applied sciences

MDPI

Article

Two-Dimensional LiDAR SLAM with Multi-Resolution Distance-Field Matching, Adaptive Keyframes, and Loop Closure

Yuxi Chen [1,2,3], Bin Zhao [1,2,*], Ruohuai Sun [1,2,*] and Qian Zhang [4]

[1] College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[2] College of Interdisciplinary Sciences, Liaoning University of Technology, Jinzhou 121001, China
[3] National School of Elite Engineers, Northeastern University, Shenyang 110819, China
[4] School of Electronics & Information Engineering, Liaoning University of Technology, Jinzhou 121001, China; zhang_q0914@163.com
* Correspondence: jckx_zhaobin@lnut.edu.cn (B.Z.); sunruohuai@stumail.neu.edu.cn (R.S.); Tel.: +86-15-9404-11119 (R.S.)

Abstract

In two-dimensional laser SLAM exploration and mapping in complex indoor environments, scan matching is sensitive to the initial pose, error accumulation, and redundant map updates. To address these issues, an MDF-ALC-SLAM method is proposed. Firstly, a multi-resolution distance field is constructed at the front end, and a coarse-to-fine scan matching strategy is adopted to optimize the pose estimation method. Second, it selects adaptive keyframes based on translation displacement, rotation angle, and matching error. Third, the back end uses polar-coordinate-space geometric descriptors to search for candidate closed loops and performs pose graph optimization after geometric verification, achieving loop detection and map correction. Finally, offline replay simulation experiments and SIASUN smart park physical experiments were conducted. The offline experiment used manually annotated reference positions and headings, with trajectory deviation ATPE being 0.042 m and ATHE being 1.54°, and the average processing time per frame was 68 ms. The on-site experiment conducted 30 autonomous return tests and used a laser tracker to measure the return pose. ATPE was 0.053 m, and ATHE was 1.94°; on the RK3399 platform, the algorithm's average processing time per frame was 73 ms. The on-site return error reflects system performance under the joint action of positioning and motion control, which is not equivalent to SLAM's overall trajectory positioning error. It demonstrates the correctness and superiority of the proposed method.

Keywords: 2D LiDAR SLAM; scan matching; multi-resolution distance field; adaptive keyframe; loop-closure detection; pose-graph optimization

Academic Editors: Yutaka Ishibashi, Manuel Filipe Pereira da Cunha Martins Costa and Sandra Franco

Received: 23 September 2026
Revised: 22 October 2026
Accepted: 27 October 2026
Published: 29 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the rapid development of artificial intelligence and robotics technology, the demand for autonomous exploration and autonomous driving systems for updating maps, autonomous positioning, and autonomous decision-making in unknown environments is increasing day by day [1,2]. Simultaneous Localisation and Mapping (SLAM) can simultaneously estimate the robot's position and construct an environmental map without prior maps, and is widely used in commercial, elderly care, industrial inspection,

warehouse logistics, intelligent manufacturing, and autonomous driving fields [3–5]. As the core capability of mobile robots, autonomous navigation accuracy and robustness directly affect task execution efficiency and system safety. Traditional navigation methods mainly rely on visual SLAM, laser SLAM, or multi-sensor fusion, using feature matching, pose estimation, and map optimization to achieve autonomous positioning and path planning in unknown or semi-unknown environments. These methods perform well in structured and textured environments, but still face problems in environments with degraded geometric features and sparse natural features [6–10]. SLAM navigation methods can be classified into visual, laser radar, and multimodal methods according to the main sensor types. Visual SLAM utilises texture and semantic information, while laser radar provides direct geometric measurements and is usually insensitive to illumination changes. Laser and multi-sensor SLAM systems often adopt methods such as scan matching, graph optimization, or a combination of both, and representative systems include Gmapping, Hector SLAM, Karto/SPA, and Cartographer. Their performance is affected by factors such as initial pose, scene geometric structure, loop closure reliability, and computing resources. Multimodal SLAM methods achieve sensor information complementarity and consistent map construction in complex environments by fusing complementary information from vision, laser radar, and inertial sensors.

Classic navigation methods usually rely on laser radar or visual SLAM to construct geometric maps and then perform path planning and motion control. Domestic and international research on improved scan matching and positioning methods for laser SLAM has been in-depth. Ran et al. [9] systematically reviewed the development of the SLAM field, pointing out that scan matching and loop closure detection are important links for reliable positioning and mapping. Tan et al. combined RGB and three-dimensional semantic features, used voxel-level semantic reconstruction and an LSTM navigation model to construct a visual language navigation framework, and reported performance improvements on the R2R and R4R datasets [11]. Jiao et al. achieved online metric semantic mapping for large-scale outdoor environments through the fusion of laser, vision, and inertial information, as well as GPU-accelerated global metric semantic grid generation [12]. Zhang et al. proposed SUI-SLAM, combining semantic segmentation and geometric uncertainty modelling to handle dynamic scenes [13]. Wang et al. achieved better results than ORB-SLAM2 by combining motion cues and semantic priors to filter visual features on the TUM RGB-D dynamic dataset, improving the utilisation of effective laser observations [14]. These studies improve the accuracy and robustness of registration from different perspectives. Loop closure detection and location identification are important for reducing cumulative drift in two-dimensional laser SLAM. These studies show that a discriminative geometric representation and reliable constraint verification are very important. Although multi-resolution scanning matching, motion-based key frame selection, and descriptor loop closure detection have been widely studied, their application in two-dimensional laser SLAM still has problems: local scanning matching is easily affected by initial pose errors, traditional motion threshold key frame strategies do not explicitly consider the quality of registration, descriptor retrieval may produce ambiguous candidates in repetitive geometric environments, and in large scene ranges, mapping may cause inaccurate positioning problems.

To address these issues, this paper proposes a two-dimensional laser SLAM algorithm that integrates multi-resolution distance field matching and adaptive loop closure optimization, named MDF-ALC-SLAM (Multi-resolution Distance Field Matching and Adaptive Loop Closure SLAM). The main contributions are as follows:

(1) A two-dimensional laser SLAM pose estimation method is proposed, which adopts a hierarchical distance field scanning matching strategy to optimize the pose level by level at multiple spatial resolutions. The coarse-layer estimate is used to initialise

subsequent fine-layer optimization, reducing local matching sensitivity to initial pose deviation while retaining fine-scale pose optimization.

(2) A redundant map update method is proposed, which uses a key frame selection strategy based on matching quality perception, jointly considering translational displacement, rotational displacement, and scan matching residuals. This enables the map update frequency to respond simultaneously to the robot's movement and matching quality, reducing redundant map updates.

(3) A loop detection and map correction method is proposed, constructing a polar grid space geometric descriptor for the retrieval of two-dimensional laser loop candidates. This descriptor characterises the spatial structure by combining point distribution, average radial distance, and angular distance variation; it fuses multiple feature channels via weighted similarity and uses geometric consistency verification to generate reliable loop constraints.

The remaining parts of this paper are arranged as follows: Section 2 introduces the front-end structure of MDF-ALC-SLAM, including the laser measurement model, distance field construction, scan matching optimization, and key frame map update strategy; Section 3 introduces loop detection and pose graph optimization based on structured laser descriptors; Section 4 conducts offline replay simulation experiments with public data and SIASUN Smart Park physical experiments, using the autonomous return test evaluation algorithm to estimate the navigation performance; Section 5 summarises the conclusions and looks forward to future work.

2. Overview of the Two-Dimensional Laser MDF-ALC-SLAM Algorithm

To verify the accuracy of the MDF-ALC-SLAM algorithm proposed in this paper for mapping in complex office environments and its positioning precision, we adopted a commercial robot experimental platform, as shown in Figure 1. This platform integrates mapping, positioning and path planning. A depth camera is used for obstacle avoidance, and sensors such as laser radar are used for environmental perception. This device supports the embedded deployment of deep learning models. The multi-sensor fusion of the 2D laser SLAM front end estimates pose from laser observations and uses the results to update the map. Loop closure detection provides robot pose estimation, thereby generating high-precision and low-drift grid maps.



Figure 1. Experimental robotic platform used for 2D LiDAR SLAM.

The core features of laser SLAM include: (1) odometer correction based on scan matching; (2) global consistency based on graph optimization; (3) loop closure detection. Its main goal is to generate high-precision, low-drift grid maps while providing robot pose

estimation. In mobile robot navigation, the optimization process usually first estimates the current frame's pose through local scan matching and integrates consecutive frames to build a local map subgraph. On this basis, it combines global scan matching and loop-closure detection to extract constraint information, then minimises global pose error through graph optimization to improve overall map consistency. Figure 2 shows the overall framework of MDF-ALC-SLAM based on multi-resolution distance field matching and adaptive loop closure optimization: A two-dimensional laser SLAM pose estimation method is proposed, which is based on coarse-to-fine scan matching of multi-resolution distance fields to reduce the sensitivity to initial pose errors; adaptive key frame selection is carried out by jointly considering robot translation displacement, rotation angle, and scan matching error, reducing redundant map updates; a method combining polar grid space geometric descriptors, geometric verification, and pose graph optimization is adopted to achieve loop closure detection and map correction.

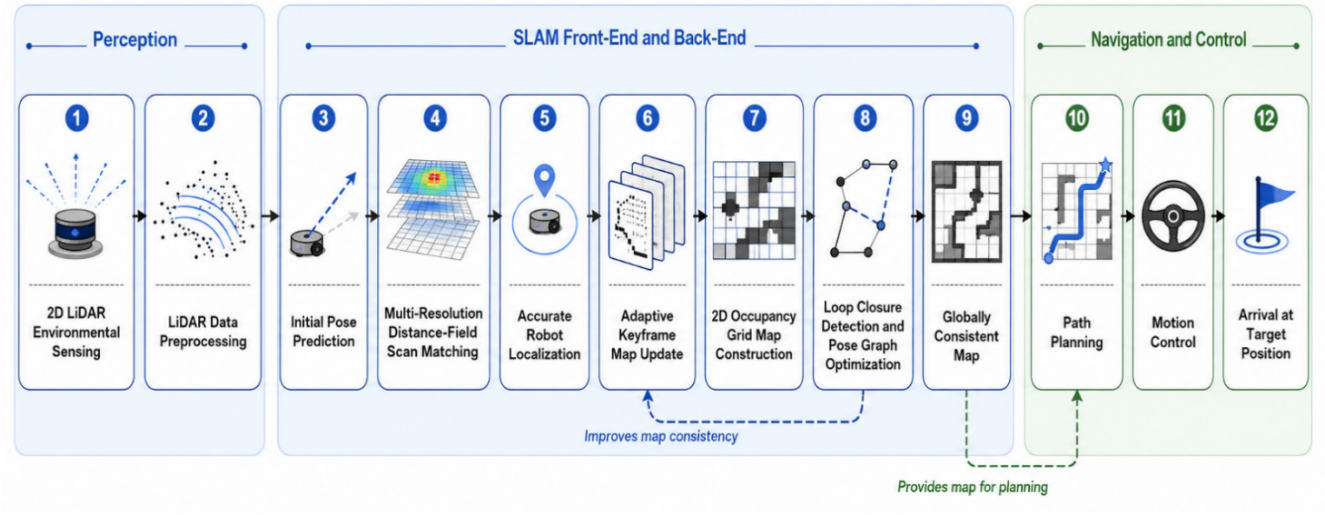


Figure 2. Overall framework of the proposed 2D LiDAR SLAM algorithm based on multi-resolution distance-field matching and adaptive loop-closure optimization.

