

II-NVM: Enhancing Map Accuracy and Consistency With Normal Vector-Assisted Mapping

Chengwei Zhao^{ID}, Yixuan Li, Yina Jian^{ID}, Jie Xu^{ID}, Linji Wang^{ID}, Yongxin Ma, and Xinglai Jin

Abstract—SLAM technology plays a crucial role in indoor mapping and localization. A common challenge in indoor environments is the “double-sided mapping issue”, where closely positioned walls, doors, and other surfaces are mistakenly identified as a single plane, significantly hindering map accuracy and consistency. To address this issue, this letter introduces a SLAM approach that ensures accurate mapping using normal vector consistency. We enhance the voxel map structure to store both point cloud data and normal vector information, enabling the system to evaluate consistency during nearest neighbor searches and map updates. This process distinguishes between the front and back sides of surfaces, preventing incorrect point-to-plane constraints. Moreover, we implement an adaptive radius KD-tree search method that dynamically adjusts the search radius based on the local density of the point cloud, thereby enhancing the accuracy of normal vector calculations. To further improve real-time performance and storage efficiency, we incorporate a Least Recently Used (LRU) cache strategy, which facilitates efficient incremental updates of the voxel map. The code is released as open-source and validated in both simulated environments and real indoor scenarios. Experimental results demonstrate that this approach effectively resolves the “double-sided mapping issue” and significantly improves mapping precision. Additionally, we have developed and open-sourced the first simulation and real-world dataset specifically tailored for the “double-sided mapping issue”.

Index Terms—SLAM, Mapping, indoor, double-sided mapping issue, normal vector-assisted.

Received 25 December 2024; accepted 10 April 2025. Date of publication 16 April 2025; date of current version 24 April 2025. This article was recommended for publication by Associate Editor G. D. Tipaldi and Editor S. Behnke upon evaluation of the reviewers’ comments. This work was supported by the State Key Laboratory of Robotics and Systems (HIT) (SKLRS-2025-KF-16) and the National Natural Science Foundation of China under Grant 62073101. (*Chengwei Zhao and Yixuan Li are co-first authors.*) (*Corresponding author: Jie Xu.*)

Chengwei Zhao is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China, and also with Hangzhou Qisheng Intelligent Technology Company Ltd., Hangzhou 311217, China (e-mail: chengweizhao0427@gmail.com).

Yixuan Li is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: 2191312118@stu.xjtu.edu.cn).

Yina Jian is with the Department of Computer Science, Columbia University in the City of New York, New York, NY 10027 USA (e-mail: yj2713@columbia.edu).

Jie Xu is with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology Harbin 150001, China, and also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jeff_xu_0503@foxmail.com).

Linji Wang and Yongxin Ma are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: lwang44@gmu.edu; yxma@mail.sdu.edu.cn).

Xinglai Jin is with Hangzhou Qisheng Intelligent Technology Company Ltd., Hangzhou 311217, China.

Code —<https://github.com/chengwei0427/II-NVM>.

Video —<https://www.youtube.com/watch?v=qso39uI7l38>.

Digital Object Identifier 10.1109/LRA.2025.3561568

I. INTRODUCTION

In SLAM tasks, systems are required to simultaneously perform pose estimation and map construction within unknown environments [1]. Accurate mapping is critical for robotic navigation and autonomous driving, particularly in complex indoor settings [2]. Additionally, the use of SLAM technology in indoor mapping is increasingly gaining attention due to its ability to provide high-precision spatial information, including local geometric details [3], [4], which supports the digital management and maintenance of buildings and facilities [5].

The distinct structural features of indoor environments, including thin walls, doors, and windows, complicate LiDAR scan data analysis. Existing LiDAR-Inertial Odometry (LIO) algorithms, which are widely used outdoors, often underperform indoors [6], [7]. Complex scenarios featuring multiple rooms and walls increase the difficulty of SLAM systems, as point clouds from both sides of thin walls may be erroneously matched as a single plane due to their proximity, causing the double-sided mapping issue (as illustrated in Fig. 1). Many existing SLAM methods experience reduced mapping accuracy due to the lack of normal vector information or inadequate handling of normal vector consistency and map management.

To effectively resolve the double-sided mapping issue, this study introduces a method based on the calculation of point cloud plane normal vectors. This method is called II-NVM, where “II” looks like the double-sided wall, emphasizing its ability to accurately distinguish between the front and back sides of a wall during mapping. “NVM” stands for Normal Vector-Assisted Mapping, indicating that this method solves the double-sided mapping issue through normal vector consistency. This innovation markedly enhances the matching accuracy between point cloud data and object surface features, thereby improving the overall precision and consistency in mapping of map construction.

The main contributions of this paper include:

- Our paper enhances the storage of normal vectors within voxel maps, enabling each voxel block to not only contain point cloud data but also record normal vector information for both the front and back sides. This design facilitates the use of normal vector consistency in subsequent matching and map updates, effectively preventing the double-sided mapping issue. By incorporating this dual-sided storage approach, the method supports efficient incremental voxel map updates, ensuring real-time performance and improved storage efficiency.

