban dataset with multi-level sensors from highly diverse urban environments.* The International Journal of Robotics Research, 2019. **38**(6): p. 642-657.

[104] Ramezani, M., et al. *The newer college dataset: Handheld lidar, inertial and vision with ground truth*. in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020. IEEE.

[105] *2D LiDAR sensor - picoScan100*. Available from: https://www.sick.com/us/en/catalog/products/lidar-and-radar-sensors/lidar-sensors/picoscan100/c/g574970?tab=overview.

[106] Wu, T., et al., *Detailed analysis on generating the range image for lidar point cloud processing.* Electronics, 2021. **10**(11): p. 1224.

[107] *Bumblebee XB3 stereo camera*. Available from: https://www.flir.com/support/products/bumblebee-xb3-firewire/#Overview.

[108] Wasenmüller, O., M. Meyer, and D. Stricker. *CoRBS: Comprehensive RGB-D benchmark for SLAM using Kinect v2*. in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2016. IEEE.

[109] Funke, J. and T. Pietzsch. *A framework for evaluating visual slam*. in *Proceedings of the British Machine Vision Conference (BMVC)*. 2009.

[110] Nardi, L., et al. *Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM*. in *2015 IEEE international conference on robotics and automation (ICRA)*. 2015. IEEE.

[111] Geiger, A., et al., *Vision meets robotics: The kitti dataset.* The International Journal of Robotics Research, 2013. **32**(11): p. 1231-1237.

[112] Zhang, Q. and R. Pless. *Extrinsic calibration of a camera and laser range finder (improves camera calibration)*. in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. 2004. IEEE.

[113] Zhang, Z. *Flexible camera calibration by viewing a plane from unknown orientations*. in *Proceedings of the seventh ieee international conference on computer vision*. 1999. Ieee.

[114] Moré, J.J. *The Levenberg-Marquardt algorithm: implementation and theory*. in *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*. 2006. Springer.

[115] Kassir, A. and T. Peynot. *Reliable automatic camera-laser calibration*. in *Proceedings of the 2010 Australasian Conference on Robotics & Automation*. 2010. Australian Robotics and Automation Association (ARAA).

[116] Pandey, G., et al., *Extrinsic calibration of a 3d laser scanner and an omnidirectional camera.* IFAC Proceedings Volumes, 2010. **43**(16): p. 336-341.

[117] Geiger, A., et al. *Automatic camera and range sensor calibration using a single shot*. in *2012 IEEE international conference on robotics and automation*. 2012. IEEE.

[118] Scaramuzza, D., A. Harati, and R. Siegwart. *Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes*. in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007. IEEE.

[119] Moghadam, P., M. Bosse, and R. Zlot. *Line-based extrinsic calibration of range and image sensors*. in *2013 IEEE International Conference on Robotics and Automation*. 2013. IEEE.

[120] Boughorbal, F., et al. *Registration and integration of multisensor data for photorealistic scene reconstruction*. in *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*. 2000. SPIE.

[121] Williams, N., et al. *Automatic image alignment for 3D environment modeling*. in *Proceedings. 17th Brazilian Symposium on Computer Graphics and Image Processing*. 2004. IEEE.

[122] Alempijevic, A., et al. *Mutual information based sensor registration and calibration*. in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006. IEEE.

[123] Taylor, Z. and J. Nieto. *A mutual information approach to automatic calibration of camera and lidar in natural environments*. in *Australian Conference on Robotics and Automation*. 2012.

[124] Mastin, A., J. Kepner, and J. Fisher. *Automatic registration of LIDAR and optical images of urban scenes*. in *2009 IEEE conference on computer vision and pattern recognition*. 2009. IEEE.

[125] Wang, R., F.P. Ferrie, and J. Macfarlane. *Automatic registration of mobile LiDAR and spherical panoramas*. in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2012. IEEE.

[126] Pandey, G., et al., *Automatic extrinsic calibration of vision and lidar by maximizing mutual information.* Journal of Field Robotics, 2015. **32**(5): p. 696-722.

