

# GBA-LIOM: Global Bundle Adjustment Enhanced LiDAR–Inertial Odometry and Mapping

Shangjia Liu<sup>ID</sup>, Jianzhu Huai<sup>ID</sup>, Member, IEEE, Binliang Wang<sup>ID</sup>, Yulong Han<sup>ID</sup>,  
and Yuan Zhuang<sup>ID</sup>, Senior Member, IEEE

**Abstract**—An accurate and consistent LiDAR point cloud map has a wide range of applications, such as in building information modeling, online retailing, and autonomous vehicles. However, existing methods face various drawbacks and challenges. First, odometry divergence often occurs in degraded, narrow corridors due to the lack of features. In addition, large-scale global optimization with LiDAR point clouds remains challenging because of memory limitations and the complex derivations involved in the process. Toward removing these barriers, this article proposes a consistent and efficient large-scale LiDAR odometry and mapping method. First, we use distribution-to-multidistribution constraints in odometry to deal with challenging scenes. Second, we propose a novel storage structure for LiDAR features, significantly reducing the excessive memory consumption of existing optimization methods. Moreover, we reformulate the error term in the optimization problem, enabling the use of generic least squares solvers and eliminating the need for complex derivative derivations. These improvements result in a substantial acceleration of LiDAR bundle adjustment (BA). Third, the optimization focuses on minimizing the roughness of surfels within windows of selected adjacent LiDAR frames. By assigning frames around the loop to a window, our method naturally integrates loop closure into the optimization. This allows our method to effectively incorporate loop constraints in BA. Finally, the proposed method is validated in terms of memory efficiency, computational speed, mapping consistency, and accuracy using a public benchmark and our own datasets.

**Index Terms**—Consistent LiDAR mapping, LiDAR bundle adjustment (BA), memory efficiency, optimization, SLAM.

## I. INTRODUCTION

HOW to achieve a globally consistent 3-D map has attracted increasing research attention in the field of robot vision in recent years [1]. From a practical perspective,

Received 18 December 2024; revised 10 April 2025; accepted 29 April 2025. Date of publication 26 May 2025; date of current version 5 June 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 42374047, in part by the Key Research and Development Program of Hubei Province through the International Scientific and Technological Cooperation Program under Grant 2023EHA036, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022B1515120067, and in part by the Wuhan Artificial Intelligence (AI) Innovation Program under Grant 2023010402040011. The Associate Editor coordinating the review process was Dr. Yan Zhuang. (*Corresponding author: Yuan Zhuang*.)

Shangjia Liu, Jianzhu Huai, Binliang Wang, and Yulong Han are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China.

Yuan Zhuang is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China, also with Wuhan Institute of Quantum Technology, Wuhan 430206, China, and also with Wuhan University Shenzhen Research Institute, Shenzhen 518057, China (e-mail: zhuangyuan@whu.edu.cn).

Digital Object Identifier 10.1109/TIM.2025.3571159

a well-constructed, high-precision 3-D map is essential for applications such as robot navigation [2], [3], [4], autonomous driving [5], and UAV navigation [6], [7], [8], [9]. Such maps are critical for various instrumentation and measurement applications, especially in the context of developing high-performance measurement systems that require accurate spatial and environmental data. As the cost of LiDAR sensors decreases [10], [11] and their scanning frequency increases relative to traditional 3-D laser scanners [12], LiDAR technology is increasingly used for building accurate and consistent 3-D maps. In this article, we focus on developing an efficient method for constructing a consistent 3-D LiDAR point cloud map, aimed at advancing measurement methodologies and supporting the design and evaluation of instrumentation systems used for monitoring and recording in dynamic environments.

Rich LiDAR SLAM algorithms have been proposed to solve the problem of LiDAR mapping, such as FAST-LIO2 [13], lio-sam [14], MULLS [15], and SuIn-LIO [16]. While these algorithms can effectively build reliable local maps in real time, they often suffer from cumulative drift in large environments such as outdoors or underground. In some degraded scenes such as stairwells, algorithm sometimes even fails. To reduce motion estimation drift, both pose graph optimization (PGO) and bundle adjustment (BA) are commonly used methods. However, PGO is generally preferred due to its higher efficiency. Taking the adjacent frame poses as constraints, PGO requires proper weighting and cannot deal with hard cases with only pose constraints, e.g., sudden drift.

To compensate the shortcomings of LiDAR SLAM and PGO methods, numerous LiDAR mapping algorithms [12], [17], [18], [19] have been proposed as they simultaneously optimize the scan pose and the geometry of the scene. Unlike visual BA, LiDAR sensors capture a significantly larger number of feature points, posing substantial challenges for memory and computational efficiency. In [17], the plane parameters are first solved analytically so that the final optimization problem is only related to the pose of the LiDAR. In [19], the plane parameters are eliminated in each iteration of the optimization to achieve real-time performance. Considering both mapping consistency and time efficiency, [12] design a pyramid structure combining BA and PGO. Although the existing LiDAR adjustment methods greatly improve the computational efficiency in the optimization process, there are still three major challenges. First, associating a vast number of LiDAR feature points is a challenge for storage. For example,

the open-source algorithm [18] suffers from memory explosion in large scenario datasets. Second, the existing BA methods struggle to effectively correct drift errors. Third, the deduction of Jacobians and even Hessians, e.g., [18], for point-cloud-based optimization can be formidable and error-prone.

In response to the challenges mentioned above, we propose a global BA enhanced SLAM system that achieves high-precision LiDAR localization and mapping in large scenarios. First, to address failure of classical LiDAR odometry methods in degraded scenes, we introduce a new distribution-to-multidistribution loss function in the matching process from odometry frames to submaps, significantly enhancing the robustness of the odometry in degraded scenes. Second, by introducing an efficient representation, we show that we can perform global optimization for LiDAR with a lower memory footprint and a higher consistency. We carefully design the data structure for LiDAR feature points to greatly improve the storage efficiency and specially formulate the optimization problem to make it amenable to generic least-squares solvers. In addition, we add place recognition in the feature association process, thereby proving that our adjustment method reduces drift better than state-of-the-art (SOTA) BA and PGO methods. Numerous experiments have been conducted on the proposed innovations, and the effectiveness of the method has been validated on both the real-world datasets and simulation experiments. In summary, our contributions are as follows.

- 1) We designed a LiDAR front-end tailored for degraded scenes and adaptively adjusted key parameters. In addition, we introduced a new distribution-to-multidistribution loss in frame-to-submap matching, effectively resolving tracking failures of LiDAR odometry in such environments.
- 2) Significant improvements in both memory efficiency and derivative speed for LiDAR BA. By adaptively extracting point cloud features and storing only the frame information relevant to the LiDAR features during the feature association stage, we effectively resolve the memory challenges that SOTA methods face. In addition, by redefining the cost terms of the optimization problem and using automatic differentiation, we eliminate the need for complex analytic derivation of the Hessian matrix, leading to substantial improvements in BA solving speed.
- 3) Feature association is refined through loop closure. To our knowledge, our method is the first to incorporate loop closure information directly into global LiDAR BA rather than PGO. This approach effectively corrects multisensor drift errors while maintaining a high level of consistency in the point cloud map. We merge window frames forming a loop closure and extract features together with the other window frames, using plane features as constraints in the global optimization process.

## II. RELATED WORK

### A. LiDAR(-Inertial) Odometry

LiDAR(-inertial) odometry is typically classified into two categories: direct methods and indirect methods. Direct

methods use the entire point cloud from the scan for pose tracking and estimation, while indirect methods track features extracted from the point cloud. Regarding indirect methods, LOAM [20] introduced a feature-based plane smoothing approach. Later works like Lego-LOAM [21] leverage ground information to enhance the robustness and efficiency of pose estimation, LiLi-OM [22] proposed a tightly integrated LiDAR-inertial odometry (LIO) solution with hierarchical LiDAR/IMU keyframe-based sliding window optimization in the back-end.

Early direct methods, such as ICP [23] and GICP [24], estimate the relative pose between two consecutive frames. However, relying solely on two frames makes it difficult to suppress odometry drift. To address this issue, Xu et al. [13] proposed FAST-LIO2, tightly coupled iterative Kalman filter-based method that matches frames to a local submap, enhancing global consistency. Building on this, FASTER-LIO [25] introduced an incremental voxel-based approach to accelerate the process.

To tackle the problem of degraded environments, Adalio [26] extended FASTER-LIO with an adaptive parameter adjustment strategy to overcome such scenarios. Similarly, Locus2 [27] proposed an adaptive voxel grid approach to improve robustness in degraded environments. DLIO [28] proposed a lightweight LIO system with continuous motion correction, specifically designed for degraded environments. Despite these advancements, failures in long-distance underground degraded scenes remain inevitable.