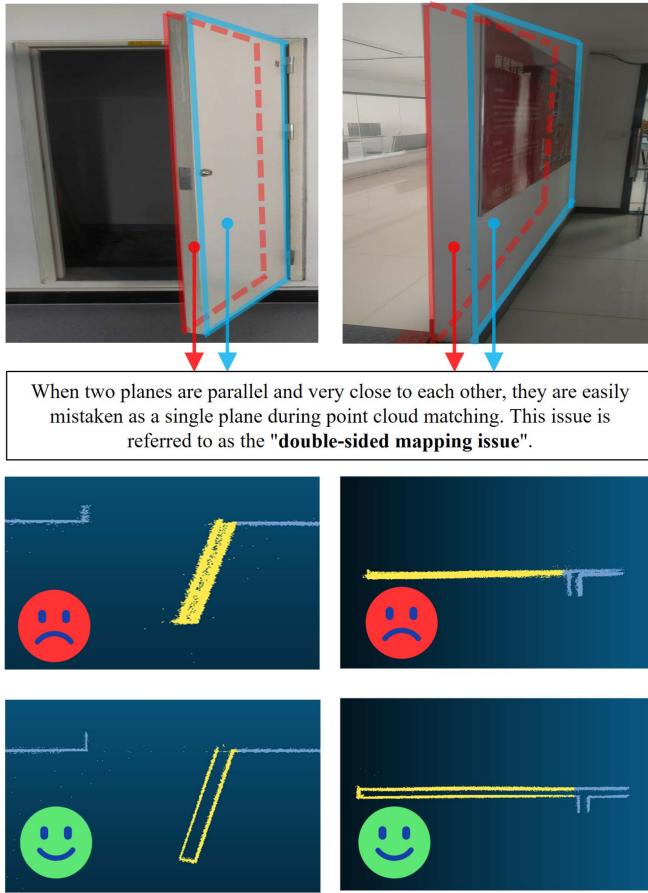


Fig. 1. illustrates a real-world scenario where the double-sided mapping issue arises, presenting examples of both erroneous and correct mappings.

- We propose a new adaptive radius KD-tree search method for calculating normal vectors, which dynamically adjusts the neighborhood search radius based on local point cloud density. This method analyzes the consistency of normal vector directions in point clouds to accurately distinguish the front and back of planar point clouds, effectively addressing mapping errors commonly encountered in traditional SLAM systems in double-sided scenarios.
- We conducted both simulation and real-world experiments to verify the effectiveness of the proposed method. To advance this research area, we have open-sourced the associated code and dataset, creating the first dataset specifically designed for the double-sided mapping issue, which provides a standardized evaluation benchmark and an important reference for future research.

II. RELATED WORK

A. Traditional LIO Mapping Algorithm

LOAM [8] is a foundational LiDAR SLAM system that facilitates pose estimation and map construction through geometric feature extraction and point cloud matching. Building on this, FAST-LIO [9] provides an efficient and robust LiDAR-Inertial

Odometry solution, employing tightly coupled, iterative extended Kalman filtering (EKF) for real-time pose estimation and map construction, while significantly reducing computational demands. In a similar vein, I2EKF-LO [10] utilizes dual iterative EKF to handle point cloud motion distortion, dynamically adjust process noise, and support diverse sensor platforms, achieving high-precision and efficient state estimation. For multimodal sensor fusion, M-DIVO [11] integrates visual, depth, and inertial modules from multiple ToF RGB-D cameras, utilizing an odometry system based on IEKF to enhance robustness, precision, and real-time performance through a multimodal redundancy scheduling mechanism and improved sensor calibration. In contrast, SuMa [12] employs a surface model-based mapping technique that uses dense point clouds to create accurate maps, while LIPS [13] leverages 3D indoor scenes by introducing nearest points to parameterize planes and utilizes a plane-to-plane cost for precise pose estimation.

Although geometric feature matching is commonly used in the aforementioned algorithms, planes are not effectively considered in double-sided areas. This can lead to pose errors and reduced accuracy in pose estimation. To address this limitation, we propose an adaptive radius KD-tree normal vector calculation method based on normal vector data to better manage the double-sided mapping issue.

B. Voxel Map-Based Mapping Method

Faster-LIO [14] uses sparse voxel hash mapping for point cloud data storage, significantly reducing storage requirements while maintaining computational efficiency. In a similar vein, VoxelMap [15] adapts to different environmental structures through an adaptive voxel size construction method, improving robustness for sparse and irregular LiDAR point clouds, and enhancing the efficiency of voxel construction, updating, and querying. CT-ICP [16] further accelerates real-time processing by storing dense point cloud local maps within a sparse voxel framework.

In contrast, our method utilizes normal vector information to address the double-sided mapping issue, employing incremental voxel updates and LRU cache strategies to manage point clouds more efficiently, thereby significantly enhancing real-time capabilities.

C. Normal Vector-Based SLAM Algorithm

The LiDAR SLAM with Plane Adjustment method [17] resolves the double-sided mapping issue by calculating the plane normal vector [18] from the local surface geometry of the point cloud and aligning it towards the LiDAR center to differentiate the front and back of objects. Similarly, LOG-LIO [19] is a robust and precise LiDAR-Inertial Odometry system that emphasizes real-time estimation of the normal vectors of LiDAR scan points and the distribution of map points, ensuring effective utilization of these components. It also proposes a ring-based fast approximate least squares method for improved efficiency. On the other hand, NV-LIOM [20] extracts normal vectors from LiDAR scans and applies them in correspondence searches to improve point cloud registration. By analyzing the distribution of

