IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 1, March 2026, pp. 52~62
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i1.pp52-62

ISTD-LIOM: Direct LiDAR-inertial odometry and mapping with intensity-enhanced stable triangle descriptor

Lixiao Yang[1], Sheng Hua[1], Youbing Feng[1], Shangzong Yang[2], Jie Wang[3]
[1]Ocean College, Jiangsu University of Science and Technology, Zhenjiang, China
[2]Computer College, Jiangsu University of Science and Technology, Zhenjiang, China
[3]School of Materials Science and Technology, Shanghai University, Shanghai, China

Article Info

Article history:

Received Jul 12, 2025
Revised Nov 21, 2025
Accepted Feb 9, 2026

Keywords:

LIDAR intensity
Loop detection
Sensor fusion
Simultaneous localization and mapping
Triangular descriptor

ABSTRACT

To address the cumulative drift problem of light detection and ranging (LiDAR)-inertial odometry (LIO) in long-duration localization and mapping tasks, this paper proposes a LiDAR-inertial odometry and mapping system, intensity-enhanced stable triangle descriptor-LiDAR-inertial odometry and mapping (ISTD-LIOM), based on the intensity-enhanced stable triangle descriptor (ISTD). This system, built on the FAST-LIO2 front-end architecture, achieves global consistency localization through loop closure detection and global optimization. First, we design the ISTD descriptor by combining geometric descriptors of triangles (including vertex plane normal vectors and edge lengths) with local intensity distribution descriptors to form a compact, rotation-invariant feature representation. Next, an adaptive keyframe management mechanism is constructed, which filters keyframes based on inter-frame relative poses and generates a descriptor database. A hybrid retrieval strategy is then proposed, which combines descriptor similarity matching and spatial distance filtering, forming an efficient loop closure candidate recognition mechanism. After applying plane iterative closest point (ICP) refinement and geometric-intensity consistency validation, the loop closure constraints are integrated into a pose graph optimization framework, correcting odometry drift. Experiments on the KITTI dataset demonstrate that the ISTD-LIOM system significantly enhances map global consistency while maintaining real-time computational performance.

Corresponding Author:

Youbing Feng
Ocean College, Jiangsu University of Science and Technology
2 Mengxi Road, Jingkou District, Zhenjiang, Jiangsu 212000, China
Email: yzfyb@just.edu.cn

1. INTRODUCTION

Light detection and ranging (LiDAR)-inertial odometry (LIO) [1]–[4] and LiDAR-inertial odometry and mapping (LIOM) [5]–[8] have gained significant prominence in the fields of robotic navigation and autonomous driving in recent years. These technologies enable real-time environmental perception and map construction by fusing data from light detection and ranging (LiDAR) and inertial measurement units (IMU). This multi-sensor fusion approach not only enhances the accuracy and robustness of localization but also provides a foundational framework for navigation. The increasing application of LIO and LIOM technologies in autonomous driving [9], [10], unmanned aerial vehicles (UAVs) [11], [12], and robotics highlight their growing importance, offering powerful technical support for achieving autonomous navigation and environmental perception.

LIO offers a significant advantage for autonomous robots [13] operating in dynamic environments by combining LiDAR's precise geometric sensing with the high-frequency motion tracking provided by IMUs. However, drift errors still emerge over extended periods of operation. These errors arise due to factors such as sensor noise, environmental changes, and the inherent accumulation of errors within the algorithm itself. Cumulative drift can degrade the robot's localization accuracy and map quality, thus limiting its performance in long-distance navigation and complex environments. Some studies have addressed this issue by using loop closure detection to identify when a robot revisits a previously encountered location, thereby eliminating cumulative errors through registration and improving the consistency of localization and mapping. Early loop closure methods primarily relied on distance metrics, comparing the current map with historical maps to detect loops. While simple and intuitive, this approach is prone to false positives and missed detections in complex environments. To improve the recall rate of loop closure detection, researchers have proposed various descriptors to extract spatial structural features from LiDAR point clouds. By retrieving similar descriptors from a database, loop closure frames can be identified with low computational cost.

Scan context [14] is a point cloud description algorithm whose core idea is to project 3D point cloud data into a 2D polar coordinate space, forming a ring-sector matrix. Due to the use of polar coordinates, similar scenes can still be matched by horizontally shifting the matrix, even when the point cloud data contains angular deviations. Scan context++ [15] is an improved version of scan context, which enhances the discriminative power and robustness of the descriptor by introducing more sophisticated feature encoding techniques. Binary tree code (BTC) [16] is an algorithm designed for environment recognition and loop closure detection using LiDAR point cloud data. It constructs a binary tree structure to represent point cloud data efficiently and match it, enabling fast environment recognition and loop detection. Reference [17] proposes a global descriptor for 3D place recognition, where the core idea is to describe local key points in the 3D point cloud using the edge lengths and angles of triangles, thereby achieving efficient place recognition and loop closure detection.

