2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Two-step Nonlinear Factor Sparsification for Scalable Long-term SLAM Backend

Binqian Jiang and Shaojie Shen

Abstract— **This paper proposes a new nonlinear factor sparsification paradigm for general feature-based long-term SLAM backend. Given a pose sparsification policy, we aim to scale the SLAM problem with space explored instead of time in a principled way so that the number of time-indexed poses can be limited. At the same time, their influence and the long-lived landmarks are appropriately maintained. To do this, we propose a new two-step sparsification pipeline. Given a pose node to remove, the first step is performed in the Markov blankets of affected landmarks. It transforms pose-landmark constraints into pose-pose constraints while preserving observability and minimizing information loss in the blanket. Moreover, since landmarks are conditionally independent, we can do this in parallel, disconnecting a pose from all the landmarks. The second step marginalizes the pose of interest with pure pose-wise constraints without affecting landmarks. Our method decouples the management of landmarks from pose-only measurements, making it general for any feature-based SLAM. We also give a practical example of how our backend works by concatenating it to a monocular VIO frontend. In simulation and real-world dataset, our sparsified backend is accurate and efficient. We open-source our backend, along with the VIO+Backend example, to contribute to the community's betterment.**

I. Introduction

Towards the goal of long-term autonomy, an essential task in simultaneous localization and mapping (SLAM) is to achieve scalability in state and environment representation [1], [2]. An odometry system provides efficient and accurate state estimation in the robot's immediate surroundings by relying on locally available and recent data [3], [4]. However, odometry inevitably accumulates drift over time, and modern SLAM frameworks typically have a loop detection module and a separate backend to maintain global consistency by pose graph optimization (PGO) [5], [6].

When a robot operates in a particular area over a prolonged duration, e.g., a neighborhood community or a campus, the traditional solution will suffer from unbounded growth in computational cost even with a lightweight backend pose graph. Marginalization can be utilized to constrain the node size of a graph, while sparsification and reduction techniques [7]–[9] are further proposed to reduce the dense fill-in after marginalization. However, due to the cost of approximating the distribution of a local Markov blanket, sparsification is typically done in pose graphs.

Pose graphs are not obviously translated from low-level and full-scale factor graphs of SLAM. Modeling loop closures as relative pose constraints can lead to information reuse, violating the widely adopted assumption of factor independence. This is because loop closure is a situation where poses observe the same landmarks, and all loop edges connected to a given pose are calculated with the exact copy of measurement by that pose, which means these loop constraints are correlated. Batch offline optimization [10] circumvents this problem, but it is hard to avoid it when manipulating pose graphs online. Also, maps recovered from sparse pose graphs with loop edges can still contain aliasing [11], [12], since relative pose factors cannot capture the high nonlinearity of constraints imposed by exteroceptive sensors. Alternatively, large-scale bundle adjustment (BA) is more accurate, but its cost renders it impractical for online use. Therefore, dynamically maintaining a combination of poses and landmarks seems better suited for the purpose of robust and scalable long-term localization and mapping.

To overcome the challenges above, we propose a two-step sparsification pipeline that limits the number of poses while their influence and the long-lived landmarks are properly preserved in a backend graph. Our key idea is to separate the processing of landmark measurements from pure pose-wise constraints. Given a pose node to remove, we first transform pose-landmark constraints into pose-pose constraints in the Markov blankets of the landmarks observed by that pose. We explicitly construct measurement Jacobians that preserve observability before solving for new measurement information via KL divergence minimization. Moreover, since landmarks are conditionally independent, the Markov blankets can be processed in parallel, mitigating the complexity of previous sparsification methods. After the first step, the pose of interest is disconnected from all landmarks. Then, we can marginalize that pose with pure pose-wise constraints (e.g., relative pose, roll-pitch) in a much smaller Markov blanket. We highlight the key capabilities of our method:

- Nonlinear factor and landmark maintenance.
- Complexity scaling with space explored.
- Natural and implicit loop closure constraint.
- Smooth transition from full pose-landmark factor graph to sparse pose graph, given marginalization policies.
- Applicable to most kinds of feature based SLAM.