[127] Sommer, H., et al. *A low-cost system for high-rate, high-accuracy temporal calibration for LIDARs and cameras*. in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017. IEEE.

[128] Li, M. and A.I. Mourikis. *3-D motion estimation and online temporal calibration for camera-IMU systems*. in *2013 IEEE International Conference on Robotics and Automation*. 2013. IEEE.

[129] Liu, Y. and Z. Meng, *Online temporal calibration based on modified projection model for visual-inertial odometry.* IEEE Transactions on Instrumentation and Measurement, 2019. **69**(7): p. 5197-5207.

[130] Sturm, J., et al. *Towards a benchmark for RGB-D SLAM evaluation*. in *Rgb-d workshop on advanced reasoning with depth cameras at robotics: Science and systems conf.(rss)*. 2011.

[131] *Vicon Motion Capture System*. Available from: https://ps.is.mpg.de/pages/motion-capture.

[132] *Optitrack Studio*. 2019; Available from: https://commons.wikimedia.org/wiki/File:Optitrack_Studio.jpg.

[133] *Vicon in use*. Available from: https://ps.is.mpg.de/pages/motion-capture/, https://www.vicon.com/resources/case-studies/a-simple-motion-capture-system-delivering-powerful-results/.

[134] Olson, E. *AprilTag: A robust and flexible visual fiducial system*. in *2011 IEEE international conference on robotics and automation*. 2011. IEEE.

[135] Ceriani, S., et al., *Rawseeds ground truth collection systems for indoor self-localization and mapping.* Autonomous Robots, 2009. **27**(4): p. 353-371.

[136] FARO Technologies, I. *FARO Vantage Laser Tracker*. 2012; Available from: https://www.youtube.com/watch?v=ibEGQB-v9HI&t=84s.

[137] Abdallah, S.M., D.C. Asmar, and J.S. Zelek. *Towards benchmarks for vision SLAM algorithms*. in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* 2006. IEEE.

[138] Abdallah, S.M., D.C. Asmar, and J.S. Zelek, *A benchmark for outdoor vision SLAM systems.* Journal of Field Robotics, 2007. **24**(1‐2): p. 145-165.

[139] Liu, Y., et al., *Simultaneous localization and mapping related datasets: A comprehensive survey.* arXiv preprint arXiv:2102.04036, 2021.

[140] Meister, S., et al. *When can we use kinectfusion for ground truth acquisition*. in *Proc. Workshop on Color-Depth Camera Fusion in Robotics*. 2012. IEEE.

[141] Chekhlov, D., et al. *Robust real-time visual SLAM using scale prediction and exemplar based feature description*. in *2007 IEEE conference on computer vision and pattern recognition*. 2007. IEEE.

[142] Klein, G. and D. Murray. *Parallel tracking and mapping for small AR workspaces*. in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. 2007. IEEE.

[143] Handa, A., et al. *Real-time camera tracking: When is high frame-rate best?* in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*. 2012. Springer.

[144] Handa, A., et al. *Real-time camera tracking: When is high frame-rate best?* in *European Conference on Computer Vision*. 2012. Springer.

[145] Schofield, S., A. Bainbridge-Smith, and R. Green, *An improved semi-synthetic approach for creating visual-inertial odometry datasets.* Graphical Models, 2023. **126**: p. 101172.

[146] Marchisotti, D. and E. Zappa, *Virtual simulation benchmark for the evaluation of simultaneous localization and mapping and 3D reconstruction algorithm uncertainty.* Measurement Science and Technology, 2021. **32**(9): p. 095404.

[147] Lu, F. and E. Milios, *Globally consistent range scan alignment for environment mapping.* Autonomous robots, 1997. **4**: p. 333-349.

[148] Bar-Shalom, Y., X.R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. 2004: John Wiley & Sons.

[149] Eade, E. and T. Drummond. *Scalable monocular SLAM*. in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. 2006. IEEE.