### B. Pose Graph Optimization

PGO formulates a nonlinear optimization problem by incorporating relative transformations between frames as pose constraints, with error terms weighted by the information matrix, and solves it using a nonlinear optimization method such as the LM algorithm. In [29], [30], and [15], loop detection and PGO are conducted for global consistency. In [31], a factor graph has been proposed that minimizes the registration error and the IMU constraints, ensuring robust optimization in feature-less environments. However, those pose graph methods still cannot handle outliers very well. Yang et al. [32] proposed a registration method that allows drastic outliers, but it is hard to extend to globally PGO. In summary, these PGO methods have some disadvantages. First, PGO struggles to obtain a globally consistent point cloud by just optimizing the frame poses. Second, it requires weight tuning and struggles to handle outliers.

### C. LiDAR BA

LiDAR BA has recently become a research trend as it compensates for the shortcomings of PGO. The biggest challenge faced by LiDAR optimization is that it is computationally intensive. For example, the correlation (or entropy)-based scan registration in [33] is pairwise available, but very hard to scale up and cannot extend to multiple scans easily due to computational limitations. Zhou et al. [34] and Kaess [35] propose plane adjustment, which optimizes only the planar feature points perceived by LiDAR, greatly reducing the

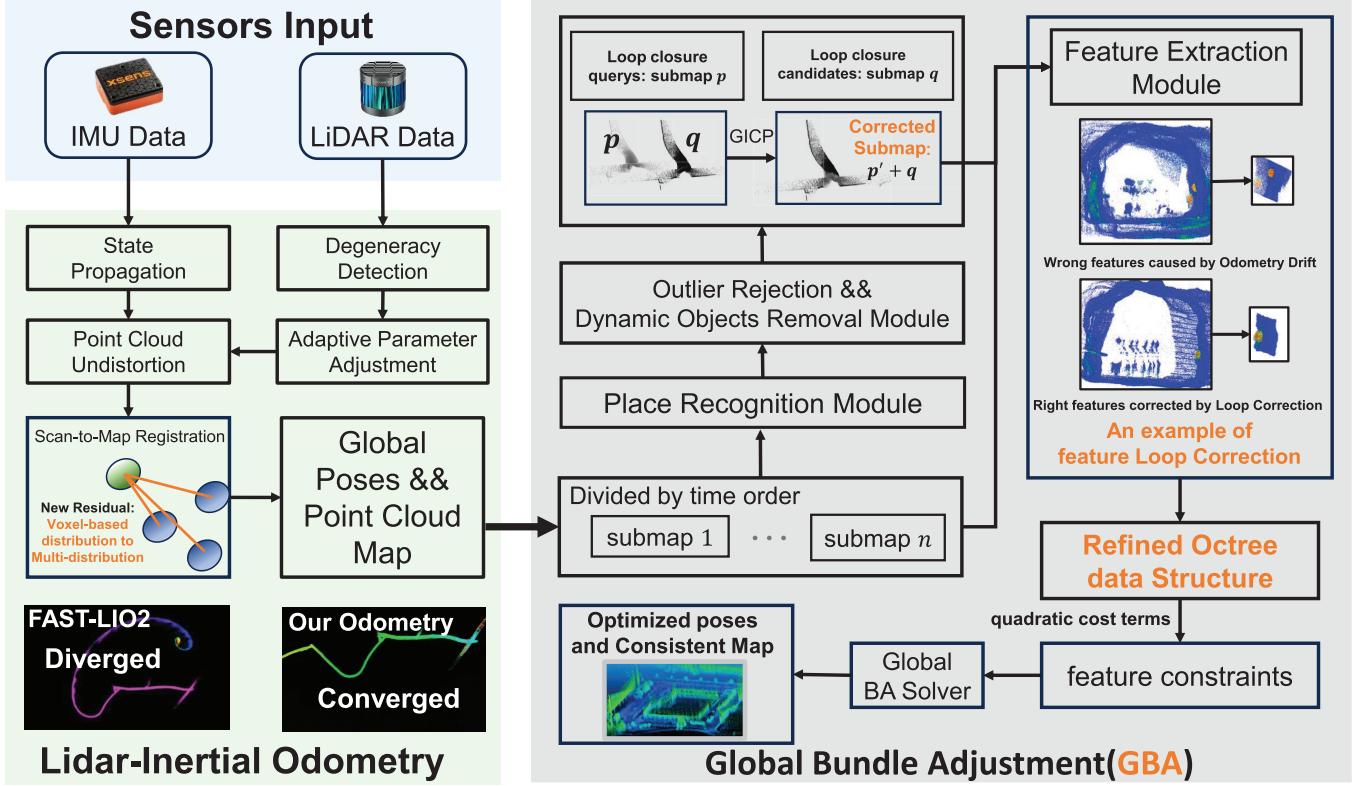


Fig. 1. GBA-LIOM overview. The innovations are highlighted with red fonts in the picture.

computation effort. In [36], eigenfactor was formulated to make the optimization independent of the number of points to accelerate the optimization. Zhou et al. [37] present  $\pi$ -factor to significantly reduce the computational complexity of BA. To further improve the computational efficiency, [17] show that plane feature can be analytically solved and removed from BA. Liu et al. [18] develop an efficient second-order BA solver and avoid enumeration of each raw point in all the steps of optimization. But the second-order solver needs complex Jacobians and Hessians. The above methods have great improvement in enhancing the computational efficiency, but do not well solve the problem of excessive memory usage for feature association. Then, Liu et al. [12] proposed a compromise solution by developing a pyramid structure to combine LiDAR BA and PGO, greatly improving computational and memory efficiency. However, the essential restriction of [12] is still using constraints of adjacent frames' pose, and although LiDAR BA results of the layer-by-layer submap provide the weight matrix of PGO, it still fails to optimize the point cloud directly, and the loop constraints may have negligible effect on the point clouds.

### III. METHODOLOGY

#### A. Overview

This article proposes a global BA enhanced LiDAR odometry and mapping method. The inputs to the algorithm are LiDAR and IMU data, and the output is the optimized poses and consistent map. Our method can be divided into two parts: LIO and global BA process. The workflow of our proposed method is shown in Fig. 1.

Our custom LIO is designed to address degradation in long-range scenarios typical in degraded environments. By incorporating a degradation detection module and introducing a novel residual constraint in the scan-to-map matching module, our front-end effectively handles situations where classical odometry methods struggle.

In the global BA process, we perform keyframe selection from the raw data, and then group the initial keyframes into several submaps, ensuring a fixed overlap between each adjacent submap. We input these submaps to extract plane features of the point cloud in the global frame. If the place recognition module finds two submaps  $u$  and  $v$  are loop closure candidates and pass through an outlier rejection module, we would compute the relative pose between these two windows through the GICP algorithm [24], unify these two windows' point clouds into a global coordinate system, and then add them to the feature extraction module, and thus the feature association process would extract correct features and give relatively good initial poses to the optimization process. And then, all the feature constraints extracted from the feature association process is stored in our refined octree data structure. Then we made our cost terms quadratic and feed them to the least-squares BA solver, along with the weight of each feature to get optimized poses and consistent point cloud map.

#### B. LiDAR-Inertial Front-End

In this section, we introduce a new scan-map registration constraint for LIO. We observe that in narrow degraded environments, the SOTA LIO methods, such as Fast-LIO2,

TABLE I  
PARAMETER SETTINGS FOR DIFFERENT CASES

Parameters	In general cases	In degeneracy cases
<i>down_sample_leaf_size</i>	0.5	<b>0.25</b>
<i>blind</i>	1.0	<b>0.3</b>
<i>plane_threshold</i>	0.1	<b>0.5</b>
<i>ICP_threshold</i>	0.8	<b>0.6</b>

Faster-LIO, and DLIO, often fail to converge. Therefore, we first perform degradation detection of the environment and adaptively adjust the parameters when encountering degraded scenarios. In addition, we note that normal vector estimation is required for surfel-based registration, but this estimation is often inaccurate in degraded scenarios. Consequently, we introduce a new constraint, the distribution-to-multidistribution distance constraint. Experiments demonstrate that this new constraint significantly improves registration performance in degraded scenarios compared with the traditional point-to-plane distance constraint method.

