Motivation Contribution Methodology Analysis Weakness

2025 IEEE International Conference on Robotics and Automation (ICRA 2025)
May 19-23, 2025. Atlanta, USA

Scalable Multi-Session Visual SLAM in Large-Scale Scenes with Subgraph Optimization

Xiaokun Pan, Zhenzhe Li, Tianxing Fan, Hongjia Zhai, Hujun Bao, Guofeng Zhang†
State Key Lab of CAD&CG, Zhejiang University

***Abstract*— Multi-session visual SLAM systems enable 6-DoF camera localization along with long-term maintenance and expansion of the global map, by utilizing image data from different sessions. However, in large-scale environments, these systems often suffer from severe scale drift. While modern SLAM systems attempt to maintain global map consistency through loop detection and correction, they still face challenges in terms of convergence and accuracy. In this paper, we propose a robust large-scale multi-session SLAM system for long-term localization and mapping that achieves global consistency. Furthermore, to address the backend optimization problem in large-scale environments, we introduce a hierarchical optimization strategy based on the graph structure. More specifically, a subgraph structure is introduced to reduce the size of problem while effectively propagating scale correction information. In addition, a hierarchical strategy enables coarse-to-fine updates of the graph states. Experimental results not only demonstrate that our method efficiently optimizes the pose graph and maintains map consistency in large-scale environments, but also highlight the effectiveness and scalability of the proposed approach.**

Feels more abstract than practical

I. INTRODUCTION

Implementing long-term visual localization and navigation in large-scale environments is crucial for various applications, including robotics, autonomous driving, virtual reality (VR), and augmented reality (AR). Existing methods have attracted considerable attention, spanning purely vision-based approaches [1], [2], [3], multi-sensor fusion techniques [4], [5], [6], and learning-based methods [7], [8], [9]. However, these approaches are typically constrained to indoor or small-scale environments. As the scale of the scene increases, challenges emerge concerning the efficiency and storage of map representations. The continuous expansion of the system's map description leads to a significant increase in map size for long-term mapping tasks. The optimization efficiency in large-scale environments often deteriorates, compromising real-time performance and, subsequently, accuracy due to the high computational cost of optimization. This raises a critical question: can we design a SLAM system tailored for large-scale environments that maintains a compact map representation while still fulfilling the requirements for long-term visual localization and navigation?

In large-scale SLAM systems, the continuously increasing size of the map over time poses a significant challenge. To address this, pose graph optimization has been utilized in graph-based SLAM for backend map representation, which mitigates the growth of the graph size by using keyframes as graph nodes. Additionally, scale error correction based on loop closure offers a feasible solution for maintaining global consistency in large-scale maps. To further address the linear growth of the backend optimization problem size over time, early works employed graph reduction techniques [10], [11], [12], [13], which necessitate a high level of complexity in the graph's topological structure. These methods aim to reduce the graph size by detecting and marginalizing redundant measurements within the graph structure. However, in visual-based large-scale SLAM systems, the local observations from the sliding window often cannot efficiently associate with landmarks in the global map, resulting in a highly sparse pattern. Then this graph typically exhibits a pose chain structure [14], in which case graph reduction approaches are generally ineffective at substantially reducing the graph size. Moreover, these reduction-based methods often lead to a loss of visual observation information, which can adversely affect the system's relocalization performance. Consequently, graph reduction strategies fail to meet the efficiency requirements of backend optimization in large-scale scenarios. An alternative approach is to employ hierarchical optimization schemes [15], [16], [17]. A common strategy involves constructing a spanning tree of pose nodes or building hierarchical maps that are progressively refined in a top-down manner. While these methods aim to approximate the problem using a smaller-scale sparse representation, they do not always ensure high accuracy. Nonetheless, these structural improvements provide valuable insights into addressing the challenges of large-scale SLAM systems.

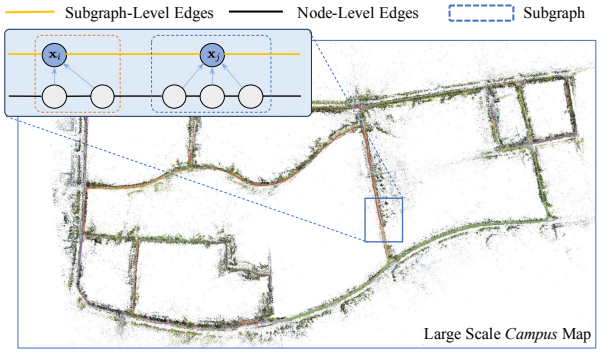

Fig. 1. Large-Scale SLAM with Subgraph. We processed multiple video sequences through our system to obtain a globally consistent map of a large-scale scene, covering an area of about $0.72km^2$, utilizing a subgraph-based backend optimization.