[150] Civera, J., A.J. Davison, and J.M.M. Montiel. *Inverse depth to depth conversion for monocular slam*. in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007. IEEE.

[151] Clemente, L.A., et al. *Mapping Large Loops with a Single Hand-Held Camera*. in *Robotics: Science and Systems*. 2007.

[152] Williams, B., G. Klein, and I. Reid. *Real-time SLAM relocalisation*. in *2007 IEEE 11th international conference on computer vision*. 2007. IEEE.

[153] Eade, E. and T. Drummond. *Unified loop closing and recovery for real time monocular SLAM*. in *BMVC*. 2008.

[154] Klein, G. and D. Murray. *Improving the agility of keyframe-based SLAM*. in *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II 10*. 2008. Springer.

[155] Zhang, Y., et al., *Improved separated-parameter calibration method for binocular vision measurements with a large field of view.* Optics Express, 2020. **28**(3): p. 2956-2974.

[156] Jang, T.Y., S.D. Kim, and S.S. Hwang, *Visual Hull Tree: A New Progressive Method to Represent Voxel Data.* IEEE Access, 2020. **8**: p. 141850-141859.

[157] Verbin, D. and T. Zickler. *Toward a universal model for shape from texture*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[158] Paragios, N., Y. Chen, and O.D. Faugeras, *Handbook of mathematical models in computer vision*. 2006: Springer Science & Business Media.

[159] Nayar, S.K., M. Watanabe, and M. Noguchi, *Real-time focus range sensor.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996. **18**(12): p. 1186-1198.

[160] Sabater, N., et al. *Review of low-baseline stereo algorithms and benchmarks*. in *Image and Signal Processing for Remote Sensing XVI*. 2010. SPIE.

[161] Furukawa, Y. and C. Hernández, *Multi-view stereo: A tutorial.* Foundations and Trends® in Computer Graphics and Vision, 2015. **9**(1-2): p. 1-148.

[162] Theia-sfm.org, *Structure from Motion (SfM) — Theia Vision Library.* 2016.

[163] Kanade, T. and M. Okutomi, *A stereo matching algorithm with an adaptive window: Theory and experiment.* IEEE transactions on pattern analysis and machine intelligence, 1994. **16**(9): p. 920-932.

[164] Okutomi, M. and T. Kanade, *A multiple-baseline stereo.* IEEE Transactions on pattern analysis and machine intelligence, 1993. **15**(4): p. 353-363.

[165] Ohta, Y. and T. Kanade, *Stereo by intra-and inter-scanline search using dynamic programming.* IEEE Transactions on pattern analysis and machine intelligence, 1985(2): p. 139-154.

[166] Cox, I.J., et al., *A maximum likelihood stereo algorithm.* Computer vision and image understanding, 1996. **63**(3): p. 542-567.

[167] Chellappa, R. and S. Theodoridis, *Academic Press Library in Signal Processing, Volume 6: Image and Video Processing and Analysis and Computer Vision*. 2017: Academic Press.

[168] Black, M.J. and P. Anandan. *A framework for the robust estimation of optical flow*. in *1993 (4th) International Conference on Computer Vision*. 1993. IEEE.

[169] Rousseeuw, P.J. and A.M. Leroy, *Robust regression and outlier detection*. 2005: John wiley & sons.

[170] Szeliski, R. and R. Zabih. *An experimental comparison of stereo algorithms*. in *International Workshop on Vision Algorithms*. 1999. Springer.

[171] Scharstein, D. and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.* International journal of computer vision, 2002. **47**: p. 7-42.

[172] Scharstein, D. and R. Szeliski. *High-accuracy stereo depth maps using structured light*. in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* 2003. IEEE.

[173] Hirschmuller, H. and D. Scharstein. *Evaluation of cost functions for stereo matching*. in *2007 IEEE conference on computer vision and pattern recognition*. 2007. IEEE.

[174] Scharstein, D. and C. Pal. *Learning conditional random fields for stereo*. in *2007 IEEE conference on computer vision and pattern recognition*. 2007. IEEE.