Although existing descriptor-based methods can perform rapid loop closure detection [18]–[20], they still suffer from missed detections and false positives in complex environments. To further enhance loop closure detection performance, this paper proposes several improvements. We design the intensity-enhanced stable triangle descriptor (ISTD), which innovatively integrates a triangular geometric descriptor—based on vertex plane normal vectors and edge length features—with a local intensity distribution descriptor to construct a compact, rotation-invariant feature representation. This multimodal fused descriptor enhances traditional geometric structures by incorporating intensity information, thereby significantly improving scene discriminability. Furthermore, this paper proposes the LiDAR SLAM framework ISTD-LIOM, which integrates dynamic keyframe management, descriptor matching, and distance-constrained loop closure detection to build a globally consistent back-end optimization system. Based on the FAST-LIO2 [21] front end, the framework extracts descriptors from each keyframe by fusing intensity and geometric features. It then performs loop closure candidate identification through descriptor similarity matching and spatial proximity constraints, effectively suppressing odometry drift over long-term sequences and greatly enhancing overall system consistency. Experimental results on the KITTI 05/07/08 datasets [22], [23] demonstrate that the ISTD-LIOM system, through high-precision loop closure detection and factor graph optimization, significantly improves localization accuracy and enhances the global consistency of large-scale point cloud maps, providing reliable technical support for autonomous navigation in complex environments.

The remainder of this paper is organized as follows. Section II introduces and analyzes the proposed method. Section III presents the experimental results on the KITTI dataset, and Section IV concludes this work.

2. METHODOLOGY

2.1. Framework overview

As shown in Figure 1, the system framework presented in this paper is a complete SLAM system that includes both a front-end LIO and a back-end factor graph optimization. The front end is based on the FAST-LIO2 architecture, utilizing an iterative Kalman filter-based state estimation method to achieve a tightly coupled fusion of LiDAR and IMU data. After processing each LiDAR frame, the front-end module transmits the undistorted point cloud and its corresponding pose estimate to the back end for loop closure detection and global optimization. The back-end module adaptively selects keyframes based on the relative pose changes and time intervals. For each newly selected keyframe, the system accumulates point clouds from all frames between the current and previous keyframes, followed by voxel-based down sampling to form an enhanced point cloud with richer spatial information. This accumulation strategy effectively expands the spatial coverage of a single frame point cloud, providing a more comprehensive environmental representation for subsequent descriptor generation and loop closure detection. It also significantly reduces

the probability of mismatches caused by the viewpoint limitations of individual point clouds. Based on the enhanced point cloud, the back-end module extracts ISTD descriptors. It builds a descriptor database while also recording the relative pose constraints between keyframes as the initial edges in the factor graph. The loop closure detection module identifies potential loop closure candidates through a dual-path candidate frame selection mechanism, which combines descriptor similarity matching with spatial distance constraints. These candidates are then verified for geometric consistency using a point-to-plane iterative closest point (ICP) registration algorithm. Loop closure constraints that meet the geometric consistency criteria are added to the factor graph, and the global poses are corrected through nonlinear optimization. Finally, the system re-registers the keyframe point clouds using the optimized pose trajectory, generating a globally consistent 3D LIDAR points map.

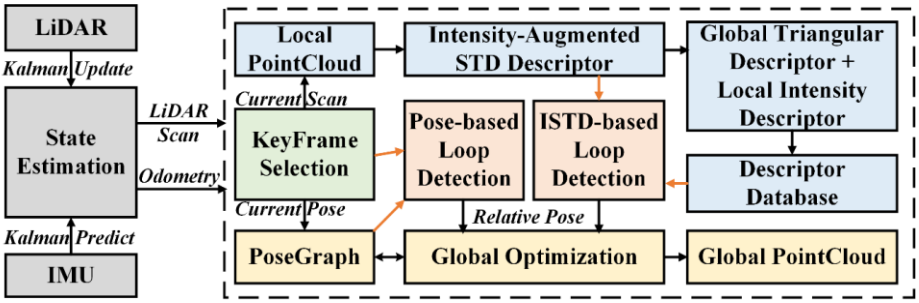

Figure 1. System overview of ISTD

2.2. Stable triangle descriptor

Inspired by [24], to improve segmentation stability, loop closure detection is performed on keyframes, which consist of points accumulated from several consecutive scans, thus resulting in increased point cloud density regardless of the specific LiDAR scanning pattern. Specifically, we utilize LiDAR odometry [25] to register each new input point cloud with the current keyframe. New keyframes are created when the number of subframes accumulates to a certain threshold.

