

SC-FAST-LIVO2: A Cross-Modal Loop Closure Framework for Long-Term SLAM with Reduced Drift

Yuang Zhou*

*Dundee International Institute
Central South University
Changsha, P.R.China
2617376@dundee.ac.uk*

Xiankun Jiang*,†

*International School
Beijing University of Posts and Telecommunications
Beijing, P.R.China
jiangxiankun@bupt.edu.cn
†Corresponding author*

Hongze Wang

*Changkong College
Nanjing University of Aeronautics and Astronautics
Nanjing, P.R.China
wanghongze@nuaa.edu.cn*

Zihan Zhao

*Dundee International Institute
Central South University
Changsha, P.R.China
2617375@dundee.ac.uk*

Xuanbo Wang

*School of Mechanical Engineering
Dalian University of Technology
Dalian, P.R.China
xuanbo@mail.dlut.edu.cn*

Abstract—This paper presents a hybrid loop closure detection framework for LiDAR-Visual SLAM systems, integrating a robust Scan Context descriptor and a Pose Graph Optimization module into a framework named SC-FAST-LIVO2**. The proposed module performs keyframe extraction and pose graph construction, and enables threshold-based loop closure detection using normalized similarity scoring for candidate matching. This module also facilitates the final global map establishment and optimization while eliminating drifts for global consistency. The Pose Graph Optimization module is integrated using SE(3) constraints, subscribing the point cloud message, the pose information, and publishing the feedback with adaptive covariance scaling. Extensive experiments show the proposed method reduces the absolute translational errors, demonstrating an effective drift correction in large-scale environments. The modular system design enables independent enhancement of individual components while maintaining overall SLAM consistency.

Index Terms—SLAM, Pose Graph Optimization, Loop Closure Detection, Scan Context

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) [1] technology stands as the foundational pillar enabling autonomous robots [2], self-driving [3], and unmanned aerial vehicles (UAV) [4] to navigate and interact with their surroundings reliably even in environments where GPS is unavailable, such as subterranean tunnels [5], dense urban canyons [6], or indoor facilities [7], as well as in entirely unexplored environments. In these scenarios, where external positioning infrastructure is unavailable or unreliable, SLAM systems must autonomously construct environmental maps while precisely tracking the robot pose in real time. Over the past decade, advancements in

sensor technology and algorithmic innovation have propelled SLAM performance to new heights such as high resolution LiDARs [8], economical IMUs [9], making it a cornerstone of modern autonomous systems.

While state-of-the-art LiDAR-Inertial Odometry (LIO) [10] systems have achieved impressive pose estimation accuracy in the short term, long-term navigation remains hindered by persistent error accumulation. Drifts as small as 1–2% of the distance traveled accumulate over time, leading to substantial positional inaccuracies in extended operation. For example, in some sequences of public dataset NTU VIRAL, LIO-SAM [11] results in 1257.159 m drift in maximum [12], which can render maps inconsistent, disrupt task planning, and even compromise the safety of autonomous operations. This limitation substantially constrains the applicability of SLAM in missions of long duration.

Recent LIO work splits into two complementary directions. On the tight coupling side, LIO-SAM [11] fuses LiDAR and IMU with robust loop closure to curb drift and preserve global consistency for long-term navigation, and applies Pose Graph Optimization (PGO) to accelerate loop closure and pose graph construction while meeting real-time constraints. On the other hand, FAST-LIVO2 [10] has emerged as a benchmark in real-time performance and computational efficiency. By employing a tightly optimized, sparse direct data processing pipeline and a sequentially updated error-state iterated Kalman filter (ESIKF) [10], FAST-LIVO2 excels in dynamic environments, where obstacles or moving objects are prevalent, as well as on resource-constrained platforms, such as small robots or drones with limited computational power. However, this focus on speed and efficiency comes at a cost: FAST-LIVO2

** Code: https://github.com/jxk6575/SC_FAST_LIVO2

lacks a dedicated loop closure mechanism, leading to reduced long-term consistency. In large scale deployments, where the trajectory of the robot spans hundreds of meters or more, the absence of drift correction can result in significant map distortion and pose inaccuracies.

The purpose of this work is to retain the low latency odometry of FAST-LIVO2 while addressing its lack of a dedicated loop closure mechanism, which degrades long-term consistency on large trajectories. Therefore, the robust loop closure module and Pose Graph Optimization backend are designed and integrated into the FAST-LIVO2 system. This complements the original front end, turning it into a full SLAM system capable of correcting drift at scale. The proposed system, SC-FAST-LIVO2, is a hybrid framework that seamlessly integrates a lightweight yet robust loop closure detection mechanism based on Scan Context descriptors and Pose Graph Optimization into FAST-LIVO2. Therefore, the main contributions of the paper include:

- 1) We propose a unified loop closure and Pose Graph Optimization that augments the FAST-LIVO2 front end, improving long-term consistency in large-scale environments.
- 2) We formulate SE(3) Pose Graph Optimization with adaptive covariances and robust loss, solved incrementally via Incremental Smoothing and Mapping 2 (iSAM2) [13], and propagate corrections to the shared voxel map.
- 3) We propose a robust loop closure detection scheme based on a cross-modal Scan Context descriptor that reduces absolute translation error and improves trajectory accuracy.

This research advances the field of autonomous navigation by showcasing that careful system integration can balance real-time performance and mapping accuracy. Moreover, the modular design of our architecture ensures backward compatibility with existing FAST-LIVO2 applications, allowing easy adoption in existing robotics systems, while greatly expanding its operational envelope to support long-duration, large-scale missions from industrial inspection to urban exploration.