[175] Scharstein, D., et al. *High-resolution stereo datasets with subpixel-accurate ground truth*. in *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*. 2014. Springer.

[176] Nakamura, Y., et al. *Occlusion detectable stereo-occlusion patterns in camera matrix*. in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1996. IEEE.

[177] Martull, S., M. Peris, and K. Fukui. *Realistic CG stereo image dataset with ground truth disparity maps*. in *ICPR workshop TrakMark2012*. 2012.

[178] Peris, M., et al. *Towards a simulation driven stereo vision system*. in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012. IEEE.

[179] Menze, M. and A. Geiger. *Object scene flow for autonomous vehicles*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[180] Frohlinghaus, T. and J.M. Buhmann. *Regularizing phase-based stereo*. in *Proceedings of 13th International Conference on Pattern Recognition*. 1996. IEEE.

[181] Egnal, G., M. Mintz, and R.P. Wildes, *A stereo confidence metric using single view imagery with comparison to five alternative approaches.* Image and vision computing, 2004. **22**(12): p. 943-957.

[182] Anke, B., et al., *A Benchmark dataset for performance evaluation of shape-from-X algorithms.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2008. **16**: p. 26.

[183] Mulligan, J., V. Isler, and K. Daniilidis. *Performance evaluation of stereo for tele-presence*. in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. 2001. IEEE.

[184] Morales, S. and R. Klette. *Ground truth evaluation of stereo algorithms for real world applications*. in *Computer Vision–ACCV 2010 Workshops: ACCV 2010 International Workshops, Queenstown, New Zealand, November 8-9, 2010, Revised Selected Papers, Part II 10*. 2011. Springer.

[185] Liu, Z. and R. Klette. *Approximated ground truth for stereo and motion analysis on real-world sequences*. in *Advances in Image and Video Technology: Third Pacific Rim Symposium, PSIVT 2009, Tokyo, Japan, January 13-16, 2009. Proceedings 3*. 2009. Springer.

[186] Ley, A., R. Hänsch, and O. Hellwich. *Syb3r: A realistic synthetic benchmark for 3d reconstruction from images*. in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. 2016. Springer.

[187] Bielova, O., et al. *A digital image processing pipeline for modelling of realistic noise in synthetic images*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019.

[188] Jeon, H.-G., et al. *Disc: A large-scale virtual dataset for simulating disaster scenarios*. in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019. IEEE.

[189] Cruz, S.D.D., et al. *Sviro: Synthetic vehicle interior rear seat occupancy dataset and benchmark*. in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020.

[190] Liu, Z., et al. *Benchmarking large-scale multi-view 3D reconstruction using realistic synthetic images*. in *Eleventh International Conference on Graphics and Image Processing (ICGIP 2019)*. 2020. SPIE.

[191] Jeon, H.-G., et al., *A large-scale virtual dataset and egocentric localization for disaster responses.* IEEE transactions on pattern analysis and machine intelligence, 2021.

[192] Koch, S., et al. *Hardware Design and Accurate Simulation of Structured-Light Scanning for Benchmarking of 3D Reconstruction Algorithms*. in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[193] Li, Y., et al., *V2x-sim: A virtual collaborative perception dataset for autonomous driving.* arXiv preprint arXiv:2202.08449, 2022.

[194] Wu, Z., Y. Monno, and M. Okutomi. *Are Realistic Training Data Necessary for Depth-from-Defocus Networks?* in *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*. 2022. IEEE.

[195] Haeusler, R. and R. Klette. *Benchmarking stereo data (not the matching algorithms)*. in *Joint Pattern Recognition Symposium*. 2010. Springer.

[196] Hirschmuller, H. and D. Scharstein, *Evaluation of stereo matching costs on images with radiometric differences.* IEEE transactions on pattern analysis and machine intelligence, 2008. **31**(9): p. 1582-1599.

[197] Leclercq, P. and J. Morris. *Robustness to noise of stereo matching*. in *12th International Conference on Image Analysis and Processing, 2003. Proceedings.* 2003. IEEE.