Same issue is being repeated

† Corresponding author: Guofeng Zhang (zhangguofeng@zju.edu.cn)
This work was partially supported by the NSF of China (No.6242500063).

Look into multi sessions

Additionally, in monocular SLAM systems, scale drift remains a significant challenge. Traditional pose graph constraints address drift by "distributing" the loop closure error across the loop, thereby making the entire loop appear consistent. However, the use of consecutive constraints, derived from local map estimates, often results in a non-uniform distribution of errors. These errors can stem from various factors, such as the number and quality of feature points tracked by the frontend, substantial local errors caused by abrupt camera movements, or inaccuracies during relocalization. As a result, relying solely on pose graph optimization may not provide sufficient reliability in terms of accuracy. While a viable approach is to use global bundle adjustment (BA) to jointly optimize both landmarks and camera poses, this method is typically confined to post-processing and is not well-suited for real-time applications, such as robotics and AR. Therefore, a more efficient and accurate solution is needed to effectively address these challenges in real-time scenarios.

To address these issues mentioned above, in this work, we first propose a monocular SLAM framework for large-scale scenes with scalable multi-session capability, and then introduce a novel backend optimization scheme based on a subgraph that enhances the efficiency of optimizing large-scale maps by aggregating local nodes into subgraphs, as shown in Fig. 1. Furthermore, by dynamically adjusting the subgraph structure, we effectively propagate scale errors across different hierarchical levels, thereby improving the overall accuracy of the map representation. Unlike previous approaches, our method focuses on optimizing the backend pose graph structure to enhance both global consistency and computational efficiency, which are crucial for the scalability of SLAM in large-scale environments. The main contributions of this paper are as follows:

- We present a robust and scalable multi-session SLAM system tailored for large-scale environments, achieving high levels of both efficiency and accuracy.
- We introduce a hierarchical pose graph optimization approach based on subgraphs, enabling efficient backend pose optimization.
- We demonstrate state-of-the-art performance in large-scale environments, validating the effectiveness of the proposed method for incremental mapping in expansive scenes and its practical application in AR.

SOTA performance feels vague

II. Related Work

A. Mulit-Session SLAM

Multi-Session SLAM addresses the problem of merging results from repeated executions of SLAM in the same environment, aiming to achieve consistent registration of multiple maps within a global common metric and incremental mapping, which is a promising approach for achieving life-long SLAM. Contrast to multi-agent SLAM [19], [20], [21], [22] which emphasizes the online collaboration of multiple robots,multi-session SLAM focuses on merging the results of multiple independent sessions over different time periods. It has significant applications for long-term SLAM, large-scale scene localization and navigation, and incremental mapping under resource-constrained conditions (*e.g.*, multiple mapping sessions due to battery limitations) [23], [24], [25], [26]. [4], [27] provides multi-session functionality based on pose graph optimization, utilizing the scale recovery capability of IMU to achieve SE(3) relocalization and global trajectory alignment across multiple sessions. Recently [28] extends this by integrating IMU information for more robust system performance. Recent advancements [26] have introduced multi-session two-view matching under wide baselines, along with end-to-end training using a differentiable solver, enabling robust Sim(3) relative pose estimation.

B. Graph Optimization in Visual SLAM

Graph-based approach is widely used as the backend of SLAM to optimize the poses of a robot within a global map, alongside with their relative constraints. This technique refines the poses of nodes solely based on the relationships defined by these constraints to achieve global consistency. Pose graph optimization (PGO) differs from bundle adjustment primarily in their optimization objectives: PGO concentrates on refining node poses, often leading to enhanced efficiency. The pose graph problem in SLAM has been extensively studied and applied in various studies [29], [10], [30], [31]. [32] employs a covisibility-based optimization strategy for a constant time operation, while [33] utilizes a Minimum Spanning Tree (MST) to prune covisibility graphs, thereby creating a more simplified but precise essential graph for global optimization, balancing both efficiency and accuracy. Moreover, [23] discusses the use of Tree-based Network Optimizer (TORO) for addressing multi-map errors in pose graph optimization. Several studies [34], [35], [11], [12], [13] have highlighted the challenge of managing increasing node counts in pose graphs due to discrete-time sampling, particularly in life-long SLAM systems that operate continuously over extended periods.

