SLAM-TSM: Enhanced Indoor LiDAR SLAM With Total Station Measurements for Accurate Trajectory Estimation

Dedong Zhang, *Member, IEEE*, Weikai Tan, John Zelek, *Member, IEEE*, Lingfei Ma, *Member, IEEE*, and Jonathan Li, *Fellow, IEEE*

***Abstract*— Simultaneous Localization and Mapping (SLAM) is a crucial task in various domains, including intelligent robotics, computer vision, and indoor navigation. Accurate and robust trajectory estimation is especially challenging in indoor environments due to the presence of feature-poor or repetitive scenes, limited visibility, and dynamic objects. Obtaining highly accurate machine platform odometry is also an important basis for solving the "last mile" problem in intelligent transportation. This paper proposes a novel algorithm that combines LiDAR-based SLAM, total station measurements, and graph optimization to optimize the robot's trajectory in indoor environments. By integrating highly accurate positional data from total station measurements as additional constraints, the proposed method enhances the performance of indoor LiDAR SLAM, effectively addressing the challenges of drift and trajectory offsets. Moreover, the proposed algorithm can provide trajectory optimization even in the absence of loop closure detection, making it more robust and suitable for a broader range of indoor environments. Experimental results validated the effectiveness of the proposed approach in reducing drift and improving trajectory estimation for low-cost indoor LiDAR devices, demonstrating its potential in various applications such as autonomous navigation, facility management, and augmented reality. It provides targeted ground truth for autonomous driving of machine platforms in intelligent transportation scenarios.**

***Index Terms*— Mobile laser scanning, indoor SLAM, graph optimization, low-cost platform, trajectory optimization.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) has been a critical research topic in the field of intelligent robotics, autonomous systems, and intelligent transportation systems (ITS) for many years [1]. One of the most common applications of SLAM is in indoor environments, where Global Navigation Satellite System (GNSS) signals are usually unavailable or unreliable [2]. Corresponding methods are also needed to solve the tasks of high-definition map generation and environmental perception to complete the "last mile" problem in the task of autonomous driving and intelligent navigation. In such scenarios, alternative sensors, such as Light Detection and Ranging (LiDAR) sensors, cameras, or inertial measurement units (IMUs), are often used to estimate the robot's positions and build a map of this GNSS-denied environment [3].

LiDAR-based SLAM has gained increasing popularity due to its advantages over other sensing modalities, such as its high accuracy, long range, and remarkable robustness to various lighting conditions [4]. However, LiDAR SLAM in indoor environments often suffers from drift and trajectory offsets, which can lead to inaccurate localization and mapping results [5]. To address these challenges, several approaches have been proposed, including loop closure detection [6], graph optimization [7], and the fusion of multiple sensors [8]. In particular, graph optimization techniques have shown great potential in improving the accuracy and consistency of SLAM results by minimizing the errors between the estimated poses and the constraints from sensor measurements [9].

In this paper, we propose a LiDAR SLAM method, named SLAM-TSM, which incorporates graph optimization and total station measurements to improve the accuracy of the estimated trajectory without relying on loop closure detection in indoor environments. This is especially relevant for low-cost indoor LiDAR devices, which may have limitations in terms of range, resolution, or field-of-view (FoV), making loop closure detection challenging [10]. To be more specific, data preprocessing was first conducted to decrease noise, outliers, and computational complexity in the raw point clouds. Second, the Iterative Closest Point (ICP) algorithm [7] was employed to register consecutive point cloud scans and estimate the relative motions between different scans. Third, the total station measurements were integrated into the proposed algorithm to improve the accuracy of the estimated trajectory. Fourth, the key points collected by the total station were utilized as landmarks, which were incorporated into the SLAM system as additional constraints. Then, graph optimization was performed to refine

Received 1 November 2023; revised 31 May 2024, 8 October 2024, and 5 November 2024; accepted 16 November 2024. Date of publication 4 December 2024; date of current version 4 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 42101451, in part by the Natural Science and Engineering Research Council of Canada under Grant RGPIN-2022-03741, and in part by the Emerging Interdisciplinary Project of the Central University of Finance and Economics from China. The Associate Editor for this article was C. Wen. *(Corresponding authors: Lingfei Ma; Jonathan Li.)*

Dedong Zhang and John Zelek are with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: dedong.zhang@uwaterloo.ca; jzelek@uwaterloo.ca).

Weikai Tan is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: weikai.tan@uwaterloo.ca).

Lingfei Ma is with the School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 102206, China (e-mail: l53ma@cufe.edu.cn)

Jonathan Li is with the Department of Geography and Environmental Management and the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Digital Object Identifier 10.1109/TITS.2024.3504487

the robot's trajectory by minimizing the errors between the estimated poses and the constraints from scan registration and total station measurements. Finally, an alternative optimization approach without loop closure detection was carried out for further error minimization.

The main contributions of this work are as follows:

1) We propose a novel algorithm for indoor LiDAR SLAM that leverages total station measurements and graph optimization to optimize the robot's trajectory.
2) We demonstrate the effectiveness of the proposed method in reducing drift and improving trajectory estimations, even in challenging indoor GNSS-denied environments.
3) We show that the proposed algorithm can provide trajectory optimization without relying on loop closure detection, making it more robust in feature-poor or repetitive indoor scenes.

The remainder of this paper is organized as follows: Section II provides an overview of related work in the field of indoor SLAM, LiDAR-based SLAM, and total station integration. Section III describes the proposed methodology, including preprocessing, scan registration, total station measurement integration, and graph optimization. Section IV presents experimental results and evaluations of the proposed method. Finally, Section V concludes the paper and discusses future research directions.

II. Related Work

Various SLAM techniques have been proposed and developed over the years, with a wide range of applications in indoor environments. In this section, we review some of the most relevant works related to LiDAR-SLAM, loop closure detection, and graph optimization.