[198] Kolmogorov, V. and R. Zabih. *Computing visual correspondence with occlusions using graph cuts*. in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. 2001. IEEE.

[199] Zitnick, C.L. and T. Kanade, *A cooperative algorithm for stereo matching and occlusion detection.* IEEE Transactions on pattern analysis and machine intelligence, 2000. **22**(7): p. 675-684.

[200] Cabezas, I., V. Padilla, and M. Trujillo. *BMPRE: An error measure for evaluating disparity maps*. in *2012 IEEE 11th International Conference on Signal Processing*. 2012. IEEE.

[201] Wang, Z., et al., *Image quality assessment: from error visibility to structural similarity.* IEEE transactions on image processing, 2004. **13**(4): p. 600-612.

[202] Wang, Z., E.P. Simoncelli, and A.C. Bovik. *Multiscale structural similarity for image quality assessment*. in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. 2003. Ieee.

[203] Malpica, W.S. and A.C. Bovik. *Range image quality assessment by structural similarity*. in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2009. IEEE.

[204] Szeliski, R. *Prediction error as a quality metric for motion and stereo*. in *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999. IEEE.

[205] Gherardi, R. *Confidence-based cost modulation for stereo matching*. in *2008 19th International Conference on Pattern Recognition*. 2008. IEEE.

[206] Mordohai, P. *The self-aware matching measure for stereo*. in *2009 IEEE 12th International Conference on Computer Vision*. 2009. IEEE.

[207] Hermann, S. and T. Vaudrey. *The gradient-a powerful and robust cost function for stereo matching*. in *2010 25th International Conference of Image and Vision Computing New Zealand*. 2010. IEEE.

[208] Merrell, P., et al. *Real-time visibility-based fusion of depth maps*. in *2007 IEEE 11th International Conference on Computer Vision*. 2007. Ieee.

[209] Wedel, A., et al., *Stereoscopic scene flow computation for 3D motion understanding.* International Journal of Computer Vision, 2011. **95**: p. 29-51.

[210] Waechter, M., et al., *Virtual rephotography: Novel view prediction error for 3D reconstruction.* ACM Transactions on Graphics (TOG), 2017. **36**(1): p. 1-11.

[211] Morales, S. and R. Klette. *A third eye for performance evaluation in stereo sequence analysis*. in *International Conference on Computer Analysis of Images and Patterns*. 2009. Springer.

[212] Shin, B.-S., D. Caudillo, and R. Klette, *Evaluation of two stereo matchers on long real-world video sequences.* Pattern Recognition, 2015. **48**(4): p. 1113-1124.

[213] Varekamp, C., K. Hinnen, and W. Simons. *Detection and correction of disparity estimation errors via supervised learning*. in *2013 International Conference on 3D Imaging*. 2013. IEEE.

[214] Haeusler, R., R. Nair, and D. Kondermann. *Ensemble learning for confidence measures in stereo vision*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013.

[215] Little, J.J. and W.E. Gillett, *Direct evidence for occlusion in stereo and motion.* Image and Vision Computing, 1990. **8**(4): p. 328-340.

[216] Luo, A. and H. Burkhardt, *An intensity-based cooperative bidirectional stereo matching with simultaneous detection of discontinuities and occlusions.* International Journal of Computer Vision, 1995. **15**(3): p. 171-188.

[217] Chang, C., S. Chatterjee, and P.R. Kube. *On an analysis of static occlusion in stereo vision*. in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1991. IEEE.

[218] Konolige, K. *Small vision systems: Hardware and implementation*. in *Robotics Research: The Eighth International Symposium*. 1998. Springer.

[219] Trapp, R., S. Drüe, and G. Hartmann. *Stereo matching with implicit detection of occlusions*. in *Computer Vision—ECCV'98: 5th European Conference on Computer Vision Freiburg, Germany, June 2–6, 1998 Proceedings, Volume II 5*. 1998. Springer.