III. Method

A. System Overview

In this work, we develop a vision-based SLAM system in which each submap is constructed using consecutive frames from a single video sequence. Building upon this framework, we have implemented a multi-session capability to enhance global consistency in mapping when processing multiple video sequences. The framework is illustrated in Fig. 2

Frontend: We adopt a state-of-the-art monocular visual odometry as the frontend of our system [36]. It employs a sliding window-based frontend optimization scheme for pose optimization and map points maintenance. Considering the set of keyframes in the sliding window, $\mathcal{K} = \{K_1, \cdots, K_n\}$, with corresponding camera-to-world pose set $\mathcal{T} = \{\xi_1, \cdots, \xi_n, |, \xi_i \in \mathfrak{se}(3)\}$, we represent the 3D point $\mathbf{X}_j^k \in \mathbb{R}^3$ corresponding to the j-th 2D observation $\mathbf{x}_k^j \in \mathbb{R}^2$ in the k-th keyframe using its inverse depth, *i.e.*, $\mathbf{X}_j^k = \mathbf{T}_k \cdot \Pi^{-1}\left(\bar{\mathbf{x}}_j^k \cdot d_j^k\right)$, where $(\bar{\cdot})$ denotes the homogeneous coordinates, and d_j^k represents the inverse depth. We define

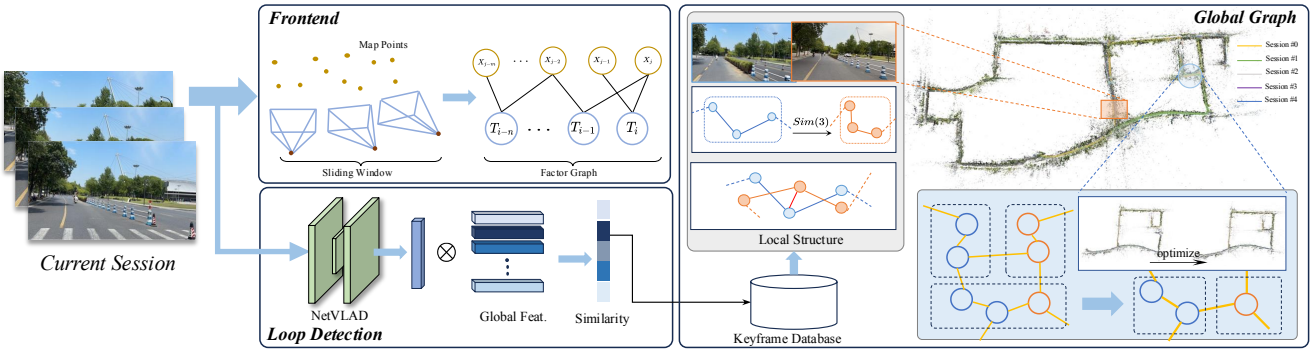

Fig. 2. Framework: The current video sequence is input into the system, where it undergoes local tracking in the frontend. Following this, loop closure detection based on NetVLAD [18] is performed, and keyframes are localized and optimized within the backend global graph. Subsequently, scale correction and global optimization are carried out within the global graph.

the set of edges in the sliding window factor graph as $\mathcal{E}$, representing corresponding keypoints between the j_1-th keypoint in keyframe k_1 and the j_2-th keypoint in keyframe k_2. The sliding window states can be solved by minimizing the reprojection error:

$$\arg\min_{\mathcal{T},\mathcal{D}} \sum_{(k_1,j_1,k_2,j_2)\in\mathcal{E}} \left\| \Pi\left(\xi_{k_2}^{-1}\cdot\xi_{k_1}\cdot\Pi^{-1}\left(\bar{\mathbf{X}}_{j_1}^{k_1}\right)-\mathbf{x}_{j_2}^{k_2}\right)\right\|^2 \tag{1}$$

In this work, we specifically focus on the components of the frontend odometry that interact with the backend graph, particularly the keyframes selected by the frontend, which serve as optimization targets in the backend for loop closure detection and global optimization.

Loop Detection: A relocalization system for detecting single-session and multi-session loop closures. A common approach for visual loop detection is vocabulary-based methods [37], as employed in works such as [1], [4], [28]. While this method is lightweight, its retrieval performance significantly degrades as the size of the image database increases. To address this limitation, we adopt a more robust and accurate learning-based approach NetVLAD [18] for efficient backend image retrieval. NetVLAD extracts a global descriptor for the context of the image, which is then compared against all features in the global graph database to compute similarities. This process identifies the most similar descriptors, thereby facilitating the discovery of corresponding loop closure image pairs (i, j).