We also implemented a monocular VIO+Backend system, and our sparsified backend is shown to be accurate and efficient in simulation and real-world dataset.

Before delving further, it is important to underscore that the long-term SLAM problem is multifaceted, while the focus of this paper is the sparsification of pose-landmark

This work was supported by HKUST Postgraduate Studentship and HKUST-DJI Joint Innovation Laboratory. The authors are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. E-mail: bjiangah@connect.ust.hk, eeshaojie@ust.hk

factor graph only. We will opensource our implementation[1]. The autonomy for frontend design, loop detection, and marginalization policy is vested in the hands of users.

II. Related Work

In light of the inherent complexity of long-term SLAM, we choose to narrow our research focus to addressing scalability concerns, which is also reflected in our literature review.

A. Information Sparsification

Marginalization is a well-known technique that limits a graph's node size and preserves its information at the linearization point. Marginalizing a node causes dense fill-in in the nodes' Markov blanket. To solve this problem, GLC [7] is proposed to model the dense fill-in as either a sparse Chou-Liu tree or a dense n-ary factor. NFR [8], [13] sparsifies the graph into new nonlinear factors and solves for the measurement information by minimizing the KL divergence between the original distribution and the designed new factors. The KLD minimization can also be solved using factor descent [14].

The idea of NFR is applied to real-world systems. Basalt [10] extracts pose-pose and roll-pitch factors from the frontend for backend optimization. The dense prior of a sliding window is approximated using sparsification in VIO [15] and VO [16]. NFR is also used in underwater vehicles [17] to alleviate communication burden. All four applications of NFR mentioned above use the special case of a closed-form solution, which is inflexible since the dimension of nonlinear measurements is the lowest and cannot be increased.

The sparse topology is optimized in [18], [19] by selecting significant off-diagonal blocks of the target information matrix that approximates the original. However, the threshold is highly problem-dependent and may result in an undesirable connection that is hard to model using nonlinear functions. Several predefined topology design policies and their KLD performance are discussed in the pose graph case [20]. How to design nonlinear factors for novel node connections, e.g., landmark-landmark constraints, is still being determined.

On top of sparsification, it is possible to enforce conservative constraint at the same time [18], [21], [22] to avoid overestimating accuracy in state estimation.

B. Measurement Selection

Another class of methods improves the scalability of SLAM by selecting a subset of raw measurements for optimization, which is free of approximation error by design [23]. Different criteria are used to guide the selection, for example, D-optimality and t-optimality of the graph [23]. [24] selects proper edges by maximizing algebraic connectivity, a metric statistically related to lower error. A geometric criterion for selecting both edges and vertices of a dense graph is proposed in [25], which focuses on the even distribution of nodes in space. [26] proposes an information-theoretic algorithm to select landmarks and poses, which evaluates the information loss before and after the deletion of the selected node. Greedy selection proves close to the optimal choice of measurement subset [27]. Selection-based methods cannot enjoy the benefit of observations over a long horizon, which stably summarizes all historical information.

[1]https://github.com/lewisjiang/ops

C. Long-term SLAM Examples

Here, we primarily focus on the techniques used in the backend that contribute to the scalability of the whole system.

In [28], a sound probabilistic approach combining a Rao-Blackwellized particle filter with a hidden Markov model is used for lifelong localization. Temporal scalability is achieved with a pose node reduction policy [9], and an impressive long-term visual SLAM experiment is conducted to showcase the efficacy. A submap-based graph sparsification is utilized to limit computation and memory complexity over time [29]. SLAM++ [30] incrementally maintains a backend graph and uses proximity and information gain as the criteria to accept new nodes into the system. In these methods, pose graph is a popular choice for managing sparsity, but it is unclear how to incorporate landmarks while still keeping the complexity manageable. More recently, ObVi-SLAM [31] relies on a sparse objects map for robust and long-term localization. Currently, their map topology is too simple, and maintenance (e.g., sparsification) is done offline.