1) *Adaptive Parameter Setting Strategy*: In narrow and degraded long-distance underground scenarios, we first use a degradation detection module to identify when the number of valid LiDAR points is insufficient for scan-to-submap matching. Upon detecting degradation, we use an adaptive parameter setting strategy. To address the limitations of using a fixed voxel size for downsampling feature points in each scan, we introduce the parameter *down\_sample\_leaf\_size*, which dynamically adjusts the leaf size for voxel-based downsampling. In narrow and degenerate environments, the number of LiDAR points available for matching is significantly lower than in open spaces. To mitigate this issue, we detect degenerate conditions based on two criteria: 1) most points in the current scan are concentrated near the LiDAR sensor origin and 2) the total number of raw points in the scan is relatively low. When such conditions are met, *down\_sample\_leaf\_size* is reduced to retain more feature points and improve robustness in degenerate scenarios.

In addition, we adjust the *blind* parameter, which specifies a radius around the LiDAR sensor within which points are discarded. In degenerate scenarios, *blind* is reduced to avoid removing useful feature points. Both *blind* and *down\_sample\_leaf\_size* are adaptively adjusted to enhance feature preservation in scan-to-submap matching. Furthermore, in the scan-to-submap registration stage, thresholds for planar determination and ICP optimization are also adapted based on the degenerate environment. The specific parameter adjustment strategy is summarized in Table I.

This overall approach constitutes our adaptive parameter setting strategy, where we first assess whether the current frame is degenerate and then apply an empirically designed set of parameters accordingly—one set for normal environments and another for degenerate scenarios.

2) *Scan-to-Submap Registration*: In scan-to-map matching, we adaptively set the ICP threshold and the plane determination threshold, as more map points are needed for matching in

narrow spaces [26], [27]. It is also worth noting that in scan-to-map registration, the normal vectors are typically derived from partial wall segments located in close proximity, thereby limiting the diversity of geometric correspondences. Therefore, constructing the distance constraints from scan points to submap planes often introduces significant errors. To address this issue, we adopted the distribution-to-multidistribution residual similar to the VGICP [38] algorithm.

We use two types of scan-to-submap matching residuals: point-to-point residuals for normal, nondegraded scenarios, and distribution-to-multidistribution residuals for degraded scenarios. Denote  $A = \{p_i, i = 1, \dots, m\}$  as the points in the current scan. For each LiDAR point  $p_j$ , we use the  $K$ -nearest neighbors (KNN) algorithm to find the five nearest points  $B = \{p_r, r = 1, \dots, m\}$  in the submap, forming a Gauss distribution  $p_r \sim \mathcal{N}(\hat{p}_r, C_{p_r})$ . The point-to-plane residual is described using the plane and normal vector that define this surface, while the distribution-to-multidistribution residual is described using a Gauss distribution to characterize the surface.

- 1) *Point-to-Plane Residual (No Degradation)*: When the neighboring points can be accurately represented by a plane, we use a point-to-plane residual to strengthen the constraint along the normal direction of the local region. This point-to-plane residual is defined as follows:

$$\mathbf{r}_{\text{surface}} = (p_i - p_r)^T \mathbf{n} \quad (1)$$

where  $\mathbf{n}$  is the normal vector of the surface.

- 2) *Distribution-to-Multidistribution Residual (Degradation Detected)*: When the neighboring points do not form a well-defined plane, but their density is sufficient to suggest a meaningful local surface, we use a voxel-based distribution-to-multidistribution residual, defined as follows. Assume there are  $j$  Gaussian distributions in a voxel. The sum of the errors from all the distributions is defined as  $\hat{d}_i N_i$  is the number of neighbor points. The final residual, incorporating the weights of the Gaussian distributions, is defined as  $\mathbf{r}_{\text{dis}}$  as follows:

$$\hat{d}_i = \frac{\sum_j \hat{p}_i}{N_i} - \hat{p}_r \quad (2)$$

$$\mathbf{r}_{\text{dis}} = \sum_i (N_i \hat{d}_i^T C_{p_r}^{-1} \hat{d}_i). \quad (3)$$

### C. Feature Association

1) *Keyframe Selection and Partition*: Note that the raw observations could have a lot of redundant information, it was necessary to perform keyframe selection on the raw data. We set a frequency  $f$  to temporally downsample all the frames of the raw data. And then we calculate threshold based on the angular velocity  $w$  and velocity  $v$  of all the frames, and once the angular or linear velocity of the current frame is larger than the threshold, it is added as a keyframe. Then we group the keyframes into fixed sized windows for feature extraction.

2) *Feature Extraction Module*: For feature extraction, we propose a novel voxelization method, to significantly improve the memory performance of octree structure for storing feature points.

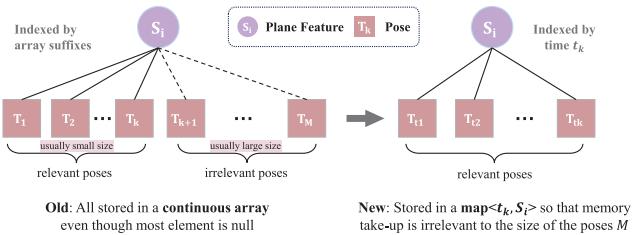


Fig. 2. Memory modification.

Suppose we now have a set of raw point clouds  $S = \{P_i \mid i = 1, \dots, M\}$  from  $M$  scans, and each scan has a point cloud  $P = \{\mathbf{p}_j \mid j = 1, \dots, N\}$ . Denote the poses of the  $M$  scans as  $\mathbf{T} = (\mathbf{T}_1, \dots, \mathbf{T}_M)$ , where  $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{t}_i) \in SE(3) \times \mathbb{R}^3$ , and  $j \in \{1, \dots, N\}$ . Then we can transform each point  $\mathbf{p}_j$  in the raw point clouds to the same global frame. The global point  $\mathbf{p}_j^G$  in the  $i$ th scan is computed as

$$\mathbf{p}_j^G = \mathbf{R}_i \mathbf{p}_j + \mathbf{t}_i; \quad j = 1, \dots, N. \quad (4)$$

Initial segmentation of all the points is performed with a voxel grid, which are stored in an octree structure, and then each voxel is iteratively segmented, with planarity check performed for each segmentation, and the iteration terminated by finding a planar feature or by reaching the maximum number of iterations. As for the criteria of planarity check, we first use principal component analysis (PCA) [39] to calculate eigenvalues ( $\lambda_1 \leq \lambda_2 \leq \lambda_3$ ) from points' covariance matrix, and a planar feature is found if  $(\lambda_1/\lambda_2)$  is less than a threshold  $\theta$ . As the depth of the iteration increases, we gradually set the threshold  $\theta$  more strictly, because the false-positive rate is higher for small planes compared with large ones. By adaptively adjusting the threshold for planar judgment, we are able to obtain more accurate features

$$\lambda_1 \leq \lambda_2 \leq \lambda_3, \quad \frac{\lambda_1}{\lambda_2} < \theta. \quad (5)$$

As shown in Fig. 2, noting that the number of scans associated with each extracted feature is limited, the innovation of our approach is to store only the feature points and its observing frames (normally no more than ten scans). This greatly reduces the burden on the program's memory comparing to the implementation of the start-of-art algorithm [17], [18], which for a single plane feature, storing all the scan's data even though most of the scans are irrelevant to this feature. Thus, the improvement on the memory allows the LiDAR feature extraction process to support a large number of point cloud frames.

3) *Place Recognition Module and Outlier Rejection Module:* We use PointNetVlad [40] as the place recognition module to generate descriptors for several frames of each window and then find the loop closure pairs using KNN search. The choice of  $K$  in KNN search directly determines how many loop closure candidates are selected for further verification. A smaller  $K$  value tends to increase precision by reducing the number of false positives, but it may significantly lower recall, potentially missing valid loop closures—especially in challenging or ambiguous environments. In our experiments,

we set  $K = 10$ , which provides a good balance between recall and precision. After detecting a loop closure, we eliminate false positives by pairwise consistent measurement (PCM) [41] set maximization algorithm. First, we construct an adjacency matrix between poses. Then, based on the PCM algorithm, we compute the maximum clique of the adjacency matrix to find a set of mutually consistent loop closures, which effectively eliminates the majority of false loop closures. And then we incorporated the Huber robust kernel into the optimization function to mitigate the effects of the remaining few false loop closures. Next, the ICP algorithm is used to compute the relative transformation between the loop closure windows and register them to the global coordinate system for feature extraction. In this way, the feature association process would give relatively good initial poses to the optimization process.

4) *Dynamic Object Removal Module:* Dynamic objects can have a significant impact on mapping accuracy, particularly when their motion introduces noise into the point cloud data. To mitigate this, we have introduced a dynamic object removal module (M\\_detector) [42] in the data preprocessing stage, which is applied before the odometry phase and the subsequent postprocessing.