Drift Estimation: Once a loop closure image pair (i, j) is identified, we perform a 7-DoF similarity transformation with scale correction based on two-view image matching. Our approach follows the methodology of DPV-SLAM [38], which avoids the dependency on repeatable keypoint detectors. To achieve accurate alignment, the method first estimates the local structure through structure-only bundle adjustment at both ends of the loop closure edges. This is followed by scale-aligned point cloud registration to refine the transformation.

Optimization: Pose graph optimization is employed in the backend to ensure consistency among keyframes and to mitigate scale drift resulting from the accumulation of visual

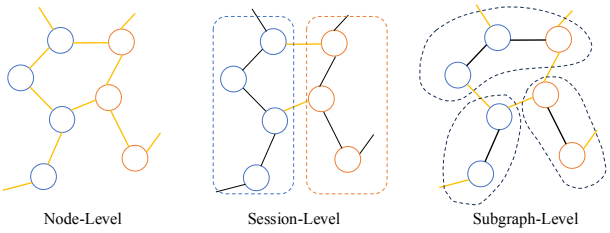

Fig. 3. Comparison of Different Graph Structures

Scale aware pose ?

estimation errors. We represent the scale-aware pose of the i-th keyframe in the global coordinate frame as $\mathbf{T}_i \in \mathrm{Sim}(3)$, and denote the relative pose between frames i and j as $\Delta\mathbf{T}_{ij} \in \mathrm{Sim}(3)$. Similar to [4], The error terms for both sequential and loop closure edges are expressed as follows:

$$e_i = \mathrm{Log}_{\mathrm{Sim}(3)}\left(\Delta\mathbf{T}_{(i,i+1)}\cdot\mathbf{T}_i^{-1}\cdot\mathbf{T}_{i+1}\right). \tag{2}$$

$$e_{jk} = \mathrm{Log}_{\mathrm{Sim}(3)}\left(\Delta\mathbf{T}_{(j,k)}\cdot\mathbf{T}_j^{-1}\cdot\mathbf{T}_k\right). \tag{3}$$

Thus, the final optimization objective can be formulated as:

$$\arg\min_{\mathbf{T}_0,\cdots,\mathbf{T}_n}\left(\sum_{(i,i+1)\in\mathcal{S}}\|e_i\|^2+\sum_{(j,k)\in\mathcal{L}}\|e_{jk}\|^2\right), \tag{4}$$

where n denotes the total number of states in the pose graph, $\mathcal{S}$ denotes the set of sequential edges, and $\mathcal{L}$ denotes the set of loop closure edges.

Global Graph: The results of each session are saved for subsequent map reuse, which is a crucial feature of life-long SLAM. The global graph contains the topological structure and node states of the map, including the poses of keyframes, the coordinates of keypoints, and their inverse depths. In addition, we store the image data on a local hard disk and load it when the corresponding images are retrieved.

B. Subgraph Structure

One of the major contributions of this work is the proposal of a subgraph optimization strategy for efficient and accurate backend optimization in large-scale scenarios. The design principle of the subgraph structure is to reduce the

number of edges in the graph optimization problem while ensuring the accuracy of the node poses. Unlike the approach proposed in [17], we emphasize the dynamic nature of the subgraph structure, where the relationships between nodes are subject to dynamic changes. This allows scale errors to be propagated across different subgraphs and hierarchical levels. As illustrated in Fig. 3, at the node-level and session-level, the number of edges are either excessive or insufficient. As discussed in [33], overly dense edges do not enhance system accuracy and instead lead to significant performance degradation. Conversely, session-level edge constraints play a crucial role in correcting scale errors, which is similar to loop closure edges. However, achieving this goal requires global scale propagation within each session.

We define a complete graph structure as $\mathcal{G} = \{\mathcal{V}_1, \cdots, \mathcal{V}_n, \mathcal{E}_1, \ldots, \mathcal{E}_m\}$ with n vertices and m edges. After introducing subgraphs, the entire backend map can be partitioned into $\{\mathcal{G}_1^{\mathrm{s}}, \ldots, \mathcal{G}_k^{\mathrm{s}}\}$. There exist several edges between these subgraphs, denoted as $\{\mathcal{E}_1^{\mathrm{s}}, \ldots, \mathcal{E}_l^{\mathrm{s}}\}$, here the superscript $(\cdot)^{\mathrm{s}}$ denotes the subgraph. The overall structure is shown at the subgraph-level in Fig. 3. We now need to consider how to partition $\mathcal{G}$. The simplest approach is to partition the nodes based on spatial proximity, with each node assigned to any subgraph. A potential issue is that the drift in scale cannot be guaranteed to be uniformly distributed among the graph nodes, which may result in the subgraphs not having globally consistent scales. Therefore, we adopt a *dynamic* partition scheme, where each node is not assigned to an individual subgraph but is instead shared across multiple subgraphs. This approach allows the fusion of biased error corrections from multiple subgraphs, thereby achieving globally consistent corrections. We define the subgraph size with a maximum of L. We first determine reference nodes from the newly received graph nodes in the frontend at fixed intervals and then the sub-nodes are assigned to subgraphs according to their distances. We use a distance threshold also L, to determine whether a sub-node is assigned to the subgraph $\mathcal{G}_i^{\mathrm{s}}$ represented by a given reference node.