3. 2D LiDAR SLAM Front End Based on Multi-Resolution Distance-Field Scan Matching

3.1. 2D LiDAR Map Modeling and Distance-Field Construction

The RPLIDAR C1 two-dimensional laser radar periodically scans the surrounding environment of the robot, measuring the relative distances of obstacles from the sensor coordinate system, providing perception observation data for positioning and mapping [15–20]. The original measurement of a single laser beam is expressed as:

$$z_i = (r_i, \theta_i) \tag{1}$$

Among them, z_i represents the original polar-coordinate measurement for the i-th laser beam, r_i is the measured distance between the radar and the obstacle, and θ_i is the corresponding angle.

The polar measurement is converted to Cartesian coordinates in the LiDAR frame as

$$p_i^L = \begin{bmatrix} x_i^L \\ y_i^L \end{bmatrix} = \begin{bmatrix} r_i \cos\theta_i \\ r_i \sin\theta_i \end{bmatrix} \tag{2}$$

In the formula, p_i^L represents the i-th point in the laser radar coordinate system L; x_i^L and y_i^L are the Cartesian coordinate position values of the i-th point.

In view of this, the complete laser scanning data Z can be expressed as:

$$Z = \{p_1^L, p_2^L, \ldots, p_N^L\} \tag{3}$$

In the formula, N represents the number of valid measurement points currently being scanned; p_i^L refers to the i-th point in the laser radar coordinate system.

In a 2D LiDAR SLAM system, the robot state is represented as

$$\xi = [x, y, \theta]^T \tag{4}$$

In the formula, ξ represents the pose vector of the planar robot, x and y are the position coordinates in the world coordinate system, and θ is the heading angle (the direction in which the robot is facing). T represents the matrix transpose.

The rotation matrix and translation vector from the robot frame to the world frame are given by

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, t = \begin{bmatrix} x \\ y \end{bmatrix} \tag{5}$$

In the formula, $R(\theta)$ is the 2D rotation matrix, t is the translation vector $t = [x, y]^T$, and x, y, and θ are the pose components defined in Equation (4).

The corresponding LiDAR point in the world frame is then expressed as

$$\begin{cases} p_i^W = T(\xi)p_i^L \\ p_i^W = R(\theta)p_i^L + t \end{cases} \tag{6}$$

Here, $T(\xi)$ denotes the rigid-body pose transformation parameterized by ξ, $R(\theta)$ is the rotation matrix, and t is the translation vector. The point p_i^L is the ith measurement in the LiDAR frame L, whereas p_i^W is the same point expressed in the world frame W.

The scan-matching objective is defined as

$$E(\xi) = \sum_{i=1}^{N} D(T(\xi)p_i^L)^2 \tag{7}$$

Among them, $E(\xi)$ is the scan-matching objective, N is the number of LiDAR points in the current scan, $D(\cdot)$ is the distance-field function, and $T(\xi)p_i^L$ is the transformed position of the ith LiDAR point in the map frame. At the correct pose, transformed points should lie close to mapped obstacle boundaries, yielding small distance-field values.

For the th LiDAR point, the residual is defined as

$$e_i(\xi) = D(T(\xi)p_i^L) \tag{8}$$

where e_i is the residual associated with the ith LiDAR point. Thus, the single-point residual equals the distance-field value evaluated at the transformed map position $T(\xi)p_i^L$.

The optimal robot pose is therefore given by

$$\xi^* = \text{argmin}_\xi E(\xi) \tag{9}$$

In the formula, ξ^* denotes the pose that minimizes the objective $E(\xi)$, and $arg\ min$ denotes the minimizing argument. The coordinate transformation maps LiDAR observations acquired at different times into a common world coordinate frame, providing a consistent representation for subsequent map construction and pose optimization.

The objective in Equation (7) is an unweighted squared-distance least-squares cost. Consequently, isolated dynamic returns or gross outliers can exert disproportionate influence on the estimate. The present formulation therefore assumes that the majority of matched returns arise from a predominantly static scene; robust losses or explicit dynamic-object rejection are natural extensions for strongly dynamic environments.

https://doi.org/10.3390/app162110726

A two-dimensional occupancy-grid map (OGM) provides a probabilistic representation of the robot workspace by discretizing the environment. Let the environment be divided into a regular grid:

$$M = \{m_1, m_2, \ldots, m_n\} \tag{10}$$

where M is the occupancy-grid map, m_i denotes the ith map cell, and n is the total number of grid cells.

The relationship between physical coordinates and grid indices is given by

$$x = ir_m, y = jr_m \tag{11}$$

Among them, i and j are the integer grid indices along the map axes; x and y are the corresponding physical coordinates; and r_m is the map resolution in meters per grid cell.

To improve the numerical stability of occupancy updates, the state of each cell is represented in log-odds form:

$$L(m) = \log \frac{P(m)}{1 - P(m)} \tag{12}$$

where $P(m)$ is the occupancy probability of cell m, and $L(m)$ is its log-odds representation. This form converts Bayesian occupancy updates into additive increments and improves numerical stability.

For a cell traversed by a LiDAR ray, the free-space update is

$$L_t(m) = L_{t-1}(m) + l_{\text{free}} \tag{13}$$

For a cell at which the ray terminates on an obstacle, the occupancy update is

$$L_t(m) = L_{t-1}(m) + l_{occ} \tag{14}$$

Among them, $L_t(m)$ and $L_{t-1}(m)$ are the current and previous log-odds values of cell m; t is the update index; and l_{free} and l_{occ} are the free-space and occupied-space update increments, respectively. The condition $l_{\text{free}} < 0$ decreases occupancy probability, whereas $l_occ > 0$ increases it.

Continuous fusion of laser observations can yield a two-dimensional occupancy map. Occupancy grid mapping and scan matching are widely used in two-dimensional laser SLAM [21–26]. Although occupancy value interpolation can also be used for continuous pose evaluation, this paper directly represents the proximity of the map position to the obstacle by the distance field. It builds the matching residuals based on this.

The occupancy map is transformed into a distance field through Euclidean distance transformation. An efficient and accurate distance transformation algorithm on sampled grids provides a standard computational basis for this operation [27,28], and the resulting distance metric is used for multi-resolution pose estimation.

The distance field function at any map position is defined as:

$$\begin{aligned} D(q) &= \min_{o \in O} \|q - o\|_2 \\ D(x, y) &= \min_{(u,v) \in O} ((x-u)^2 + (y-v)^2)^{1/2} \end{aligned} \tag{15}$$

where q denotes an arbitrary map position, O is the set of obstacle cells, and o is an obstacle-cell position. In Cartesian form, (x, y) denotes the query position and (u, v) an obstacle-cell coordinate. The function $D(q)$ gives the Euclidean distance to the nearest obstacle; $D(q) = 0$ indicates that the query position lies on an obstacle boundary.

3.2. Scan-Matching Optimization Based on Multi-Resolution Distance Fields

The obstacle distance field provides spatial residuals for aligning laser points with the map, and its local derivatives are used for pose optimization. The precise distances of the discrete occupancy point set are not necessarily differentiable at positions where there are occupancy points or multiple nearest points. Therefore, the actual gradient used by the optimizer depends on the numerical field representation and interpolation method [29–33].

The gradient of the distance field is defined as:

$$\nabla D(q) = \begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \end{bmatrix} \tag{16}$$

Among them, $\nabla D(q)$ is the spatial gradient of the distance field at position q, and $\partial D/\partial x$ and $\partial D/\partial y$ are its partial derivatives along the map x and y axes. The gradient points in the direction of the steepest local increase in distance and enters pose optimization through the residual Jacobian.

Given $e_i(\xi) = D(T(\xi)p_i^L) = D(p_i^W)$, the derivative of the distance-field residual with respect to the robot pose follows from the chain rule:

$$\frac{\partial D}{\partial \xi} = \frac{\partial D}{\partial p_i^W}\frac{\partial p_i^W}{\partial \xi} = \frac{\partial D}{\partial p_i^W}\frac{\partial T(\xi)p_i^L}{\partial \xi} \tag{17}$$

In the formula, $\partial D/\partial p_i\hat{}W$ is the spatial derivative of the distance field evaluated at the transformed point p_i^W, and $\partial p_i\hat{}W/\partial \xi$ is the derivative of that point with respect to the pose vector ξ. A first-order Taylor expansion about the current pose estimate gives

$$\begin{gathered} e(\xi + \Delta\xi) \approx e(\xi) + J\Delta\xi \\ J_i = (\nabla D(p_i^W))^T \frac{\partial (T(\xi)p_i^L)}{\partial \xi} \\ \frac{\partial (T(\xi)p_i^L)}{\partial \xi} = \begin{bmatrix} 1 & 0 & -x_i^L sin\,\theta - y_i^L cos\,\theta \\ 0 & 1 & x_i^L cos\,\theta - y_i^L sin\,\theta \end{bmatrix} \end{gathered} \tag{18}$$

Here, e is the stacked scan-residual vector, J is its Jacobian, J_i is the Jacobian row associated with the ith LiDAR point, and $\Delta\xi$ is the pose increment. The quantities x_i^L and y_i^L are the LiDAR-frame coordinates of point i. The spatial gradient in Equation (16) is a column vector, so its transpose appears in Equation (18).

The Gauss–Newton increment satisfies the normal equations:

$$(J^T J)\Delta\xi = -J^T e \tag{19}$$

where J^T denotes the transpose of the Jacobian, e is the stacked residual vector, and $\Delta\xi$ is the Gauss–Newton pose increment. A unique increment requires J to have full column rank; otherwise the normal equations can become ill-conditioned.