A. LiDAR-Based SLAM

LiDAR-based SLAM has been a prominent area of study in the domain of robotics and autonomous vehicles. The primary aim of such algorithms is to generate a map of the environment and concurrently localize the sensor within this map. This section aims to survey some key contributions and developments in this field. A seminal work in the field of LiDAR-based SLAM was introduced by Levinson and Thrun [11]. Their approach was primarily designed for autonomous driving applications, leveraging 3D LiDAR to perform real-time and large-scale SLAM. This algorithm combined the scan alignment with a graph-based optimization process to generate high-fidelity maps. This method was fundamental in inspiring further research in the field. LOAM (Lidar Odometry and Mapping) [4] represented another breakthrough in the field of LiDAR-based SLAM, which exhibited real-time performance, maintaining high precision over long distances. By segregating the problem of motion estimation and scene reconstruction, LOAM efficiently handled the computational challenges of the SLAM problem. Qin et al. [12] made several experimental comparisons of a large number of SLAM algorithms in different scenarios and found that LOAM could obtain good performance in both small and large test fields. SegMatch [13] offered an alternative approach to the LiDAR SLAM by segmenting the environment into identifiable elements, matching these segments across scans, and generating a pose graph for optimization. SegMatch demonstrated superior performance in urban environments, even with sparse LiDAR data.

More recently, advanced techniques like LeGO-LOAM [14], SuMa [15] and SuMa++ [16] have greatly improved previous algorithms. LeGO-LOAM was designed specifically for ground vehicles and achieved high accuracy while maintaining a low computational load. SuMa, on the other hand, employed a surfel-based representation, which was more efficient for large-scale 3D environments. SuMa++ uses the same Surfel-based Mapping (SuMa) pipeline to extend integrating semantic information, which improves the robustness and accuracy of the pose estimation by SuMa. Lastly, research on incorporating machine learning into the LiDAR SLAM, for example, Yang et al. [17] offered promising avenues for further development in the field. Their method used a deep learning network to identify salient features in the LiDAR data, reducing the reliance on manual feature engineering and potentially improving the robustness of SLAM algorithms. As LiDAR sensors become more available due to their decreasing costs, it's expected that research in LiDAR-based SLAM will continue to grow, providing even more robust and efficient solutions for mapping and localization.

B. Loop Closure Detection

Loop closure detection, also known as place recognition, is a critical component in SLAM systems. It is responsible for identifying when a robot revisits a previously mapped location, allowing for error correction in the map and trajectory over time. Angeli et al. [18] introduced a method based on a bag-of-words approach for visual loop closure detection. Their work highlighted the importance of invariance to viewpoint and lighting changes for effective loop closure identification in visual SLAM systems. FAB-MAP, a popular probabilistic technique for large-scale loop closure detection, was presented by Cummins and Newman [19]. They used a fast appearance-based mapping approach that demonstrated a high degree of robustness against perceptual aliasing and changes in the environment's appearance. Lu and Milios [20] pioneered work in scan matching techniques for loop closure detection in 2D SLAM, which was extended to 3D by Bosse and Zlot [21]. They introduced an approach based on scan context for loop closure detection in 3D LiDAR SLAM systems, focusing on achieving robustness to various types of environmental changes. Gálvez-López and Tardós [22] introduced a variant of the bag-of-words approach known as DBoW2, specifically optimized for large-scale loop closure detection, which was able to work effectively in extensive environments and provided a balance between detection performance and computational efficiency.In addition, Wang et al. [23] developed a tightly coupled lidar inertial odometry method for large-scale environments with tight loops. The framework introduces a direct point cloud registration method that does not require feature extraction. To ensure accuracy, the framework employs strategies including inertial measurement unit

(IMU) pre-integration, local-scale direct scan matching, and an efficient fused loop closure detection method with conditional detection.

A recent and exciting development in the field is the use of deep learning techniques for loop closure detection. Gao et al. [24] proposed a method based on Convolutional Neural Networks (CNNs) to learn and match visual features for loop closure detection, showing excellent performance in challenging environments. Similarly, Zeng et al. [25] leveraged Recurrent Neural Networks (RNNs) to capture temporal information for loop closure detection. As SLAM systems continue to evolve, advancements in loop closure detection will remain essential for maintaining accurate and reliable mapping and localization over time.

C. Graph Optimization

Graph optimization is also a critical technique utilized in SLAM systems to correct accumulated drift errors and improve the overall accuracy of mapping and trajectory estimation. In the field of large-scale SLAM, Konolige et al. [26] made significant contributions with their development of the Sparse Pose Adjustment (SPA) algorithm, which is a scalable approach that has demonstrated its effectiveness in managing large pose graphs while minimizing computational costs. Olson et al. [27] proposed a stochastic gradient descent method for the graph-based SLAM, demonstrating its practicality for large-scale indoor and outdoor environments. Additionally, the g2o framework, presented by Kümmerle et al. [9], is one of the most notable contributions in this field. It provided a general framework for graph optimization, offering flexibility to incorporate various kinds of nodes and constraints, hence making it widely applicable across a broad range of SLAM applications. Grisetti et al. [28] introduced the concept of hierarchical optimization for pose graphs in their work on the Hierarchical Optimization on Manifold (HOG-Man) algorithm. This work showed how a hierarchy of coarser representations could be used to accelerate the graph optimization process. Dellaert and Kaess [29] introduced the concept of Square Root SAM (Smoothing and Mapping), showing that solving the SLAM problem can be achieved by finding a square root of the data association problem. This work has significantly influenced subsequent research in graph-based SLAM. Suenderhauf et al. [30] developed Switchable Constraints for robust pose graph SLAM, which was an approach that handled erroneous loop closures in dynamic environments by introducing switch variables. More recently, deep learning has also been applied to graph optimization problems in SLAM. Li et al. [31] presented a learning-based framework that leveraged graph neural networks for the SLAM back-end optimization. Another interesting development in the field of graph-based SLAM optimization is the work developed by Carlone et al. [32] on the Lagrangian Duality in 3D SLAM. They proposed a novel approach for solving the SLAM problem in a decentralized manner, based on dual decomposition, which led to significant improvements in terms of computation time and scalability. Yang et al. [17] devised a new state estimation factor graph that integrates a feature-specific observation uncertainty model and the intrinsic prior of the object being tracked, which is solved by sliding window optimization. Rui et al. [33] introduced a graph optimization framework for urban scenarios with unstable GPS signals, featuring a direct point cloud registration system for fine environmental structures and an improved strong tracking extended Kalman filter (STEKF) to handle anomalous GPS data, achieving high accuracy and efficiency compared to existing methods. Chou and Chou [34] propose a tightly-coupled visual-lidar SLAM (TVL-SLAM), where independent visual and lidar frontends are integrated with all visual and lidar measurements in backend optimizations, featuring efficient lidar residual compression for large-scale bundle adjustments.