Map-centric SLAM can directly scale the problem with space explored instead of time. Elastic LiDAR fusion [32], [33] models the map as nodes interconnected with neighbors, and continuously localizing and updating this map will not increase overall complexity. However, how to maintain graph edges in case of scene change needs to be clearly addressed.

Some learning-based methods suitable for long-term SLAM, e.g., neural implicit representation, are worth mentioning. One notable work called NeRF-W [34] models lighting and transient objects alongside the static environment, which is appealing for long-term map representation.

III. Preliminaries

A. Nonlinear Factor Recovery

Nonlinear Factor Recovery (NFR) [8], [13] uses a set of new and possibly sparse nonlinear factors to approximate the original distribution stored in a Markov blanket by solving a KL divergence minimization problem. While its initial purpose is to use sparse factors to approximate a dense linear prior distribution, we use it to re-distribute information among manually allocated measurements within a Markov blanket, another type of "sparsification". In this problem, the prior multivariate distribution $p(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ is approximated by another distribution $p_a(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Lambda}_a^{-1})$, and their similarity is measured by KL divergence [8]:

$$D_{KL}(p(\mathbf{x})|p_a(\mathbf{x})) = \\ \frac{1}{2}\left(\langle \boldsymbol{\Lambda}_a, \boldsymbol{\Sigma}_0 \rangle - \log\det(\boldsymbol{\Lambda}_a \boldsymbol{\Sigma}_0) + \|\boldsymbol{\mu}_a - \boldsymbol{\mu}_0\|^2_{\boldsymbol{\Lambda}_a^{-1}} - d\right) \tag{1}$$

where $\langle \cdot, \cdot \rangle$ is matrix inner product and d is a constant. Obviously, to minimize $D_{KL}(p(\mathbf{x})|p_a(\mathbf{x}))$, the mean of old and new distributions should be equal, and $\boldsymbol{\Lambda}_a$ should be positive-definite. More specifically, if we know Jacobians

of the new set of independent nonlinear factors, we can calculate the information of each factor by solving:

$$\begin{aligned} &\min \langle \mathbf{J}_a^\top \mathbf{\Omega}_a \mathbf{J}_a, \mathbf{\Sigma}_0 \rangle - \log\det(\mathbf{J}_a^\top \mathbf{\Omega}_a \mathbf{J}_a) \\ &s.t. \quad \mathbf{\Omega}_a = \mathrm{diag}(\cdots, \mathbf{\Omega}_i, \cdots),\ \mathbf{\Omega}_i \succ \mathbf{0} \\ &\quad\ \ \mathbf{J}_a = [\cdots, \mathbf{J}_i^\top, \cdots]^\top \end{aligned} \tag{2}$$

where $\mathbf{J}_i$ and $\mathbf{\Omega}_i$ have the same row dimension, $\mathbf{J}_a$ is known and we want to solve for its corresponding information $\mathbf{\Omega}_a$. The problem in (2) can be solved analytically if $\mathbf{J}_a$ is invertible, and generally it can be solve with Interior-point (IP) method. Please see [8], [14], [20] for detailed discussion.

IV. Methodology

A. Overview

Inspired by many sparsification theories and applications [7], [8], [10], [13], [15]–[17], [19], we develop a backend framework that uses a graph of both poses and landmarks as the map representation for long-term SLAM. New poses connect with the past via long-lived landmarks, so we only keep necessary types of factors in the graph, namely relative pose, global roll-pitch (as prior, or made observable in aided inertial navigation systems), global position + yaw (as prior, or made observable with GNSS), and the measurement functions of exteroceptive sensors (e.g., cameras, LiDARs) that connect poses with landmarks. Also, we assume the marginalization decision is given outside our framework, and the sparse pattern is also user-defined, as long as it does not cause wrong degrees of freedom for the constraints.