[220] Weng, J., N. Ahuja, and T.S. Huang. *Two-view matching*. in *Second Int Conf on Comput Vision*. 1988. Publ by IEEE.

[221] Hannah, M.J., *Computer matching of areas in stereo images*. 1974: Stanford University.

[222] Smitley, T. and R. Bajcsy. *Stereo processing of aerial, urban images*. in *Proc. Seventh Int. Conference on Pattern Recognition*. 1984.

[223] Anandan, P. *Computing dense displacement fields with confidence measures in scenes containing occlusion*. in *Intelligent Robots and Computer Vision*. 1985. SPIE.

[224] Anandan, P., *A computational framework and an algorithm for the measurement of visual motion.* International Journal of Computer Vision, 1989. **2**(3): p. 283-310.

[225] Hannah, M.J. *Detection of errors in match disparities*. in *Proc. Image Understanding Workshop*. 1982.

[226] Leclerc, Y.G., *Constructing simple stable descriptions for image partitioning.* International journal of computer vision, 1989. **3**(1): p. 73-102.

[227] Samaras, D., et al. *Variable albedo surface reconstruction from stereo and shape from shading*. in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. 2000. IEEE.

[228] Scharstein, D. *Stereo vision for view synthesis*. in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1996. IEEE.

[229] Matthies, L., *Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation.* International Journal of Computer Vision, 1992. **8**(1): p. 71-91.

[230] Scharstein, D. and R. Szeliski, *Stereo matching with nonlinear diffusion.* International journal of computer vision, 1998. **28**: p. 155-174.

[231] Hu, X. and P. Mordohai. *Evaluation of stereo confidence indoors and outdoors*. in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010. IEEE.

[232] Haeusler, R. and R. Klette. *Evaluation of stereo confidence measures on synthetic and recorded image data*. in *2012 International Conference on Informatics, Electronics & Vision (ICIEV)*. 2012. IEEE.

[233] Hu, X. and P. Mordohai, *A quantitative evaluation of confidence measures for stereo vision.* IEEE transactions on pattern analysis and machine intelligence, 2012. **34**(11): p. 2121-2133.

[234] Strecha, C., et al. *On benchmarking camera calibration and multi-view stereo for high resolution imagery*. in *2008 IEEE conference on computer vision and pattern recognition*. 2008. Ieee.

[235] Jensen, R., et al. *Large scale multi-view stereopsis evaluation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[236] Schops, T., et al. *A multi-view stereo benchmark with high-resolution images and multi-camera videos*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[237] Knapitsch, A., et al., *Tanks and temples: Benchmarking large-scale scene reconstruction.* ACM Transactions on Graphics (ToG), 2017. **36**(4): p. 1-13.

[238] Koutsoudis, A., et al., *Multi-image 3D reconstruction data evaluation.* Journal of cultural heritage, 2014. **15**(1): p. 73-79.

[239] Stathopoulou, E.K., M. Welponer, and F. Remondino, *Open-source image-based 3D reconstruction pipelines: Review, comparison and evaluation.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W17, 2019: p. 331-338.

[240] Eid, A.H., S.S. Rashad, and A.A. Farag. *A general-purpose platform for 3-D reconstruction from sequence of images*. in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*. 2002. IEEE.

[241] Levoy, M., *Stanford Spherical Gantry.* 2002.

[242] Schonberger, J.L. and J.-M. Frahm. *Structure-from-motion revisited*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[243] Schönberger, J.L., et al. *Pixelwise view selection for unstructured multi-view stereo*. in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. 2016. Springer.

[244] Moulon, P., et al. *Openmvg: Open multiple view geometry*. in *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1*. 2017. Springer.

[245] Wu, C., *VisualSFM: A visual structure from motion system.* http://www. cs. washington. edu/homes/ccwu/vsfm, 2011.

[246] Sırma, İ.E., *VisualSFM : 3D Construction of images.* 2014.

[247] Schöning, J. and G. Heidemann. *Evaluation of multi-view 3D reconstruction software*. in *Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015, Proceedings, Part II 16*. 2015. Springer.