In conclusion, graph optimization continues to be a vital component in SLAM systems, with ongoing research continually improving the speed, scalability, and accuracy of these methods. This paper aims to bridge the gap in the literature by proposing a novel algorithm for indoor LiDAR SLAM that integrates total station measurements and graph optimization. The method is designed to address the challenges of drift and trajectory offsets in indoor environments and is particularly suited for low-cost indoor LiDAR devices.

III. Method

In advanced SLAM (Simultaneous Localization and Mapping) systems, integrating odometry and total station constraints enhances mapping accuracy and robustness. Odometry constraints, derived from onboard sensors like LiDAR or IMUs, ensure local consistency by providing continuous updates on the robot's movement, although they are prone to cumulative errors or drift over time. Conversely, total station constraints utilize precise measurements from a total station instrument to provide exact coordinates of specific points, serving as fixed references that help correct global errors and maintain the true scale of the map. The fusion of these data types within a graph optimization framework allows for the strategic correction of drift from odometry with pinpoint accuracy from total station measurements, balancing responsiveness with precision and achieving a robust, accurate mapping and navigation system. This integration ensures that SLAM systems maintain both local and global consistency, effectively managing the trade-offs between rapid sensor feedback and long-term measurement accuracy. The proposed method integrates total station measurements and graph optimization to optimize the robot's trajectory in indoor GNSS-denied environments. In this section, we describe the main steps of these developed algorithms in detail, including data preprocessing, scan registration, total station measurement integration, key point integration, graph optimization, and an alternative optimization approach without loop closure detection.

A. Data Preprocessing

Data preprocessing is an essential step to reduce noise, outliers, and computational complexity in the raw point cloud data. The preprocessing step consists of two main components: filtering and downsampling.

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

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

$$\mu_i = \frac{1}{k}\sum_{j=1}^{k} \text{dist}\left(p_i, p_{k_i,j}\right) \tag{1}$$

$$\sigma_i = \left|\frac{1}{k}\sum_{j=1}^{k} \text{dist}\left(p_i, p_{k_i,j}\right) - \mu_i\right| \tag{2}$$

where $\text{dist}\left(p_i, p_{k_i,j}\right)$ is the distance between points p_i and $p_{k_i,j}$.

If the distance from p_i to the mean distance μ_i is greater than a user-defined threshold T, which is determined as follows:

$$T = \alpha * \sigma_i \tag{3}$$

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

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

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

$$q_i = \frac{1}{|v_i|}\sum_{p\in v_i} p \tag{4}$$

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

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

B. Scan Registration

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

To be more specific, given two point clouds $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_m\}$, the objective of the ICP algorithm is to find the optimal transformation matrix T (composed of a rotation matrix R and a translation vector t) that minimizes the mean squared error between these two point clouds using the following equation:

$$E(T) = \frac{1}{n}\sum_{i=1}^{n} T\left(p_i\right) - q_{\text{i}}' \tag{5}$$

where $E(T)$ is the error function to be minimized. The variable T represents the transformation applied to the source point cloud. $T\left(p_i\right)$ represents the transformed point p_i and q_{i}' is the corresponding point in the point cloud Q.

C. Total Station Measurements

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

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

$$l_j = \arg\min_{l\in L} \|k_i - l\| \tag{6}$$

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

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

$$e_i = k_i - T\left(p_i\right) * l_j \tag{7}$$

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

D. Key Point Integration

To integrate the key points measured by the total station, we identify the corresponding points in the LiDAR point

cloud data. This is achieved by using a nearest-neighbor search algorithm, which finds the LiDAR points closest to the total station measurements. Once the corresponding points are identified, these key points are incorporated into the SLAM system as additional nodes in the proposed graph structures.

E. Graph Optimization

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

Let $G = (V, E)$ be a pose graph, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes representing the robot poses, and $E = \{e_1, e_2, \ldots, e_m\}$ is a set of edges representing the constraints between these poses. Each edge e_{ij} is associated with a transformation matrix T_{ij} and an information matrix Ω_{ij}, which represents the inverse of the covariance matrix of the constraint.

In this paper, we employ two types of constraints in the pose graph: odometry constraints and total station constraints. Odometry constraints are derived from the relative pose transformations T_i obtained from the scan registration process (see Section III-B), which connect consecutive poses and enforce the consistency between the estimated poses and the relative motion estimated from the scan registration. Total station constraints are derived from the total station measurements (see Section III-C), which connect the robot poses to the key points and enforce the consistency between the estimated poses and the absolute positions of these key points.

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

$$E(V) = \sum_{i,j \in E} e_{ij}^T * \Omega_{ij} * e_{ij} \tag{8}$$

where e_{ij} is the error between the connected poses v_i and v_j, given the transformation matrix T_{ij} and the information matrix Ω_{ij}.

We employ the Gauss-Newton algorithm to solve the graph optimization problem, which iteratively refines the robot poses by minimizing the total error in the pose graph. Algorithm 1 describes the Gauss-Newton algorithm in detail.

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

F. Alternative Optimization Without Loop Closure Detection

In SLAM, loop closure detection identifies whether the robot has returned to a location where it has previously visited. If a loop closure is detected, a constraint is added to the pose graph that connects the current robot pose with the previous pose at the same location, helping correct the accumulated drift in the robot's trajectory over time. In the absence of loop closure detection, the method of optimization becomes more reliant on accurate relative pose estimations. As such, alternative techniques need to be employed to minimize the drift in the trajectory. We detail the workflow to carry out the graph-based SLAM optimization without relying on loop closure detection as follows:

Algorithm 1 Gauss-Newton Optimization

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

Initial Pose Estimation: As always, the process begins with an initial pose estimation. This is done through initial scan-matching methods. These initial pose estimates will serve as the nodes in the pose graph.

Edge Creation: For every consecutive pair of poses, an edge is created in the graph. The transformation between the poses, along with the measure of uncertainty (captured by the covariance matrix), is stored in the edge.

Graph Optimization: Without loop closures, the graph remains a chain. Each edge in this chain graph represents a spatial constraint between two robot poses. The objective of the graph optimization step is to adjust these poses to minimize the error introduced by these constraints. The set of equations formed to represent the constraints is solved using Gauss-Newton.