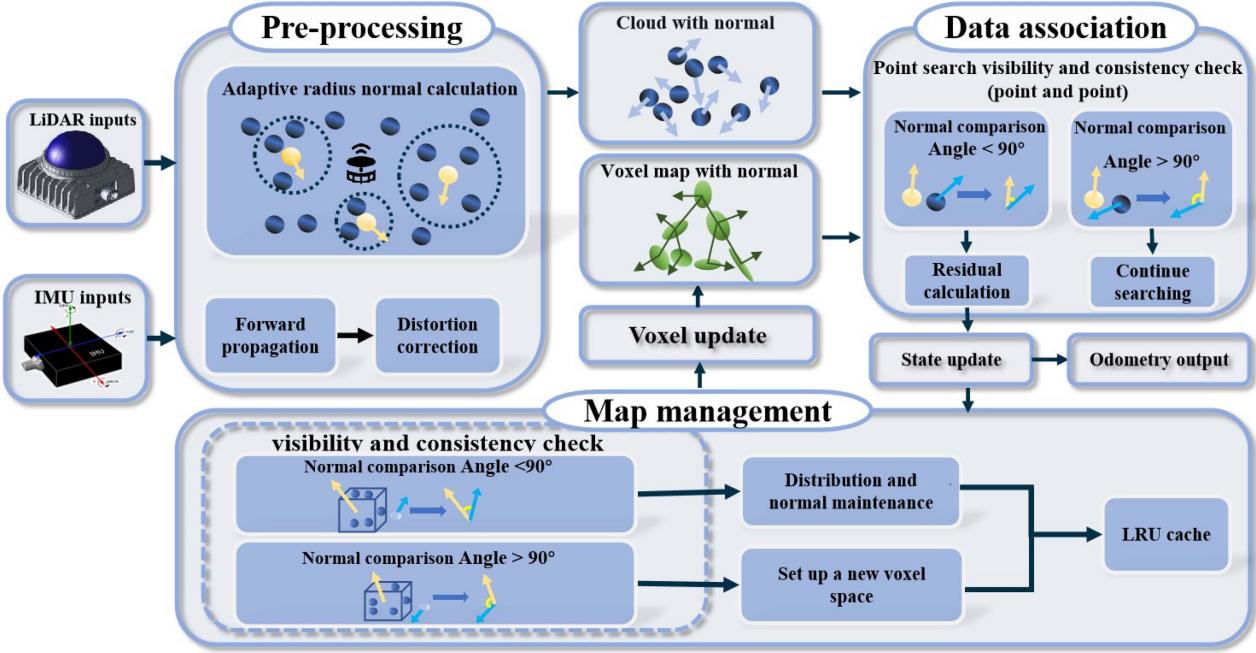


Fig. 2. System overview of II-NVM.

normal vector directions and checking for degeneracy, it adjusts correspondence uncertainty to enhance registration accuracy.

Existing normal vector-based SLAM methods face limitations in registration accuracy due to poor utilization and management of normal vectors. For example, LOG-LIO only uses normal vectors for field-of-view checks, while NV-LIOM stores normal vectors for each point and builds local maps from keyframes, but struggles with submap construction and normal vector inconsistencies. To address these issues, we propose an adaptive resolution normal vector estimation method that works with various LiDAR sensors. We also extend the voxelmap data structure to include normal vector data and enhance voxels to store both front and back side information, solving the double-sided mapping issue.

III. METHOD

The pipeline of II-NVM is shown in Fig. 2. This section presents the core SLAM method we proposed, which aims to solve the “double-sided mapping issue”. Our method ensures the consistency of plane normal vectors and improves map accuracy through adaptive search radius normal vector calculation, incremental voxel map management, and normal vector view consistency processing.

A. Data Preprocessing and Normal Vector Calculation

Accurately calculating normal vectors is crucial for maintaining consistency in plane normal vector discrimination. Traditional methods typically manage data using a KD-tree and employing a fixed-radius neighborhood search to estimate normal vectors. However, in complex indoor environments, the density of LiDAR point cloud data can vary significantly based on the

openness of the environment and the distance from the LiDAR origin. The persistent use of a fixed-radius neighborhood search method struggles to adapt to the geometric features at varying distances, leading to decreased accuracy in normal vector calculations, especially in edge and corner areas. To address this, this paper proposes a distance-adaptive robust normal estimation method.

Due to the inaccuracy of LiDAR data for distance measurements in regions that are either too close or too far, we first filter out these point clouds. The filtered data is then organized using a KD-tree structure, enabling efficient iteration through the dataset to identify the nearest neighbors for each point. Next, the normal vector for each point is calculated using the eigenvalue decomposition method. To address the varying densities of point cloud data at different scanning radius, we introduce a distance-adaptive search radius, where the size of the search radius is set based on the scanning radius (Dist) of the current point, as shown in (1):

$$r = \frac{D_{\text{scan}} - D_{\min}}{D_{\max} - D_{\min}} \cdot (R_{\max} - R_{\min}), \quad (1)$$

where R_{\max} and R_{\min} represent the maximum and minimum search radius, respectively. Similarly, D_{\max} and D_{\min} denote the maximum and minimum scanning radius and D_{scan} is the scanning distance of the current point (sensor-to-point range).

Additionally, due to boundary discontinuities and multiple reflections, the covariance matrix decomposition is particularly sensitive to outliers. To address this, we employ two simple and effective methods for outlier elimination. Firstly, if the number of points in the nearest neighbor set is below a specified threshold Ng_{thres} , the current point is considered an outlier. The second criterion is based on the planarity of the neighborhood set:

assuming $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of the plane covariance decomposition, arranged in descending order, and δ_p represents its planarity. If δ_p exceeds the threshold δ_{thres} , the current point is deemed an outlier. The planarity δ_p is calculated as shown in (2):

$$\delta_p = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}. \quad (2)$$

Data identified as outliers will not be included in subsequent calculations.