The robot pose is then updated as:

$$\xi_{k+1} = \xi_k + \Delta\xi \tag{20}$$

Among them, ξ_k and ξ_{k+1} are the pose estimates at iterations k and $k+1$, respectively, and $\Delta\xi$ is the increment obtained from Equation (19). Iterative optimization seeks a local pose estimate that reduces the scan-to-map residuals; it does not guarantee a globally optimal alignment.

Local distance-field optimization can be sensitive to initialization. Large inter-scan motion, occlusion, or repetitive geometry can cause convergence to an incorrect alignment, a general limitation of local scan registration that motivates coarse-to-fine or broader-search initialization strategies.

A coarse-to-fine multi-resolution optimization strategy is therefore proposed.

Distance fields are first constructed at multiple resolutions, where denotes the scale level.

The distance-field pyramid is constructed as

$$D_l(x, y) = \text{Downsample}(D_{l-1}(x, y)) \tag{21}$$

In the formula, D_0 is the original finest-resolution distance field, D_l denotes the distance field at pyramid level l, $Downsample(\cdot)$ is the downsampling operator, and L is the index of the coarsest level. The level index satisfies $l = 0,1, \dots, L$. Distance values are expressed in physical units at every level, and optimization traverses the levels from coarse to fine.

Optimization proceeds from the coarsest level to the finest level. At each level, the search region is described by a vector of nonnegative translational and angular half-widths:

$$b_l = \left[b_{x,l}, b_{y,l}, b_{\theta,l}\right]^T \tag{22}$$

where b_l is the search half-width vector at pyramid level l; $b_{x,l}$, $b_{y,l}$, and $b_{\theta,l}$ are the allowable half-widths in the x, y, and heading directions, respectively. The coarsest level provides the initial pose estimate for the next finer level.

As the optimization moves to a finer level, the search half-widths do not increase:

$$b_{l-1} \preccurlyeq b_l, \quad l = L - 1, \dots, 1 \tag{23}$$

In the formula, the symbol $\preccurlyeq$ denotes component-wise inequality, so every search half-width at level $l-1$ is no larger than the corresponding half-width at level l. The final refinement minimizes the matching objective at the finest level:

$$\xi^* = arg\min_{\xi} E_0(\xi) \tag{24}$$

Among them, $E_0(\xi)$ is the finest-level matching objective and ξ^* is the final optimized pose. This hierarchical strategy is intended to improve initialization for fine-scale scan matching, although ambiguity can remain in repetitive or weakly constrained scenes.

3.3. Adaptive Keyframe-Based Map Updating Using Motion and Matching Error

The traditional SLAM navigation algorithm integrates each frame of laser scanning into the map, which easily leads to the introduction of redundant observations, thereby causing problems such as increased map maintenance costs [34]. To address this issue, this paper designs an adaptive keyframe selection method.

The definition of the robot's translational displacement is:

$$\Delta s = \sqrt{\Delta x^2 + \Delta y^2} \tag{25}$$

where Δs is the translational displacement relative to the last selected keyframe, and Δx and Δy are the corresponding coordinate changes.

The keyframe selection score is defined as:

$$Q = \alpha \Delta s + \beta |\Delta \theta| + \gamma E_m \tag{26}$$

Here, α, β, and γ weight translational motion, angular motion, and matching error, respectively. The implementation uses scaled coefficients so that the three terms contribute on a comparable, approximately dimensionless basis. The coefficients were designed from reference scales of 0.20 m translation, 10° rotation, and 0.05 m scan-matching error, corresponding to normalized importance weights of 0.45, 0.25, and 0.30. Accordingly, α = 2.25 m^{-1}, β = 1.43 rad^{-1}, and γ = 6.00 m^{-1}. Δs and $|\Delta \theta|$ denote motion relative to the last selected keyframe, and E_m denotes the scalar scan-matching error. The keyframe

score threshold is Q_{th} = 1.0. When $Q > Q_{th}$, the current LiDAR scan is selected as a keyframe.

The map is then updated using the selected keyframe:

$$M_k = M_{k-1} \oplus Z_k \tag{27}$$

In the formula, M_k and M_{k-1} are the updated and previous maps, respectively; Z_k is the selected keyframe scan; k is the keyframe index; and the $\oplus$ operator denotes integration of Z_k into the map using its estimated pose.

The score makes the map-update frequency responsive to robot motion and scan-to-map agreement. Its weights and threshold are fixed parameters; adaptation arises from the changing inputs to the score. Because a large matching error may indicate either a useful new observation or failed registration, the implementation additionally rejects map integration when E_m exceeds E_{max} = 0.10 m. Thus, a scan is integrated only when $Q > Q_{th}$ and $E_m \leq E_{max}$. This prevents grossly misregistered scans from being promoted solely because they produce a large residual.

4. SLAM Back End Based on a Spatial-Geometric LiDAR Descriptor and Loop-Closure Optimization

Closed-loop detection and global optimization are crucial steps for reducing long-term drift in two-dimensional laser SLAM [34,35]. The front-end scan matching estimates the local motion between adjacent scans, but sensor noise, motion estimation errors, and repeated geometric structures can lead to cumulative drift. Therefore, this paper designs a combined spatial-geometry encoding laser structure descriptor. First, candidate closed loops are generated through fast keyframe retrieval. Incorrect matches are rejected using laser scan geometric consistency, and finally the global trajectory is corrected through pose graph optimization.

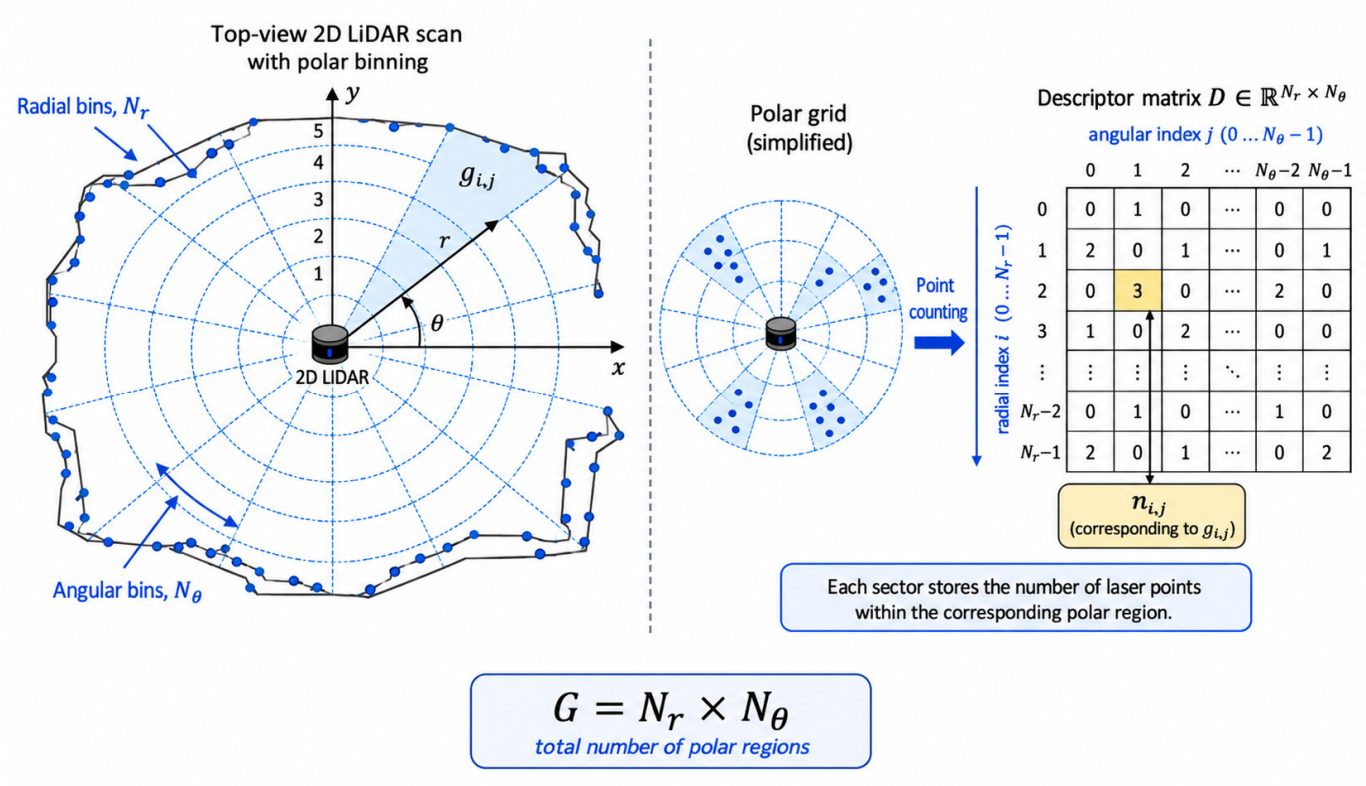
4.1. Construction of a 2D LiDAR Structural Descriptor Based on Joint Spatial-Geometric Encoding in Polar Coordinates

The core task of closed-loop detection is to determine whether the current environment of the robot has been visited before. Laser data collected at different times may change due to rotation, translation, and local occlusion. Directly matching the laser point sets incurs high computational costs and is also susceptible to environmental changes.

We construct a joint spatial-geometric descriptor from keyframe LiDAR data. Each polar cell records the point count, mean radial distance, and local structural variation, yielding a compact representation of the environment. As shown in Figure 3, the scan space is divided into polar cells:

$$Gr = N_r \times N_\theta \tag{28}$$

Here, G_r denotes the polar-grid partition, N_r and N_θ are the numbers of radial rings and angular sectors, respectively, and N_rN_θ is the total number of regions. The illustrated region in Figure 3 contains three LiDAR points.



(**a**) (**b**)

Figure 3. (**a**). Polar grid partitioning of 2D LiDAR point clouds. (**b**). Descriptor construction based on grid point counts.

The LiDAR-point count in each region is defined as

$$n_{ab} = \sum_{i=1}^{N} I(p_i \in C_{ab}) \tag{29}$$

Here, n_{ab} is the number of LiDAR points in region C_{ab}; a indexes the radial ring; b indexes the angular sector; N is the number of points in the scan; and $I(\cdot)$ is the indicator function, equal to 1 when point p_i belongs to C_{ab} and 0 otherwise.

For a nonempty region, the mean radial-distance feature is defined as

$$d_{ab} = \frac{1}{n_{ab}} \sum_{i:\, p_i \in C_{ab}} r_i, \quad n_{ab} > 0 \tag{30}$$

where d_{ab} is the mean radial distance in region C_{ab}, r_i is the radial distance of point p_i, and n_{ab} is the region point count. Equation (30) applies only when $n_ab > 0$; an empty region has no defined sample mean. Compared with point counts, this feature additionally represents obstacle-distance scale.