We then use the reference node to represent the subgraph $\mathcal{G}_i^{\mathrm{s}}$, denoted as $\mathcal{V}_i^{\mathrm{s}}$, with the corresponding pose with scale denoted as $\mathbf{T}_i^{\mathrm{s}} \in \mathrm{SE}(3)$. This can be formulated as:

$$\arg \min_{\mathbf{T}_0^{\mathrm{s}}, \cdots, \mathbf{T}_k^{\mathrm{s}}} \left(\sum_{(i,j) \in \{\mathcal{E}_1^{\mathrm{s}}, \ldots, \mathcal{E}_l^{\mathrm{s}}\}} \|e_{ij}\|^2 \right), \tag{5}$$

where e_{ij} can be obtained using Eq. (3), as the constraints between subgraphs are similar to loop edges. After obtaining the scale correction of the reference node, we propagate the scale to other nodes within the subgraph. We perform scale adjustment on the relative pose $\mathbf{dT}_{i \to r} \in \mathrm{SE}(3)$ from the sub-node $\mathcal{V}_i$ to the reference node $\mathcal{V}_r$:

$$\mathbf{dT}_{i \to r}^{\text{scaled}} = \begin{bmatrix} \mathbf{dR}_{i \to r} & s \cdot \mathbf{dt}_{i \to r} \\ 0 & 1 \end{bmatrix} \in \mathrm{Sim}(3), \tag{6}$$

where s is the scale estimated from the anchor node within the subgraph. However, since a given node may have more than one reference nodes from different subgraphs, we use a weighted scale averaging strategy for the multiple scale estimates $\{s_1, \ldots, s_k\}$:

$$s = \sum_{i=1}^{k} w_i \cdot s_i, \tag{7}$$

where w_i represents the normalized weight, which we use inverse distance from the node to the reference to weight it.

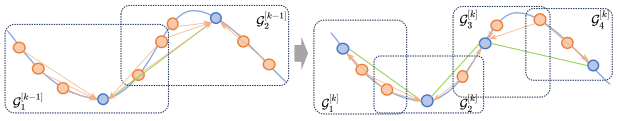

Fig. 4. Hierarichical Subgraph-based Optimization. We constrain the states to be optimized within subgraphs (green) in the graph structure. The node states in each subgraph are updated based on the relative transformations with respect to the reference nodes (orange).

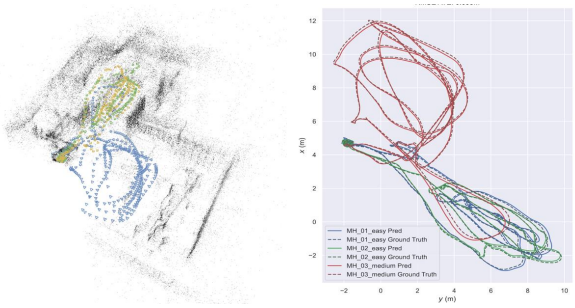

Fig. 5. Multi-Session Keyframes and Trajectory (MH01-MH03). We use different color to denote the keyframes from each sessions with the aligned point cloud with the same scale.

C. Hierarichical Optimization

Dynamic subgraphs are not only defined from the perspective of spatial partitioning but can also be optimized from the subgraph scale, *i.e.*, from a hierarchical perspective. Hierarchical optimization is a common optimization strategy [39], [17], where this coarse-to-fine approach can effectively prevent the optimization problem from falling into local minima. As shown in Fig. 4, at lower levels (with smaller subgraph structures), we re-partition the subgraphs to avoid nodes at the subgraph boundaries from falling into local optima due to a lack of consistent constraints. We achieve this goal simply by reducing the size of the subgraph L. To simplify the problem, we halve the subgraph size at each level, *i.e.*, $L := L/2$. If the coarsest level of the subgraph is denoted as $\mathcal{G}^{[k]}$, then as the hierarchy k is refined, we obtain $\mathcal{G}^{[k+1]}$. It is important to note that in the full hierarchical optimization, $\Delta\mathbf{T}_{ij}$ in Eq. (2) and Eq. (3) remains unchanged throughout. This is still an iterative optimization process, starting from the largest subgraph structure. The optimization proceeds to finer subgraph levels based on the convergence criteria.