B. Incremental Voxel Map

We adopted a voxel map management strategy combined with an LRU caching strategy to achieve incremental voxel map construction. Each voxel block not only stores point cloud data but also records normal vector information to facilitates subsequent consistency checks of normal vectors.

1) *Intra-Voxel View Consistency Judgment*: In each voxel block, point cloud data collected from different times and locations may have different normal vector distributions. Here, we describe how to judge view consistency. First, for data that needs to be inserted into a voxel, we calculate its view consistency with existing data. Assuming surface data exists within the current voxel, we determine whether the principal direction of the surface data has been calculated. If it has, we proceed to the next step, otherwise, we traverse all data on the current surface, extract all point normal vector information, construct a covariance matrix, and calculate the main direction of the normal vector distribution in the current voxel using covariance decomposition. Then, using the vector dot product formula, we compute the angle between the normal vector of the point to be inserted and the principal direction of the current voxel. If the angle is less than a threshold θ_{th} (e.g., 90°), it is considered coplanar and the point is inserted into the current surface while updating the voxel's principal direction. If the angle exceeds θ_{th} , the point is stored on the opposite side of the current plane. Notably, θ_{th} is empirically set to $90^\circ \pm 10^\circ$, where minor variations within this range do not compromise experimental robustness due to the voxel-based geometric tolerance.

2) *NVM Voxel Map Management*: Traditional voxel map management involves dividing the environment into multiple voxel blocks and using hash values for accessing each block. Each voxel block stores the coordinate information of LiDAR point cloud data, reducing storage and computing complexity. To effectively resolve the double-sided mapping issue, we extended the traditional voxel map structure, allowing each voxel block to store not only coordinate information but also normal vector data for each point cloud. Additionally, considering that each object may have two planes (front and back), we expand each voxel block to store data for both planes. By default, data is stored on the front side first, and if view inconsistency is detected in the current voxel, the data is stored on the back side of the current plane. This extension enables the voxel block to store more geometric information, facilitating in precise associative matching in subsequent point correspondence searches and view consistency judgments.

Specifically, for incoming LiDAR data and corresponding global poses, we first transform point cloud data to the global coordinate system using (3), and calculate the hash index of the voxel map corresponding to the current point using (4). When no voxel data exists for the current index, the current voxel block is initialized, and data is stored in it. If a voxel block corresponding to the current index already exists, the view consistency between the normal vector of the current point and the normal vectors of existing data in the voxel is evaluated. When the normal vector views are consistent, it indicates the point and existing points in the voxel block originate from the same surface, and the system updates the point cloud data into the front region of the voxel block. On the other hand, if normal vector views are inconsistent, suggesting the point and points stored in the voxel block may come from different sides of an object, to avoid the double-sided mapping issue, the system stores the point in the back region of the voxel block without resetting the entire voxel block.

$$P_{wi} = TP_{li}, \quad n_{wi} = Rn_{li}, \quad (3)$$

where P_{wi} and n_{wi} are the position and normal vector in the global coordinate system, and P_{li} and n_{li} are the corresponding values in the local coordinate system. T is the translation matrix and R is the rotation matrix, which together describe the pose of the current point cloud.

$$v_i = \frac{P_{wi}}{d_i}, \quad (4)$$

where v_i represents the voxel index for the current point, P_{wi} is the position of the point in the global coordinate system, and d_i is the resolution of the voxel map.

By dividing voxel blocks into front and back regions, the system can store data from different perspectives. The front region stores data consistent with the normal vector of the current observation point, while the back region is for data from the opposite surface. This approach allows a single voxel block to contain information from multiple views, effectively solving the double-sided mapping issue and preventing data loss and computational overhead associated with resetting voxel blocks. This method ensures that data stored in voxel blocks always reflects the correct geometric structure, preventing the double-sided mapping issue.

3) *LRU-Based Incremental Voxel Map Update*: In managing incremental voxel maps, we introduce the LRU caching management strategy. LRU is a common memory management algorithm suitable for scenarios involving efficient handling of large data blocks. In SLAM, due to the vast amount of point cloud data, real-time map updates require effective voxel block management. It can lead to memory overflow or decreased system performance. By introducing the LRU management strategy, we can dynamically update and maintain voxel blocks based on the actual usage of point clouds. When a voxel block has not been accessed or updated for some time, it is marked as “least recently used” and is prioritized for replacement when storage space needs to be freed. This way, the system can ensure that storage resources are concentrated on the most active voxel blocks while reducing memory consumption. This mechanism not only

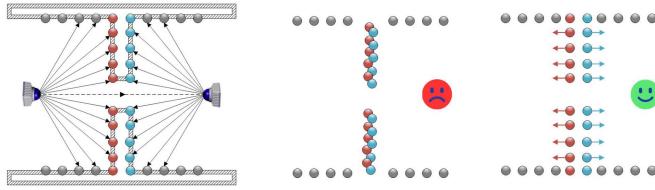


Fig. 3. When the LiDAR scans through a wall, two closely spaced walls may be erroneously identified as a single plane. Accurate matching is achieved by employing normal vector consistency calculations.

enhances the efficiency of map updates but also provides a more flexible framework for handling normal vector consistency in map management.