2.2.1. LIDAR plane and key points detection

For a given keyframe of the point cloud, plane detection is initially performed using region growing. Specifically, the entire point cloud is divided into voxels of a given size (e.g., 1 m). Each voxel contains a set of points $p_i (i = 1, \ldots, N)$, and the covariance matrix of the points is computed:

$$\overline{p} = \frac{1}{N}\sum_{i=1}^{N} p_i; \sum = \frac{1}{N}\sum_{i=1}^{N}(p_i - \overline{p})\,(p_i - \overline{p})^T \quad (1)$$

Let λ_k represent the k-th largest eigenvalue of matrix $\sum$. The plane criterion is defined as:

$$\lambda_3 < \sigma_1 and \lambda_2 > \sigma_2$$

where the default values for σ_1 and σ_2 are 0.01 and 0.05, respectively. Using this criterion, we can determine if the points within the voxel form a plane. If so, the voxel is labeled as a plane voxel. Starting from any plane voxel, a plane is initialized, and the plane is expanded by searching for adjacent voxels. If an adjacent voxel belongs to the same plane (with the same plane's normal direction and distance below a threshold), it is added to the growing plane. Otherwise, if the adjacent voxel does not belong to the same plane, it is added to the boundary voxel list of the growing plane. This process is repeated until all adjacent voxels have been added or the boundary voxels are reached. For boundary voxels, the points they contain are projected onto the corresponding plane. For each plane, we create an image where the image plane coincides with the detected plane, and each pixel represents the maximum distance of points contained within the boundary voxels of the plane. A key point is then selected as the point with the largest pixel value within its 5×5 neighborhood.

2.2.2. Triangular descriptor construction

Using the key points extracted from the keyframes, a k-D tree is constructed, and 20 nearest neighbors are searched for each point to form triangular descriptors. Redundant descriptors with identical edge lengths are removed. Each triangular descriptor consists of three vertices P_1, P_2, and P_3, along with their

corresponding projected normal vectors n_1, n_2 and n_3. Additionally, the vertices of the triangle are arranged in descending order of edge lengths (see Figure 2). This edge length sorting ensures that $l_{12} \geq l_{23} \geq l_{13}$.

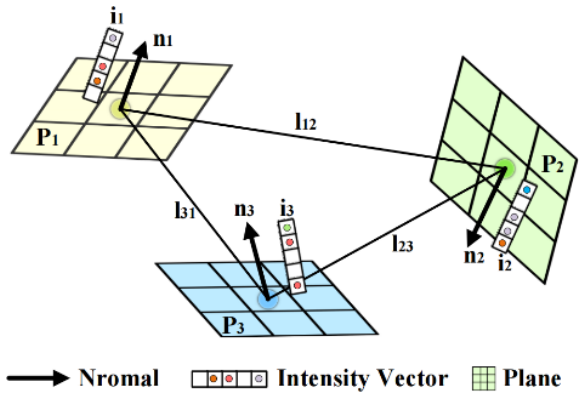

Figure 2. Triangular descriptor

2.3. Local intensity feature extraction

Although triangular descriptors are capable of capturing spatial structural features, they are prone to mismatches in regions with geometrically similar structures. LiDAR intensity information, which reflects the surface reflectance characteristics of objects, can distinguish between different materials. Therefore, intensity information is incorporated to enhance the discriminative power of the descriptors in similar scenes. For each vertex of the triangular descriptor, a local intensity descriptor is constructed. In the direction of the plane containing the vertex, the intensity distribution across different height levels reflects the material characteristics of the feature in the vertical direction, thereby improving its distinctiveness. Specifically, different objects such as buildings, vegetation, and other protrusions exhibit distinct intensity distributions along the height direction.

2.3.1. Vertical sampling strategy design

Given a key point v_i and the unit normal vector n_i of the plane it resides on, a vertical sampling axis is constructed along the direction of the normal vector. Let the vertical sampling range be denoted as $[-hmax_{max}]$ and the sampling interval as Δh. The total number of sampling layers is given by:

$$N_s = \frac{2h_{max}}{\Delta h} \tag{2}$$

For the j-th sampling layer ($j = 1,2,\ldots,N_s$), its spatial position is defined as (3).

$$P_j = v_i + \left(j - \frac{N_s+1}{2}\right) \cdot \Delta h \cdot n_i \tag{3}$$

This formula ensures that the sampling layers are symmetrically distributed around the key point v_i, with the central layer $\left(j = \frac{N_s+1}{2}\right)$ corresponding to the position of the key point, thus preserving the spatial symmetry of the local intensity features.

2.3.2. Intensity feature aggregation mechanism

At each sampling layer P_j, a spherical neighborhood with radius r is defined as:

$$\mathcal{P}_j = \{p \in \mathcal{P} \mid \|\boldsymbol{p} - \boldsymbol{p}_j\| \leq r\} \tag{4}$$

where $\mathcal{P}$ denotes the input point cloud. This neighborhood design strikes a balance between spatial resolution and computational efficiency, ensuring sufficient local intensity information is captured while mitigating noise introduced by overly large neighborhoods. For each sampling layer, the mean intensity is computed as the representative feature (5).

$$\mu_j = \begin{cases} \frac{1}{|\mathcal{P}_j|}\sum_{p\in\mathcal{P}_j} I(p), & \text{if } |\mathcal{P}_j| > 0 \\ 0, & \text{if } |\mathcal{P}_j| = 0 \end{cases} \quad (5)$$

$I(p)$ represents the intensity value of point p. A zero value is assigned in the case of an empty set, ensuring descriptor completeness and preventing numerical anomalies during computation.