IV. EXPERIMENT

We use the state-of-the-art DPVO [36] as a baseline frontend and have developed the multi-session SLAM framework based on it. Contrast to the very recent work DPV-

SLAM [38] which includes a loop closure module, we employ different loop detection and backend optimization strategies. More importantly, and our framework supports multi-session functionality for life-long SLAM implementation. All of our experiments were conducted in the same environment, using an AMD Ryzen 9 7950X 16-core CPU and an RTX 4090 24GB GPU.

Datasets: We evaluated our method on the EuRoC [40] dataset and a large-scale Campus scene that we collected ourselves. The EuRoC dataset provides camera poses in a single global coordinate system, which allows us to assess the accuracy of the multi-session system. For evaluating the efficiency of backend map optimization in large-scale scenarios, we captured 6 video sequences of a large area, totaling $0.72km^2$, with a total trajectory length over $12km$. On this dataset, we assessed the efficiency of backend optimization and conducted a qualitative comparative analysis of the map reconstruction results.

Metric: For trajectory accuracy, the system aligns with the global map at scale during each session, but monocular systems still exhibit scale discrepancies. In contrast, the comparison method [28], [4] achieves true scale recovery through IMU integration. Therefore, for evaluation, we employed the approach from [26], which first performs a 7-DoF global alignment of the final estimated trajectory, followed by evaluating the absolute trajectory error (ATE) in terms of the Root Mean Square Error (RMSE) against the ground truth. In the comparison of system performance in large-scale scenarios, we focus on pose optimization performance, including the number of pose nodes and the convergence of the optimization process.

A. EuRoC Dataset

The EuRoC [40] dataset consists of 11 sequences from three different scenarios. The MH01-MH05 sequences were collected in a Machine Hall, while Vicon1 and Vicon2 each contain three sequences. These different sequences within the three scenarios share the global coordinate system. The ground truth trajectories for each sequence were obtained using a laser tracker and the VICON motion capture system, with an image sequence frame rate of 20 FPS. We performed accuracy evaluations for both independent sessions and multi-sessions to demonstrate the effectiveness of the proposed method.

Although our approach emphasizes backend optimization performance and multi-session-based scalability in large-scale scenarios, we still conducted a brief evaluation of the single-session accuracy on the MH scenarios of the EuRoC dataset according to the experimental configurations in [38], [36], this was done to assess the basic performance of the proposed backend. As shown in Tab. I, our approach yields smaller trajectory errors in some scenarios compared to DPV-SLAM [38], with overall performance being comparable. We also show multiple trajectories in a global coordinate system in Fig. 5.

Not so robust, just comparable

In Tab. II, we report multi-session results based on the protocol from [28], [26]. We evaluate with four different

TABLE I

SIGNLE-SESSION EVALUATION ON EUROC DATASETS.

Sequence	MH01	MH02	MH03	MH04	MH05
ORB-SLAM2	0.071	0.067	0.071	0.082	-
DROID-SLAM	0.013	**0.014**	0.022	0.043	0.043
DPVO	0.087	0.055	0.158	0.137	0.114
DPV-SLAM	**0.013**	0.016	0.022	0.043	**0.041**
DPV-SLAM++	0.015	0.016	0.021	0.041	0.052
Ours	0.014	0.016	**0.019**	**0.040**	0.043

collected data lacks more info

TABLE II

MULTI-SESSION EVALUATION ON EUROC USING RMSE ATE.

Scene name	MH01-03	MH01-05	V101-103	V201-203
# Trajectories	3	5	3	3
VINS	-	0.210	-	-
ORB-SLAM3(VI)	0.037	0.065	0.040	0.048
Multi-DiffPose	0.045	0.059	0.081	0.072
CCM-SLAM	0.077	-	-	-
ORB-SLAM3(V)	**0.030**	**0.058**	**0.058**	0.284
MS-DPV-SLAM	0.129	0.139	0.122	0.112
Ours	0.082	0.109	0.092	**0.065**

configurations, where sequences were input into the system to obtain trajectories in a global coordinate system. Baseline results are from [26]. Among them, Multi-DiffPose [26] and CCM-SLAM [41] are monocular SLAM methods, while ORB-SLAM3 [28] and VINS-Mono [4] are monocular inertial SLAM methods. Due to differences in evaluation metrics, we only list the visual-inertial methods for reference. Our approach demonstrates superior performance in global mapping and trajectory consistency compared to the baseline MS-DPV-SLAM, which we extended the multi-session capability of DPV-SLAM [38]. We believe this is due to the enhanced loop detection capability and the proposed subgraph optimization strategy. While Multi-DiffPose performs comparably, it employs a per-frame global registration scheme with scale, which incurs higher computational costs (differentiable pose optimization for two-view). The accuracy advantage of ORB-SLAM3 stems from complex co-visibility maintenance and global registration, which comes at the cost of complex system maintenance. The work closest to our problem is VINS-Mono [4], which, despite having IMU scale constraints, shows similar performance to our baseline due to its chained constraints.