C. NVM-Based State Estimation

To evaluate the effectiveness of our proposed algorithm, we integrated the NVM module into our open-source project, CT-LIO.¹ The complete system workflow can be found in the project code, with this section focusing primarily on the data association process.

For data association, each point cloud datum is transformed into global coordinates using initial system-provided values. We calculate the hash index corresponding to each point using (3) and (4). This index is used to traverse the current index and its 26 neighboring indices.

We extract the data stored within the voxels corresponding to these indices. In addition to verifying that the distance from the current point meets specific criteria, we examine the angle between the normal vector of the target point and those of neighboring points to ensure they originate from the same surface. Similar to the Section III-B1), if the angle between the normal vectors of two points is below the designated threshold, they are considered valid neighboring points. Specific details are illustrated in Fig. 3.

After acquiring map data from neighboring points that satisfy the conditions, we construct the point-to-plane residual. Planar consistency is then applied to calculate the weight of the existing constraint, which is subsequently used to estimate the current state.

IV. EXPERIMENTAL RESULTS

A. Description of Dataset and Experimental Environment

In this study, directly comparing the performance of end-to-end SLAM methods presents certain challenges. One key issue is that some SLAM algorithms achieve high accuracy in an end-to-end setup without adequately addressing the double-sided mapping problem. Furthermore, obtaining ground truth data for indoor environments, particularly in large spaces spanning multiple rooms, is difficult. Traditional motion capture techniques are not capable of providing precise ground truth in such complex settings. As a result, conventional methods fall

TABLE I
SIMULATION DATASET AND LiDAR SPECIFICATIONS

Dataset Name	LiDAR Type	Duration (s)	Trajectory Length (m)	Pose
Wall_15cm_a	Livox	149.092	54.673	4473
Wall_15cm_b	Velodyne	115.300	54.304	1154
Wall_10cm_a	Livox	240.441	69.761	7214
Wall_10cm_b	Velodyne	167.000	64.533	1671
Wall_5cm_a	Livox	248.480	48.425	7215
Wall_5cm_b	Velodyne	55.300	17.435	554
Room_a	Livox	97.090	254.678	2913
Room_b	Velodyne	115.800	50.675	1159
Café_a	Livox	118.209	48.425	3546
Café_b	Velodyne	90.900	42.632	910
Garage_a	Livox	483.033	69.761	7214
Garage_b	Livox	323.400	101.790	3235
Garage_c	Livox	456.129	136.814	13684
Garage_d	Livox	425.425	141.411	12763
Garage_e	Velodyne	353.900	171.414	3540

short in properly assessing how well SLAM systems handle the double-sided mapping issue.

To address these challenges, this study adopts two evaluation approaches to assess the performance of the proposed method. First, trajectory evaluation is performed using the Gazebo simulation environment, which generates precise ground truth trajectories, enabling accurate assessment of the algorithm's performance. Second, given the difficulty of obtaining accurate ground truth for real indoor datasets, we introduce wall thickness as an alternative evaluation metric. These two evaluation methods provide a comprehensive and objective means to validate the effectiveness of the proposed method in handling the double-sided mapping issue, ensuring both scientific rigor and fairness in the evaluation process. The experimental results are demonstrated in the video² in the footnote.

Table I summarizes the characteristics of the simulation datasets used in this study, including the types of LiDAR, collection duration, trajectory length, and the number of poses. The datasets cover various environments, such as walls of different thicknesses, rooms, cafés, and garages.

B. Evaluation of Odometry in Gazebo Simulation Scenarios

In the Gazebo simulation environment, we built a complex indoor scene with various geometric structures, walls of different thicknesses, and double-sided areas, replicating real-world double-sided mapping challenges (Fig. 4). To evaluate SLAM systems' pose estimation performance, we used Absolute Trajectory Error (ATE), which measures trajectory accuracy by comparing the estimated trajectory with the ground truth from the simulation. After running each method, we calculated and compared their ATE values [21].

Tables II and III present the quantitative results of Livox LiDAR and Velodyne LiDAR in various indoor environments.

¹<https://github.com/chengwei0427/ct-li>

²Video — <https://www.youtube.com/watch?v=qso39uI7l38>

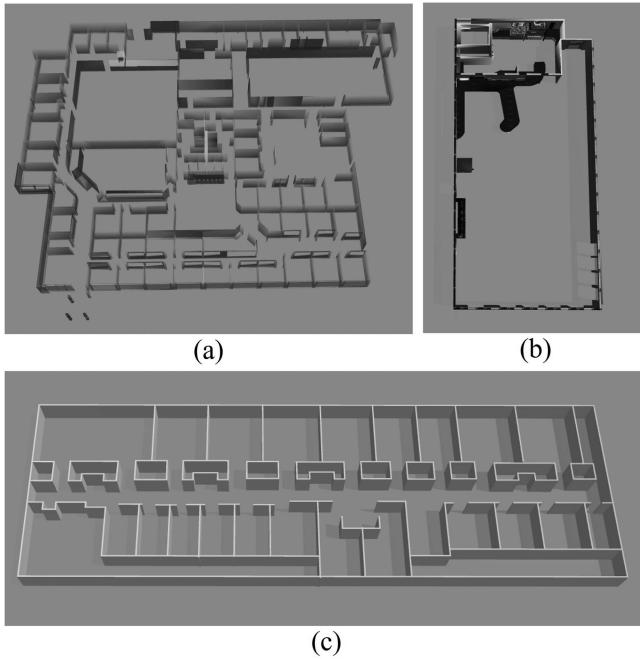


Fig. 4. Gazebo simulation scenarios show (a) garage scene, (b) café scene, and (c) fixed-distance wall scene.