For two neighboring, nonempty angular regions in the same radial ring, the mean-distance difference is defined below for $b = 2, \ldots, N_\theta$:

$$\Delta d_{ab} = d_{ab} - d_{a(b-1)} \tag{31}$$

where Δd_{ab} is the mean-distance difference between neighboring angular sectors b and $b-1$ within the same radial ring a. The regional features are arranged into the following matrices:

$$\Delta D_k = \begin{bmatrix} \Delta d_{11} & \Delta d_{12} & \cdots & \Delta d_{1b} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta d_{a1} & \Delta d_{a2} & \cdots & \Delta d_{ab} \end{bmatrix} \tag{32}$$

where ΔD_k is the distance-variation matrix of keyframe k, whose entries are Δd_{ab}. The corresponding point-count and mean-distance matrices are denoted by N_k and D_k, respectively. The structured descriptor of the kth keyframe is then expressed as

$$F_k = [N_k, D_k, \Delta D_k] \tag{33}$$

where F_k is the complete descriptor of keyframe k; N_k, D_k, and ΔD_k are the point-distribution, mean-distance, and distance-variation matrices, respectively.

The point-count channel describes the distribution of returns, while the mean-distance channel describes their radial scale. The difference channel is a deterministic transformation of the mean-distance channel and emphasizes angular contrast in the matching score. It does not add independent geometric measurements.

4.2. Loop-Closure Candidate Retrieval Based on Spatial-Geometric Similarity

For the current keyframe descriptor, potential loop closures are retrieved from the historical keyframe database

$$\mathcal{F} = \{F_1, F_2, \ldots, F_M\} \tag{34}$$

to obtain candidate revisits. Here, F is the historical descriptor database, F_i is the descriptor of historical keyframe i, and M is the number of stored historical descriptors. The current descriptor is denoted by F_c. Temporally adjacent observations are not, by themselves, evidence of a loop closure.

A weighted similarity measure combines the three feature channels. In Equations (36)–(38), each feature matrix is flattened into a row vector using the same ordering, and the norms are Euclidean norms. The cosine expressions are defined for nonzero feature vectors:

$$S(F_c, F_i) = \lambda_1 S_N + \lambda_2 S_D + \lambda_3 S_{\Delta D} \tag{35}$$

where $S(F_c, F_i)$ is the combined similarity between the current and historical descriptors; λ_1, λ_2, and λ_3 are nonnegative channel weights; and S_N, S_D, and $S_{\Delta D}$ are the point-count, mean-distance, and distance-variation similarities, respectively. The spatial-distribution similarity is defined as

$$S_N = \frac{F_c^N (F_i^N)^T}{\|F_c^N\| \|F_i^N\|} \tag{36}$$

The distance-structure similarity is defined as

$$S_D = \frac{F_c^D (F_i^D)^T}{\|F_c^D\| \|F_i^D\|} \tag{37}$$

The distance-variation similarity is defined as

$$S_{\Delta D} = \frac{F_c^{\Delta D} (F_i^{\Delta D})^T}{\|F_c^{\Delta D}\| \|F_i^{\Delta D}\|} \tag{38}$$

In Equations (36)–(38), F_c^N, F_c^D, and $F_c^{\Delta D}$ denote the flattened feature vectors of the current descriptor, with the corresponding historical vectors indexed by i. The superscript T denotes transpose and $\|\cdot\|$ denotes the Euclidean norm. For nonzero feature vectors, the point-count and nonnegative distance channels have cosine similarities in [0, 1], whereas the signed distance-difference channel has similarity in [−1, 1]. Consequently, nonnegative weights satisfying $\lambda_1 + \lambda_2 + \lambda_3 = 1$ yield a bounded aggregate score. A historical keyframe becomes a candidate when the score exceeds the loop-closure threshold.

The combined score measures agreement between the three feature channels and is used to rank candidate revisits. Descriptor comparisons require consistent angular alignment. A high aggregate score does not guarantee that every channel is similar or that the

candidate is a valid loop closure. Channel-wise cosine normalization removes each channel's overall magnitude, so the distance channel represents the relative pattern of radial distances rather than absolute metric scale alone.

4.3. Loop-Closure Constraint Verification Based on Geometric Consistency

Descriptor similarity alone may produce false loop closures in long corridors, symmetric scenes, or highly repetitive environments. The proposed method therefore applies geometric consistency verification between the LiDAR scans.

Let the relative pose between the current keyframe and a candidate keyframe be

$$\Delta\xi_{ci} = [\Delta x, \Delta y, \Delta\theta]^T \tag{39}$$

where $\Delta\xi_{ci}$ is the relative pose from candidate keyframe i to current keyframe c, and Δx, Δy, and $\Delta\theta$ are its relative translation and heading components. Distance-field scan matching estimates this relative transform from the candidate keyframe to the current keyframe.

The loop-closure matching error is defined as

$$E_{loop} = \frac{1}{N}\sum_{j=1}^{N} D_c\left(T(\Delta\xi_{ci})p_j^i\right) \tag{40}$$

where E_{loop} is the loop-closure matching error, N is the number of points in the candidate scan, D_c is the distance field associated with the current keyframe, $T(\Delta\xi_ci)$ is the candidate-to-current transformation, and p_j^i is the jth point of candidate keyframe i. The loop-closure pose-constraint residual is defined as

$$e_{ci} = v(Z_{ci}^{-1}X_c^{-1}X_i) \tag{41}$$

Here, e_{ci} is the loop-closure pose residual, Z_{ci} is the measured transform from candidate keyframe i to current keyframe c, and X_c and X_i map the respective keyframe coordinates to the world frame. The function $v(\cdot)$ extracts the two translation components and heading from an SE(2) transform.

The residual compares the measured relative transform with the relative transform implied by the current pose estimates. Its translational components are expressed in meters and its angular component in radians.

4.4. 2D Pose-Graph Optimization with Loop-Closure Constraints

After loop-closure detection, the robot trajectory is represented as a pose graph, following the standard graph-based SLAM formulation in which poses are nodes and relative-pose measurements are edges [33]. Practical nonlinear least-squares solvers such as g2o provide a general optimization framework for these graph-structured estimation problems [34].

The node set contains the keyframe poses, and each edge represents a relative-pose constraint between two keyframes.

The pose-constraint residual is defined as

$$e_{ij} = v\left(Z_{ij}^{-1}X_i^{-1}X_j\right) \tag{42}$$

Here, e_{ij} is the pose-constraint residual between keyframes i and j, Z_{ij} is the measured relative transform, and X_i and X_j are their world-frame poses. Matrix products denote composition in SE(2), and $v(\cdot)$ uses the same translation-and-heading convention as Equation (41).

Minimizing the information-weighted residuals over the pose constraints gives a nonlinear least-squares problem:

$$\min_{X} \sum_{(i,j)\in C} e_{ij}{}^{T} \Omega_{ij} e_{ij}, \quad e_{ij} = \left[e_x, e_y, e_\theta\right]^T \tag{43}$$

Here, X denotes the set of pose variables, C is the set of pose constraints, e_{ij} is the residual for the constraint between keyframes i and j, and Ω_{ij} is its information matrix, conventionally the inverse residual covariance. The components e_x, e_y, and e_θ are the translational and angular residual terms.

The optimization over the full set of adjacent-keyframe and loop-closure constraints is written as

$$X^* = \arg\min_{X} \sum_{(i,j)\in C} e_{ij}{}^{T} \Omega_{ij} e_{ij} \tag{44}$$

Here, X^* is the optimized set of keyframe poses, X is the pose-variable set, C is the set of relative-pose constraints, and Ω_{ij} weights residual e_{ij}. The constraint set includes both adjacent-keyframe and accepted loop-closure measurements.

The node poses are iteratively adjusted to reduce the total weighted residual. Because relative constraints leave the global reference frame undetermined, a reference pose must be fixed. Nonzero residuals generally remain because the measurements contain noise.

The optimized pose set defines the global robot trajectory. Pose-graph optimization reduces drift accumulated during long-term operation and improves the overall consistency of the 2D map.

The back end combines descriptor-based candidate retrieval, geometric verification, and pose-graph optimization. Retrieval proposes possible revisits, verification assesses their geometric consistency, and graph optimization distributes accepted loop-closure corrections across the trajectory. These corrections complement the local estimates produced by the front end. A globally corrected map additionally requires the scan data to be placed consistently with the optimized keyframe poses.

5. Experimental Results and Analysis

5.1. Experimental Setup and Evaluation Methods

The experiment consisted of an offline playback of publicly available two-dimensional laser SLAM data from German museums and a real robot field test. The former compared the computational costs and trajectory deviation relative to the manually annotated reference for Gmapping, Hector SLAM, Karto SLAM, Cartographer, and MDF-ALC-SLAM, and analyzed the map results and module functions; the latter was conducted in the SIASUN Smart Park, evaluating the repeat mapping of the complete system and the position and heading errors upon returning to the starting target after mapping. The offline experiment used the complete data package b0-2014-07-11-10-58-16. bag, bag, without extracting local fragments. All offline algorithms and ablation configurations were run on the same computer equipped with an Intel Core i7-12700H and 16 GB of memory. Table 1 lists the offline and field platform configurations, and Table 2 lists the algorithm parameters of this paper. The amplitude of the translational disturbance is set to a maximum of 0.1 m, and the amplitude of the heading disturbance is set to a maximum of 5°.

All algorithms used the two-dimensional laser data in the same data package, and the odometer and IMU inputs were selected according to the actual configuration of each algorithm. The sensor differences belong to system configuration differences, so the comparison conclusion is only applicable to the tested configurations. The offline evaluation used the self-developed annotation tool, manually annotating the reference position and heading on the grid map. All algorithms shared the same fixed annotation, and the trajectory points output were paired with the reference points in the order of the route, with each pair of points corresponding to the same position on the route. The manual

annotation has uncertainty and cannot be regarded as an independent high-precision measurement of the true value.

Table 1. Platform configurations for offline data replay and field experiments.

Configuration	Description
Offline replay computing platform	Intel Core i7-12700H; 16 GB RAM
Onboard computing platform for field experiments	RK3399 board
Software environment for offline replay	Ubuntu 20.04; ROS Noetic; C++
Sensors and motion inputs for field experiments	360° 2D LiDAR; wheel odometry; inertial measurement unit (IMU)
LiDAR used in field experiments	RPLIDAR C1; 10 Hz
Reference measurement system for field experiments	Leica AT960 with T-Mac; 800 Hz

Table 2. Parameter settings for the MDF-ALC-SLAM framework.