#### D. Global BA Optimization

In this section, we formulate the LiDAR mapping optimization problem. First, we formulate the map adjustment problem as reducing roughness on all the small plane features and show that the feature parameters can be eliminated from the formulated optimization. Then, we analyze defined residual terms could lead to easy implementation of the optimization algorithm. Finally, we present our linear solver based on the LM optimization method.

1) *BA Formulation:* Assume there are  $W_f$  features, each denoted by feature  $\mathbf{f}_i (i = 1, \dots, W_f)$ , observed by  $W_p$  LiDAR scans, there pose each expressed as  $\mathbf{T}_j = (\mathbf{R}_j, \mathbf{t}_j) (j = 1, \dots, W_p)$ . A traditional BA problem simultaneously optimize all the feature parameters  $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_{W_f})$  and all the LiDAR scan poses  $\mathbf{T} = (\mathbf{T}_1, \dots, \mathbf{T}_{W_p})$ , to make the produced global map more consistent with all the LiDAR observations. We denote  $m(\mathbf{f}_i, \mathbf{T})$  as the sum of squared distance between the  $i$ th feature extracted from LiDAR measurements  $\mathbf{f}_i$  and all the involved feature points (controlled by the pose of their scan  $\mathbf{T}$ ). Then a direct BA formulation is

$$\min_{\mathbf{T}, \mathbf{f}} \left( \sum_{i=1}^{W_f} m(\mathbf{f}_i, \mathbf{T}) \right). \quad (6)$$

Suppose through the feature extraction module, we get feature point set  $P_{f_{ij}} = (\mathbf{p}_{f_{ijk}}, k = 1, \dots, N_{ij})$  that has  $N_{ij}$  feature points extracted on the  $i$ th feature at the  $j$ th LiDAR pose, each feature point denoted by  $\mathbf{p}_{f_{ijk}}$ . And then the predicted coordinate  $\mathbf{p}_{f_{ijk}}^G$  in the global frame is

$$\mathbf{p}_{f_{ijk}}^G = \mathbf{R}_j \mathbf{p}_{f_{ijk}} + \mathbf{t}_j. \quad (7)$$

For a plane feature, denote  $\mathbf{n}_i$  as the plane normal vector and  $\mathbf{o}_i$  as a point on the plane, then it is parameterized by  $\mathbf{f}_i = (\mathbf{n}_i, \mathbf{o}_i)$ . Then the Euclidean distance between the extracted global feature point  $\mathbf{p}_{f_{ijk}}$  and the plane is  $\|\mathbf{n}_i^T (\mathbf{p}_{f_{ijk}} - \mathbf{o}_i)\|^2$ , as

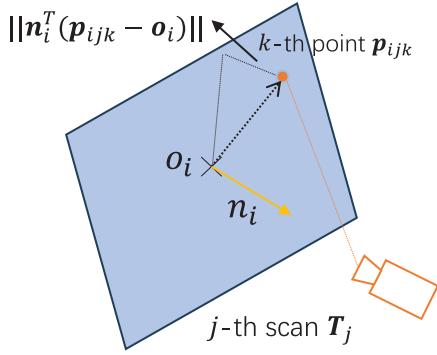


Fig. 3.  $i$ th plane feature  $\mathbf{f}_i = (\mathbf{n}_i, \mathbf{o}_i)$ .

shown in Fig. 3. So  $m(\mathbf{f}_i, \mathbf{T})$  for the  $i$ th plane feature could be computed as

$$m(\mathbf{f}_i, \mathbf{T}) = \frac{1}{N_i} \sum_{j=1}^{W_p} \sum_{k=1}^{N_{ij}} \|\mathbf{n}_i^T(\mathbf{p}_{ijk} - \mathbf{o}_i)\|^2 \quad (8)$$

where  $N_i$  is the total number of feature points for all the LiDAR scans. Feature parameter  $\mathbf{f}$  can be analytically solved before the optimization process [17]. So the direct BA formulation in (6) reduces to

$$\min_{\mathbf{T}, \mathbf{f}} \left( \sum_{i=1}^{W_f} m(\mathbf{f}_i, \mathbf{T}) \right) = \min_{\mathbf{T}} \left( \sum_{i=1}^{W_f} \min_{\mathbf{f}_i} m(\mathbf{f}_i, \mathbf{T}) \right) \quad (9)$$

where  $\lambda_l(\mathbf{A}_i)$  denotes the  $l$ th largest eigenvalue of matrix  $\mathbf{A}_i$ , its

$$\mathbf{A}_i \triangleq \frac{1}{N_i} \sum_{j=1}^{W_p} \sum_{k=1}^{N_{ij}} (\mathbf{p}_{ijk} - \bar{\mathbf{p}}_i), \quad \bar{\mathbf{p}}_i \triangleq \frac{1}{N_i} \sum_{j=1}^{W_p} \sum_{k=1}^{N_{ij}} \mathbf{p}_{ijk}. \quad (10)$$

Here,  $\mathbf{u}_l(\mathbf{A}_i)$  denotes the corresponding eigenvector. We perform singular value decomposition (SVD) of  $\mathbf{A}_i$ , and note that (9) holds when  $\mathbf{n}_i^* = \mathbf{u}_3(\mathbf{A}_i)$ ,  $\mathbf{o}_i^* = \bar{\mathbf{p}}_i$ .

2) *Least-Squares Solver:* We introduce cost function  $C(\mathbf{T})$  which minimizes the sum of the distances of all the original observed feature points to the plane features, as in (11). Since  $\mathbf{f}_i$  has been eliminated before the optimization, we omit  $\mathbf{f}_i$  in the following deductions:

$$C(\mathbf{T}) = \sum_{i=1}^{W_f} m(\mathbf{f}_i, \mathbf{T}). \quad (11)$$

Since  $C(\mathbf{T})$  is a nonlinear function of  $\mathbf{T}$ , solving the problem directly with nonlinear optimization algorithms such as the Levenberg-Marquardt (LM) algorithm requires a second-order Taylor expansion of  $\mathbf{T}$ , with the incremental update of  $\Delta\mathbf{T}$

$$C(\mathbf{T} + \Delta\mathbf{T}) \approx C(\mathbf{T}) + \mathbf{J}\Delta\mathbf{T} + \frac{1}{2}\Delta\mathbf{T}^T \mathbf{H}\Delta\mathbf{T}. \quad (12)$$

Note that solving (12) requires deriving and computing the complex Hessian matrix  $\mathbf{H}$ , which presents a challenge in terms of optimization speed. To address this issue, we define plane factor residual term  $r$  for  $T$  as (13), converting the

---

### Algorithm 1 LM Optimization

---

```

input : Initial poses  $\mathbf{T}$ 
output: Final optimized states  $\mathbf{T}$  minimizing  $\sum \|r^2\|$ 

1  $\mu = 0.01, \nu = 2, j = 0;$ 
2 repeat
3    $j = j + 1$ 
4    $\mathbf{J} = \mathbf{0}_{1 \times 6W_p}, \mathbf{A} = \mathbf{J}^T \mathbf{J}, r$  as in cost computed from
    (13)
5   Solve  $(\mathbf{A} + \mu\mathbf{I})\delta\mathbf{T} = \mathbf{J}^T r$ 
6    $\mathbf{T}' = \mathbf{T} \boxplus \delta\mathbf{T}$ 
7   Compute current cost  $c = r(\mathbf{T})^2$  and the new cost
     $c' = r(\mathbf{T}')^2$  from (13)
8    $\rho = (c - c') / (\delta\mathbf{T} \cdot (\mu\delta\mathbf{T} + \mathbf{J}^T r))$ 
9   if  $\rho > 0$  then
10    |  $\mathbf{T} = \mathbf{T}'$ 
11    |  $\mu = \mu * \max(\frac{1}{3}, 1 - (2\rho - 1)^3); \nu = 2;$ 
12   else
13    |  $\mu = \mu\nu; \nu = 2\nu$ 
14   end
15 until  $\|\delta\mathbf{T}\| < \epsilon$  or  $j \geq j_{max};$ 

```

---

original constraints into a problem that can be solved using least-squares methods

$$r_i(\mathbf{T}) = \sqrt{m(\mathbf{f}_i, \mathbf{T})} \quad (13)$$

$$C(\mathbf{T}) = \frac{1}{2} \sum_{i=1}^{W_f} r_i(\mathbf{T})^2. \quad (14)$$

For simplicity, the residuals can be combined into vector form  $\mathbf{r}(\mathbf{T})$

$$\mathbf{r}(\mathbf{T}) = \begin{bmatrix} r_1(\mathbf{T}) \\ \vdots \\ r_{W_f}(\mathbf{T}) \end{bmatrix}. \quad (15)$$