2.3.3. Local intensity descriptor construction

Finally, the local intensity descriptor for a key point v_i is constructed as (6).

$$F_I^{(i)} = \left[\mu_1, \mu_2, \ldots, \mu_{N_s}\right]^T \quad (6)$$

This N_s-dimensional vector descriptor effectively captures the vertical intensity variation pattern around the key point. Compared to traditional single-point intensity values, this layered representation provides a more comprehensive characterization of the local material properties and geometric structure, offering richer and more robust intensity information for subsequent feature matching.

2.4. ISTD descriptor definition

By combining geometric features and intensity features, the complete ISTD is defined as (7).

$$D_{ISTD} = \left\{D_{geo}, F_I^{(1)}, F_I^{(2)}, F_I^{(3)}\right\} \quad (7)$$

$D_{geo} = \{l_1, l_2, l_3, \alpha, \beta, \gamma\}$ denotes the original geometric features of the STD descriptor and $F_I^{(i)}$ are the local intensity features corresponding to the three vertices of the triangle, respectively. This hybrid descriptor preserves the spatial expressiveness of the geometric features while significantly enhancing sensitivity to material properties, thereby improving feature discriminability in geometrically similar environments.

2.5. Intensity-consistency-based matching verification

2.5.1. Search loop candidate

Since hundreds of descriptors can be extracted from a single keyframe, we employ a hash table to store and retrieve descriptors during matching efficiently. Six attributes invariant to rotation and translation are used to compute the hash key: the side lengths l_{12}, l_{23}, l_{13}, and the dot products of the projected normal vectors $n_1 \cdot n_2$, $n_2 \cdot n_3$, $n_1 \cdot n_3$. Descriptors sharing all six similar attributes are assigned the same hash key and stored in the same container. For a query keyframe, all descriptors are extracted. For each descriptor Δ_i, its hash key is computed, and the corresponding container in the hash table is located. Each keyframe that has at least one descriptor in the matched container receives one vote. After all descriptors Δ_i from the query keyframe are processed, the top 10 keyframes with the highest vote counts are selected as loop closure candidates. The matched descriptors are retained for subsequent loop detection.

2.5.2. Intensity feature similarity measurement

During the triangular matching process, traditional geometric constraints are effective in eliminating the most obvious mismatches. However, they are limited in scenes where the geometric structures are similar but differ in material properties. The introduction of intensity features provides a reliable basis for resolving such ambiguous matches. For a candidate pair of matched triangles (T_q, T_r), the consistency of intensity features between corresponding vertices is evaluated using cosine similarity. For the i-th pair of corresponding vertices, the intensity feature similarity is defined as (8).

$$S_{int\,ensity}^{(i)} = \frac{F_I^{q(i)} \cdot F_I^{r(i)}}{\left\|F_I^{q(i)}\right\| \cdot \left\|F_I^{r(i)}\right\|} \quad (8)$$

The choice of cosine similarity offers the following theoretical advantages: First, it is invariant to the absolute scale of intensity values, making it robust to calibration differences across different LiDAR sensors. Second, cosine similarity emphasizes the pattern of intensity variation rather than absolute values, which aligns with our design goal of capturing local material characteristics. Finally, it is computationally efficient, making it suitable for real-time applications. To evaluate the overall intensity consistency of a triangle, the aggregated intensity similarity is defined as (9).

$$S_{int\,ensity} = \frac{1}{3}\sum_{i=1}^{3} S_{int\,ensity}^{(i)} \quad (9)$$

2.5.3. Intensity verification criterion

The intensity-based matching verification employs a dual-threshold strategy, ensuring both overall consistency and local reliability. The global consistency constraint requires that the average intensity similarity of the triangle satisfies:

$$S_{int\,ensity} > \Gamma_I \quad (10)$$

The local reliability constraint requires that the intensity similarity of each vertex satisfies:

$$\min_{i=1,2,3} S_{int\,ensity}^{(i)} > \Gamma_{min} \quad (11)$$

where Γ_I is the global intensity similarity threshold and Γ_{min} is the minimum similarity threshold for each individual vertex.

2.6. Loop closure detection and global pose graph optimization

2.6.1. Loop closure detection

This paper adopts a loop closure detection approach that combines a distance-threshold-based keyframe selection strategy with a dual-path candidate screening mechanism. The system selects keyframes based on a predefined spatial distance threshold d_{th}. When the robot's traveled distance exceeds this threshold, the current frame is designated as a new keyframe. All frames within this threshold are associated with their corresponding keyframe, forming a point cloud set for each keyframe. During loop candidate selection, two complementary filtering paths are employed: i) feature-based screening, which selects the top n historical keyframes with the highest similarity to the current keyframe based on the ISTD descriptor; and ii) spatial proximity-based screening, which selects the top m historical keyframes that are closest in Euclidean space to the current keyframe's pose.