Parameter	Value
Map resolution	0.05 m
Distance-field update frequency	10 Hz
Keyframe translation reference scale	0.2 m
Keyframe angular reference scale	10°
Loop-closure candidate similarity threshold	0.85
Maximum optimization iterations	20
Keyframe weight α	2.25 m^{-1}
Keyframe weight β	1.43 rad^{-1}
Keyframe weight γ	6.00 m^{-1}
Keyframe score threshold Q_{th}	1.0
Matching-error rejection threshold E_{max}	0.10 m
Distance-field pyramid levels	3 (0: finest; 2: coarsest)
Scan-matching solver	Gauss–Newton
Keyframe matching-error reference scale	0.05 m

The scan matching employs a three-level distance field pyramid and Gaussian-Newton optimization. The 0th layer is the finest layer, with a map resolution of 0.05 m; the 2nd layer is the coarsest layer. The optimization proceeds from coarse to fine, and the previous layer's result is used to initialize the next layer. The key frame reference scale participates in the weighted score, not as independent insertion triggering conditions; only when the score exceeds 1.0 and the matching error is no greater than 0.10 m will the scan be fused into the map. The closed-loop candidate similarity threshold of 0.85 does not replace the geometric verification in Section 3.3.

The computational cost is evaluated by the average processing time per frame of the algorithm, including scan matching, map update, closed-loop detection, and back-end optimization. The statistical range includes scan matching, map update, closed-loop detection, and back-end optimization, excluding the waiting time for data playback. The sum of processing times for each frame within the statistical range is divided by the actual number of processed frames to obtain the average processing time per frame. All offline algorithms and ablation configurations are replayed once with the data packets, and the frame average values of this run are reported.

The average processing time per frame of MDF-ALC-SLAM is 68 ms, which is lower than 85 ms of Karto SLAM, 75 ms of Gmapping, and 95 ms of Cartographer, but higher than 60 ms of Hector SLAM. This comparison is only for the used data, computing platform, and algorithm configuration.

Offline playback uses absolute trajectory position error (ATPE) and absolute heading error (ATHE) to evaluate the deviation of SLAM output from the manually annotated reference. Both indicators are statistically calculated based on the root mean square value of N valid corresponding points. ATPE first calculates the Euclidean distance between each pair of position points in the same coordinate system, then takes the average of the squared distances and takes the square root, with units in meters. ATHE first normalizes each pair of heading differences to [−π, π], then takes the average of the squared values and takes the square root, and finally converts to degrees. Offline correspondence relationships are established based on the route sequence and corresponding positions, without using timestamp synchronization for pairing.

Offline ATPE and ATHE reflect the trajectory deviation relative to the manually annotated reference; in the field test, the deviation of the robot after returning relative to the starting target is evaluated based on the actual poses obtained from external measurement equipment. The reference basis and statistical objects of the two types of evaluations are different, so they are reported separately.

The field return position error is used to evaluate the deviation between the measured return position and the starting target position after the robot completes the closed-loop route. Its calculation formula is:

$$e_{p,j} = \| p_{\text{return},j} - p_{\text{target},j} \|_2, \quad \overline{e}_p = \frac{1}{N}\sum_{j=1}^{N} e_{p,j} \tag{45}$$

In the formula, p_{target} represents the initial target position, and p_{return} is the actual position of the robot after returning, as measured by Leica AT960 + T-Mac. Both refer to the position of the robot base and are uniformly expressed in the same reference base coordinate system, with the position error unit being meters. Here, $j = 1, \ldots, 30$; each pair of the initial target and return positions is matched in the same reference coordinate system. Formula (45) defines the single error and the arithmetic mean of the errors over 30 trials.

The on-site return heading error is used to evaluate the deviation between the actual heading after the robot's return and the initial target heading. Its calculation formula is:

$$e_{\psi,j} = \| \psi_{\text{return},j} - \psi_{\text{target},j} \|_2, \quad \overline{e}_\psi = \frac{1}{N}\sum_{j=1}^{N} e_{\psi,j} \tag{46}$$

In the formula, ψ_{target} represents the initial target heading, and ψ_{return} is the return heading measured by Leica AT960 + T-Mac and converted to the robot base coordinate system. Both represent the direction angles of the robot coordinate system relative to the same reference base coordinate system, and adopt the same angle convention. The angular difference in Equation (46) is first normalized to the range [−π, π], then the absolute value is taken and converted to degrees; subsequently, the arithmetic mean of the 30 results is calculated.

To ensure reproducible, comparable experiments comparing different SLAM algorithms, this paper runs the experiments in a unified software and hardware environment. It configures the sensor inputs and key parameters for each baseline algorithm. All algorithms use the complete sequence (b0-2014-07-11-10-58-16.bag) from the public dataset of the Deutsches Museum for two-dimensional laser radar scanning data as the main perception input, and select wheel odometry and inertial measurement unit (IMU) data according to the actual requirements of each algorithm. The experimental platform uses an Intel Core i7-12700H processor and 16 GB of memory, and the software environment is Ubuntu 20.04 and ROS Noetic. Each baseline algorithm generally follows the parameter configuration recommended by the original literature or open-source implementation,

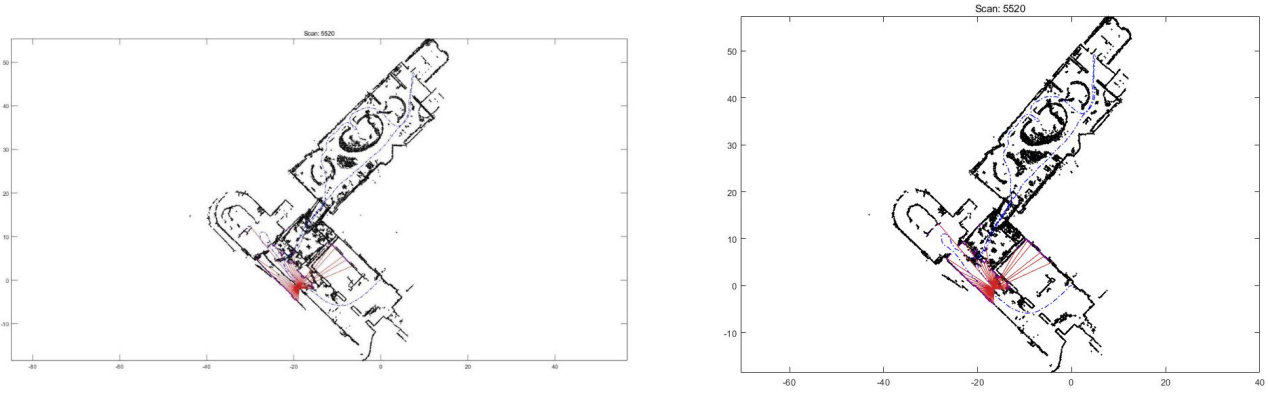
and uniformly sets the map resolution to 0.05 m. At the same time, necessary adaptations are made to the sensor topics and coordinate transformations. Specifically, CoreSLAM adopts a lightweight mapping method based on laser scan matching, with the translation search range set to ±0.20 m and the angle search range set to ±10°; GMapping adopts a mapping method based on particle filtering, with the particle number set to 30, the translation update threshold set to 1.0 m, the rotation update threshold set to 0.5 rad, and the resampling threshold set to 0.5; Hector SLAM adopts a pose estimation method based on laser scan matching, with the multi-resolution map layer number set to 3, the map update distance threshold set to 0.4 m, and the angle threshold set to 0.9 rad; Cartographer adopts a local scan matching and global pose graph optimization framework, with the distance data quantity of each sub-map set to 90, the laser scan cumulative quantity set to 1, the scan matching translation weight set to 10, the rotation weight set to 40, and the maximum optimization iteration number set to 20; Karto SLAM adopts a mapping method based on scan matching and pose graph optimization, with the minimum scan update distance set to 0.5 m, the minimum rotation angle set to 0.5 rad, the search space dimension of correlation set to 0.5 m, the search resolution set to 0.01 m, and the maximum loop search distance set to 3.0 m. The MDF-ALC-SLAM proposed in this paper uses a map resolution of 0.05 m, a key-frame translation reference scale of 0.2 m, a rotation reference scale of 10°, an 85% loop-candidate similarity threshold, and a maximum optimization iteration count of 20. All algorithms use the same data sequence and computing platform for offline playback to reduce the influence of external experimental conditions on the results. It should be noted that different algorithms have different dependencies on odometry and IMU, so the comparison in this paper belongs to the system-level performance comparison of each algorithm under the actual sensor configuration, rather than an independent performance comparison of algorithm modules under the same sensor input conditions.

5.2. Public Data Playback Mapping Results and Ablation Analysis

The publicly available dataset used is the 2D laser SLAM data from the Deutsches Museum, and the link is as follows:

https://google-cartographer-ros.readthedocs.io/en/latest/data.html

Figure 4 shows the maps before and after loop-closure optimization in the same data playback of the Deutsches Museum, using the same map resolution and display scale. Figure 4a shows phenomena such as wall misalignment, repeated contours, and insufficient boundary coincidence in the revisit area, which we use to qualitatively assess the impact of cumulative errors on the mapping structure.



(**a**) (**b**)

Figure 4. (**a**). Map before loop closure optimization. (**b**). Map after loop closure optimization.

In Figure 4b, the repeated boundaries and local misalignment of the revisit area have been alleviated. We jointly use candidate retrieval, geometric verification, and pose graph optimization to establish loop constraints and adjust the poses of historical keyframes. This comparison reflects the visible changes in the map structure and is not used for calculating trajectory accuracy or the proportion of error reduction.

The qualitative evaluation in Figure 4 focuses on wall overlap, repeated contours, and overall structural continuity, and does not replace quantitative evaluation based on reference data using map screenshots. Figure 5 shows the mapping screenshots at different times during the same complete data package replay, which are used to observe map expansion, connectivity of the observed areas, and changes in obstacle boundaries. This is not used to determine positioning accuracy or real-time performance.