Fishy

B. Large Scale Campus Dataset

In Tab. III, we report the performance on a large-scale dataset composed of multiple video sequences we captured. This dataset was captured using iPhone 15 video recording, where the original resolution is 1920×960 with a frame rate of 30 fps, and the total duration of the complete video sequence is over 1 hour. The challenges on this dataset include the efficiency of the backend optimization in multi-session system. We attempted to use COLMAP [42] to reconstruct these sequences, but found that neither the exhaustive search method nor the NetVLAD-based [18], [43] image matching

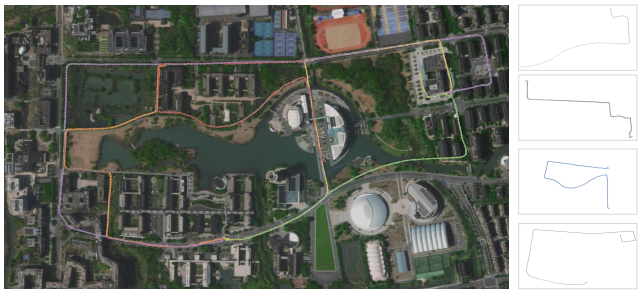

Fig. 6. Multi-Session Trajectories in a large-scale map. We aligned the multi-session trajectories with the satellite map and plotted them together. In addition to correcting scale errors, this approach also ensures global structural consistency.

TABLE III
TIME COST(S) OF POSE GRAPH OPTIMIZATION.

Graph Node	323	870	1062	1508	3143	5234
MS-DPV-SLAM	0.534	1.043	1.225	1.690	2.895	5.535
Ours	0.256	0.423	0.586	0.782	1.301	2.345

and registration approach succeeded even after up to 30 hours of running time. In contrast, our method enables real-time localization and mapping for large-scale scenes. The final output of multi-session trajectories are aligned with Google Maps and shown in Fig. 6. We also randomly visualize 4 trajectories with drift in single session mode.

We compared the impact of pose graph scale on optimization efficiency. We use the number of nodes involved in the optimization to reflect the size of the optimization problem, which is determined by the topological positions of the detected loops in the sequence data. These data are intermediate results from running the collected sequences. For each set of samples, we performed 10 runs to obtain an average time cost. Our comparison method is MS-DPV-SLAM, as it is an extension of the multi-session capabilities from DPV-SLAM, with a vanilla pose graph optimization strategy. The results are shown in Tab. III. We did not consider early stopping after convergence but ran a full 30 iterations. As can be seen, the time consumption of our approach is significantly lower than that of traditional full pose graph optimization, with the overall time cost being around 50% of that for the pose graph, demonstrating the performance advantage of subgraph-based optimization.

V. APPLICATION

Incremental Mapping: As shown in Fig. 1, the proposed multi-session approach emphasizes both accuracy and real-time performance, which is crucial for applications targeting real-time large-scale mapping comparing to [42], which failed to reconstruct the map from these sequences. On one hand, it ensures the scalability of the map through incremental mapping, which is more robust and efficient than single-sequence reconstruction. On the other hand, the system's final pose estimation graph topology can be used as input for SfM to achieve more detailed large-scale scene reconstruction, thereby improving the overall efficiency of SfM pipeline.

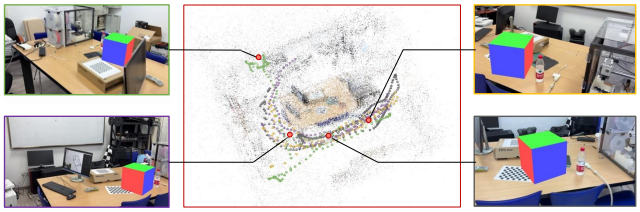

Fig. 7. Long-Term Augmented Reality

Long-Term Augmented Reality: A key application of multi-session features in AR is the ability to reuse scene maps, enabling camera localization within a global coordinate system for persistent virtual object placement and interaction. By leveraging map optimization results from previous sessions, the current system state is enhanced including both the geometric structure of the map and the camera pose. Consequently, our proposed approach facilitates multi-session-based interactions in AR by reusing the map, as illustrated in Fig. 7.