TABLE II
RESULTS OF POSE ESTIMATION COMPARISON ON LIVOX LiDAR

Sequence	II-NVM	CT-LIO	II-NVM(1m)	FAST-LIO2	IG-LIO	DLIO
Wall_15cm_a	0.0179	0.0236	0.0186	0.0545	0.0817	0.1161
Wall_10cm_a	0.0137	0.0155	0.0145	0.0492	0.0824	0.1145
Wall_5cm_a	0.0163	0.1669	0.0168	0.0364	0.0766	0.0680
Room_a	0.0144	0.0150	0.0147	0.0570	0.0858	0.0729
Café_a	0.0168	0.0180	0.0171	0.0694	0.0846	0.1334
Garage_a	0.0162	0.0171	0.0165	0.0849	0.1096	0.1049
Garage_b	0.0158	0.0187	0.0188	0.0430	0.0446	0.0926
Garage_c	0.0133	0.0515	0.0135	0.1674	1.1304	0.1637
Garage_d	0.0138	0.0168	0.0147	1.3726	0.2051	0.1152

In all our experiments, we employed a voxel size of 0.3 m and successfully addressed the distortion issue that was absent in the simulation. CT-LIO and II-NVM (1 m) demonstrated the trajectory accuracy without using normal vectors and the adaptive radius KD-tree.

Fig. 5 presents a comparison of trajectories from various algorithms for qualitative analysis. The experimental results demonstrate that in indoor environments with multiple walls, the II-NVM method significantly enhances trajectory estimation accuracy by effectively resolving double-sided mapping issues. It achieved the lowest ATE value, outperforming all other algorithms. Ablation experiments with CT-LIO and II-NVM (1 m) further demonstrate the effectiveness of the adaptive radius KD-tree search method and the normal vector consistency check. In contrast, the other methods showed larger errors in complex

TABLE III
RESULTS OF POSE ESTIMATION COMPARISON ON VELODYNE LiDAR

Sequence	II-NVM	CT-LIO	LOG-LIO	FAST-LIO2	IG-LIO	NV-LIOM	DLIO
Wall_15cm_b	0.0475	0.1466	0.3963	0.0686	0.2152	0.2344	0.0532
Wall_10cm_b	0.0474	0.0543	0.4026	0.0881	0.2030	0.3648	0.1145
Wall_5cm_b	0.0410	0.0491	0.1693	0.1748	0.3067	0.1038	0.0986
Room_b	0.0641	0.1869	0.1078	0.0950	0.1506	0.1123	0.1083
Café_b	0.0206	0.0703	0.0930	0.1526	0.1260	0.0552	0.0958
Garage_e	0.2153	0.2460	0.6651	0.9372	0.6651	0.4602	0.2914

settings and struggled with the challenges posed by double-sided mapping. The findings highlight the superior mapping accuracy and reliability of the II-NVM method in multi-wall environments.

C. Evaluation of Wall Thickness

We first compared the performance of various SLAM algorithms in addressing the double-sided mapping issues. The experimental results indicate that other SLAM algorithms have significant deficiencies when addressing double-sided mapping issues, as they are unable to effectively distinguish between the front and back sides of walls, leading to reduced mapping accuracy. In contrast, the II-NVM method successfully resolves the double-sided mapping issue, accurately identifying the front and back sides of walls and generating precise maps. The mapping results of different algorithms in the simulation environment are shown in Fig. 6.

To validate the proposed method, we evaluated II-NVM using real and simulated data, focusing on its accuracy in restoring wall thickness in double-sided areas.

In the experiment, point cloud data from indoor scenes was collected and processed using the II-NVM algorithm. To evaluate the algorithm's performance in double-sided areas, we extracted the double-sided regions from the point cloud after building the map and used a plane fitting method to determine the plane equation of one side of the wall. Based on the fitting results, the average distance between the plane of one side of the wall and the other was calculated as an evaluation metric for wall thickness, providing a quantitative assessment of the algorithm's accuracy in real scenarios. Since other SLAM algorithms cannot effectively address double-sided mapping issues and cannot measure wall thickness, this experiment further demonstrated the effectiveness of the proposed method in handling double-sided areas and restoring wall thickness through comparison with real point cloud data and some simulated point cloud data. Fig. 7 shows the mapping results in real scenarios, with numbered walls selected for further analysis and comparison.

Since the simulation environment offers precise control over wall geometry and thickness, we conducted additional experiments to evaluate II-NVM's performance across walls of varying thicknesses. By simulating seven wall areas with varying thicknesses, we further validated the adaptability and robustness of the proposed method under different conditions.