Let the current keyframe pose be T_i, and the set of historical keyframe poses be $\{T_1, T_2, T_3, \ldots, T_{i-1}\}$. Then, the spatial proximity-based candidate set is defined as:

$$C_2 = arg\min_{j<1} \ll \|t_i - t_j\|_2, |C_2| = m \quad (12)$$

where t_i and t_j denote the position vectors of keyframes i and j, respectively. For the selected candidate keyframes, the system performs precise registration using a point-to-plane ICP algorithm, which minimizes the point-to-plane distances iteratively to estimate the optimal relative transformation. In the geometric verification stage, the system comprehensively evaluates indicators such as registration error, point cloud overlap ratio, and transformation plausibility to identify candidates with high geometric consistency. The final loop constraint is established using the candidate with the minimum registration error that also meets predefined quality thresholds, and its relative pose transformation is incorporated into back-end optimization.

2.6.2. Factor graph optimization

In this paper, a factor graph framework is employed for global pose optimization, formulating a nonlinear optimization problem that incorporates odometry factors and loop closure factors. In the factor graph representation, relative pose constraints between adjacent keyframes are encoded as odometry factors, while constraints obtained from loop closure detection are encoded as loop closure factors. Let the set of keyframe poses be denoted as $\mathcal{X} = \{x_1, x_2, \ldots, x_K\}$, the set of odometry constraints as $\mathcal{O} = \{o_{i,i+1}\}$, and the set of loop closure constraints as $\mathcal{L} = \{l_{i,j}\}$. The factor graph optimization problem can then be formulated as (13).

$$\mathcal{X}^* = \underset{\mathcal{X}}{\text{argmin}} \left[\sum_{i=1}^{K-1} \left\| x_i^{-1} \cdot x_{i+1} - o_{i,i+1} \right\|_{\sum_o}^2 + \sum_{(i,j)\in\mathcal{L}} \left\| x_i^{-1} \cdot x_j - l_{i,j} \right\|_{\sum_l}^2 \right] \quad (13)$$

Here, the first term represents the odometry factors, which constrain the relative poses between adjacent keyframes, and the second term represents the loop closure factors, which constrain the relative poses between loop-closure keyframe pairs. $\sum_o$ and Σ_l denote the covariance matrices associated with odometry and loop closure observations, respectively. The system solves the optimization using the Levenberg–Marquardt algorithm, which adaptively adjusts the damping factor to strike a balance between convergence speed and stability. An incremental update mechanism is also implemented: when a new loop closure

constraint is detected, optimization is performed only on the local subgraph, significantly improving computational efficiency. After optimization, the system re-registers the point clouds based on the corrected poses to generate a globally consistent map of the environment.

2.6.3. Map construction

The map construction module applies the globally consistent poses obtained from factor graph optimization to the concatenation of keyframe point clouds, resulting in a high-precision 3D representation of the environment. Let the local point cloud of the i-th keyframe be denoted as $\mathcal{P}_i$, and its optimized global pose as T_i^*. Then, the construction of the global map M can be expressed as (14).

$$M = \bigcup_{i=1}^{K} T_i^* \cdot \mathcal{P}_i \tag{14}$$

Here, $T_i^* \cdot \mathcal{P}_i$ represents the transformation of the local point cloud $\mathcal{P}_i$ into the global coordinate system. This process aligns originally drifted local point cloud segments to a unified global frame, effectively eliminating the negative impact of accumulated pose errors on map consistency. To ensure map quality, the system employs adaptive voxel filtering for downsampling the merged point cloud:

$$M_{filtered} = VoxelFilter(M, v_{size}) \tag{15}$$

where v_{size} denotes the adaptively adjusted voxel size, which reduces redundant data while preserving essential geometric features.

3. RESULTS AND DISCUSSION

3.1. Precision-recall evaluation

In this paper, the precision-recall curves of three descriptors are evaluated using sequences 05, 06, and 08 from the KITTI dataset. Ground-truth loop closures are manually annotated for these sequences. The annotation results are shown in Figure 3, where blue poses represent loop closure frames that revisit previously explored locations.