Iterative Refinement: Due to the absence of loop closure detection, it becomes critical to frequently run the optimization process to reduce the accumulation of drift.

Error Propagation: In this method, after each graph optimization, the errors (or adjustments) are back-propagated through the graph to adjust previous pose estimates. This operation helps reduce the drift and keeps the trajectory estimation accurate.

By iteratively refining pose estimations and backpropagating these refinements back through the graph, the system can still maintain a reasonable estimate of the robot's trajectory, even without loop closure detection. However, it's important to note that the lack of loop closure detection makes the system more prone to long-term drift, especially in large or complex environments.

IV. Experimental Results

In order to evaluate the performance of the proposed LiDAR SLAM method in indoor GNSS-denied environments, a series

of experiments were conducted in different indoor scenarios. These experiments aim to test the accuracy, consistency, and robustness of the proposed method under various conditions, such as long corridors, complex underground parking lots, and indoor environments lacking 3D textures. As a criterion to check the integrity of the generated trajectory and the stitched 3D point cloud, two parts of data were used. The first was predefined movement trajectory checkpoints, which were on the trajectory of the moving platform. For each point passed, the moving platform stopped for a few seconds to record the position of that point. Secondly, two markers were set near the trajectory every 3 meters. These two parts of the marker points were recorded by both the point cloud data and the total station data for the calculation of the relative position of the meridian for comparison.

A. Data Collection Platform Setup

Given the unique characteristics and financial affordability of various sensors and devices, we believe that it is more reasonable to use commercial-grade LiDARs (e.g., Livox and Velodyne) compared with survey-grade sensors (e.g., RIEGL and Trimble) in indoor environments, especially the non-repetitive scanning LiDAR (i.e., Livox Horizon). Because of the variety and complexity of indoor scenarios, a single form of mobile device (e.g., LiDAR carts or backpacks) is not suitable for all scenarios. For example, a single LiDAR-based backpack will not work in relatively narrow scenes, such as underground walkways or office corridors lacking 3D textures, which should be replaced with a combination of LiDAR, camera, and IMU sensors mounted on a cart to collect the desired data smoothly. Thus, a modular, low-cost, and mobile platform that can integrate multiple sensors is more suitable for indoor scenarios than the commercial indoor/outdoor collection devices on the market today. The problems of sparse point cloud densities and small viewing angles of LiDARs like Livox Horizon can be solved by using multiple sensors and repeatedly scanning the same areas. Simultaneously, data upsampling and artificial intelligence algorithms can also be employed to increase the point cloud density for extensive applications.

On this basis, we propose a modular indoor scanning system with a multi-sensor combination, called Wat-MMP. Fig. 2 shows the modular multi-sensor system platform designed for collecting indoor data in this paper. This platform can add or remove sensors according to the changes of the carrier (e.g., cart, backpack, and handheld). This Wat-MMP system includes a non-repetitive scanning 32-line equivalent 3D LiDAR with a horizontal FoV of 81.7° and a vertical FoV of 25.1°. The point cloud $P(x, y, z)$ coordinate system acquired by the LiDAR is the same as the local coordinate system of the LiDAR itself.

For different scenarios in indoor environments, this Wat-MMP modular platform is extended in four different forms. Fig. 1(a) shows the structural design of the mobile platform, on which different modules can be switched as needed. Fig. 1(b) shows that in the backpack form, the LiDAR can be replaced from a nonrepetitive scanning 3D LiDAR to a 360° mechanically scanning 16-line LiDAR, and a 360° camera is

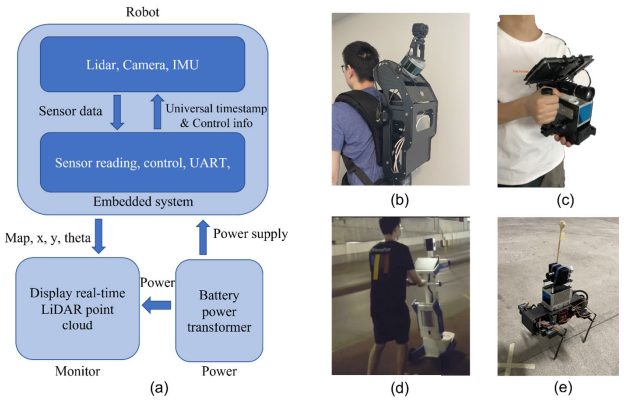


Fig. 1. Different Wat-MMP data collection platforms. (a) The architecture of the modular mobile platform, (b) Wat-MMP on a backpack device, (c) Wat-MMP on a handheld device, (d) Wat-MMP on a cart device, and (e) Wat-MMP on a robot dog device.

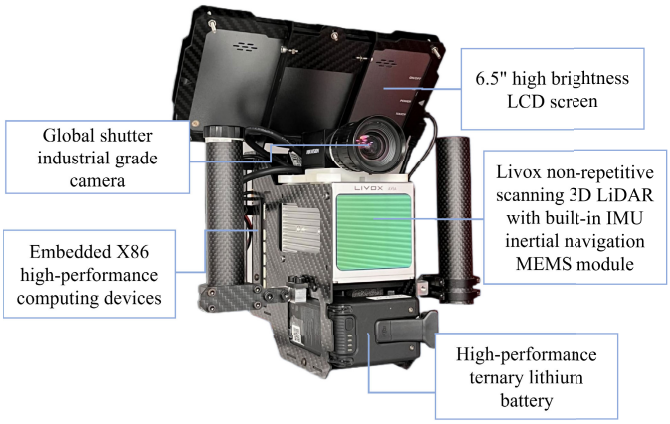


Fig. 2. Waterloo modular mobile platform (Wat-MMP) used in this paper for multi-sensor integration and data collection.

selected to replace the fisheye camera for environmental data collection in buildings, corridors, underground walkways, etc. Fig. 1(c) presents a handheld platform integrating a fisheye camera and a 360° mechanically scanning 16-line LiDAR. As shown in Fig. 1(d), a larger size battery is added and placed on a cart for data collection in large-scale indoor scenarios including airport parking lots, underground parking lots, and other large-scale indoor environments. Moreover, as shown in Fig. 1(e), adding a mechanical motion structure to place the module on a robot dog for data collection.