[248] Opitz, R., et al. *Close-range photogrammetry vs. 3D scanning: Comparing data capture, processing and model generation in the field and the lab*. in *CAA 2012*. 2012.

[249] Nguyen, H.M., et al., *3D models from the black box: investigating the current state of image-based modeling.* 2012.

[250] Maes, F., et al., *Multimodality image registration by maximization of mutual information.* IEEE transactions on Medical Imaging, 1997. **16**(2): p. 187-198.

[251] Zollei, L., et al. *2D-3D rigid registration of X-ray fluoroscopy and CT images using mutual information and sparsely sampled histogram estimators*. in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001. IEEE.

[252] Zöllei, L., J.W. Fisher, and W.M. Wells. *A unified statistical and information theoretic framework for multi-modal image registration*. in *Information Processing in Medical Imaging: 18th International Conference, IPMI 2003, Ambleside, UK, July 20-25, 2003. Proceedings 18*. 2003. Springer.

[253] Troccoli, A.J. and P.K. Allen. *A shadow based method for image to model registration*. in *2004 Conference on Computer Vision and Pattern Recognition Workshop*. 2004. IEEE.

[254] Zhao, W., D. Nister, and S. Hsu, *Alignment of continuous video onto 3D point clouds.* IEEE transactions on pattern analysis and machine intelligence, 2005. **27**(8): p. 1305-1318.

[255] Liu, L., et al. *Multiview geometry for texture mapping 2d images onto 3d range data*. in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. 2006. IEEE.

[256] Vasile, A., et al. *Automatic alignment of color imagery onto 3d laser radar data*. in *35th IEEE Applied Imagery and Pattern Recognition Workshop (AIPR'06)*. 2006. IEEE.

[257] Ding, M., K. Lyngbaek, and A. Zakhor. *Automatic registration of aerial imagery with untextured 3d lidar models*. in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008. IEEE.

[258] Liu, L. and I. Stamos, *A systematic approach for 2D-image to 3D-range registration in urban environments.* Computer Vision and Image Understanding, 2012. **116**(1): p. 25-37.

[259] Cignoni, P., et al. *Meshlab: an open-source mesh processing tool*. in *Eurographics Italian chapter conference*. 2008. Salerno, Italy.

[260] Nikolov, I. and C. Madsen. *Benchmarking close-range structure from motion 3D reconstruction software under varying capturing conditions*. in *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection: 6th International Conference, EuroMed 2016, Nicosia, Cyprus, October 31–November 5, 2016, Proceedings, Part I 6*. 2016. Springer.

[261] Jiang, S., C. Jiang, and W. Jiang, *Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools.* ISPRS Journal of Photogrammetry and Remote Sensing, 2020. **167**: p. 230-251.

[262] Bianco, S., G. Ciocca, and D. Marelli, *Evaluating the performance of structure from motion pipelines.* Journal of Imaging, 2018. **4**(8): p. 98.

[263] Martell, A., H.A. Lauterbach, and A. Nuchtcer. *Benchmarking structure from motion algorithms of urban environments with applications to reconnaissance in search and rescue scenarios*. in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2018. IEEE.

[264] Roncella, R., C. Re, and G. Forlani, *Performance evaluation of a structure and motion strategy in architecture and cultural heritage.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2012. **38**: p. 285-292.

[265] Green, S., A. Bevan, and M. Shapland, *A comparative assessment of structure from motion methods for archaeological research.* Journal of Archaeological Science, 2014. **46**: p. 173-181.

[266] Wallace, L., et al., *Assessment of forest structure using two UAV techniques: A comparison of airborne laser scanning and structure from motion (SfM) point clouds.* Forests, 2016. **7**(3): p. 62.

[267] Marteau, B., et al., *Application of Structure ‐ from ‐ Motion photogrammetry to river restoration.* Earth Surface Processes and Landforms, 2017. **42**(3): p. 503-515.