Then we have

$$C(\mathbf{T} + \Delta\mathbf{T}) = \frac{1}{2} \mathbf{r}^T(\mathbf{T} + \Delta\mathbf{T}) \mathbf{r}(\mathbf{T} + \Delta\mathbf{T}). \quad (16)$$

Since  $\mathbf{r}(\mathbf{T})$  is a nonlinear function of  $\mathbf{T}$ , performing a first-order Taylor expansion yields

$$\mathbf{r}(\mathbf{T} + \Delta\mathbf{T}) \approx \mathbf{r}(\mathbf{T}) + \mathbf{J}\Delta\mathbf{T}. \quad (17)$$

Substituting (17) into (16) yields

$$C(\mathbf{T} + \Delta\mathbf{T}) \approx C(\mathbf{T}) + \Delta\mathbf{T}^T \mathbf{J}^T \mathbf{r} + \frac{1}{2} \Delta\mathbf{T}^T \mathbf{J}^T \mathbf{J} \Delta\mathbf{T}. \quad (18)$$

Comparing (12) and (18), we observe that with the redesigned residuals and the least-squares solution, we can avoid deriving the complex Hessian matrix and only Jacobians are needed and can be obtained by Ceres's auto-diff. And then we use a Levenberg-Marquardt algorithm to solve. The complete algorithm is listed in Algorithm 1.

## IV. EXPERIMENTAL EVALUATION

We experimentally validate the innovations proposed in feature association and optimization process. First, experiments were conducted to illustrate the improvement of memory



Fig. 4. AgileX robot equipped with Hesai 32 LiDAR and Xsens IMU for data collection.

efficiency and derivative speed on the public dataset. Then, the enhancement of the mapping quality is verified. Finally, compared with recent algorithms, we validate the improvement of the trajectory accuracy on real-world datasets.

#### A. Implementations

We implemented our proposed method in C++ and tested it on Ubuntu 20.04 running on a desktop with Intel i5-9400H CPU and 32-GB RAM.

For the feature association process, we implemented the octree structure as depicted above. PointNetVlad generates descriptors of the window and use KNN search to find the loop closure candidates. For the optimization process, since the refactored optimization problem (13) is a standard least-squares problem, to which the existing solvers apply, we solved the problem using Ceres [43]. The termination conditions are max iterations 50 (i.e.,  $j_{\max} = 50$ ), rotation update below  $10^{-6}$  rad, and translation update below  $10^{-6}$ . We take the SOTA odometry results of the LiDAR odometry algorithm FAST-LIO2 [13] as input to our proposed work.

#### B. Dataset Introduction

We conducted experiments on four real-world datasets. We evaluated map consistency on the Newer College dataset [44] and performed accuracy experiments on the ColoRadar [45], Kitti [46], and our own collected datasets. We used a robotic platform for data collection, specifically the AgileX robot, equipped with a Hesai 32-line LiDAR and an Xsens IMU, as shown in Fig. 4. We chose the Newer College dataset because it provides a ground-truth map, allowing us to assess map consistency using the Chamfer distance (CD) [47]. The other three datasets were selected because they cover various environments, including underground and outdoor scenes, as well as different data collection methods, such as handheld and vehicle-mounted, providing a comprehensive evaluation of trajectory accuracy.

#### C. Evaluation on Odometry

We demonstrated the effectiveness of our improvements to the odometry module through ablation experiments. By comparing the performance of FAST-LIO2, DLIO, and our odometry integrated with the adaptive parameter adjustment module and the distribution-to-multidistribution residual module on self-collected data, we validated the effectiveness of

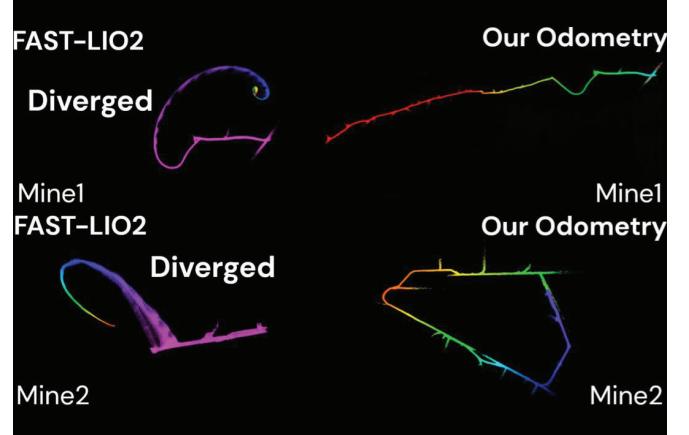


Fig. 5. Comparison of Mine1 and Mine2 between fastlio2 and our odometry.

TABLE II  
EVALUATION ON DIFFERENT ODOMETRY METHODS

Scenario	FAST-LIO2	DLIO	Ours (with A)	Ours (with B)	Ours (with A+B)
Mine1	✗	✗	✗	✓	✓
Mine2	✗	✓	✓	✓	✓
Tunnel1	✗	✗	✗	✓	✓
Tunnel2	✗	✓	✓	✗	✓

✗: Diverge, ✓: Converge, A: Adaptive Parameter Setting Strategy, B: New scan-submap Residual

TABLE III  
MEMORY CONSUMPTION (GB, TOTAL MEMORY 32 GB) OF DIFFERENT METHODS ON DIFFERENT SEQUENTIAL DATA

Sequence	frame number $N$	BALM2.0 (GB)	Ours (GB)
edgar classroom run0	100	2.496	<b>1.872</b>
	500	11.232	<b>3.392</b>
	1000	Killed	<b>4.432</b>
	1500	Killed	<b>4.688</b>
outdoors run0	100	2.272	<b>1.712</b>
	500	12.672	<b>3.984</b>
	1000	Killed	<b>4.459</b>
	1500	Killed	<b>5.024</b>

each module in our system, as shown in Table II. From Fig. 5, it can be seen that FAST-LIO2 failed to converge on the Mine1 and Mine2 sequences, whereas our odometry remained stable and did not diverge.

#### D. Memory Usage and Derivative Speed Evaluation

In this section, we compare the memory usage and derivative speed of BALM2.0 [18] and our method. We select several sequences from ColoRadar and first run FAST-LIO2 [13] to get the pose of each LiDAR scan and the undistorted point cloud frames. FAST-LIO2 [13] saves the frames at a frequency of about 10 Hz. To run a large scene (e.g., 10 min of data, downsampled to 2.5 Hz), the feature extraction module needs

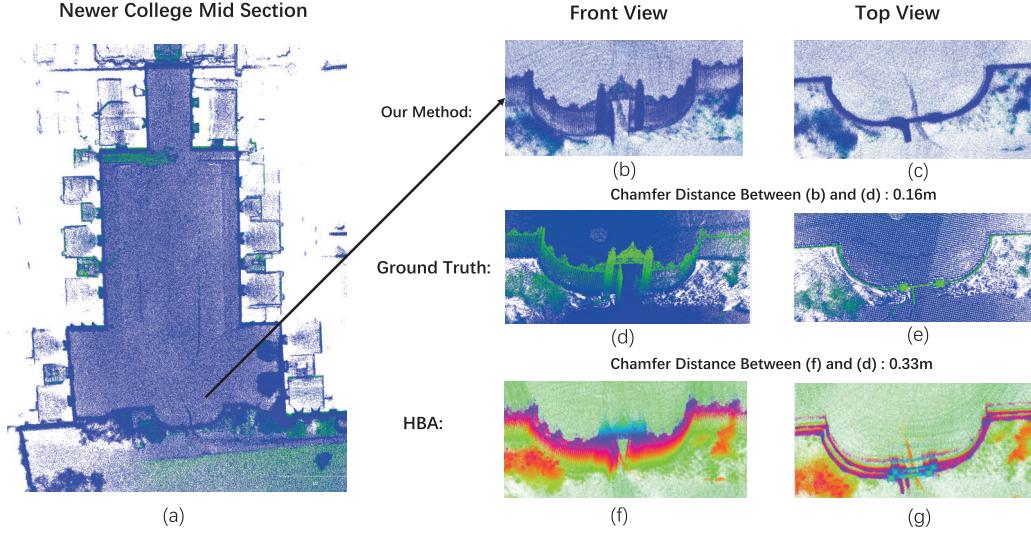


Fig. 6. Consistency results of Newer College mid-section. (a) Optimized point cloud of the Newer College mid section. (b) Front view of the local area optimized by our method. (c) Top-down view of the local area optimized by our method. (d) Front view of the ground-truth local point cloud. (e) Top-down view of the ground-truth local point cloud. (f) Front view of the local area optimized by the HBA algorithm. (g) Top-down view of the local area optimized by the HBA algorithm.