In this paper, we collected data from several scenarios, including a small building parking lot, a large complex underground parking lot, and a long building flyover, which can usually be used for the validation of indoor 3D reconstruction algorithms. The frame of signal data that had been synchronized was provided for each scene in the dataset. We used the LOAM algorithm [4] to generate large-scale 3D point cloud data as a comparison for 3D reconstruction. Moreover, three types of data were compared separately, which were raw point cloud data, trajectory data, and reference point data, respectively. As shown in Fig. 3, three parts of data samples show (a) the reduction of the complete point cloud by the LOAM algorithm, (b) the optimization of the generated

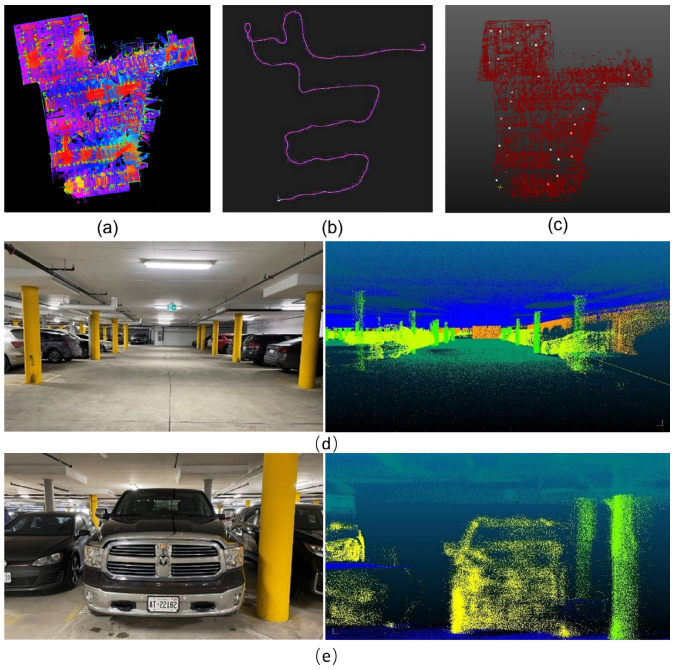


Fig. 3. (a) The dataset of the complete point cloud after large-scale stitching of an underground parking lot, (b) platform trajectory generated by SLAM algorithm, (c) the location of the manually set reference points on the frame extraction point cloud, (d) and (e) more diverse display of the collected point clouds in the scene.

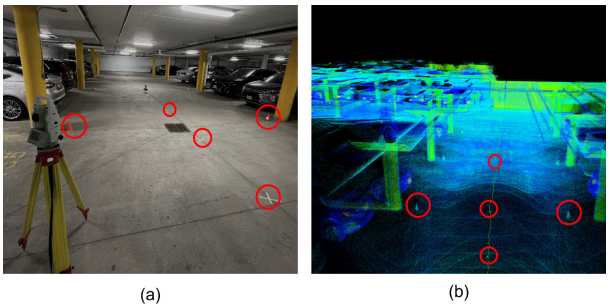


Fig. 4. Position schematic in our field survey. (a) The mobile platform moves along predetermined track marks in the underground parking lot. (b) In the point cloud data, the location of the predefined markers and the trajectory are obtained by *SLAM-TSM*.

trajectories, and (c) the drift degree and error of the reference points. Accordingly, multiple cones were placed in each indoor environment where data were collected to increase the number of observable reference points. Based on the collected data, the relative distances between all cones were checked using a total station for error analysis. Note that the labeled reference point in Fig. 3(c) is only a schematic representation of the reference point positions in the algorithm synthesis, and these positions cannot be used as the ground truth values of the relative positions of the reference points. Such ground truth points were manually obtained by the Leica MS50 total station in the field environment, with a relative measurement accuracy of 2 mm + 2ppm. Fig. 4 shows the 3D positions of the reference points and their relative position acquisition operations. Considering that this dataset can be added by the modular platform with new point cloud data, the local coordinate system of the dataset will be changed accordingly. Therefore, for the reference points on this basis, the relative distances between the point cloud points were chosen to replace the point positions in the local coordinate system as the criterion for judging the point cloud stitching. It should be noted that in order to better distinguish the parts to which the point clouds belong, the point clouds of different parts such as the ceiling, floor, columns, vehicles, and walls have been marked with different colors in Fig. 3 and Fig. 4. The color of the background in these figures is the color of the wall and ceiling as the background in different perspectives.

B. Performance in Small Building Parking Lot

In this experiment, the mobile platform was navigated in a small building parking lot, presenting unique challenges for SLAM algorithms due to the presence of multiple levels, ramps, occlusions, and poor lighting conditions. This data collection environment also includes numerous similar-looking parked cars, which can make feature extraction and scan registration more difficult. The proposed method was able to accurately estimate the platform's trajectory, even in the presence of occlusions, ramps, and repetitive structures. The integration of total station measurements allowed the proposed method to maintain accurate positional estimates despite the aforementioned challenging conditions. The graph optimization process helped to correct the trajectory offset and drift, ensuring a consistent and accurate trajectory estimate throughout the parking facility.

In the evaluation experiments, n points of interest for reference ($Q = [q_1, q_2, \ldots, q_n]$) were selected from the point clouds obtained by the proposed method and the LOAM algorithm, respectively. Also, the same position reference points obtained based on the total station measurements were $P = [p_1, p_2, \ldots p_n]$. We adopted the manual alignment of point pairs and calculated their distances between corresponding points. Based on the assumption of rigid invariance of trajectory and corresponding point clouds, this paper takes the distances of interest points on the point cloud data and the distances of reference points as the trajectory error. The specific calculation formula is as follows:

$$e_t(p_i, q_i) = \frac{\sum_{i=0, j=0, i<j}^{n} \left\| p_i - p_j \right\|^2}{\sum_{i=0, j=0, i<j}^{n} \left\| q_i - q_j \right\|^2} \tag{9}$$

Fig. 5 shows the interest point selection and reference point selection of the static reference data, and the compared experimental results between the proposed method and LOAM. The distances between the interest points and the reference points are also closer. Table I shows the results of error calculation (i.e.,err_{cp}) between the developed method (i.e., front-end odometer and back-end closed-loop optimization of the SLAM-TSM) and the LOAM method for map building using control points and measurement points. According to the error data analysis, it can be seen that the proposed method in this paper is more accurate than the LOAM method, especially for the parking lot scenes with high occlusion and obvious