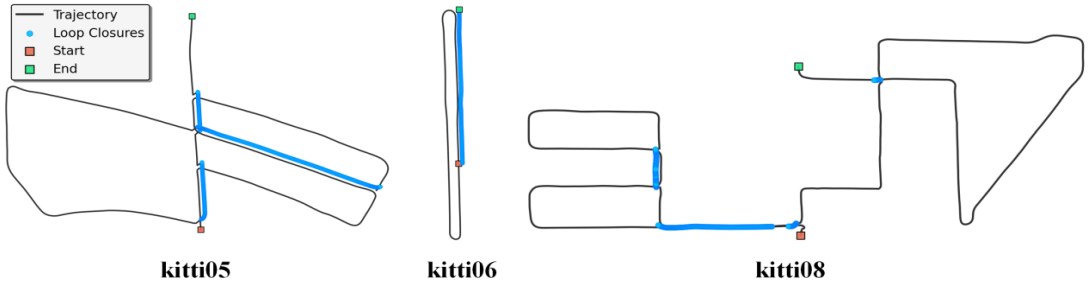

Figure 3. Visualization of KITTI sequences

After ground-truth annotation, the proposed ISTD descriptor is evaluated on the three datasets by plotting its precision-recall curve. Multiple precision and recall values are obtained by adjusting parameters to construct the precision-recall curve. For each sequence, if a detected loop closure lies within 20 meters of the query frame, it is marked as a true positive (TP); if no loop closure is detected at an annotated loop closure frame, it is marked as a false negative (FN); and if a detected loop closure does not correspond the query frame within 20 meters, it is marked as a false positive (FP). precision (P) and recall (R) are then calculated as (16).

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN} \tag{16}$$

To validate the effectiveness of the proposed method, it is compared with scan context [12] and STD [15] on three sequences of the publicly available KITTI dataset. The experimental results on the three sequences are shown in Figure 4. It can be observed that the proposed ISTD descriptor achieves a higher area under the curve (AUC), indicating superior performance.

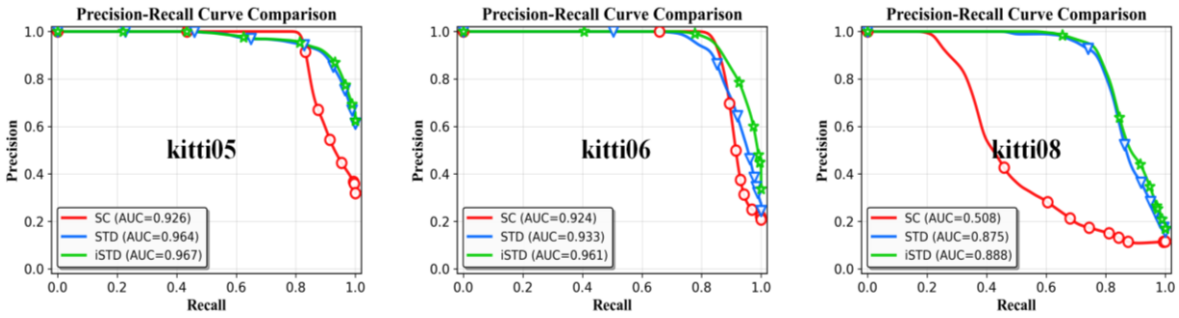

Figure 4. Performance comparison on the KITTI dataset

3.2. Quantitative comparison of localization accuracy

To further verify the accuracy improvement brought by ISTD to global optimization within the overall system, a quantitative evaluation of localization accuracy is conducted. As shown in Figure 5, the results are obtained on three test sequences from the KITTI dataset (Sequences 05, 07, and 08). The evaluated trajectories include the FAST-LIO2 [19] poses LIO, the back-end optimization based on STD (loop-STD), and the back-end optimization based on ISTD (Loop-ISTD). The red boxes highlight the deviation of the trajectory endpoints from the ground truth for each method. It can be observed that odometry exhibits significant drift in large-scale scenarios, while the trajectory optimized using ISTD is closer to the ground truth compared to that of STD, indicating higher localization accuracy. The quantitative results are presented in Table 1, where ISTD achieves the lowest trajectory error across all three test cases.

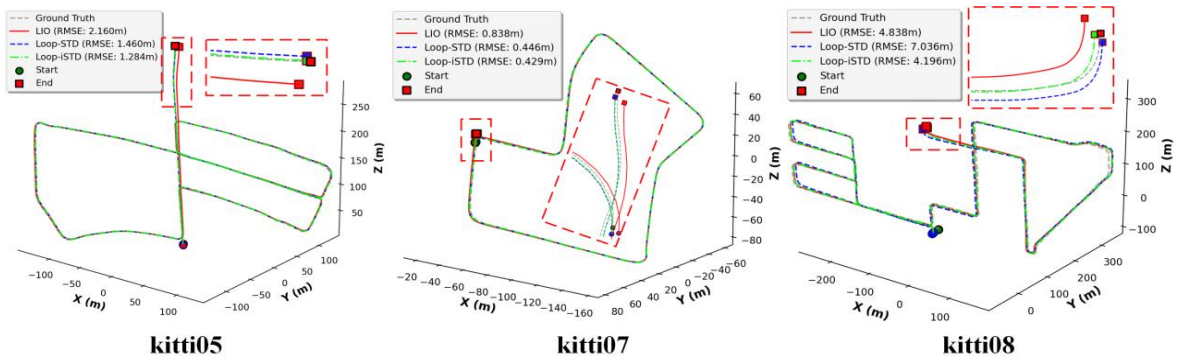

Figure 5. Visual comparison of KITTI sequences

Table 1. Absolute position errors in KITTI Datasets with different methods

Method	Rmse mean median		
	KITTI 05	KITTI 07	KITTI 08
FAST-LIO2	2.16 1.99 1.89	0.84 0.71 0.65	4.83 4.14 3.94
Loop-STD	1.46 1.32 1.33	0.45 0.39 0.43	7.03 6.15 6.20
Loop-iSTD	**1.28 1.16 1.17**	**0.43 0.38 0.42**	**4.20 3.60 3.34**