The remainder of this paper is organized as follows. Section II reviews the related work on LiDAR-Inertial-Visual SLAM and loop closure. Section III presents the methodology of the proposed SC-FAST-LIVO2 framework, beginning with the preliminaries of the Scan Context descriptor and the similarity computation used for place recognition, and then detailing the overall system architecture, the loop closure module, pose graph optimization, and their interaction with the FAST-LIVO2 front end. Section IV reports the experimental setup, datasets, benchmarking protocols, and both quantitative and qualitative results. Section V concludes the paper and discusses future research directions.

II. RELATED WORK

Over the past few years, LiDAR SLAM has improved rapidly due to its robustness to lighting variation and faithful geometric mapping even in complex environments. This progress has been particularly driven by three key technological trends: the development of solid-state LiDAR with higher point density and lower power consumption, the emergence of

efficient point cloud processing algorithms, and the increasing computational power of embedded platforms.

A. LiDAR Odometry Methods

The seminal LOAM framework [8] established an approach with two tiers by separating high-frequency odometry from low-frequency mapping, improving real-time performance and accuracy through registration based on features. However, LOAM is limited by its lack of explicit loop closure, resulting in accumulated drift over long missions. For ground robots, performance of LeGO-LOAM [14] is further enhanced by using ground segmentation and a robust optimization scheme, but both frameworks remain loosely coupled and lack global consistency across large and repetitive environments.

Tightly coupled direct registration approaches mark significant progress. FAST-LIO [15] employs an iterated Kalman filter on raw LiDAR points, eliminating the need for feature extraction. The subsequent FAST-LIO2 [16] introduces incremental KD-tree map maintenance to achieve lightweight, high-frequency, and accurate local odometry. Despite these advances, these systems primarily function as odometry front ends, and in the absence of loop closure, suffer from accumulated drift in long-term deployments.

B. Multi-Sensor Fusion

To further improve robustness, particularly in sensor-challenging scenarios, multi-source SLAM [17]–[19] frameworks such as VINS-Mono [9], FI-SLAM [20], and FAST-LIVO [21] have fused visual, LiDAR, and inertial data. Notably, the tightly coupled Visual-Inertial subsystem in these frameworks can maintain tracking during short-term LiDAR dropout. FAST-LIVO2 [10] exemplifies this by jointly optimizing these modalities via a lightweight front-end and an efficient ESIKF. However, these systems still lack a global loop-closure mechanism and remain susceptible to cumulative error over long trajectories.

C. Toward Global Consistency Approaches

Addressing long-term drift, graph SLAM frameworks have become mainstream. LIO-SAM [11] incorporates global graph optimization by fusing LiDAR odometry, IMU pre-integration, GPS information and loop closure through a graph factor. However, real-time performance may be affected in large-scale scenarios because maintaining and optimizing a global map introduces noticeable computational overhead.

Among loop closure mechanisms, Scan Context descriptors [22] generate rotation-invariant descriptors for efficient LiDAR place recognition, while in the visual domain, the bag of words (BoW) method [23] enables scalable visual loop closure through hierarchical vocabulary trees for binary descriptors. Both approaches exploit compact and discriminative features coupled with efficient search, greatly aiding drift correction and global consistency, and have influenced many recent SLAM systems.

Overall, while the frontier LiDAR-Inertial-Visual-Odometry (LIVO) systems such as FAST-LIVO2 excel in real-time performance and robustness in dynamic or resource-constrained

environments, they inherently suffer from cumulative drift errors over long-term operations due to the lack of a global loop closure mechanism. This limitation becomes particularly critical in large-scale applications, where even minor drifts compound significantly, leading to substantial pose inaccuracies and inconsistent maps.

III. METHODOLOGY

We will describe the proposed SC-FAST-LIVO2 framework in this section. This section describes the proposed SC-FAST-LIVO2 framework, with emphasis on the integration of the descriptor-based loop-closure module into FAST-LIVO2.

A. Preliminaries

Scan Context is a lightweight, non-histogram global descriptor that has proven effective in LiDAR-based loop closure, and has been adopted by SC-A-LOAM [24], SC-LIO-SAM [25], SC-LeGO-LOAM [26], and FAST-LIO-SLAM [27]. In this work we integrate Scan Context [22] into FAST-LIVO2, yielding SC-FAST-LIVO2, embedded in the SC-PGO Module which will be introduced in the next part. The basic principles of Scan Context are summarized below:

1) *Point-Cloud Partitioning*: Let the point cloud be partitioned by N_r concentric rings and N_s azimuthal sectors:

$$\mathcal{P} = \bigcup_{\substack{i \in [N_r] \\ j \in [N_s]}} \mathcal{P}_{ij}, \quad (1)$$

where $\mathcal{P}_{ij}$ denotes the sub-cloud in ring i and sector j .

2) *Bin Encoding*: Each bin is mapped to a real value via

$$\phi: \mathcal{P}_{ij} \rightarrow \mathbb{R}, \quad \phi(\mathcal{P}_{ij}) = \max_{\mathbf{p} \in \mathcal{P}_{ij}} z(\mathbf{p}). \quad (2)$$

where $z(\cdot)$ is the z -coordinate.

3) *Scan Context Matrix*: Stacking the bin values yields the SC matrix

$$I = (a_{ij}) \in \mathbb{R}^{N_r \times N_s}, \quad a_{ij} = \phi(\mathcal{P}_{ij}). \quad (3)$$

Root-shift augmentation can be applied to improve robustness.