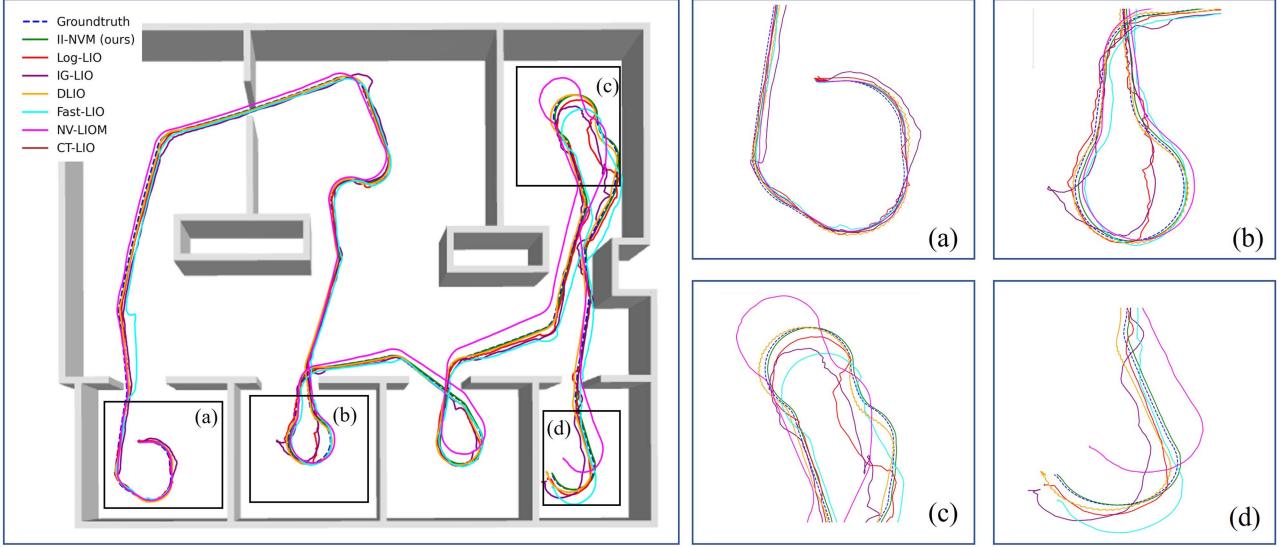


Fig. 5. Comparison of localization estimates from different algorithms.

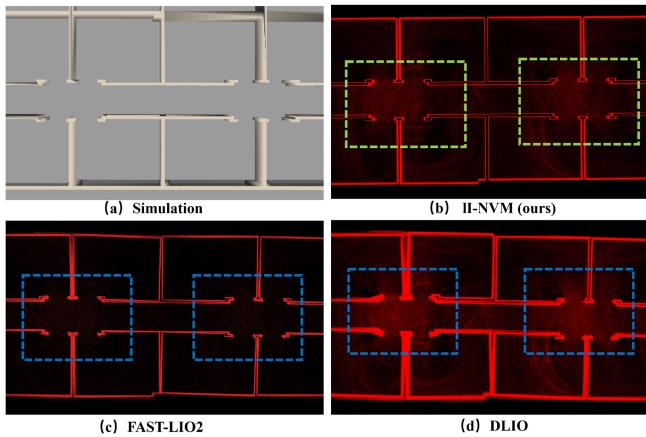


Fig. 6. Mapping results of different algorithms in simulation.

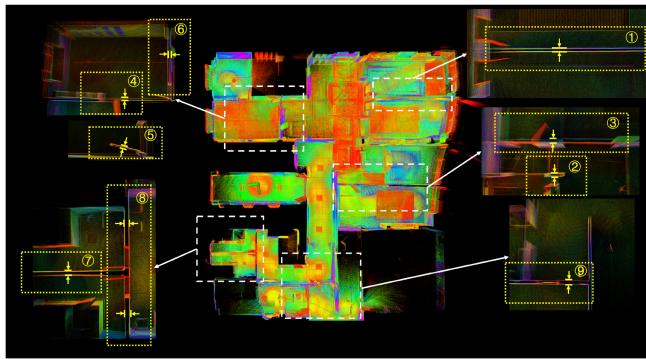


Fig. 7. Mapping results of walls and doors in a real scenario.

The experimental results, presented in Tables IV and V, show the differences and relative errors between the actual wall thickness and the thickness estimated by the II-NVM method. Overall, the proposed method demonstrates high accuracy in

TABLE IV
COMPARISON OF II-NVM DOUBLE-SIDED MAPPING PERFORMANCE ACROSS DIFFERENT WALL THICKNESSES IN REAL-WORLD SCENARIOS

Area	Real Thickness (cm)	II-NVM (cm)	Percentage Change
1	12.0	10.554	-12.05%
2	12.0	12.242	+2.02%
3	11.9	12.026	+1.06%
4	11.9	11.652	-2.08%
5	4.0	4.585	+14.63%
6	9.1	9.550	+4.95%
7	15.5	15.937	+2.82%
8	15.9	14.737	-7.32%
9	12.3	12.754	+3.69%

TABLE V
COMPARISON OF II-NVM DOUBLE-SIDED MAPPING ACCURACY ACROSS DIFFERENT WALL THICKNESSES IN SIMULATION

Area	Real Thickness (cm)	II-NVM (cm)	Percentage Change
1	15.0	15.073	+0.49%
2	13.0	13.127	+0.98%
3	11.0	11.002	+0.02%
4	9.0	9.161	+1.79%
5	7.0	7.342	+4.89%
6	5.0	5.032	+0.64%
7	3.0	2.789	-7.03%

estimating wall thickness in most scenarios, confirming its effectiveness in mapping double-sided areas. While some errors were observed in certain cases, the method exhibited good stability and reliability overall. In real-world wall thickness tests, the II-NVM method maintained measurement errors within 1% for thicker targets (such as 15 cm and 13 cm), highlighting its excellent reliability. For thinner targets (such as 3 cm and 4 cm),

TABLE VI
COMPARISON OF SLAM OPERATION PERFORMANCE WITH AND WITHOUT
LRU CACHE

Operation	II-NVM (s)	II-NVM_w/o LRU (s)	Percentage Change
Map Update	0.19	0.70	+72.85%
Optimize	3.34	4.60	+27.39%
Pose Estimate	3.58	5.29	+32.33%
Process Measurement	3.96	5.68	+30.28%
Total Time	12.07	20.21	+40.28%

although the error slightly increased, it remained below 10%. These results further validate the effectiveness and practicality of the II-NVM method in addressing centimeter-level double-sided mapping challenges.