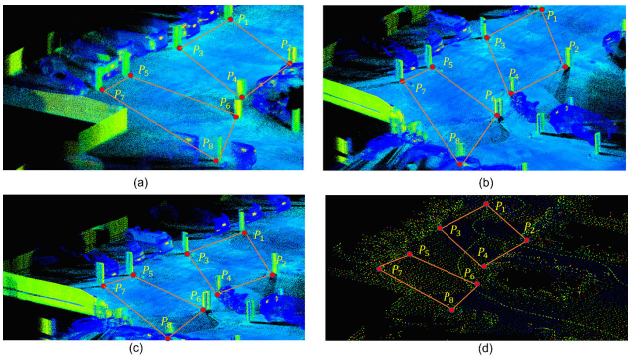


Fig. 5. Point cloud quality evaluation and analysis: error point selection and calculation on three different sets of algorithm results. (a) LOAM algorithm, (b) front-end results of our algorithm, (c) closed-loop detection and optimization results of our method, and (d) optimization results of total station using reference points, which are used here as comparison data for the real reference standard.

TABLE I
ERROR ANALYSIS OF THE SMALL PARKING LOT RESULTS OF OUR ALGORITHM AND LOAM ALGORITHM

Dataset	Type	LOAM	Our (only front-end)	Our (Optimized)
Small parking lot #1	err_{cp1}(m)	1.6643	0.1764	0.0572
	err_{cp2}(m)	2.8256	0.1648	0.0411
	err_rate	5.47%	0.49%	0.19%
Small parking lot #2	err_{cp1}(m)	0.8246	0.3284	0.1123
	err_{cp2}(m)	1.3273	0.1648	0.0856
	err_rate	2.71%	0.58%	0.32%

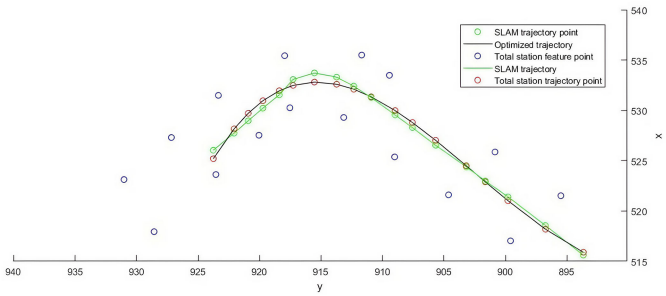


Fig. 6. Trajectory optimization of a mobile platform in a small building parking lot.

features. Meanwhile, Fig. 6 shows the trajectory comparison in the parking lot environment. It can be seen that the trajectory optimized by our proposed method nearly coincides with the feature points measured by the total station, while the optimization of the motion trajectory is more obvious. The key difference between SLAM trajectory points and the SLAM trajectory is the scope: one refers to individual poses, and the other to the entire path. The distinction between total station feature points and total station trajectory points lies in the application of total stations for static versus dynamic measurements, with the latter being less common. Finally, the difference between a SLAM trajectory point and an optimized trajectory point lies in the accuracy and refinement of the position data, with optimization processes enhancing trajectory accuracy by reducing errors.

TABLE II
ERROR ANALYSIS OF THE BUILDING FLYOVER RESULTS OF OUR ALGORITHM AND LOAM ALGORITHM

Dataset	Type	LOAM	Our (only front-end)	Our (Optimized)
Building flyover #1	err_{cp1}(m)	8.9918	0.0653	0.0294
	err_{cp2}(m)	8.9946	0.1102	0.065
	err_rate	27.03%	0.34%	0.24%
Building flyover #2	err_{cp1}(m)	2.0516	0.2944	0.0053
	err_{cp2}(m)	2.3076	0.1482	0.0297
	err_rate	7.65%	1.39%	0.09%

C. Performance in Long Building Flyover

The mobile platform traversed a long building overpass, which is a challenging environment for SLAM due to the lack of distinguishable features and potential drift accumulation. Fig. 7 shows the SLAM results obtained using different algorithms. The results show that the proposed method effectively corrects the drift and trajectory offsets by combining total station measurements with a graph optimization process.

Table II shows the error analysis of building flyover results, where the proposed method outperforms the LOAM algorithm. In particular, the proposed method achieved 0.09% localization accuracy (236 m travel distance and 0.17 m cumulative error) in the data of Building flyover #2. Fig. 7 shows the different 3D point cloud results of our method and the LOAM algorithm on the scanned data of the building flyover. It is obvious that the results of the LOAM algorithm are more blurred in the window structure reduction of type A. Moreover, the SLAM-TSM algorithm optimized by closed-loop detection outperforms the LOAM algorithm in the type B environment that has air conditioner chassis shading and is symmetric from left to right.

D. Performance in Large Underground Parking Lot

The mobile platform was navigated in a large underground parking lot, characterized by an extensive area with multiple levels, ramps, and a larger number of parked vehicles. Large-scale underground parking environments can be particularly challenging for SLAM algorithms due to the increased likelihood of drift accumulation and the presence of vast and repetitive structures that make feature extraction and scan registration more challenging. The experimental setup was similar to the previous experiments, with the robot following predefined trajectories throughout the parking facility. The LiDAR data and total station measurements were collected and processed in real time by the proposed SLAM-STM method. The estimated trajectories were compared to ground truth data obtained from an external motion capture system. The capture system consists of a white blob fixed above the mobile robot dog, a ground motion trajectory marker point, two markers set every three meters, and a total station for dynamic capture. During the movement of the Wat-MMP platform, the total station continuously captures the 3D spatial position of the top-placed ball as the robot dog passes each ground trajectory marker point. The data was recorded into the local coordinate system that has all marker points and ground track points.