3.3. Qualitative comparison of mapping results

As illustrated in Figure 6, the mapping results for KITTI sequences 05, 07, and 08 are presented to qualitatively compare the effectiveness of the proposed method. In each subfigure, the red dashed box located in the lower left corner highlights the point cloud map constructed using raw odometry poses. Due to the accumulation of pose drift over time, the maps generated in this manner exhibit evident duplicate shadows and misalignment, which negatively impact overall mapping quality. In contrast, the lower right subfigure shows the mapping result obtained using globally consistent poses refined by back-end optimization. The use of optimized poses significantly reduces duplicate shadow effects. It produces a more coherent and accurate global map, demonstrating the advantage of the proposed ISTD-based loop closure and global optimization in maintaining global consistency.

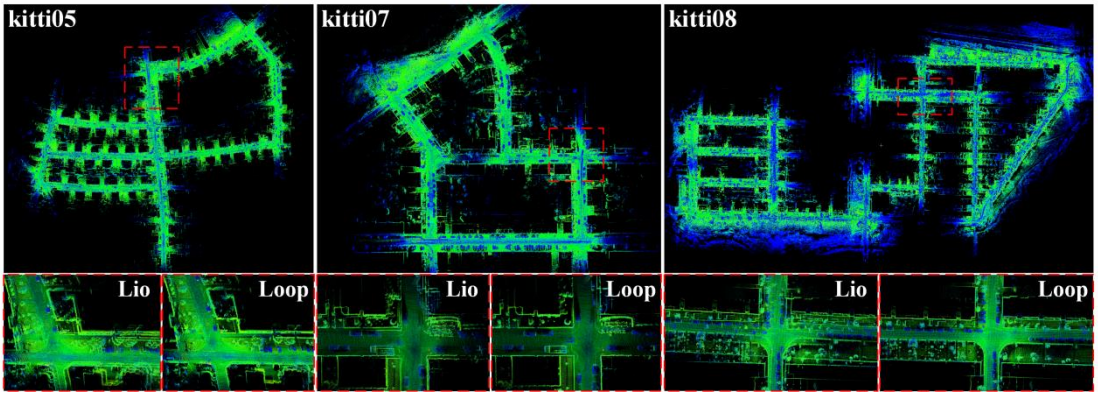

Figure 6. Comparison of mapping results before and after optimization

4. CONCLUSION

This study addresses the issue of cumulative drift in long-sequence tasks involving LIO by proposing a loop closure detection framework based on intensity-enhanced descriptor, called ISTD-LIOM. By integrating point cloud intensity distribution with geometric spatial features, we designed a rotation-invariant ISTD descriptor. Combined with a dynamic keyframe triggering mechanism and a hybrid loop closure retrieval strategy, this approach enables the systematic incorporation of intensity information into loop closure detection. Experimental results demonstrate that the method significantly enhances loop closure robustness and global map consistency on the KITTI dataset, validating the effectiveness of intensity features in correcting LiDAR-inertial odometry drift. However, the current research still has limitations in dynamic scene adaptability and multi-sensor collaboration. Future work will focus on designing semantically enhanced descriptors and cross-modal data fusion to further enhance the system's generalization in complex environments.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Lixiao Yang	✓	✓	✓		✓		✓	✓	✓		✓		✓	
Sheng Hua					✓				✓					
Youbing Feng				✓		✓				✓		✓		
Shangzong Yang						✓				✓				
Jie Wang		✓								✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author [FY] on request.

REFERENCES

[1] W. Xu and F. Zhang, "FAST-LIO: a fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021, doi: 10.1109/LRA.2021.3064227.

[2] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: versatile LiDAR SLAM via multi-metric linear least square," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2021, pp. 11633–11640. doi: 10.1109/ICRA48506.2021.9561364.

[3] 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 Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022, doi: 10.1109/LRA.2022.3152830.

[4] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-LIO: robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200459, Jul. 2023, doi: 10.1002/aisy.202200459.

[5] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D LiDar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, May 2019, pp. 3144–3150. doi: 10.1109/ICRA.2019.8793511.

[6] Z. Wang, L. Zhang, Y. Shen, and Y. Zhou, "D-LIOM: tightly-coupled direct LiDAR-inertial odometry and mapping," *IEEE Transactions on Multimedia*, vol. 25, pp. 3905–3920, 2023, doi: 10.1109/TMM.2022.3168423.