4) *Similarity*: Given a query I^q and a candidate I^c , define column vectors c_j^q and c_j^c . The cosine distance is

$$d(I^q, I^c) = \frac{1}{N_s} \sum_{j=1}^{N_s} \left(1 - \frac{c_j^q \cdot c_j^c}{\|c_j^q\| \|c_j^c\|} \right). \quad (4)$$

To handle yaw, choose the best circular column shift of I^c :

$$D(I^q, I^c) = \min_{n \in [N_s]} d(I^q, I_n^c), \quad n^* = \operatorname{argmin}_{n \in [N_s]} d(I^q, I_n^c). \quad (5)$$

5) *Two-Phase Search*: Compute the ring key from row sparsity:

$$k = (\psi(r_1), \dots, \psi(r_{N_r})), \quad \text{where } \psi: r_i \rightarrow \mathbb{R}, \quad (6)$$

$$\psi(r_i) = \frac{\|r_i\|_0}{N_s}.$$

where r_i is the i -th row of I . First retrieve K nearest neighbors of the query key in a KD-tree, then verify each candidate by $D(I_q, I_c)$. Accept the loop if

$$c^* = \operatorname{argmin}_{c_k \in \mathcal{C}} D(I^q, I^{c_k}), \quad \text{s.t. } D < \tau. \quad (7)$$

and the optimal shift n^* gives a coarse yaw estimate. More detailed information can be obtained from the Scan Context [22].

B. System Overview

The overall architecture of SC-FAST-LIVO2 is shown in Fig 1, which contains 6 components: Sensor Data Input, Forward Propagation, Backward Propagation, State Estimation, Map Construction and Output Layer.

The main parts of the FAST-LIVO2 are Sensor Input, Forward Propagation, State Estimation, Map Construction and Output Layer parts. The details have been discussed in FAST-LIVO2 [10]. The backward propagation process is described in detail below.

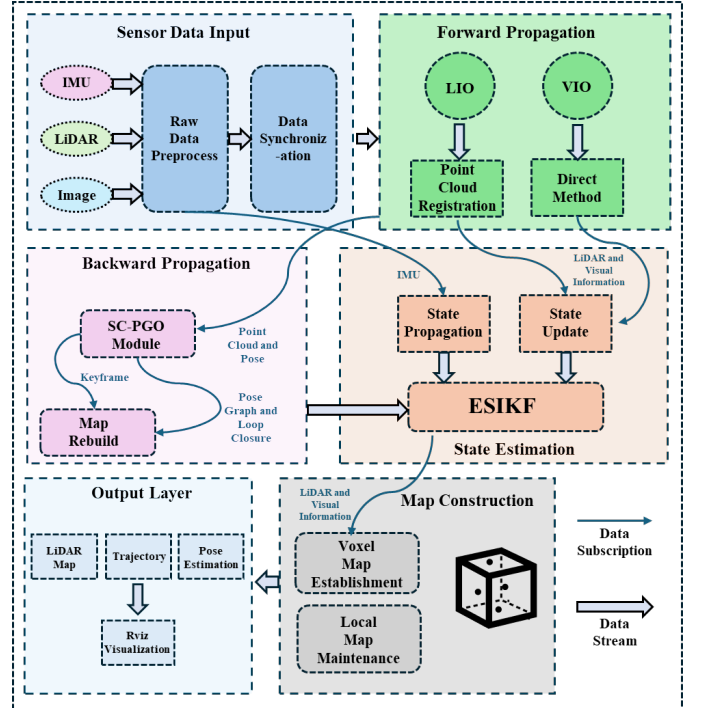


Fig. 1: Basic Architecture of SC-FAST-LIVO2

C. Scan Context-Pose Graph Optimization (SC-PGO)

The Scan Context-Pose Graph Optimization module is the core component for ensuring global positioning consistency, and the whole structure about the SC-PGO module is shown as pseudocode in Algorithm 1. It corrects cumulative drift errors from the LiDAR-Inertial-Visual Odometry over extended operation. The process involves five key steps: keyframe selection, loop closure detection and verification, pose graph construction, incremental Pose Graph Optimization, and map update and drift optimization. By integrating local poses into a

Algorithm 1 SC-PGO Module for Global Consistency

Require: Keyframe pose $\mathbf{T}_k$; point cloud $\mathcal{P}_k$; translation threshold τ_t ; rotation threshold τ_r ; similarity threshold τ_s ; fitness threshold τ_f