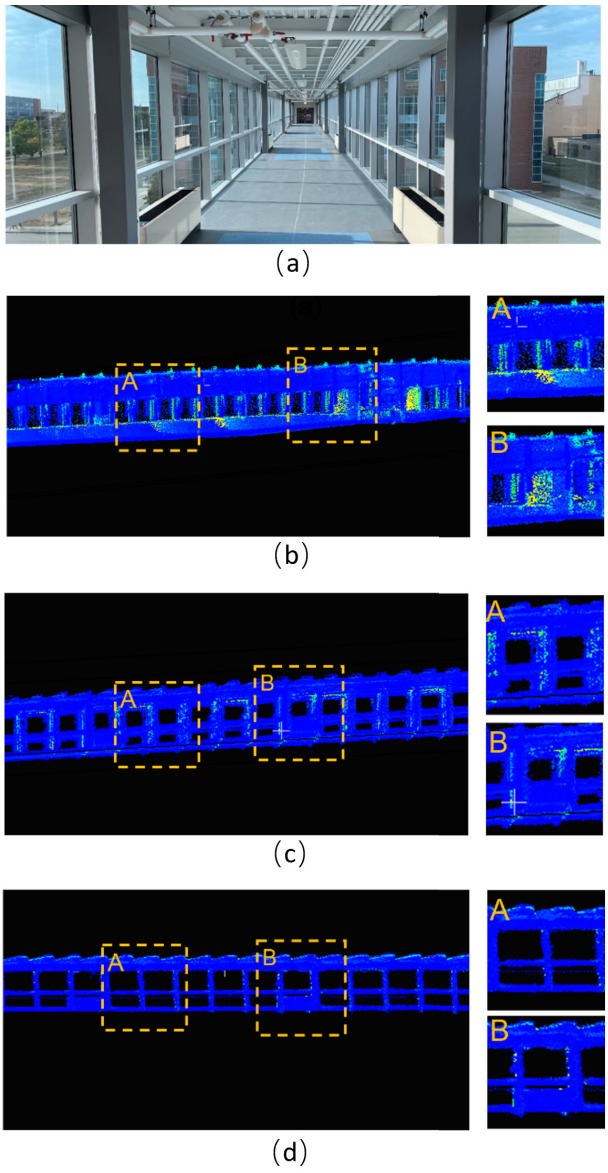


Fig. 7. Experimental results comparison from the building layover scan. (a) Images of the building flyover. (b) 3D point cloud obtained using LOAM algorithm. (c) Point cloud data using our algorithm front-end. (d) Point cloud data after optimization using our algorithm.

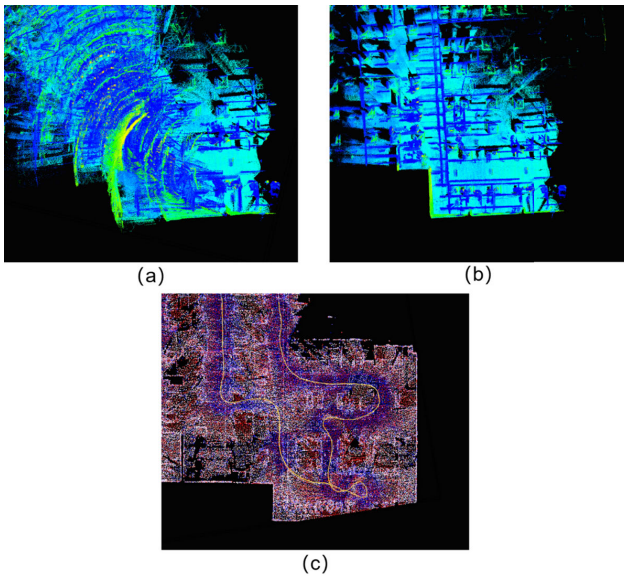


Fig. 8. Point cloud quality evaluation and analysis: error point selection and calculation on three different sets of algorithm results (a) LOAM algorithm, (b) front-end results of our algorithm in this paper, (c) closed-loop detection and optimization results of our algorithm in this paper.

TABLE III

ERROR ANALYSIS OF THE LARGE PARKING LOT RESULTS OF OUR ALGORITHM AND LOAM ALGORITHM

Dataset	Type	LOAM	Our (only front-end)	Our (Optimized)
Large parking lot #1	err_{cp1}(m)	4.0204	0.2461	0.0494
	err_{cp2}(m)	3.8976	0.1758	0.105
	err_rate	12.6%	0.82%	0.15%
Large parking lot #2	err_{cp1}(m)	1.0564	0.1056	0.0136
	err_{cp2}(m)	1.0804	0.0658	0.0156
	err_rate	3.11%	0.74%	0.12%

Fig. 8 shows that the proposed method can effectively address the challenges that occur in the largescale underground parking environments. By incorporating total station measurements and graph optimization techniques, the developed method was able to maintain accurate and consistent trajectory estimates. Even in areas with repetitive structures, such as rows of parked cars, our method can successfully correct the trajectory offset and drift, ensuring precise trajectory estimates. Table III shows the results of the error calculation. It is worth noting that since the underground parking lot environment is extremely challenging for the LOAM algorithm, the LOAM data used in the testing has drifted severely to the point where the markers and path trajectories cannot be discerned. Therefore, only the points that can be manually identified were compared in the experiment, so the actual LOAM results should be even worse than the results in Table III. In comparison to other SLAM methods, the proposed method demonstrated its superior performance in terms of accuracy and consistency, particularly in environments with largescale underground parking facilities. The ability to provide accurate trajectory estimates without relying on loop closure detection further highlights the suitability of the proposed method for environments where loop closures are scarce or difficult to detect.

V. CONCLUSION

In this paper, we present a LiDAR SLAM method for complex indoor environments, which incorporates graph optimization and total station measurements to improve the accuracy of the estimated trajectory without relying on loop closure detection. The model performance was demonstrated through a series of experiments in various indoor environments. In small compact parking lots, the proposed method has an error rate as low as 0.49% without loop closure detection and 0.19% with loop closure detection, respectively. These values are 0.34% and 0.09% in building flyover scenarios, and 0.74% and 0.12% in large parking lots, respectively. The results showed that the proposed method was capable of producing accurate and consistent trajectories, even in challenging conditions, such as long corridors, complex rooms, and environments with dynamic objects, making the

proposed method more suitable for data environments where loop closures are scarce or difficult to detect. Future work will focus on improving the scalability of the proposed method for large-scale environments, as well as exploring the integration of other sensors, such as cameras or inertial measurement units, to further enhance the accuracy and robustness of the SLAM system. Additionally, the development of efficient techniques for map representation and online map updating will be investigated to enable real-time applications in various domains, such as autonomous navigation, robotic exploration, and augmented reality.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[2] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[3] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[4] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. Conf.*, Jul. 2014, pp. 1–9.