B. Landmark-centered Markov Blanket

As mentioned in Sec.I, using pose-pose constraints in case of loop closure causes double counting, and a pure pose graph can have low accuracy. As a result, we decide to model landmarks in the graph to act as a medium between current and historical poses in case of loop closure or relocalization. We start by considering landmark information. Throughout this section (Sec.IV), we use the notations in Fig.1 to illustrate our idea and derive equations.

We first consider Markov blankets centered at landmarks. The state of a landmark solely depends on poses in the blanket (see Fig.1). Therefore, it is reasonable to manipulate the local graph when seeking sparsity [8]. The sparse topology can be manually designed, and we aim to transform pose-landmark constraints into pose-pose constraints in a Markov blanket. Let us take landmark $\mathbf{f}_1$ in Fig.1a for example: we can remove measurement $\mathbf{z}_{31}$ and adjust the information of $\mathbf{z}_{11}$, $\mathbf{z}_{12}$, $\mathbf{u}_{12}$, $\mathbf{u}_{23}$ using NFR [8] to achieve sparsity, and the nonlinear functions for new factors are readily available–just the same as original $\mathbf{z}$ and $\mathbf{u}$'s.

1) Observability-aware nonlinear pose-pose constraint: The above mentioned method need to adjust $\mathbf{u}_{12}$, $\mathbf{u}_{23}$ for each landmark, which disables parallel processing of Markov blankets. One way to enable it is to keep $\mathbf{u}_{12}$, $\mathbf{u}_{23}$ unaltered until all the landmark blankets finish, at the cost of figuring out which nonlinear functions the additional factors $\mathbf{v}_{13}$ and $\mathbf{v}_{23}$ are from. This is not trivial, and readers can take Fig.1b as a bearing-only measurement case, and try to imagine what nonlinear constraints should $\mathbf{v}_{13}$ and $\mathbf{v}_{23}$ satisfy. The good news is that we can first find the Jacobian of $\mathbf{v}_{13}$ and $\mathbf{v}_{23}$ by construction. Let us still use Fig.1 as an example. In the original Markov blanket, we have:

$$\mathbf{J}_{mb} = \begin{bmatrix} \mathbf{J}_{\mathbf{z}_{11}}(\mathbf{x}_1) & & & \mathbf{J}_{\mathbf{z}_{11}}(\mathbf{f}_1) \\ & \mathbf{J}_{\mathbf{z}_{21}}(\mathbf{x}_2) & & \mathbf{J}_{\mathbf{z}_{21}}(\mathbf{f}_1) \\ & & \mathbf{J}_{\mathbf{z}_{31}}(\mathbf{x}_3) & \mathbf{J}_{\mathbf{z}_{31}}(\mathbf{f}_1) \end{bmatrix} \tag{3}$$

where $\mathbf{J}_{\mathbf{z}_{11}}(\mathbf{x}_1) = \frac{\partial \mathbf{z}_{11}}{\partial \mathbf{x}_1}$ and the others are likewise. After disconnecting $\mathbf{z}_{31}$ and adding $\mathbf{v}_{13}$ and $\mathbf{v}_{23}$, we have the new Jacobian:

$$\mathbf{J}'_{mb} = \begin{bmatrix} \mathbf{J}_{\mathbf{z}_{11}}(\mathbf{x}_1) & & & \mathbf{J}_{\mathbf{z}_{11}}(\mathbf{f}_1) \\ & \mathbf{J}_{\mathbf{z}_{21}}(\mathbf{x}_2) & & \mathbf{J}_{\mathbf{z}_{21}}(\mathbf{f}_1) \\ \mathbf{J}_{\mathbf{v}_{13}}(\mathbf{x}_1) & & \mathbf{J}_{\mathbf{v}_{13}}(\mathbf{x}_3) & \\ & \mathbf{J}_{\mathbf{v}_{23}}(\mathbf{x}_2) & \mathbf{J}_{\mathbf{v}_{23}}(\mathbf{x}_3) & \end{bmatrix} \tag{4}$$