Ensure: Optimized global poses $\{\mathbf{T}_k^*\}$; updated global map $\mathcal{M}$

```
1: Keyframe Selection:
2: if  $d_{\text{trans}} > \tau_t$  OR  $d_{\text{rot}} > \tau_r$  then
3:   Create new keyframe with  $\mathbf{T}_k$ ,  $\mathcal{P}_k$ 
4:   Add node  $\mathbf{T}_k$  to pose graph
5: Loop Closure Detection (Coarse):
6: Extract Scan Context descriptor  $\mathbf{SC}_k$  from  $\mathcal{P}_k$ 
7: Compute similarity  $s(\mathbf{SC}_k, \mathbf{SC}_j)$  for all past keyframes
8: if  $\exists j$  such that  $s(\mathbf{SC}_k, \mathbf{SC}_j) > \tau_s$  then
9:   Add candidate  $(k, j)$  to loop closure list  $\mathcal{L}$ 
10: Loop Closure Verification (Fine):
11: for all  $(k, j) \in \mathcal{L}$  do
12:   Align  $\mathcal{P}_k$  and  $\mathcal{P}_j$  using ICP:
13:    $\min_{\mathbf{R}, \mathbf{t}} \sum \left\| \mathbf{R} \cdot \mathbf{p}_i^{\text{world}} + \mathbf{t} - \mathbf{q}_i^{\text{body}} \right\|_2^2$ 
14:   Compute fitness score:  $\text{fitness} = \frac{1}{M} \sum \left\| \mathbf{R} \cdot \mathbf{p}_i + \mathbf{t} - \mathbf{q}_i \right\|$ 
15:   if  $\text{fitness} < \tau_f$  then
16:     Add loop closure factor  $\mathbf{f}_{\text{loop}}(\mathbf{T}_k, \mathbf{T}_j)$  to graph
17: Pose Graph Optimization:
18: Construct graph with factors: prior, odometry, loop closure
19: Solve:  $\{\mathbf{T}_k^*\} = \arg \min_{\{\mathbf{T}_k\}} \sum \left\| \mathbf{f}(\mathbf{T}_i, \mathbf{T}_j) \right\|_\Sigma^2$  using iSAM2
20: Propagate corrections:  $\Delta \mathbf{T}_k = \mathbf{T}_k^* \cdot \mathbf{T}_k^{-1}$ 
21: Map Update:
22: for all keyframes do
23:   Transform point cloud:  $\mathbf{p}_i^{\text{world}} = \mathbf{T}_k^* \cdot \mathbf{q}_i^{\text{body}}$ 
24:   Fuse into global voxel map  $\mathcal{M}$  with downsampling
25: return  $\{\mathbf{T}_k^*\}$ ,  $\mathcal{M}$ 
```

globally consistent framework, this module enhances mapping accuracy and robustness in complex environments, making it essential for robotics, autonomous navigation, and large-scale mapping applications.

1) *Keyframe Selection:* To balance computing speed and accuracy, the system uses a smart keyframe selection method. It tracks robot movement in real time and marks the current pose as a keyframe when certain geometric conditions are met. The system uses a factor graph to represent constraints between keyframes. Each keyframe pose is a node in $\text{SE}(3)$, with transformation matrix:

$$\mathbf{T}_k = \begin{bmatrix} \mathbf{R}_k & \mathbf{t}_k \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (8)$$

where $\mathbf{R}_k \in \text{SO}(3)$ denotes the current rotation matrix, and $\mathbf{t}_k \in \mathbb{R}^3$ denotes the current translation vector.

Two geometric criteria are applied to evaluate motion extent: For translation, the Euclidean distance is expressed as:

$$d_{\text{trans}} = \|\mathbf{t}_k - \mathbf{t}_{\text{last}}\|_2, \quad (9)$$

where the Euclidean distance between the current translation vector $\mathbf{t}_k$ and that of the last keyframe $\mathbf{t}_{\text{last}}$ is computed.

For rotation, the angular difference between rotation matrices $\mathbf{R}_k$ and $\mathbf{R}_{\text{last}}$ is measured by:

$$d_{\text{rot}} = \arccos \left(\frac{\text{tr}(\mathbf{R}_k \cdot \mathbf{R}_{\text{last}}^T) - 1}{2} \right) \times \frac{180^\circ}{\pi}. \quad (10)$$

A new keyframe is created when movement or rotation exceeds the threshold, capturing major pose changes. Each keyframe contains full pose data, point clouds, and feature descriptors, providing strong constraints for pose optimization. This approach balances processing cost and trajectory precision.

2) *Loop Closure Detection and Verification:* Loop closure detection identifies robot revisits to prior areas, offering essential global constraints for PGO. The system uses a two-stage coarse-to-fine architecture to ensure efficiency and reliability.

In coarse matching, rotation-invariant Scan Context descriptors extract global features from LiDAR point clouds by projecting 3D points onto a polar grid and analyzing height distribution per cell, forming a 2D environmental feature matrix. Scene similarity between keyframes is computed via:

$$s(\mathbf{SC}_k, \mathbf{SC}_j) = 1 - \frac{\sum_{i,j} |\mathbf{SC}_k(i, j) - \mathbf{SC}_j(i, j)|}{N_r \cdot N_\theta \cdot h_{\text{max}}}. \quad (11)$$

where $\mathbf{SC}_k, \mathbf{SC}_j$ are the feature matrices of keyframes, N_r and N_θ are radial and angular grid sizes, and h_{max} is the maximum detection range of the LiDAR. A similarity score above a threshold suggests a loop closure, triggering fine verification.

In fine verification, the system uses the Iterative Closest Point (ICP) algorithm [28] for precise registration between candidate frames. ICP iteratively establishes point correspondences and optimizes pose estimation, effectively capturing geometric details to converge to an optimal transformation. The optimization objective is:

$$\min_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^M \left\| \mathbf{R} \cdot \mathbf{p}_i^{\text{world}} + \mathbf{t} - \mathbf{q}_i^{\text{body}} \right\|_2^2, \quad (12)$$

where $\mathbf{p}_i^{\text{world}}$ denotes a point in the current frame, and $\mathbf{q}_i^{\text{body}}$ denotes the corresponding point in the candidate sensor frame. $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector to be determined, and M indicates the number of point pairs involved in the registration calculation. Registration quality is assessed via mean fitting error:

$$\text{fitness} = \frac{1}{M} \sum_{i=1}^M \left\| \mathbf{R} \cdot \mathbf{p}_i^{\text{world}} + \mathbf{t} - \mathbf{q}_i^{\text{body}} \right\|_2. \quad (13)$$

A loop closure is accepted only if the error is below a threshold, and the relative pose is added to the pose graph. Temporal consistency checks are incorporated to eliminate unstable matches, further enhancing robustness.