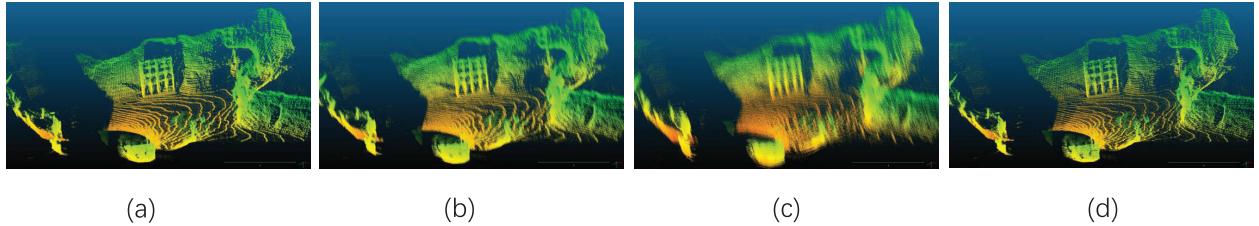


Fig. 7. Point cloud with different noises and the optimized result. (a) Point cloud with noise (0.01 m, 0.03 rad). (b) Point cloud with noise (0.04 m, 0.09 rad). (c) Point cloud with noise (0.1 m, 0.17 rad). (d) Optimized point cloud of first three (converged to the same place).

to be able to support about  $2.5 \times 10 \times 6 = 1500$  frames of scans. Therefore, for each sequence of ColoRadar data, we choose 100, 500, 1000, 1500 as the data input to the feature extraction module and assess the performance of our method and BALM2.0 in terms of memory usage. In addition, we use input sizes of 100, 200, 300, 500 for the derivative module to assess performance in terms of derivative computation speed. The results are shown in Tables III and IV.

From Table III, we can see that the experimental results are consistent with those analyzed in Section III-C: As the number of frames increases, the memory usage of our method does not increase proportionally because the number of frames associated with each plane feature remains essentially constant. As shown in Table IV, the derivative computation speed of BALM 2.0 is significantly impacted by the Hessian matrix, resulting in a rapidly decrease as the number of frames increases. In contrast, our algorithm demonstrates a significant advantage in derivative computation speed compared with BALM 2.0.

#### E. Consistency Analysis

In this section, we experimentally illustrate that the proposed BA method can effectively improve the consistency of mapping as quantitatively evaluated on the built maps. We conducted experiments on the Newer College dataset,

TABLE IV  
DERIVATIVE TIME OF DIFFERENT METHODS ON DIFFERENT  
SEQUENTIAL DATA

Sequence	frame number $N$	Derivative Time(s) (BALM2.0)	Derivative Time(s) (Ours)
edgar classroom run0	100	92	<b>37</b>
	200	271	<b>89</b>
	300	1453	<b>148</b>
	500	2175	<b>239</b>
outdoors run0	100	56	<b>18</b>
	200	130	<b>42</b>
	300	897	<b>71</b>
	500	1372	<b>102</b>

TABLE V  
COMPARISON OF MAPPING RESULTS USING CD

Sequence	Chamfer Distance(m) (HBA)	Chamfer Distance(m) (Ours)
Mid-Section	0.33	<b>0.16</b>
Quad	0.42	<b>0.17</b>
park	0.83	<b>0.42</b>

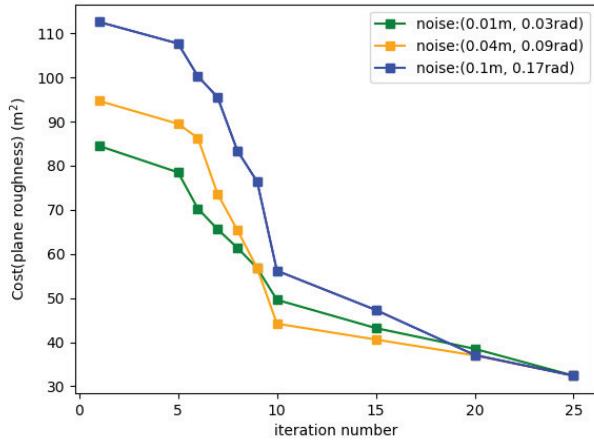


Fig. 8. Cost change through iterations.

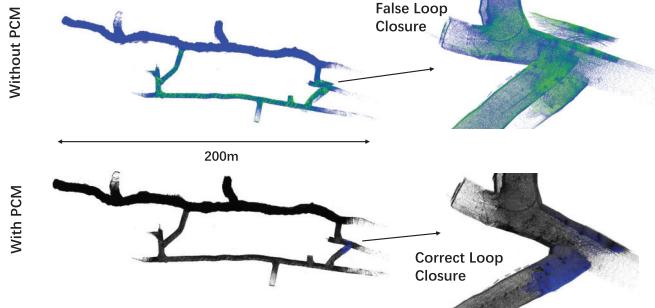


Fig. 9. Comparison of optimization results on the Campus2 dataset with and without using PCM.

calculating the CD between the optimized maps generated by our algorithm and the HBA algorithm, compared with the ground-truth map. As shown in Table V and Fig. 6, our algorithm integrates loop closure information into BA and optimizes both the poses and features, achieving better map consistency compared with HBA. In addition, the improvement of consistency should be more intuitive by comparing the maps before and after the optimization. Therefore, we selected a section of data in ColoRadar with a very small error, added high-frequency random noise with mean values of (0.01 m, 0.03 rad), (0.04 m, 0.09 rad), and (0.1 m, 0.17 rad) to the pose of each frame, and then ran our optimization algorithm. The optimized map was compared with the initial map without optimization, as shown in Fig. 7, and it can be clearly seen that the consistency of the map after optimization is greatly improved and the resolution is obviously increased as the cost decreases through iteration, which are shown in Fig. 8.

#### F. Ablation Study of the Outlier Rejection Module

We validated the outlier rejection module on our collected dataset, specifically assessing the performance of the PCM algorithm in handling false loop closures. The experiments demonstrate that this module is highly effective, successfully eliminating false loop closures in the Campus1, Campus2, and Campus3 sequences. Fig. 9 illustrates the comparison between

TABLE VI  
COMPARISON OF RMSE PRECISION WITH AND WITHOUT DYNAMIC OBJECT REMOVAL ACROSS DIFFERENT SEQUENCES

Sequence	RMSE (With Removal)	RMSE (Without Removal)
Campus Canteen	<b>0.075</b>	0.086
Hall	<b>0.024</b>	0.027
longboard_Run0	<b>0.182</b>	0.219

the optimization results with and without the PCM module. It is evident that incorporating PCM effectively removes false loop closures, thereby ensuring better map consistency in the optimized results.

#### G. Ablation Study of the Dynamic Object Removal Module

We have conducted experiments to evaluate the effectiveness of M\_detector on three different datasets that contain a substantial amount of dynamic objects, such as pedestrians and vehicles. These datasets include the following:

- 1) Campus Canteen (self-collected);
- 2) Hall (self-collected);
- 3) ColoRadar Dataset: (longboard\_Run0).

Our results show that the M\_detector module effectively filters out dynamic objects across all the three datasets. The odometry pose information is used by M\_detector to identify dynamic objects based on their inconsistent motion patterns across frames. After detection, the dynamic points are filtered out before being passed to the mapping module. This systematic approach ensures that dynamic objects such as pedestrians and vehicles are accurately identified and removed, enhancing the overall robustness and accuracy of the mapping process.

Fig. 10 shows the real-time odometry results of the Campus Canteen sequence. In this sequence, we can observe a large number of dynamic pedestrians, which are moving objects that can impact the accuracy of mapping. As shown in Fig. 11, M\_detector effectively removes the interference caused by dynamic objects. After the removal of dynamic objects using M\_detector, we performed GBA optimization and compared the results with those obtained before the dynamic object removal. The comparison results are shown in Table VI. The removal of these objects leads to a slight improvement in mapping accuracy.

#### H. Accuracy Analysis

In this section, we experimentally show that our global optimization method can effectively reduce the trajectory drift better than the SOTA BA and PGO methods. We set the window size as  $n = 10$ , and fixed overlap between windows overlap = 5 for the settings. By running three datasets, ColoRadar, Kitti, and our own dataset, recent algorithms such as SC-PGO (Scan Context [48]-based loop detection and GTSAM [49]-based PGO) and HBA [12] are compared with our method.