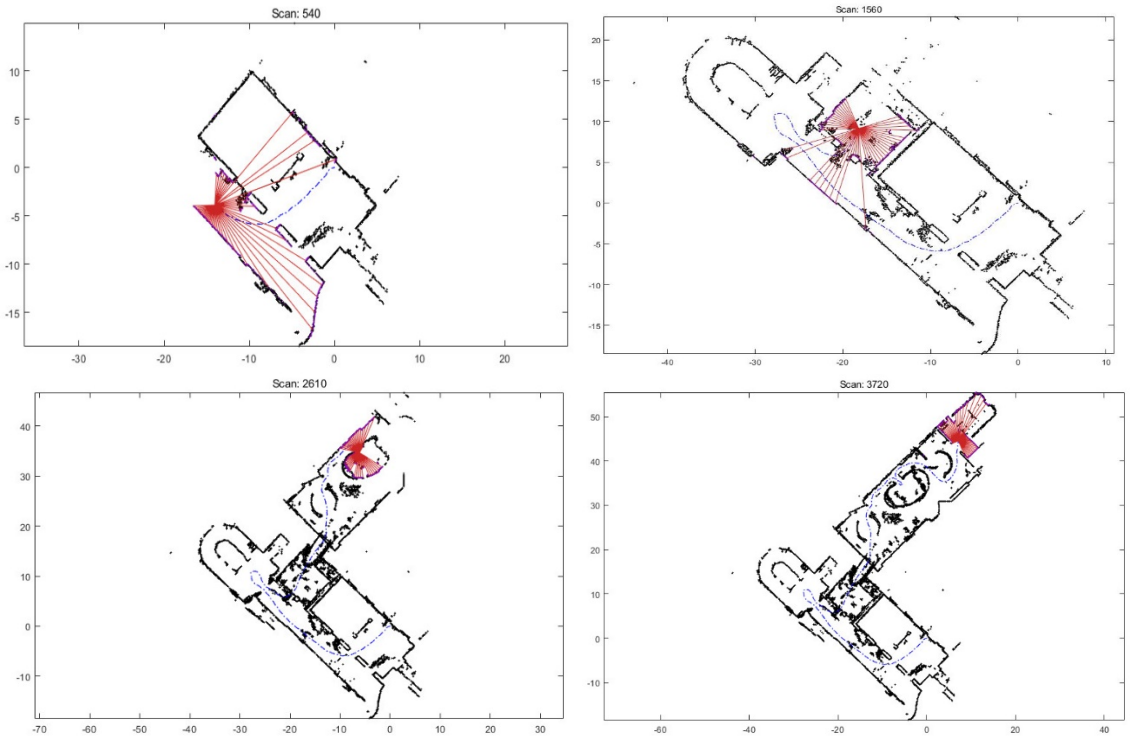


Figure 5. Mapping results at different time steps during offline replay of the Deutsches Museum dataset.

Table 3 presents the reduction ratios of ATPE and ATHE of the proposed method compared with each of the comparison algorithms. Taking the most basic CoreSLAM as the baseline, the algorithms are compared. CoreSLAM is a simplified scan matching and grid map update navigation algorithm. The computational cost and the trajectory deviation relative to the manual annotation reference are evaluated, respectively.

The reduction ratios are compared based on CoreSLAM as the baseline, and the ATPE and ATHE in the offline replay evaluation are calculated to reflect the deviation changes relative to the manual annotation reference. In the article, ATPE and ATHE are the absolute trajectory position error and the absolute heading error.

$$\text{Error reduction rate} = \frac{E_{\text{baseline}} - E_{\text{proposed}}}{E_{\text{baseline}}} \times 100\% \quad (47)$$

Among them, E_{baseline} denotes the root mean square deviation of the comparison algorithm in the same offline replay evaluation, while E_{proposed} denotes the root mean square deviation of the method presented in this paper.

Table 3. Percentage reductions in ATPE and ATHE relative to baseline SLAM methods using manually annotated reference data during public dataset replay.

Method	ATPE (m)	ATHE (°)	ATPE Reduction (%)	ATHE Reduction (%)
CoreSLAM	0.076	4.34	—	—
Gmapping	0.044	2.72	42.1	37.3
Hector SLAM	0.054	3.66	28.9	15.7
Cartographer	0.056	2.72	26.3	37.3
Karto SLAM	0.062	4.01	18.4	7.6
MDF-ALC-SLAM	**0.042**	**1.54**	**44.7**	**64.5**

Only the MDF-ALC-SLAM algorithm comparison and ablation were completed in the offline replay of the public data. This paper reports the arithmetic mean of the position error modulus and the absolute value of the normalised heading error returned by 30 trials. The offline experiment used manually annotated reference positions and headings; the trajectory deviation ATPE was 0.042 m, ATHE was 1.54°, and the average processing time per frame was 68 ms. Table 4 uses the complete MDF-ALC-SLAM as the benchmark and removes the distance-field scan matching, multi-resolution optimization, adaptive keyframe, and closed-loop optimization modules one by one for single-module removal ablation. "w/o" indicates that the corresponding module is not used.

When other modules and experimental conditions remain the same, replacing the joint keyframe scoring in this paper with the motion threshold rule yields ATPE and ATHE values of 0.069 m and 2.63°, respectively, and an average processing time of 69 ms per frame. In contrast, the joint scoring configuration in this paper achieves ATPE and ATHE values of 0.042 m and 1.54°, respectively, a reduction of 39.13% and 41.44% compared to the motion-threshold-only rule, with an average processing time of 68 ms per frame. The results show that under these testing conditions, keyframe scoring that jointly considers motion changes and matching residuals reduces trajectory deviation relative to the manual annotation reference compared with using only the motion threshold rule, while maintaining similar average computational costs.

Table 4. Ablation experiments of different module combinations.

Method	ATPE (m)	ATHE (°)	Average Processing Time (ms/Frame)
w/o adaptive keyframe	0.043	1.57	72
w/o distance field scan matching	0.064	2.41	86
w/o multi-resolution optimization	0.057	2.24	91
w/o closed-loop optimization	0.081	4.53	93
MDF-ALC-SLAM	**0.042**	**1.54**	**68**
MDF-ALC-SLAM with Movement threshold	0.069	2.63	69

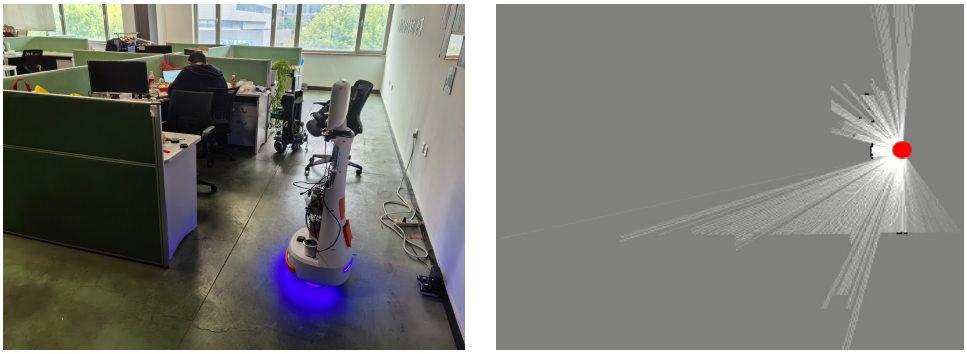
Table 4 presents the comparison results of the complete MDF-ALC-SLAM method and each variant after removing individual modules. The ATPE of the complete method is 0.042 m, the ATHE is 1.54°, and the average processing time per frame is 68 ms, all of which are the lowest values among the listed configurations. After removing the adaptive keyframe, ATPE and ATHE increase to 0.043 m and 1.57°, respectively, and the processing time increases to 72 ms, indicating that this module mainly shows a reduction in computational overhead in this test, with a relatively small impact on trajectory error. After removing the distance field scan matching or multi-resolution optimization, both error values increase, suggesting that these two modules have a positive effect on the pose estimation under the current configuration. Among them, after removing the distance field scan matching, ATPE and ATHE are 0.064 m and 2.41°, respectively; after removing the multi-

resolution optimization, they are 0.057 m and 2.24°, respectively. The error after removing the closed-loop optimization is the largest, with ATPE reaching 0.081 m and ATHE reaching 4.53°, indicating that closed-loop optimization is more effective in suppressing the cumulative drift in this playback.

5.3. Field Mapping and Autonomous Return Experiment in SIASUN Smart Park

The field test was conducted in the open office area and adjacent corridors of SIASUN Smart Park, covering an area of approximately 500 m², with a total fixed route length of about 93 m. Each test consisted of two phases: manual remote control for mapping, followed by allowing the robot to autonomously start from the initial position and return to that position, using external equipment to evaluate the return position and heading deviation. We carried out 30 tests, reinitialising the mapping each time, not using the previous map, and keeping the algorithm parameters consistent. The robot was set to a travelling speed of 0.3 m/s. RPLIDAR C1 two-dimensional laser radar was used on-site, with an actual scanning frequency of 10 Hz, and a wheel odometer and IMU were equipped. The algorithm was deployed on the RK3399 board, and the parameters are shown in Table 2. The reference pose was obtained from the Leica Absolute Tracker AT960 laser tracker and the T-Mac six-degree-of-freedom target, with the actual data output frequency being 800 Hz. The test route was arranged around the tracker, with a total length of approximately 93 m. The measurements were synchronised with the robot's clock, and the on-site data were associated by timestamp. Based on the target installation relationship, we converted the measured poses to the robot base; the heading angle represents the robot coordinate system's direction relative to the reference base coordinate system. To display the trajectory, the reference pose was unified to the SLAM map coordinate system through fixed rotation and translation.

Figures 6–8 show on-site scenes of adjacent workstation passageways, passageway intersections, and corridors, with partitions, walls, columns, and door openings providing geometric constraints; desks and chairs forming local obstructions; and glass partitions and repetitive structures possibly increasing scan-matching ambiguity. The initial workstation partitions, walls, and seats limited the visible range, and the map mainly covered the adjacent free space and some obstacle boundaries, as shown in Figure 6. After entering the intersection area, the multi-directional visible space increased, and the newly observed boundaries expanded the map into a connected passage, as shown in Figure 7. Subsequently, continuous walls and openings on both sides of the corridor entered the observation range, and the map presented the structure of the corridor and adjacent areas, as shown in Figure 8. The experiment covered straight paths, turns, and changes in visible space to demonstrate continuous mapping. The red dots represent the robot's position at the moment of the screenshot. The screenshots show the expansion of the space and the structure's connectivity; we calculated the position and heading error for returning to the starting target based on external measurement data.



(**a**) (**b**)

Figure 6. (**a**). Initial scene in the passageway adjacent to the workstation. (**b**). Local mapping results during the initial stage.



(**a**) (**b**)

Figure 7. (**a**). Real-world scene at the passageway intersection. (**b**). Map expansion results in the intersection area.



(**a**) (**b**)

Figure 8. (**a**). Real-world scene in the corridor. (**b**). Mapping results of the corridor and adjacent areas.