3) *Pose graph construction*: Three factor types form the constraint framework, and they are listed below:

The **prior factor** anchors the initial keyframe:

$$\mathbf{f}_{\text{prior}}(\mathbf{T}_0) = \log(\mathbf{T}_{\text{prior}}^{-1} \cdot \mathbf{T}_0). \quad (14)$$

where $\log: \text{SE}(3) \mapsto \text{se}(3)$ represents the mapping from the Lie group to the Lie algebra, and $\mathbf{T}_{\text{prior}} = \mathbf{I}$, $\mathbf{T}_0$ is the pose of the initial frame.

The **odometry factor** links consecutive keyframes using relative pose $\hat{\mathbf{T}}_{k,k+1}$:

$$\mathbf{f}_{\text{odom}}(\mathbf{T}_k, \mathbf{T}_{k+1}) = \log(\hat{\mathbf{T}}_{k,k+1}^{-1} \cdot (\mathbf{T}_k^{-1} \cdot \mathbf{T}_{k+1})). \quad (15)$$

where $\hat{\mathbf{T}}_{k,k+1} \in \text{SE}(3)$ is the relative pose observation from frame k to frame $k+1$ obtained from odometry measurements.

The **loop closure factor** imposes strong constraints when a loop is detected:

$$\mathbf{f}_{\text{loop}}(\mathbf{T}_k, \mathbf{T}_j) = \log(\hat{\mathbf{T}}_{k,j}^{-1} \cdot (\mathbf{T}_k^{-1} \cdot \mathbf{T}_j)). \quad (16)$$

where $\hat{\mathbf{T}}_{k,j}$ is the relative pose observation value calculated through point cloud registration.

4) *Incremental Pose Graph Optimization*: Pose Graph Optimization aims to find the optimal pose set minimizing global constraint errors, equivalent to maximum a posterior estimation:

$$\{\mathbf{T}_k^*\} = \arg \min_{\{\mathbf{T}_k\}} \sum_{\text{factors}} \|\mathbf{f}(\mathbf{T}_i, \mathbf{T}_j)\|_{\Sigma}^2. \quad (17)$$

where $\{\mathbf{T}_k^*\}$ is the collection of all optimal keyframe poses, the summation Σ is over all constraint factors (prior, odometry and loop closure) in the graph, $\|\cdot\|_{\Sigma}^2$ is the Mahalanobis distance and Σ is the covariance matrix associated with each constraint. Note that k indexes all keyframes, while each factor $\mathbf{f}$ involves a specific pair of keyframes (i, j) .

The system uses the iSAM2 [13] algorithm, which maintains a Bayesian tree to optimize only affected subgraphs upon new data. It preserves sparsity through marginalization for real-time operation. The process is initialized with fused odometry, and the output is the corrected set of global poses.

5) *Map Update and Drift Compensation*: After pose graph optimization, the system updates the global map and compensates drift using the optimized poses. For each keyframe, the pose correction is calculated as:

$$\Delta \mathbf{T}_k = \mathbf{T}_k^* \cdot \mathbf{T}_k^{-1} \quad (18)$$

Here, $\Delta \mathbf{T}_k$ represents the correction amount for transforming the pose $\mathbf{T}_k$ before optimization to the pose $\mathbf{T}_k^*$ after optimization. The raw point cloud is then transformed into the global frame via:

$$\mathbf{p}_i^{\text{world}} = \mathbf{T}_k^* \cdot \mathbf{q}_i^{\text{body}}. \quad (19)$$

which transforms the point cloud in the sensor coordinate system to the global coordinate system through the optimized pose.

Finally, voxel grid filtering down samples and fuses the corrected point cloud, building a globally consistent dense map. This eliminates accumulated drift and ensures global consistency.

D. Interaction Between SC-PGO Module and Other Components

The interaction of SC-PGO module with other system components is crucial for improving accuracy and reducing odometry drift.

1) *Subscribe Forward Information from FAST-LIVO2*: The PGO module receives external data to construct the constraints. By subscribing the topic concerning LiDAR odometry, PGO module is capable of capturing the real-time pose calculated by LIO and Visual-Inertial Odometry (VIO) in forward propagation, which includes the position in 3D framework and roll, pitch, yaw angles. This subscription is used for extracting keyframes and constructing primary pose graph.

The complete point cloud message is also subscribed by the PGO module. The raw point cloud information is preprocessed originally, then goes into the Point Cloud Registration module in the forward propagation section for further feature extraction. After this procedure, the point clouds are fed into the LiDAR odometry module for subsequent processing, then subscribed by the PGO module as the constraints for keyframe storage and ICP registration.

2) *Publish Optimized Information to FAST-LIVO2*: Upon receiving a new keyframe from the PGO module, the Scan Context generates the Ring Key and Sector Key. A KD tree is built to filter the candidate keyframes based on similarity scoring. Detected loop closure will be added into the pose graph. The integration of the Scan Context and PGO enhances global pose optimization via iSAM2 algorithm, after which visualization messages are published to FAST-LIVO2 with synchronized keyframe index. The PGO module at backend will concentrate more on the accumulated odometry errors by FAST-LIVO2 at the front end, which includes keyframe extraction, pose graph construction, loop closure detection and global optimization.