We emphasize conducting accuracy analysis using HBA instead of BALM2.0 for several reasons. As outlined in Section IV-D, BALM2.0 suffers from memory explosions in

TABLE VII  
ABSOLUTE TRAJECTORY ERROR (RMSE/ATE OF COLORADAR AND KITTI, CE OF OWN DATASET, METERS)  
FOR FAST-LIO2, SC-PGO, HBA, AND OURS

Datasets	Sequences	FAST-LIO2	SC-PGO	HBA	Ours
ColoRadar (RMSE)	arpg_lab_run0	0.597	0.323	0.172	<b>0.118</b>
	arpg_lab_run1	0.295	0.111	<b>0.063</b>	0.152
	arpg_lab_run2	0.214	0.087	0.078	<b>0.042</b>
	outdoors_run0	0.370	0.294	0.183	<b>0.102</b>
	outdoors_run1	0.312	0.286	0.172	<b>0.142</b>
	outdoors_run2	0.489	0.312	0.282	<b>0.152</b>
	aspen_run0	0.164	0.059	<b>0.042</b>	0.064
	aspen_run1	0.176	0.101	0.034	<b>0.024</b>
	aspen_run2	0.154	0.104	0.112	<b>0.071</b>
	ec_hallways_run0	0.587	0.506	0.342	<b>0.283</b>
Own DataSet (CE)	ec_hallways_run1	1.707	<b>0.732</b>	0.972	0.824
	ec_hallways_run2	2.363	1.423	0.823	<b>0.629</b>
	edgar_classroom_run0	2.581	1.732	1.44	<b>1.08</b>
	edgar_classroom_run1	2.542	2.434	1.982	<b>1.36</b>
Kitti DataSet ATE( $^{\circ}$  m)	edgar_classroom_run2	1.932	1.782	1.495	<b>0.92</b>
	XingHu Lab	0.872	0.795	0.683	<b>0.072</b>
	Campus 1	0.253	0.143	0.159	<b>0.063</b>
	Campus 2	0.112	0.092	0.083	<b>0.085</b>
	Campus 3	1.029	0.874	0.724	<b>0.453</b>
	kitti00	-	0.5/0.8	0.7/0.8	<b>0.6/0.8</b>
	kitti01	-	0.9/1.8	0.9/1.9	<b>0.8/1.2</b>
	kitti02	-	1.6/8.2	1.2/5.1	<b>0.9/3.7</b>
	kitti03	-	1.2/1.4	0.7/0.6	<b>0.8/0.7</b>
	kitti04	-	0.5/0.9	<b>0.1/0.5</b>	0.1/0.7

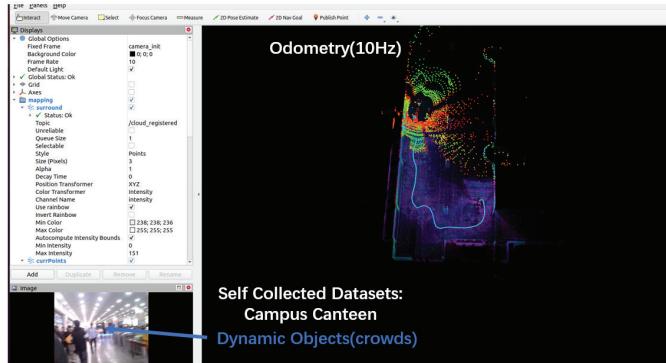


Fig. 10. Campus Canteen sequence with a large number of dynamic pedestrians, real-time odometry results.

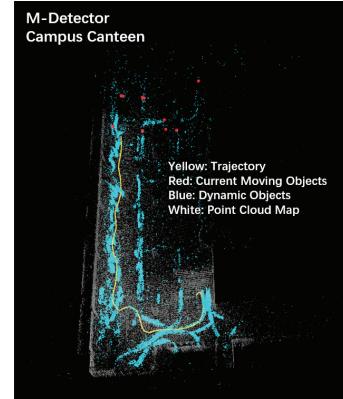


Fig. 11. M\_detector for dynamic object removal in the Campus Canteen sequence.

large-scale BA scenarios. HBA, derived from BALM2.0, offers a method specially for global BA in large-scale datasets.

Table VII shows that our method is effective in improving the accuracy of the drift trajectory better than the SOTA BA and PGO methods. Compared with these algorithms, our algorithm improves the accuracy in each sequence since we incorporate global information into the optimization. SC-PGO and HBA also improved accuracy compared with FAST-LIO2.

However, since SC-PGO did not optimize the map globally, its result may lack consistency. And the statistics clearly shows that HBA's accuracy is weaker than ours.

In most of the ColoRadar and Kitti sequences, the data return to the starting point, the drift for odometry results often reach a meter in terms of closure error (CE), and our optimization method can close the CE to cm levels, as shown

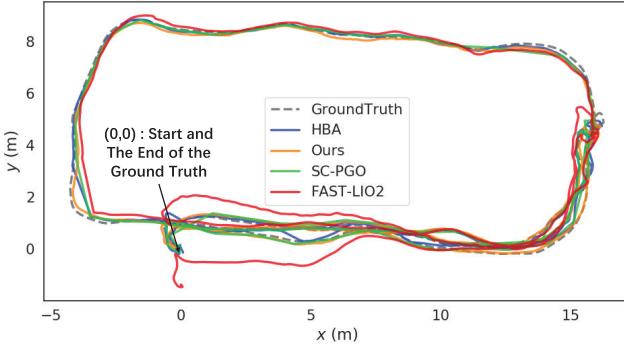


Fig. 12. Trajectory generated from different methods of sequence arpg\_lab\_run0.

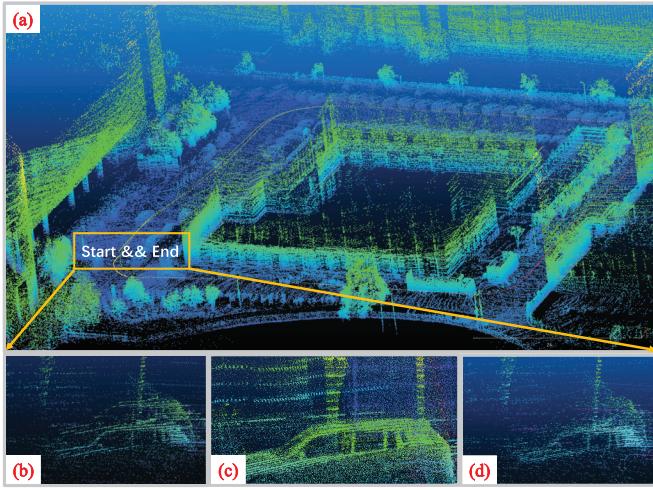


Fig. 13. Map closeups before and after global optimization. (a) Map and trajectories of sequence XingHu Lab generated from odometry. (b) Car before optimization has duplicate outline. (c) After running our method, the duplicate outline has been aligned. (d) After running HBA, the duplicate outline has been aligned, but still had some noise points.

in Fig. 12. Moreover, the experimental results showed that our optimized map resolves the duplicate outline problems. As shown in Fig. 13, in the XingHu Lab data, a car at the beginning of the data has an overlapping shadow due to CE before optimization, and the duplicate silhouette of the car disappears in the optimized map. Compared with BALM 2.0 or HBA, GBA-LIOM improves memory efficiency, ensuring that memory consumption remains manageable even when processing data from medium- to large-scale environments. However, we acknowledge certain tradeoffs. The superior accuracy and mapping consistency of GBA-LIOM in large-scale environments with drifted data rely on loop closure information to accelerate convergence. In the absence of loop closures, GBA-LIOM may not achieve the same level of performance within a typical number of iterations. Addressing this limitation is an important direction for our future work, where we aim to enhance GBA’s convergence speed and practicality even in scenarios without loop closures.

## V. CONCLUSION

This article presents a novel localization and mapping methods for LiDAR point clouds. We introduce a new scan-to-map

residual during tracking process to mitigate the divergence occurred in the challenging datasets and carefully design the storage structure for LiDAR features and significantly improve memory efficiency of feature extraction. By refactoring the cost term, we convert the nonlinear optimization problem into a least-squares problem, which can be solved by generic least-squares solvers. Finally, we add loop closure information in the feature extraction process, showing for the first time that the LiDAR adjustment methods could reduce large trajectory drifts better than the SOTA BA and PGO methods. Through a number of experiments, we have validated our method’s consistency, accuracy, and efficiency of memory consumption.

In the future, we plan to include cameras to provide further constraints in the optimization process. Another worthwhile avenue is to extend our method to other point cloud sensors such as 4-D radars [50].