Note that the Jacobian for new measurements $\mathbf{z}'_{11}$ and $\mathbf{z}'_{21}$ are the same as those for $\mathbf{z}_{11}$ and $\mathbf{z}_{21}$, since they have the same measurement model (nonlinear function) at the same linearization point. Next we show how to construct a solution to $\mathbf{J}_\mathbf{v}$'s, and a sufficient condition to get this solution. Take $\mathbf{J}_{\mathbf{v}_{13}}$ for example. Suppose the minimal parameterization of landmark $\mathbf{f}_1$ is n_f-dimensional, and the matrix

$$\mathbf{J}_{\mathbf{z}_{11},\mathbf{z}_{31}}(\mathbf{f}_1) = \begin{bmatrix} \mathbf{J}_{\mathbf{z}_{11}}(\mathbf{f}_1) \\ \mathbf{J}_{\mathbf{z}_{31}}(\mathbf{f}_1) \end{bmatrix} \tag{5}$$

has m_f ($> n_f$) rows and $\mathrm{rank}(\mathbf{J}_{\mathbf{z}_{11},\mathbf{z}_{31}}(\mathbf{f}_1)) = n_f$. Then we can calculate the left null space of $\mathbf{J}_{\mathbf{z}_{11},\mathbf{z}_{31}}(\mathbf{f}_1)$ denoted as $\mathbf{U}_{\mathbf{z}_{11},\mathbf{z}_{31}}(\mathbf{f}_1)$ with orthonormal columns that satisfies:

$$\mathbf{J}_{\mathbf{z}_{11},\mathbf{z}_{31}}(\mathbf{f}_1)^\top \mathbf{U}_{\mathbf{z}_{11},\mathbf{z}_{31}}(\mathbf{f}_1) = \mathbf{0} \tag{6}$$

Now we stack the first and third block rows of (3) and project to the left null space of landmark Jacobians, which yields

$$[\mathbf{J}_{\mathbf{v}_{13}}(\mathbf{x}_1), \mathbf{J}_{\mathbf{v}_{13}}(\mathbf{x}_3)] = \mathbf{U}^\top_{\mathbf{z}_{11},\mathbf{z}_{31}}(\mathbf{f}_1) \begin{bmatrix} \mathbf{J}_{\mathbf{z}_{11}}(\mathbf{x}_1) & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\mathbf{z}_{31}}(\mathbf{x}_3) \end{bmatrix} \tag{7}$$

This gives a solution to the Jacobians of nonlinear function $\mathbf{v}_{13}$, and the sufficient condition is in our assumption: $m_f > n_f$. This condition is easily met, for example in monocular camera, 2 observations give 4 dimensions of measurements, while the DoF of a feature point is 3. The new Jacobian in (4) spans the same row space as the original measurements and ensures $\mathrm{rank}(\mathbf{J}_{mb}) = \mathrm{rank}(\mathbf{J}'_{mb})$, which means no additional or deficient degree of freedom exists in the new measurements, and the observability is preserved.

2) Information approximation: It turns out that with the designed Jacobians in (4), the measurement information in the Markov blanket can be *exactly* re-expressed without approximation.

Proposition 1: Given $\mathbf{J}_{mb}$ from (3) and $\mathbf{J}'_{mb}$ from (4) with new rows calculated as (7), and given a full rank information matrix $\mathbf{\Omega}_{mb}$, there exists a $\mathbf{\Omega}'_{mb}$ such that:

$$\mathbf{J}_{mb}^\top \mathbf{\Omega}_{mb} \mathbf{J}_{mb} = \mathbf{J}'^\top_{mb} \mathbf{\Omega}'_{mb} \mathbf{J}'_{mb} \tag{8}$$

Proof: Null space projection used in (7) is equivalent to linear row operations, so we can find a matrix $\mathbf{A}$ such that:

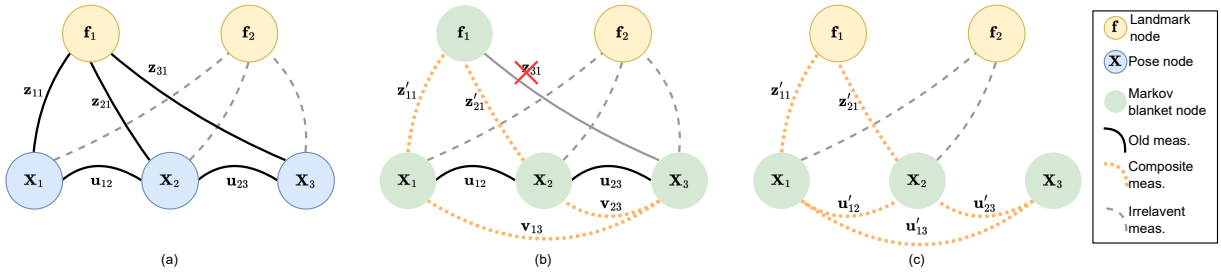

Fig. 1: Illustration of sparsification around pose $\mathbf{X}_3$. (a) The Markov blanket of landmark $\mathbf{f}_1$ observed by pose $\mathbf{X}_3$, and all the measurements $\mathbf{u}, \mathbf{z}$ in the blanket. (b). In the Markov blanket of $\mathbf{f}_1$, we transform pose-landmark constraint $\mathbf{z}_{31}$ into pose-pose constraints, and also recalculate the information of each new measurement (orange dotted curves) using sparsification techniques. (c) We do the same for all the landmarks observed by $\mathbf{X}_3$, and $\mathbf{X}_3$ is then disconnected from landmarks.

$\mathbf{J}'_{mb} = \mathbf{A}\mathbf{J}_{mb}$. Therefore a solution is $\mathbf{A}^\top \mathbf{\Omega}'_{mb}\mathbf{A} = \mathbf{\Omega}_{mb}$, which can be solved by methods like eigen decomposition. ■

Remark 1: Proposition 1 justifies the effectiveness of our Jacobian construction method in terms of information preservation. It is possible to use trivial Jacobians (e.g., one-hot rows) in (7) while keeping their rank unchanged. However, that cannot guarantee an $\mathbf{\Omega}'_{mb}$ for (8) to hold.

Note that $\mathbf{J}_{mb}^\top \mathbf{\Omega}_{mb}\mathbf{J}_{mb}$ is the information matrix of the measurements in the Markov blanket by definition. The exact solution of $\mathbf{\Omega}'_{mb}$ is generally not block diagonal, which means undesirable correlation exists among $\mathbf{v}_{13}, \mathbf{v}_{23}, \mathbf{z}'_{11}, \mathbf{z}'_{21}$. To minimize information loss, we formulate a sparsification problem as (2) to find the independent information matrices of measurements $\mathbf{v}_{13}, \mathbf{v}_{23}, \mathbf{z}'_{11}, \mathbf{z}'_{21}$, given their Jacobians and the original distributions.

Another concern in approximation is rank-deficient information matrices. Solving with minimum parameterization can avoid the problem [19], e.g., using relative pose in an anchored local pose graph. However, the states in the landmark's Markov blanket are generally extremely underconstrained, and it is not trivial to derive a set of new parameterizations for every problem we solve. For this reason, we adopt the projection-based method [8] to handle the information approximation in a lower dimensional space and reproject the results back to the original space where the solution is physically meaningful.

C. Pose-centered Markov Blanket

After transforming all the pose-landmark measurements into pose-pose constraints for $\mathbf{X}_3$ (see Fig.1c), we proceed to the second step of approximation: either do another sparsification over the Markov blanket of $\mathbf{X}_3$, or just marginalize it, affecting only a handful of poses. If we decide to keep $\mathbf{X}_3$ in the states, we can create relative pose constraints $\mathbf{u}'_{12}, \mathbf{u}'_{13}, \mathbf{u}'_{23}$, by merging the original relative pose constraints $(\mathbf{u}_{12}, \mathbf{u}_{23})$ and the new pose-pose constraints $(\mathbf{v}_{13}, \mathbf{v}_{23})$. At this step, keeping the linearization point as in the previous step is crucial since the extracted pose-pose constraints are yet to be subsumed into relative pose factors at their operating point. It is clear that $\mathbf{v}_{13}, \mathbf{v}_{23}$ only transmit information from the landmark's Markov blanket to the pose's, and their exact nonlinear form is insignificant.