IV. EXPERIMENT RESULTS

This section presents the experimental setup and results, including baseline comparisons to baselines and evaluations in terms of Absolute Translational Errors and qualitative mapping quality.

A. Implementation and System Configuration

The proposed SC-FAST-LIVO2 system is implemented in ROS. The default FAST-LIVO2 configuration is used, where exposure time estimation and reference patch update are both enabled, while normal vector refinement is disabled. LiDAR points in each scan are downsampled at a 1:3 ratio. The root voxel size for the voxel map is set at 0.5 m, and the maximum layer of the internal octree is 3. The image patch size is 8×8 for image alignment and 11×11 for normal refinement. In the sequential ESIKF settings, the camera photometric noise is set to a constant value of 100. In the PGO settings, the update frequency is set to 20. Our system uses identical parameters across all sequences and datasets with the same sensor configuration. The experimental platform is equipped with an Intel i9-13900H CPU and 32 GB RAM.

FAST-LIVO2

SC-FAST-LIVO2

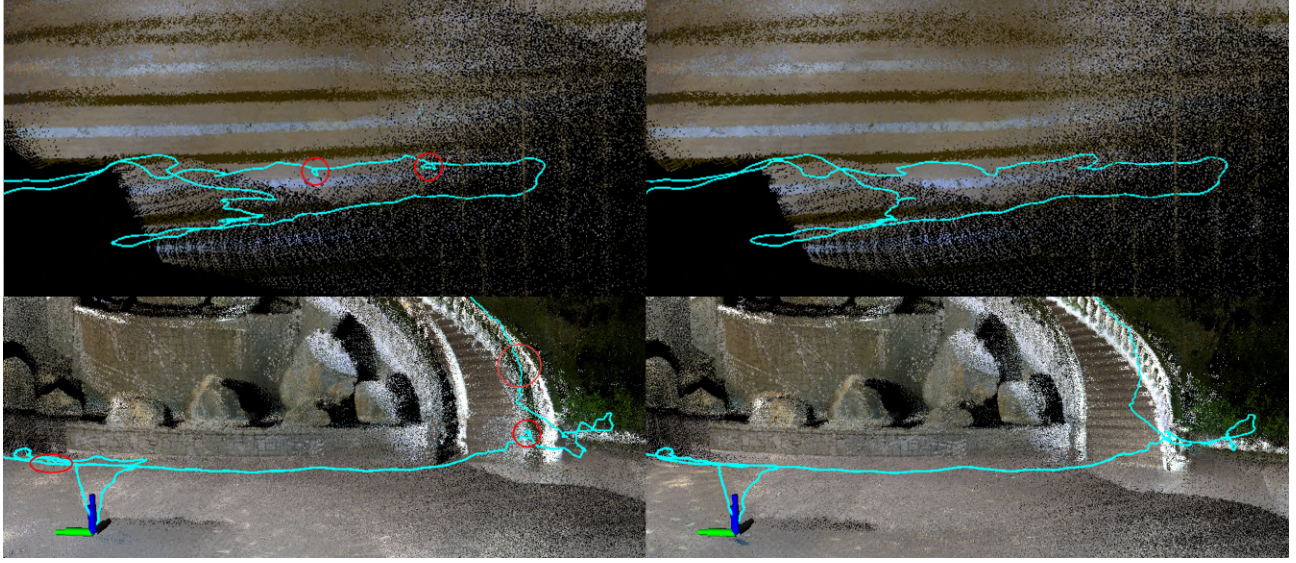


Fig. 2: Loop closure comparison results of colored point cloud and path constructed by the original FAST-LIVO2 system and SC-FAST-LIVO2. The point clouds from top to bottom correspond to "HKU Lecture Center" and "HKU Cultural Center".

B. Datasets

1) *Hilti datasets*: The Hilti'22 dataset [29] and Hilti'23 dataset [30] comprise indoor and outdoor sequences acquired across heterogeneous settings, including sparse and conventional construction sites, corridors, rooms, and the galleries of the Sheldonian Theatre in Oxford. These sequences present various challenging environments characterized by textureless surfaces, varying illumination conditions, and insufficient LiDAR planar constraints. For each sequence, millimeter-accurate ground truth obtained through either a motion capture system (MoCap) or a Total Station [31] is provided. Note that the ground truth of the Hilti datasets is not publicly available. Therefore, algorithmic results on these datasets are evaluated through the official Hilti evaluation platform.

2) *FAST-LIVO2 Private Dataset*: To evaluate the performance of the PGO module, we conducted additional experiments using the private dataset constructed by FAST-LIVO2. The FAST-LIVO2 private dataset consists of 20 sequences captured across a diverse range of environments, such as campus buildings, corridors, basements, and mining tunnels, characterized by structural sparsity, clutter, poor illumination, significant lighting variations, and weak texture. Each sequence is designed with a return-to-start trajectory, enabling accurate drift evaluation.

C. Benchmark Experiments

In this experiment, evaluations are conducted on sequences from the Hilti'22 and Hilti'23 datasets. Our method is benchmarked against several state-of-the-art odometry systems, including SDV-LOAM [32], a semi-direct LiDAR-Visual

Odometry system; FAST-LIO2 [16], a direct LiDAR-Inertial Odometry system; FL2BIPS [33], a LiDAR-Inertial Odometry system with Bayesian ICP pose optimization that combines FAST-LIO2 filtering with batch Pose Graph Optimization; FL2BIPS-GTSFM [33], a multi-modal SLAM system that extends FL2BIPS with visual bundle adjustment using a distributed structure-from-motion pipeline powered by Georgia Tech Structure from Motion (GTSAM); Leapmotor, a LiDAR-Inertial Odometry system based on FAST-LIO with a real-time slide-window optimization; FrankenPharos, a LiDAR-Inertial SLAM system that enables simultaneous optimization of poses and IMU biases while incorporating online de-skewing and X-ICP-based degeneracy awareness. FAST-LIVO [21], a fast and tightly-coupled sparse-direct LiDAR-Inertial-Visual odometry system; and FAST-LIVO2 [10], an improved version of FAST-LIVO employing a sequentially-updated ESIKF.