## REFERENCES

- [1] V. Niculescu, T. Polonelli, M. Magno, and L. Benini, “NanoSLAM: Enabling fully onboard SLAM for tiny robots,” *IEEE Internet Things J.*, vol. 11, no. 8, pp. 13584–13607, Apr. 2024.
- [2] S. Thrun et al., “Stanley: The robot that won the DARPA grand challenge,” *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.
- [3] C. Urmson et al., “Autonomous driving in urban environments: Boss and the urban challenge,” *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, Aug. 2008.
- [4] X. Li, Y. Zhou, and B. Hua, “Study of a multi-beam LiDAR perception assessment model for real-time autonomous driving,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.
- [5] J. Levinson et al., “Towards fully autonomous driving: Systems and algorithms,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.
- [6] F. Gao, W. Wu, W. Gao, and S. Shen, “Flying on point clouds: Online trajectory generation and autonomous navigation for quadrotors in cluttered environments,” *J. Field Robot.*, vol. 36, no. 4, pp. 710–733, 2019.
- [7] F. Kong, W. Xu, Y. Cai, and F. Zhang, “Avoiding dynamic small obstacles with onboard sensing and computation on aerial robots,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7869–7876, Oct. 2021.
- [8] Y. Ren et al., “Bubble planner: Planning high-speed smooth quadrotor trajectories using receding corridors,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Kyoto, Japan, Oct. 2022, pp. 6332–6339.
- [9] Y. Zhuang et al., “Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches,” *Inf. Fusion*, vol. 95, pp. 62–90, Jul. 2023.
- [10] Z. Liu, F. Zhang, and X. Hong, “Low-cost retina-like robotic LiDARs based on incommensurable scanning,” *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 1, pp. 58–68, Feb. 2022.
- [11] Y. Dong, L. Li, Y. Liu, S. Xu, and Z. Zuo, “PVE-LIOM: Pseudo-visual enhanced LiDAR-inertial odometry and mapping,” *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.
- [12] X. Liu, Z. Liu, F. Kong, and F. Zhang, “Large-scale LiDAR consistent mapping using hierarchical LiDAR bundle adjustment,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1523–1530, Mar. 2023.
- [13] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “FAST-LIO2: Fast direct LiDAR-inertial odometry,” *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [14] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.
- [15] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, “MULLS: Versatile LiDAR SLAM via multi-metric linear least square,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2021, pp. 11633–11640.
- [16] H. Zhang, R. Xiao, J. Li, C. Yan, and H. Tang, “A high-precision LiDAR-inertial odometry via invariant extended Kalman filtering and efficient surfel mapping,” *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024.

- [17] Z. Liu and F. Zhang, "BALM: Bundle adjustment for LiDAR mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3184–3191, Apr. 2021.
- [18] Z. Liu, X. Liu, and F. Zhang, "Efficient and consistent bundle adjustment on LiDAR point clouds," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4366–4386, Dec. 2023.
- [19] L. Zhou, D. Koppel, and M. Kaess, "LiDAR SLAM with plane adjustment for indoor environment," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7073–7080, Oct. 2021.
- [20] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot. Sci. Syst. X*, Berkeley, CA, USA: Robotics: Science and Systems Foundation, Jul. 2014.
- [21] 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.
- [22] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-LiDAR-inertial odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5167–5174, Jul. 2021.
- [23] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [24] A. Segal, D. Hähnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot. Sci. Syst. V*, Cambridge, MA, USA: MIT Press, Jun. 2009.
- [25] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.
- [26] H. Lim, D. Kim, B. Kim, and H. Myung, "AdaLIO: Robust adaptive LiDAR-inertial odometry in degenerate indoor environments," in *Proc. 20th Int. Conf. Ubiquitous Robots (UR)*, Jun. 2023, pp. 48–53.
- [27] A. Reinke et al., "LOCUS 2.0: Robust and computationally efficient LiDAR odometry for real-time 3D mapping," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9043–9050, Oct. 2022.
- [28] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct LiDAR-inertial odometry: Lightweight LIO with continuous-time motion correction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 3983–3989.
- [29] S. Liang, Z. Cao, C. Wang, and J. Yu, "A novel 3D LiDAR SLAM based on directed geometry point and sparse frame," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 374–381, Apr. 2021.
- [30] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot. Sci. Syst. XIV*, Pittsburgh, PA, USA: Robotics: Science and Systems Foundation, Carnegie Mellon Univ., Jun. 2018.
- [31] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally consistent and tightly coupled 3D LiDAR inertial mapping," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Philadelphia, PA, USA, May 2022, pp. 5622–5628.
- [32] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2021.
- [33] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Computer Vision—ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11–14, 2004. Proceedings, Part III* (Lecture Notes in Computer Science), vol. 3023, T. Pajdla and J. Matas, Eds., Berlin, Germany: Springer-Verlag, May 2004, pp. 558–569.
- [34] L. Zhou, D. Koppel, H. Ju, F. Steinbruecker, and M. Kaess, "An efficient planar bundle adjustment algorithm," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Nov. 2020, pp. 136–145.
- [35] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4605–4611.
- [36] G. Ferrer, "Eigen-factors: Plane estimation for multi-frame and time-continuous point cloud alignment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1278–1284.
- [37] L. Zhou, S. Wang, and M. Kaess, "P-LSAM: LiDAR smoothing and mapping with planes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5751–5757.
- [38] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11054–11059.
- [39] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, Aug. 1987.
- [40] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4470–4479.
- [41] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2916–2923.
- [42] H. Wu, Y. Li, W. Xu, F. Kong, and F. Zhang, "Moving event detection from LiDAR point streams," *Nature Commun.*, vol. 15, no. 1, p. 345, Jan. 2024.
- [43] S. Agarwal, K. Mierle, *Ceres Solver*. [Online]. Available: <https://github.com/ceres-solver/ceres-solver>
- [44] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld LiDAR, inertial and vision with ground truth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4353–4360.
- [45] A. Kramer, K. Harlow, C. Williams, and C. Heckman, "ColoRadar: The direct 3D millimeter wave radar dataset," *Int. J. Robot. Res.*, vol. 41, no. 4, pp. 351–360, Apr. 2022.
- [46] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9296–9306.
- [47] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Balanced chamfer distance as a comprehensive metric for point cloud completion," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 29088–29100.
- [48] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4802–4809.
- [49] F. Dellaert and G. Contributors (May 2022). *Borglab/GTSAM*. [Online]. Available: <https://github.com/borglab/gtsam>
- [50] Y. Zhuang, B. Wang, J. Huai, and M. Li, "4D iRIOM: 4D imaging radar inertial odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3246–3253, Jun. 2023.



**Shangjia Liu** received the B.Eng. degree in geomatics engineering from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 2022, where he is currently pursuing the master's degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.

His research interests include LiDAR SLAM, optimization, mapping, and computer vision.



**Jianzhu Huai** (Member, IEEE) received the B.Eng. and M.Sc. degrees in civil engineering from Beihang University, Beijing, China, in 2009 and 2012, respectively, and the Ph.D. degree in geodetic engineering from The Ohio State University, Columbus, OH, USA, in 2017.

He is an Associate Researcher at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. He subsequently spent two and a half years with the Autonomous Personal Robot Group at Segway Robotics, Nashua, NH, USA, where he developed software for localization, mapping, and sensor calibration, before joining Wuhan University. His research focuses on low-cost large-area mapping and autonomous exploration in challenging environments.



**Binliang Wang** received the B.Sc. degree from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2019, and the M.Sc. degree in photogrammetry and remote sensing from Wuhan University in 2022, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.

His research interests include photogrammetric processing of Mars remote sensing image, satellite and robot pose determination, and location, navigation and mapping technology based on multisensor fusion.



**Yuan Zhuang** (Senior Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Calgary, Calgary, AB, Canada, in 2015.

He is a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. He has co-authored more than 130 academic papers and 30 patents. His current research interests include multisensors' integration, real-time location system, wireless positioning, and Internet of Things (IoTs).

Dr. Zhuang received over ten academic awards. He is an Associate Editor of IEEE ACCESS, the Guest Editor of IEEE INTERNET OF THINGS JOURNAL and *Satellite Navigation*, and a reviewer of more than 20 IEEE journals.



**Yulong Han** received the bachelor's degree in surveying and mapping engineering from Shandong University of Science and Technology, Qingdao, China, in 2019, and the master's degree in surveying and mapping engineering from the Capital Normal University, Beijing, China, in 2022. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Surveying, Mapping and Remote Sensing Information Engineering, Wuhan University, Wuhan, China.

His research interests include 3-D point cloud acquisition and application, multisensor fusion-based localization and mapping, navigation, and surveying technology.