[7] S. Zhao, Z. Fang, H. Li, and S. Scherer, "A robust laser-inertial odometry and mapping method for large-scale highway environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Nov. 2019, pp. 1285–1292. doi: 10.1109/IROS40897.2019.8967880.

[8] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: tightly-coupled LiDar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2020, pp. 5135–5142. doi: 10.1109/IROS45743.2020.9341176.

[9] J. Levinson *et al.*, "Towards fully autonomous driving: systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Jun. 2011, pp. 163–168. doi: 10.1109/IVS.2011.5940562.

[10] C. Badue *et al.*, "Self-driving cars: a survey," *Expert Systems with Applications*, vol. 165, p. 113816, Mar. 2021, doi: 10.1016/j.eswa.2020.113816.

[11] S. Liu *et al.*, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, Jul. 2017, doi: 10.1109/LRA.2017.2663526.

[12] A. Mokhtari, A. Benallegue, and Y. Orlov, "Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle," *International Journal of Robotics and Automation*, vol. 21, no. 1, pp. 39–49, 2006, doi: 10.2316/Journal.206.2006.1.206-2842.

[13] S. Parsons, "Autonomous robots: from biological inspiration to implementation and control by George A. Bekey, MIT press, 560 pp., $55.00, ISBN 0-262-02578-7," *The Knowledge Engineering Review*, vol. 20, no. 2, pp. 197–198, Jun. 2005, doi: 10.1017/S0269888905210512.

[14] G. Kim and A. Kim, "Scan context: egocentric spatial descriptor for place recognition within 3D point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2018, pp. 4802–4809. doi: 10.1109/IROS.2018.8593953.

[15] G. Kim, S. Choi, and A. Kim, "Scan context++: structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1856–1874, Jun. 2022, doi: 10.1109/TRO.2021.3116424.

[16] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, "BTC: a binary and triangle combined descriptor for 3-D place recognition," *IEEE Transactions on Robotics*, vol. 40, pp. 1580–1599, 2024, doi: 10.1109/TRO.2024.3353076.

[17] C. Yuan, J. Lin, Z. Zou, X. Hong, and F. Zhang, "STD: stable triangle descriptor for 3D place recognition," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1897–1903, May 2022, doi: 10.1109/ICRA48891.2023.10160413.

[18] M. Cummins and P. Newman, "FAB-MAP: probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, Jun. 2008, doi: 10.1177/0278364908090961.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.

[20] T. Botterill, S. Mills, and R. Green, "Bag-of-words-driven, single-camera simultaneous localization and mapping," *Journal of Field Robotics*, vol. 28, no. 2, pp. 204–226, Mar. 2011, doi: 10.1002/rob.20368.

[21] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: fast direct LiDAR-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022, doi: 10.1109/TRO.2022.3141876.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, doi: 10.1177/0278364913491297.

[24] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: segment based place recognition in 3D point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2017, pp. 5266–5272. doi: 10.1109/ICRA.2017.7989618.

[25] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online LiDAR odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8518–8525, Jul. 2022, doi: 10.1109/LRA.2022.3187250.

BIOGRAPHIES OF AUTHORS

Lixiao Yang received the B.E. degree from Jiangsu University of Science and Technology, Zhenjiang, China, in 2023. He is currently working toward the M.S. degree with the Ocean College, Jiangsu University of Science and Technology, Zhenjiang, China. His research interests include SLAM and sensor fusion. He can be contacted at 13052750039@163.com.

Sheng Hua received the B.E. degree from Nanjing University of Science and Technology Zijin College, Nanjing, China, in 2021. He is currently working toward the M.S. degree with the Ocean College, Jiangsu University of Science and Technology, Zhenjiang, China. His research interests include wireless sensor network technology. He can be contacted at 1056541275@qq.com.

Youbing Feng (CCF member) received the B.E. degree from Suzhou University, Suzhou, China, in 2000. and the Ph.D. degree from Jiangsu University, Zhenjiang, China, in 2009. He is currently an assistant professor at Ocean College, Jiangsu University of Science and Technology. His research interests include agricultural internet of things, wireless sensor network and control of UAVS. He can be contacted at yzfyb@just.edu.cn.

Shangzong Yang received the B.E. degree from Nanjing University of Science and Technology Taizhou College Taizhou, China, in 2023, He is currently working toward the M.S. degree with the computer college, Jiangsu University of Science and Technology, Zhenjiang, China, His research interests include computer vision and robotic grasping, He can be contacted at email: justysz13@163.com.

Jie Wang received the B.E. degree from Jiangsu University of Science and Technology, Zhenjiang, China, in 2024. She is currently working toward the M.S. degree with the School of Materials Science and Technology, Shanghai University, Shanghai, China. Her research interests include corrosion resistant material. She can be contacted at email: 18151919798@163.com.