For FAST-LIO2, FAST-LIVO, and FAST-LIVO2, we use the recommended settings for both indoor and outdoor environments. For SDV-LOAM, we utilize the open-source implementation of a loosely coupled integration framework that combines the vision module with LeGO-LOAM. The corresponding evaluation results are directly taken from the FAST-LIVO2 paper [10]. Furthermore, we utilize the publicly available results of FL2BIPS, FL2BIPS-GTSFM, Leapmotor and FrankenPharos from the official Hilti'22 and Hilti'23 leaderboards for objective comparison.

The results from these experiments are shown in Table I and Table II, demonstrating that our system outperforms all other methods across most sequences with an average RMSE of 0.035 m, which is 6.58% lower than the baseline FAST-LIVO2

TABLE I: Absolute Translational Errors (RMSE, meters) in Hilti'22 Sequences

Dataset	Sequence	SDV-LOAM*	FAST-LIO2*	FL2BIPS**	FL2BIPS-GTSFM**	FAST-LIVO*	FAST-LIVO2*	Ours
Hilti'22	Construction Ground	25.121	0.013	0.015	0.051	0.022	0.010	0.009
	Construction Multilevel	12.561	0.044	0.037	0.122	0.052	0.020	0.019
	Construction Stairs	9.212	0.320	1.392	2.674	0.241	0.016	0.014
	Long Corridor	19.531	0.064	0.166	0.387	0.065	0.067	0.058
	Cupola	9.231	0.250	0.915	4.180	0.182	0.180	0.161
	Lower Gallery	11.232	0.024	0.231	0.800	0.022	0.007	0.006
	Attic to Upper Gallery	4.551	0.720	0.300	0.121	0.621	0.069	0.073
	Outside Building	2.622	0.028	0.032	0.118	0.052	0.035	0.034
Average		11.758	0.183	0.386	1.074	0.157	0.505	0.468

* These models and data are provided by FAST-LIVO2 [10]

** These models and data are provided by Hilti platform [29]

TABLE II: Absolute Translational Errors (RMSE, meters) in Hilti'23 Sequences

Dataset	Sequence	SDV-LOAM*	FAST-LIO2*	Leapmotor**	FrankenPharos**	FAST-LIVO*	FAST-LIVO2*	Ours
Hilti'23	Floor 0	4.621	0.031	0.031	0.025	0.021	0.022	0.019
	Floor 1	7.951	0.031	0.041	0.023	0.022	0.023	0.021
	Floor 2	7.912	0.083	0.031	0.037	0.048	0.022	0.021
	Underground	6.151	0.038	0.025	0.017	0.035	0.016	0.015
	Stairs	9.032	0.170	0.088	0.011	0.152	0.018	0.015
	Parking 3x floors down	19.952	0.320	0.553	0.400	0.356	0.032	0.026
	Floor 1 Large room	16.781	0.028	0.066	0.048	0.031	0.026	0.028
	Floor 2 Large room - dark	15.012	0.040	0.065	0.044	0.053	0.046	0.047
Average		10.927	0.093	0.113	0.076	0.090	0.026	0.024

* These models and data are provided by FAST-LIVO2 [10]

** These models and data are provided by Hilti platform [30]

framework. Our system achieves the best performance in the majority of sequences. It is worth noting that in the Outside Building sequence, our system does not exhibit significant improvement in localization accuracy compared to the baseline FAST-LIVO2 framework. This observation can be attributed to the absence of loop closures in the ground truth trajectory of this sequence. Consequently, our system cannot enhance the trajectory accuracy through loop closure detection. Excluding this sequence, our method demonstrates superior localization performance compared to the baseline FAST-LIVO2. This improvement can be attributed to the precise loop closure detection and Pose Graph Optimization implemented in SC-FAST-LIVO2, which effectively reduces the accumulated drift error during long-term mapping, resulting in more accurate trajectory reconstruction.

D. High Accuracy Mapping

This experiment evaluates the high-precision localization and mapping capabilities of our proposed system using the FAST-LIVO2 private dataset. As shown in Figure 2, SC-FAST-LIVO2 demonstrates superior localization and mapping performance in both indoor and outdoor environments. In regions characterized by large viewing angles between the camera and scene, our method significantly improves mapping quality through pose calibration and local map reconstruction approaches. Regarding trajectory generation, our method effectively eliminates the loop closure artifacts exhibited in the baseline FAST-LIVO2 and achieves significant improvements in both trajectory accuracy and pose optimization.

V. CONCLUSIONS AND FUTURE WORK

This paper presents SC-FAST-LIVO2, a hybrid loop-closure and optimization framework in the back end that augments the FAST-LIVO2 front end with Scan Context-based place recognition and iSAM2-based pose graph optimization.