The front end performs multi-resolution registration and keyframe updates, while the back end performs retrieval, geometric verification, and pose-graph optimization. The white areas represent the observed free space, the black contours represent the occupied boundaries, and the grey areas represent the unknown space. The stripes extending along the laser rays cannot be directly interpreted as walls. The on-site map shows the gradual connection of the channel and adjacent areas, and the qualitative evaluation focuses on the continuity of the walls, repeated boundaries, and geometric relationships at the intersections. The autonomous return after mapping adopts a complete closed route. The robot starts and returns to the initial target autonomously. The return position and heading errors are calculated based on the deviation between the actual return position and heading and the starting target position and heading. Figure 9 shows the complete route, the measured reference trajectory, and the SLAM-estimated trajectory, with the starting target and return position labelled. The average processing time per frame for scan matching, map update, loop closure detection, and back-end optimization is calculated for each on-site experiment. The average frame value of 30 experiments is taken as the arithmetic mean, resulting in 73 ms. The position and heading errors of the return are calculated based on the actual return pose and the starting target pose and heading. The average return position error ATPE is 0.053 m, and the average return heading error ATHE is 1.94°.

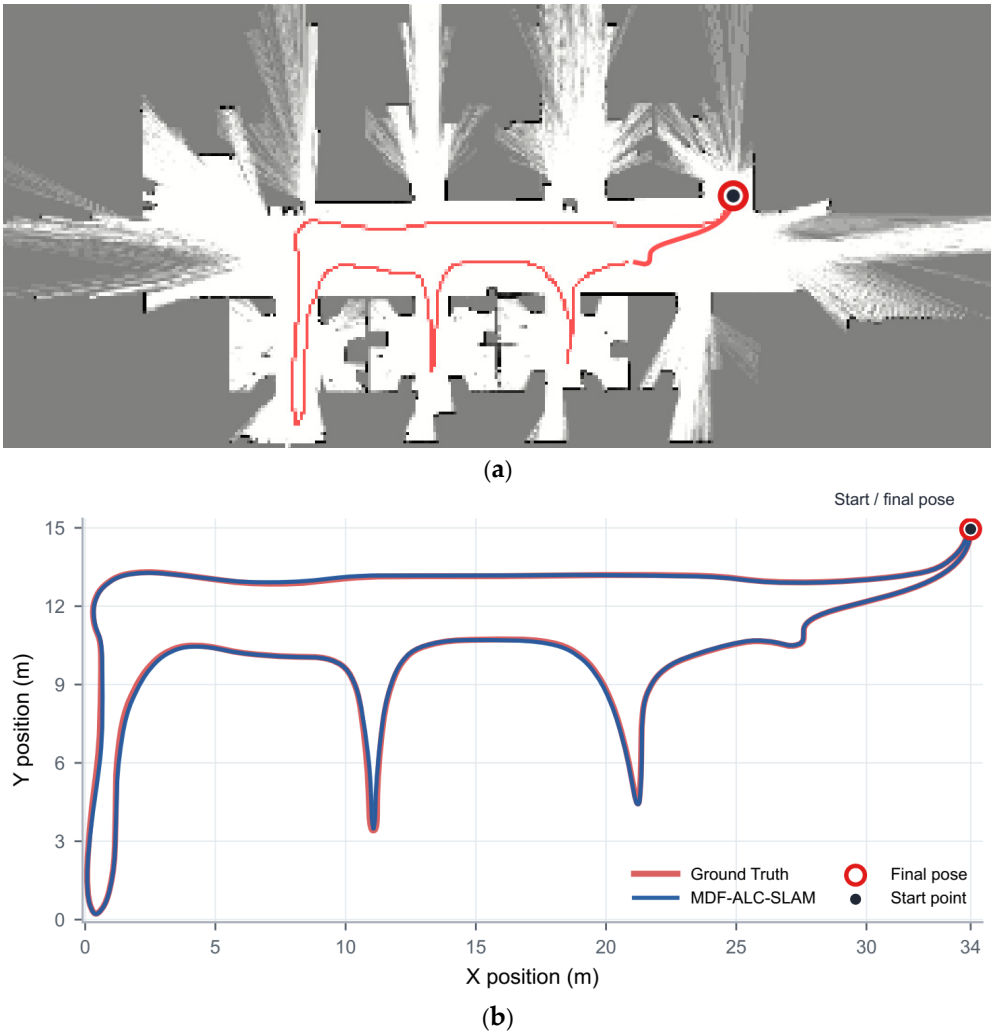


Figure 9. (**a**). Annotated navigation route with loop closure. (**b**). Comparison between the measured reference trajectory and the SLAM-estimated trajectory during the autonomous return test.

Figure 10 uses the full MDF-ALC-SLAM as the baseline. It removes the distance-field scan matching, multi-resolution optimization, adaptive keyframe, and closed-loop optimization modules one by one to conduct single-module ablation. "w/o" indicates that the corresponding module is omitted, and "Target path" denotes the manually annotated reference trajectory. The figure qualitatively shows the output of each configuration and the reference trajectory.

Figure 10 presents the trajectory comparison of the complete MDF-ALC-SLAM and each variant after the removal of individual modules. Specifically, Figure 10a shows the overall trajectory, and Figure 10b illustrates the differences between each ablation configuration and the reference trajectory. After removing the distance field scan matching, the trajectory shape shows a significant deviation; after removing the multi-resolution optimization, the deviation in some turning and path connection areas increases; after removing the adaptive keyframe, the overall shape of the trajectory is still close to the reference path, with relatively small changes; after removing the closed-loop optimization, some sections show a relatively obvious overall offset, and the generated error is the largest. Figure 10b shows the manually annotated "Target path" reference trajectory in the real scene, where the on-site return error reflects system performance under the combined effects of positioning and motion control, verifying MDF-ALC-SLAM's correctness.

As shown in Figure 11, navigation variability experiments were conducted within a confined area for 15 trials, presenting the navigation test scenario in a narrow office

corridor. The workstation partitions, desks, and chairs restricted the robot's movement space and created local obstructions. This scene was used to supplement the evaluation of the robot's performance in repeatedly navigating and returning under space-constrained conditions. The photos in the figure show the test environment and the robot's operating situation. The specific return errors and fluctuations between tests are given in Table 5. The test points were selected as the robot's target points in the figure.

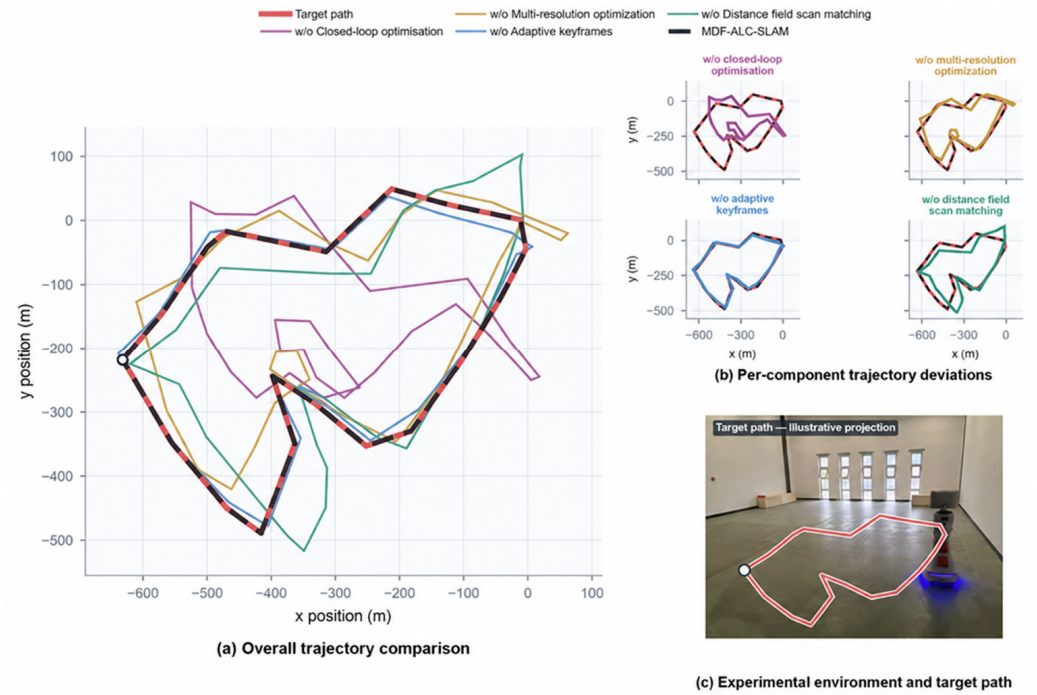


Figure 10. Trajectory comparison between the full proposed method and its variants with individual modules removed. (**a**) Overall trajectory comparison among the manually annotated target path, the complete MDF-ALC-SLAM, and the four single-module ablation variants. (**b**) Per-component trajectory deviations produced by removing closed-loop optimization, multi-resolution optimization, adaptive keyframes, or distance-field scan matching. (**c**) Experimental environment and the target path used for field validation.



Figure 11. Navigation variability experiments within a confined area.

Table 5 lists the return position error $e_{p,j}$ and return heading error $e_{\psi,j}$ for 15 trials. According to the table, the average return position error is 0.0535 m, the sample standard deviation is 0.0235 m, and the error range is 0.012~0.094 m; the average return heading error is 1.941°, the sample standard deviation is 0.463°, and the error range is 0.98~2.76°. These statistical quantities simultaneously reflect the average return deviation and the fluctuations during repeated runs of this group of tests. Among them, the return position

error shows a relatively obvious relative dispersion. These results evaluate the system's return performance under the combined effects of positioning and motion control in the narrow-scenario measurement. They are similar to the previous experimental results of ATPE = 0.053 m and ATHE = 1.94°.

Table 5. Return position and heading errors across 15 navigation trials in a confined environment.