[268] Hosseinian, S. and H. Arefi, *3D Reconstruction from Multi-View Medical X-ray images–review and evaluation of existing methods.* The international archives of the photogrammetry, remote sensing and spatial information sciences, 2015. **40**: p. 319-326.

[269] Lochhead, I. and N. Hedley, *Dry ‐ lab benchmarking of a structure from motion workflow designed to monitor marine benthos in three dimensions.* The Photogrammetric Record, 2021. **36**(175): p. 224-251.

[270] Mertes, J., T. Thomsen, and J. Gulley, *Evaluation of structure from motion software to create 3d models of late nineteenth century great lakes shipwrecks using archived diver-acquired video surveys.* Journal of Maritime Archaeology, 2014. **9**: p. 173-189.

[271] Cook, M.J. and J.B. DeSanto, *Validation of Geodetic Seafloor Benchmark Stability Using Structure ‐ From ‐ Motion and Seafloor Pressure Data.* Earth and Space Science, 2019. **6**(9): p. 1781-1786.

[272] Visser, F., et al., *An evaluation of a low-cost pole aerial photography (PAP) and structure from motion (SfM) approach for topographic surveying of small rivers.* International Journal of Remote Sensing, 2019. **40**(24): p. 9321-9351.

[273] Yang, Y., et al., *Evaluation of structure from motion (SfM) photogrammetry on the measurement of rill and interrill erosion in a typical loess.* Geomorphology, 2021. **385**: p. 107734.

[274] Cook, K.L., *An evaluation of the effectiveness of low-cost UAVs and structure from motion for geomorphic change detection.* Geomorphology, 2017. **278**: p. 195-208.

[275] Panagiotidis, D., P. Surový, and K. Kuželka, *Accuracy of Structure from Motion models in comparison with terrestrial laser scanner for the analysis of DBH and height influence on error behaviour.* Journal of Forest Science, 2016. **62**(8): p. 357-365.

[276] Bash, E.A., et al., *Evaluation of SfM for surface characterization of a snow-covered glacier through comparison with aerial lidar.* Journal of Unmanned Vehicle Systems, 2020. **8**(2): p. 119-139.

[277] Nouwakpo, S.K., et al., *Evaluation of structure from motion for soil microtopography measurement.* The photogrammetric record, 2014. **29**(147): p. 297-316.

[278] Iheaturu, C.J., E.G. Ayodele, and C.J. Okolie, *An assessment of the accuracy of structure-from-motion (SfM) photogrammetry for 3D terrain mapping.* Geomatics, landmanagement and landscape, 2020. **2**: p. 65-82.

[279] Nouwakpo, S.K., M.A. Weltz, and K. McGwire, *Assessing the performance of structure ‐ from ‐ motion photogrammetry and terrestrial LiDAR for reconstructing soil surface microtopography of naturally vegetated plots.* Earth Surface Processes and Landforms, 2016. **41**(3): p. 308-322.

[280] Yang, H., et al., *A new alternative for assessing ridging information of potato plants based on an improved benchmark structure from motion.* Computers and Electronics in Agriculture, 2023. **213**: p. 108220.

[281] Deng, H., T. Birdal, and S. Ilic. *Ppfnet: Global context aware local features for robust 3d point matching*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[282] Arun, K.S., T.S. Huang, and S.D. Blostein, *Least-squares fitting of two 3-D point sets.* IEEE Transactions on pattern analysis and machine intelligence, 1987(5): p. 698-700.

[283] Chen, Y. and G. Medioni, *Object modelling by registration of multiple range images.* Image and vision computing, 1992. **10**(3): p. 145-155.

[284] Zhang, Z., *Iterative point matching for registration of free-form curves and surfaces.* International journal of computer vision, 1994. **13**(2): p. 119-152.

[285] Stewart, G.W., *On the early history of the singular value decomposition.* SIAM review, 1993. **35**(4): p. 551-566.

[286] Mur-Artal, R. and J.D. Tardós, *Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras.* IEEE transactions on robotics, 2017. **33**(5): p. 1255-1262.