Experiments on Hilti'22, Hilti'23, and a private dataset demonstrate consistent reductions in RMSE and visibly improved map consistency across challenging scenarios, outperforming the baseline models. Nevertheless, some limitations remain. In trajectories with few or no revisits, the benefit of loop closure is naturally limited because reliable place-recognition constraints cannot be established. In addition, front-end robustness may degrade under extreme viewpoint changes.

In future work, more loop-closure detection methods will be explored to further improve performance, including approaches that combine LiDAR and visual information. The Scan Context descriptor and the pose graph optimization module will also be enhanced to better exploit multimodal fusion, thereby improving robustness and global consistency.

ACKNOWLEDGMENT

All the authors would like to thank Professor Sen Wang for his insightful discussions about this topic, which shed light on loop closure detection and pose graph optimization method and our further work.

REFERENCES

- [1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, p. 1309–1332, 2016.
- [2] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, "Autonomous robotic exploration using occupancy grid maps and graph slam based on shannon and rényi entropy," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 487–494.
- [3] A. Singandhupe and H. M. La, "A review of slam techniques and security in autonomous driving," in *2019 third IEEE international conference on robotic computing (IRC)*. IEEE, 2019, pp. 602–607.
- [4] G. Sonugür, "A review of quadrotor uav: Control and slam methodologies ranging from conventional to innovative approaches," *Robotics and Autonomous Systems*, vol. 161, p. 104342, 2023.
- [5] F. D. Lauro, D. G. Sorrenti, and M. A. Sotelo, "Multi-robot LiDAR SLAM: A practical case study in underground tunnel environments," *arXiv preprint arXiv:2507.21553*, 2025.
- [6] Y. Zhang, F. Dong, Q. Sun, and W. Song, "Laser SLAM matching localization method for subway tunnel point clouds," *Sensors*, vol. 25, no. 12, p. 3681, 2025.
- [7] R. Murai, E. Dexheimer, and A. J. Davison, "Mast3r-slam: Real-time dense slam with 3d reconstruction priors," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 695–16 705.
- [8] J. Zhang, S. Singh *et al.*, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [9] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [10] C. Zheng, W. Xu, Z. Zou, T. Hua, C. Yuan, D. He, B. Zhou, Z. Liu, J. Lin, F. Zhu, Y. Ren, R. Wang, F. Meng, and F. Zhang, "Fast-livo2: Fast, direct lidar-inertial-visual odometry," *IEEE Transactions on Robotics*, vol. 41, pp. 326–346, 2025.
- [11] 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, 2020, pp. 5135–5142.
- [12] Y. Gao and L. Zhao, "Ve-liom: A versatile and efficient lidar-inertial odometry and mapping system," *Remote Sensing*, vol. 16, no. 15, p. 2772, 2024.
- [13] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [14] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 4758–4765.
- [15] W. Xu and F. Zhang, "Fast-lío: 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, 2021.
- [16] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lío2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [17] P. Chen, X. Zhao, L. Zeng, L. Liu, S. Liu, L. Sun, Z. Li, H. Chen, G. Liu, Z. Qiao *et al.*, "A review of research on slam technology based on the fusion of lidar and vision," *Sensors*, vol. 25, no. 5, p. 1447, 2025.
- [18] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, and M. Luo, "A review of multi-sensor fusion slam systems based on 3d lidar," *Remote Sensing*, vol. 14, no. 12, p. 2835, 2022.
- [19] Z. Fan, L. Zhang, X. Wang, Y. Shen, and F. Deng, "Lidar, imu, and camera fusion for simultaneous localization and mapping: a systematic review," *Artificial Intelligence Review*, vol. 58, no. 6, pp. 1–59, 2025.
- [20] Z. Song, X. Zhang, R. Wang, S. Zhang, S. Wu, Y. Wang, and J. Yuan, "Fi-slam: Feature information-based robust and efficient lidar slam," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–12, 2025.
- [21] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry," in *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2022, pp. 4003–4009.
- [22] 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, 2018, pp. 4802–4809.
- [23] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [24] gisbi kim, "Sc-a-loam," <https://github.com/gisbi-kim/SC-A-LOAM>, 2021, accessed: 2025/7/30.
- [25] —, "Sc-lío-sam," <https://github.com/gisbi-kim/SC-LIO-SAM>, 2021, accessed: 2025/7/30.
- [26] —, "Sc-lego-loam," <https://github.com/gisbi-kim/SC-LeGO-LOAM>, 2021, accessed: 2025/7/30.
- [27] —, "Fast_lío_slam," https://github.com/gisbi-kim/FAST_LIO_SLAM, 2021, accessed: 2025/7/30.
- [28] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.
- [29] L. Zhang, M. Helmberger, L. F. T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. Fallon, "Hilti-oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 408–415, 2022.
- [30] L. Zhang, M. Helmberger, L. F. T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. F. Fallon, "Hilti-oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping," *IEEE Robotics Autom. Lett.*, vol. 8, no. 1, pp. 408–415, 2023. [Online]. Available: <https://doi.org/10.1109/LRA.2022.3226077>
- [31] C. Klug, C. Arth, D. Schmalstieg, and T. Gloor, "Measurement uncertainty analysis of a robotic total station simulation," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 2576–2582.
- [32] Z. Yuan, Q. Wang, K. Cheng, T. Hao, and X. Yang, "Sdv-loam: Semi-direct visual-lidar odometry and mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 11 203–11 220, 2023.
- [33] J. Chen, X. Hu, S. Ma, J. Jiao, M. Liu, and F. Dellaert, "Fast-lío, then bayesian icp, then gtsfm," *arXiv preprint arXiv:2210.00146*, 2022.