D. Evaluation of Processing Time

By using a normal vector voxel map with an LRU cache, we ensure efficient response and updating of the voxel map, even in complex scenarios. This method enhances storage efficiency while maintaining real-time performance, providing stronger support for the widespread use of voxel maps in real-time applications. In this section, we will examine how the LRU cache module compares to the radius-based map management module (CT-LIO) in terms of time consumption for map updating, state estimation, pose optimization, and measurement processing for each sequence, as shown in the table.

The evaluation of processing time, presented in Table VI, shows that integrating the normal vector voxel map with an LRU cache significantly enhances the processing efficiency of II-NVM and reduces overall time consumption. The processing time for all major operations was notably decreased, indicating that this method effectively enhances both the real-time performance and storage efficiency in complex scenarios, providing robust support for real-time applications.

V. CONCLUSION AND FUTURE WORK

The letter addresses the prevalent challenge of double-sided mapping in SLAM systems, particularly in indoor scenarios, by proposing a solution anchored in normal vector consistency. This solution involves extending the storage of normal vector data within voxel blocks, alongside implementing an adaptive radius KD-tree search method and a view consistency determination mechanism, which effectively resolves the difficulty in distinguishing the front and back sides of objects. Experimental results demonstrate that this method significantly improves mapping accuracy in complex indoor environments and reduces double-sided mapping errors. Furthermore, by the integration of an LRU caching strategy enables efficient incremental updates of the voxel map, boosting real-time performance while maintaining accurate mapping.

Future research could develop this method as a standalone plugin, facilitating seamless integration with other LIO systems to further enhance the mapping accuracy of various SLAM systems in complex indoor environments.

REFERENCES

- [1] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.
- [2] H. Ismail, R. Roy, L. J. Sheu, W. H. Chieng, and L. C. Tang, "Exploration-based SLAM (e-SLAM) for the indoor mobile robot using lidar," *Sensors*, vol. 22, no. 4, 2022, Art. no. 1689.
- [3] M. Palieri et al., "LOCUS: A multi-sensor lidar-centric solution for high-precision odometry and 3D mapping in real-time," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 421–428, Apr. 2021.
- [4] A. Reinke et al., "LOCUS 2.0: Robust and computationally efficient LiDAR odometry for real-time 3D mapping," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9043–9050, Oct. 2022.
- [5] J. Camiña, L. J. Sánchez-Aparicio, C. M. Corrochano, D. Sanz-Arauz, and D. González-Aguilera, "Analysis of a SLAM-based laser scanner for the 3D digitalization of underground heritage structures. A case study in the wineries of baltanas (palencia, Spain)," in *Proc. Int. Conf. Florence Heri-Tech: Future Heritage Sci. Technol.*, Springer, 2022, pp. 42–56.
- [6] 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*, IEEE, 2023, pp. 48–53.
- [7] K. Huang et al., "LOG-LIO2: A LiDAR-inertial odometry with efficient uncertainty analysis," 2024, *arXiv:2405.01316*.
- [8] J. Zhang and S. Singh, "Loam: LiDAR odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, Berkeley, CA, 2014, vol. 2, pp. 1–9.
- [9] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [10] W. Yu et al., "I²EKF-LO: A dual-iteration extended Kalman filter based LiDAR odometry," 2024, *arXiv:2407.02190*.
- [11] J. Xu et al., "M-DIVO: Multiple ToF RGB-D cameras enhanced depth-inertial-visual odometry," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 37562–37570, Dec. 2024.
- [12] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," *Robot.: Sci. Syst.*, vol. 2018, 2018, Art. no. 59.
- [13] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "LIPS: LiDAR-inertial 3D plane SLAM," in *Proc. 2018 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, IEEE, 2018, pp. 123–130.
- [14] 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. Automat. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.
- [15] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online LiDAR odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8518–8525, Jul. 2022.
- [16] P. Dellenbach, J. E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time elastic LiDAR odometry with loop closure," in *Proc. 2022 Int. Conf. Robot. Automat.*, IEEE, 2022, pp. 5580–5586.
- [17] L. Zhou, D. Koppel, and M. Kaess, "LiDAR SLAM with plane adjustment for indoor environment," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7073–7080, Oct. 2021.
- [18] H. Badino, D. Huber, Y. Park, and T. Kanade, "Fast and accurate computation of surface normals from range images," in *Proc. 2011 IEEE Int. Conf. Robot. Automat.*, IEEE, 2011, pp. 3084–3091.
- [19] K. Huang, J. Zhao, Z. Zhu, C. Ye, and T. Feng, "LOG-LIO: A LiDAR-inertial odometry with efficient local geometric information estimation," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 459–466, Jan. 2024.
- [20] D. Chung and J. Kim, "NV-LIO: LiDAR-inertial odometry using normal vectors towards robust SLAM in multifloor environments," 2024, *arXiv:2405.12563*.
- [21] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM." 2017. [Online]. Available: <https://github.com/MichaelGrupp/evo>