Trial	$e_{p,j}$ (m)	$e_{\psi,j}$ (°)	Trial	$e_{p,j}$ (m)	$e_{\psi,j}$ (°)
1	0.059	1.94	9	0.083	1.57
2	0.064	2.01	10	0.066	1.75
3	0.051	2.23	11	0.094	2.54
4	0.049	2.65	12	0.050	2.04
5	0.062	1.89	13	0.019	1.58
6	0.061	1.72	14	0.013	1.72
7	0.067	0.98	15	0.012	1.74
8	0.053	2.76			

6. Conclusions

This paper presents MDF-ALC-SLAM, a 2D laser SLAM method that integrates multi-resolution distance-field scan matching, adaptive keyframe selection that accounts for motion and matching errors, spatial geometric descriptor-based loop-closure retrieval, geometric verification, and pose-graph optimization. The public data replay demonstrates the changes in map structure, the trajectory deviation relative to the manually annotated reference, and the computational cost. The complete method takes an average of 68 ms per frame to process, which is 28.42% lower than the basic configuration of 95 ms. The field test is divided into two stages: manual remote mapping and autonomous return after mapping. The average return position error across 30 tests is 0.053 m, and the heading error is 1.94°; the average per-frame processing time on the RK3399 platform is 73 ms. Due to the different offline and field platforms and configurations, the processing times are reported separately. These results provide support for the effectiveness and deployment feasibility of the method under the tested data, environment, and system configuration. Among them, the on-site return error reflects system performance under the combined effects of positioning and motion control and verifies the correctness of MDF-ALC-SLAM.

Future research will be conducted in the following directions:

Combine dynamic target detection and map update to reduce the influence of moving objects on laser scan matching, and evaluate the system performance under different dynamic intensities.

Study more efficient laser feature representation and retrieval methods to reduce the overhead of loop closure detection in large-scale environments.

Further integrate visual, IMU, and other sensor information, and evaluate trajectory positioning accuracy, running delay, and system stability through independent reference measurements and repeated experiments.

Author Contributions: Conceptualization, B.Z. and Y.C.; methodology, Q.Z.; software, R.S. and B.Z.; validation, B.Z. and R.S.; formal analysis, Y.C.; investigation, Q.Z.; resources, Y.C.; writing—original draft preparation, Y.C.; writing—review and editing, B.Z. and Y.C.; funding acquisition, B.Z. and Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. U20A20197); the Doctoral Research Startup Foundation of Liaoning University of Technology (Grant No. XB2026024); the 2026 Fundamental Research Project of the Educational Department of Liaoning Province (Research on Multimodal Cognition and Collaboration Methods for Intelligent

Inspection Robots); the Leading Talent Project of Innovation Team Supported by the Department of Education of Liaoning Province (No. LJ222410142060); the Natural Science Foundation of Liaoning Province (No. 20180520022); and the Key Technology Research Project of Intelligent Composite Robots supported by the Department of Science and Technology of Liaoning Province (No. 2023JH2/101300225).

Data Availability Statement: The relevant code and associated materials will be made available after acceptance of the manuscript. The dataset is available from the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhou, B.; Li, C.; Chen, S.; Xie, D.; Yu, M.; Li, Q. ASL-SLAM: A LiDAR SLAM with activity semantics-based loop closure. *IEEE Sens. J.* **2023**, *23*, 13499–13510.
2. Shi, C.; Chen, X.; Xiao, J.; Dai, B.; Lu, H. Fast and accurate deep loop closing and relocalization for reliable LiDAR SLAM. *IEEE Trans. Robot.* **2024**, *40*, 2620–2640.
3. Lang, F.; Ming, R.; Yuan, Z.; Xu, X.; Wu, K.; Yang, X. Adaptive global graph optimization for LiDAR-inertial SLAM. *IEEE Robot. Autom. Lett.* **2024**, *9*, 9868–9875.
4. Gao, J.; Fan, J.; Zhai, S.; Dai, G. Wi-loop SLAM: Loop closures with wireless sensing in multipath SLAM. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 15185–15197.
5. Lee, H.; Yang, S. Real-time SLAM-guided closed-loop photodynamic therapy with pixel-accurate light-dose control. *IEEE Robot. Autom. Lett.* **2026**, *11*, 3446–3453.
6. Ma, Q.; Jian, Q.; Li, M.; Ullah, S. LiDAR-visual fusion SLAM for autonomous vehicle location. *IEEE Internet Things J.* **2025**, *12*, 25197–25210.
7. Li, X.; Wei, P.; Chen, Q.; Shao, B.; Chen, G. A bio-inspired SLAM system with multi-sensor fusion for real-world outdoor navigation. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2548608.
8. Wang, S.; Zheng, D.; Li, Y. LiDAR-SLAM loop closure detection based on multi-scale point cloud feature transformer. *Meas. Sci. Technol.* **2024**, *35*, 036305. https://doi.org/10.1088/1361-6501/ad147a.
9. Ran, Y.; Xu, X.; Tan, Z.; Luo, M. A review of 2D LiDAR SLAM research. *Remote Sens.* **2025**, *17*, 1214.
10. Ou, F.; Li, Y.; Li, N.; Zhou, J.; Zhang, W.; Miao, Z. SG-ISBP: Orchard robots localization and mapping with ground optimization and loop closure detection integration. *IEEE Sens. J.* **2024**, *24*, 13164–13174.
11. Tan, S.; Sima, K.; Wang, D.; Ge, M.; Guo, D.; Liu, H. Self-supervised 3-D semantic representation learning for vision-and-language navigation. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 6738–6751. https://doi.org/10.1109/TNNLS.2024.3395633.
12. Jiao, J.; Geng, R.; Li, Y.; Xin, R.; Yang, B.; Wu, J.; Wang, L.; Liu, M.; Fan, R.; Kanoulas, D. Real-time metric-semantic mapping for autonomous navigation in outdoor environments. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 5729–5740. https://doi.org/10.1109/TASE.2024.3429280.
13. Zhang, W.Q.; Wang, J.; Zhang, L.; Ma, Z. SUI-SLAM: A semantics and uncertainty incorporated visual SLAM algorithm towards dynamic indoor environments. *Robot* **2024**, *46*, 732–742. https://doi.org/10.13973/j.cnki.robot.230195.
14. Wang, M.Y.; Song, W. An RGB-D SLAM algorithm based on adaptive semantic segmentation in dynamic environment. *Robot* **2023**, *45*, 16–27. https://doi.org/10.13973/j.cnki.robot.210368.
15. Wang, H.; Zhou, S.C.; Fang, B.F. RGB-D SLAM algorithm based on delayed semantic information in dynamic environment. *Pattern Recognit. Artif. Intell.* **2023**, *36*, 953–966.
16. Li, Z.; Chamran, K.; Alobaedy, M.M.; Sheikh, M.A.; Siddiqui, T.; Ahad, A. An optimization on 2D-SLAM map construction algorithm based on LiDAR. *J. Intell. Robot. Syst.* **2024**, *110*, 144. https://doi.org/10.1007/s10846-024-02123-1.
17. Du, S.; Chen, T.; Lou, Z.; Wu, Y. A 2D-LiDAR-based localization method for indoor mobile robots using correlative scan matching. *Robotica* **2025**, *43*, 514–541. https://doi.org/10.1017/S026357472400198X.
18. Shi, W.; Dai, H.; Ji, M.; Cui, Y.; Liu, C.; Chen, Q. Self-supervised point cloud importance awareness network for 2-D LiDAR SLAM. *IEEE Sens. J.* **2025**, *25*, 30060–30073. https://doi.org/10.1109/JSEN.2025.3583431.
19. Douadi, L.; Dupuis, Y.; Vasseur, P. Stable keypoints selection for 2D LiDAR based place recognition with map data reduction. *Robotica* **2022**, *40*, 3786–3810. https://doi.org/10.1017/S0263574722000613.
20. Jo, J.H.; Moon, C.B. Scan matching validation scheme based on the neural network using Gaussian model input to improve robustness of graph SLAM. *Int. J. Adv. Robot. Syst.* **2025**, *22*, 17298806251406790. https://doi.org/10.1177/17298806251406790.

21. Wan, K.; Luo, J. SPVL-vSLAM: Visual SLAM for autonomous driving vehicles based on semantic patch-NetVLAD loop closure detection in semi-static scenes. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 8975–8991.
22. Cui, Y.; Xiong, Z.; Li, X.; Pan, J. Fast inertial-only SLAM: A fast pedestrian SLAM method for structured buildings based on foot-mounted IMU. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 1–13. https://doi.org/10.1109/TIM.2025.3588955.
23. Zhang, W.; Zhu, R.; Zheng, G.; Wang, J.; Wang, Y.; Li, Y.; Miao, Z. Multisensor adaptive fusion for SLAM in plant factories using deep reinforcement learning. *IEEE Trans. Ind. Inform.* **2026**, 1–12.
24. Xu, X.; Zhang, L.; Yang, J.; Cao, C.; Wang, W.; Ran, Y.; Tan, Z.; Luo, M. A review of multi-sensor fusion SLAM systems based on 3D LiDAR. *Remote Sens.* **2022**, *14*, 2835.
25. Yu, J.; Yu, T.H.; Zhang, Q.Y. BSC-Lego: A hybrid approach combining BoW3D and scan-context for loop closure detection in laser SLAM. *Results Eng.* **2025**, *27*, 106519. https://doi.org/10.1016/j.rineng.2025.106519.
26. Zhang, J.; Wang, S.; Tan, X. ODLC_SAM: A novel LiDAR SLAM system towards open-air environments with loop closure. *Ind. Robot Int. J. Robot. Res. Appl.* **2023**, *50*, 1011–1023.
27. Pan, H.; Liu, D.; Ren, J.; Huang, T.; Yang, H. LiDAR-IMU tightly-coupled SLAM method based on IEKF and loop closure detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 6986–7001.
28. Grisetti, G.; Stachniss, C.; Burgard, W. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. https://doi.org/10.1109/TRO.2006.889486.
29. Kohlbrecher, S.; von Stryk, O.; Meyer, J.; Klingauf, U. A flexible and scalable SLAM system with full 3D motion estimation. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–5 November 2011; pp. 155–160. https://doi.org/10.1109/SSRR.2011.6106777.
30. Konolige, K.; Grisetti, G.; KumMerle, R.; Burgard, W.; Limketkai, B.; Vincent, R. Efficient sparse pose adjustment for 2D mapping. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 22–29. https://doi.org/10.1109/IROS.2010.5649043.
31. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–20 May 2016; pp. 1271–1278. https://doi.org/10.1109/ICRA.2016.7487258.
32. Felzenszwalb, P.F.; Huttenlocher, D.P. Distance transforms of sampled functions. *Theory Comput.* **2012**, *8*, 415–428. https://doi.org/10.4086/toc.2012.v008a019.
33. Grisetti, G.; Kummerle, R.; Stachniss, C.; Burgard, W. A tutorial on graph-based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2010**, *2*, 31–43. https://doi.org/10.1109/MITS.2010.939925.
34. Kummerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g2o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3607–3613. https://doi.org/10.1109/ICRA.2011.5979949.
35. Kim, G.; Kim, A. Scan Context: Egocentric spatial descriptor for place recognition within 3D point cloud map. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 4802–4809. https://doi.org/10.1109/IROS.2018.8593953.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.