[5] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 473–479.

[6] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1611–1626, Oct. 2013.

[7] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Winter 2010.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artificial Intell.*, Jul. 2002, pp. 593–598.

[9] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2O: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 3607–3613.

[10] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Fast and accurate SLAM with Rao–Blackwellized particle filters," *Robot. Auto. Syst.*, vol. 55, no. 1, pp. 30–38, Jan. 2007.

[11] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4372–4378.

[12] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, "A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6907–6921, Jul. 2022.

[13] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 5266–5272.

[14] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[15] J. Behley, A. Geiger, J. Stiller, and D. Cremers, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot. Sci. Syst. Conf. (RSS)*, 2018, pp. 59–68.

[16] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537.

[17] B. Yang, A. Dai, and S. Savarese, "Learning object bounding boxes for 3D instance segmentation on point clouds," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 6740–6749.

[18] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1027–1037, Oct. 2008.

[19] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.

[20] F. Lu and Milios, "Robot pose estimation in unknown environments by matching 2D range scans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 935–938.

[21] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spin-image-based normal distribution transform," in *Proc. IEEE Int. Conf. Robot. Autom (ICRA)*, Jun. 2009, pp. 4312–4319.

[22] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[23] S. Wang, J. Zhang, and X. Tan, "PDLC-LIO: A precise and direct SLAM system toward large-scale environments with loop closures," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 626–637, Jan. 2024.

[24] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2198–2204.

[25] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 199–208.

[26] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2010, pp. 22–29.

[27] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 2262–2269.

[28] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proc. Robot., Sci. Syst. III*, Jun. 2007, pp. 65–72.

[29] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[30] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 1879–1884.

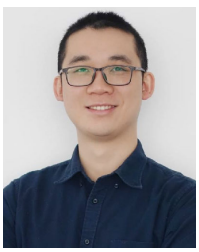
[31] Y. Li, Y. Ushiku, and T. Harada, "Pose graph optimization for unsupervised monocular visual odometry," 2019, *arXiv:1903.06315*.

[32] L. Carlone, R. Aragues, J. Castellanos, and B. Bona, "A linear approximation for graph-based simultaneous localization and mapping," in *Proc. Robotics: Sci. Syst. VII*, Jun. 2011, pp. 41–48.

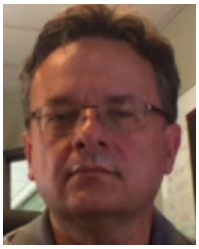
[33] Z. Rui, W. Xu, and Z. Feng, "FAST-LiDAR-SLAM: A robust and real-time factor graph for urban scenarios with unstable GPS signals," *IEEE Trans. Intell. Transp. Syst.*, early access, 2024, doi: 10.1109/TITS.2024.3455263. [Online]. Available: https://ieeexplore.ieee.org/document/10681669

[34] C.-C. Chou and C.-F. Chou, "Efficient and accurate tightly-coupled visual-LiDAR SLAM," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14509–14523, Sep. 2022.

Dedong Zhang (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and computer engineering from Baylor University, Waco, TX, USA, in 2016 and 2018, respectively, and the Ph.D. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 2023. He is currently a Post-Doctoral Researcher with the Department of Geography and Environmental Management, University of Waterloo. He specializes in optical, IMU, and LiDAR remote sensing, machine learning, artificial intelligence, and simultaneous localization and mapping. His current research interests include computer vision, localization and mapping, and multi-sensor fusion.



Weikai Tan received the B.Sc. degree from Nanjing University, China, in 2014, and the B.Sc. and M.Sc. degrees in geomatics from the University of Waterloo, Canada, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree in geography with the Department of Geography and Environmental Management, University of Waterloo. He specializes in optical, SAR, and LiDAR remote sensing, artificial intelligence, and environmental monitoring. His current research interests include urban 3D high-definition mapping and multi-sensor fusion.



John Zelek (Member, IEEE) received the Ph.D. degree from McGill University, QC, Canada, in 1996. He is currently an Associate Professor and the Co-Director of the Vision Image Processing (VIP) Laboratory, University of Waterloo, Waterloo, ON, Canada. He is formerly the Associate Graduate Chair of Systems Design Engineering from 2013 to 2017. His research interests include autonomous robotic mapping and localization, 3D scene understanding, man-made infrastructure assessment (e.g., roads, buildings, and bridges), eye (fundus and OCT) image understanding for disease, learning 3D models from single-views, and athletic sports tracking and biomechanical understanding of play and ability from video feeds.

Lingfei Ma (Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Waterloo, Waterloo, ON, Canada, in 2020. He is currently an Associate Professor with the Central University of Finance and Economics, Beijing, China. He has published more than 50 papers in refereed journals and conferences, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, ISPRS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE-CVPRW. His research interests include autonomous driving, mobile laser scanning, intelligent processing of point clouds, 3D scene modeling, and machine learning. He was a recipient of the 2020 National Best Ph.D. Thesis Award granted by Canadian Remote Sensing Society. He serves as an Editorial Board Member for *International Journal of Applied Earth Observation and Geoinformation*.

Jonathan Li (Fellow, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, South Africa, in 2000. He is currently a Professor of geomatics and systems design engineering with the University of Waterloo, Canada. He has co-authored almost 600 publications, more than 150 of which were published in top remote sensing journals, including *Remote Sensing of Environment*, *ISPRS Journal of Photogrammetry and Remote Sensing*, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and *International Journal of Applied Earth Observation and Geoinformation*. He has also published papers in flagship conferences in computer vision and AI, including CVPR, AAAI, and IJCAI. He has supervised nearly 200 master's/Ph.D. students and post-doctoral fellows/visiting scholars to completion. His research interests include AI-based information extraction from Earth observation images and LiDAR point clouds, pointgrammetry and remote sensing, GeoAI, and 3D vision for digital twin cities and autonomous driving. He is a fellow of Canadian Academy of Engineering, the Royal Society of Canada (Academy of Science), and the Engineering Institute of Canada. He is the President of Canadian Institute of Geomatics (CIG). He is the Editor-in-Chief of *International Journal of Applied Earth Observation and Geoinformation* and an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.