VI. CONCLUSION

In this paper, we propose a multi-session visual SLAM system with subgraph optimization for large-scale scenarios, aiming to achieve efficient and accurate lifelong SLAM. We present an efficient back-end optimization strategy based on dynamic subgraph optimization and a hierarchical refinement scheme that ensures precision, enabling accurate correction of scale drift in large-scale environments. The incremental reconstruction accuracy of multi-session SLAM provides an effective solution for life-long SLAM. However, its limitation lies in instability when dealing with visual appearance changes. Our future goal is to develop a more robust multi-session system by incorporating multi-sensor information.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[3] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[4] 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.

[5] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3126–3131.

[6] J. Li, X. Pan, G. Huang, Z. Zhang, N. Wang, H. Bao, and G. Zhang, "Rd-vio: Robust visual-inertial odometry for mobile augmented reality in dynamic environments," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[7] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "NICE-SLAM: Neural implicit scalable encoding for SLAM," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 786–12 796.

[8] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 499–507.

[9] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 039–18 048.

[10] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 54–61.

[11] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term SLAM," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 50–72, 2016.

[12] G. Huang, M. Kaess, and J. J. Leonard, "Consistent sparsification for graph optimization," in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 150–157.

[13] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[14] G. Dubbelman and B. Browning, "COP-SLAM: Closed-form online pose-chain optimization for visual SLAM," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1194–1213, 2015.

[15] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. E. Thorpe, "Subgraph-preconditioned conjugate gradients for large scale SLAM," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2566–2571.

[16] Y. Tazaki, "A spanning tree-based multi-resolution approach for pose-graph optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 033–10 040, 2022.

[17] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2d and 3d mapping," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 273–278.

[18] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[19] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 354–366, 2012.

[20] P. Schmuck, T. Ziegler, M. Karrer, J. Perraudin, and M. Chli, "CoVINS: Visual-inertial SLAM for centralized collaboration," in *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2021, pp. 171–176.

[21] M. Karrer, P. Schmuck, and M. Chli, "CVI-SLAM—collaborative visual-inertial SLAM," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2762–2769, 2018.

[22] X. Pan, G. Huang, Z. Zhang, J. Li, H. Bao, and G. Zhang, "Robust collaborative visual-inertial SLAM for mobile augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[23] M. Zhao, X. Guo, L. Song, B. Qin, X. Shi, G. H. Lee, and G. Sun, "A general framework for lifelong localization and mapping in changing environment," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3305–3312.

[24] Y. Wang, Y. Ng, I. Sa, A. Parra, C. Rodriguez-Opazo, T. Lin, and H. Li, "Mavis: Multi-camera augmented visual-inertial SLAM using se 2 (3) based exact imu pre-integration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1694–1700.

[25] S. Hochdorfer and C. Schlegel, "Towards a robust visual SLAM approach: Addressing the challenge of life-long operation," in *2009 International Conference on Advanced Robotics*. IEEE, 2009, pp. 1–6.

[26] L. Lipson and J. Deng, "Multi-session SLAM with differentiable wide-baseline pose optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 626–19 635.

[27] T. Qin, P. Li, and S. Shen, "Relocalization, global optimization and map merging for monocular visual-inertial SLAM," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1197–1204.

[28] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[29] R. Elvira, J. D. Tardós, and J. M. Montiel, "ORBSLAM-Altas: a robust and accurate multi-map system," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6253–6259.

[30] C. Stachniss and H. Kretzschmar, "Pose graph compression for laser-based SLAM," in *Robotics Research: The 15th International Symposium ISRR*. Springer, 2017, pp. 271–287.

[31] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[32] H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual SLAM," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2352–2359.

[33] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[34] J. Wang and E. Olson, "Robust pose graph optimization using stochastic gradient descent," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 4284–4289.

[35] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. J. Leonard, "Real-time 6-dof multi-session visual SLAM over large-scale environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1144–1158, 2013.

[36] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[37] 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.

[38] L. Lipson, Z. Teed, and J. Deng, "Deep patch visual SLAM," *arXiv preprint arXiv:2408.01654*, 2024.

[39] G. Zhang, H. Liu, Z. Dong, J. Jia, T.-T. Wong, and H. Bao, "Efficient non-consecutive feature tracking for robust structure-from-motion," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5957–5970, 2016.

[40] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[41] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[42] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[43] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 716–12 725.