Up to now, we have removed a pose from the graph, limiting the graph size while trying to minimize information loss. Future optimization will be carried out with the newly generated nonlinear factors.

V. Example: VIO + Proposed Backend

This section describes a concrete example of using our backend by concatenating it to OpenVINS [35], a filter-based visual-inertial odometry. We chose it as the frontend because it rigorously follows consistency principles by design. We include this small example in our open-source code since the monocular-inertial suite is perhaps practitioners' most widely used sensory modalities.

A. Information from Frontend with Adaptation Graph

The direct marginal pose information from the Kalman filter contains a hybrid of exteroceptive and interoceptive measurements, which is unfavorable if we want to model landmarks separately. To ensure the factors fed into the backend are traceable and abuse-free (e.g., no measurement double counting), we design a small adaptation factor graph to transform filter information into backend factors.

The small adaptation graph acts as a buffer for filter information. It has the same length as the filter clone poses, the maximum number of poses a filter can constrain at a time. It does *not* optimize state values. Instead, it sparsely approximates constraints with values directly from the filter. Fig.2a shows that when a new clone is created in the filter, we marginalize old inertial states and add newly produced linear MSCKF constraints (i.e., pose-pose constraints from observing transient features [35]). Note that the prior factor contains 6 DoF pose information from initialization. Therefore, this small factor graph is well-constrained. Fig.2b illustrates the marginalization with involved states. Fig.2c does a sparsification over poses affected by MSCKF constraints and the prior factor. At this step, we can include a roll-pitch factor [10] in the approximating factors (P3 factor also has a copy, and NFR resolves this), which is globally observable and can constrain the backend graph.

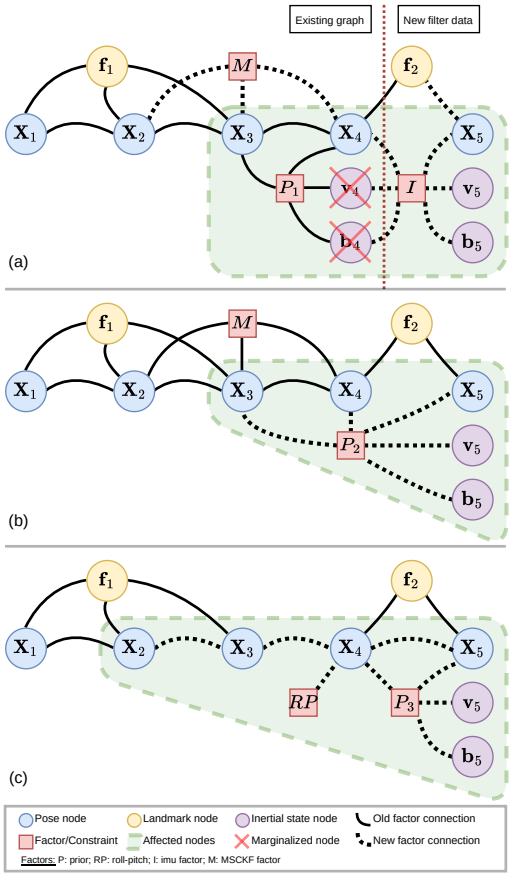

Fig. 2: Workflow in case the arrival of new measurement data from the filter at the small adaptation graph.

TABLE I: Simulation configuration

Parameter	Value	Parameter	Value
IMU Freq. (hz)	200	Max IMU Clones	11
Camera Freq. (hz)	10	Max Cam. Obs. Dist. (m)	7.0
Gyro. White Noise	1.7e-4	Max Cam. Track. Pts.	250
Accel. White Noise	2e-3	Max SLAM Feat. in State	50
Gyro. Rand. Walk	1.9e-5	Sim. Duration (s)	800
Accel. Rand. Walk	3e-3	Approx. Traj. Length (m)	2040

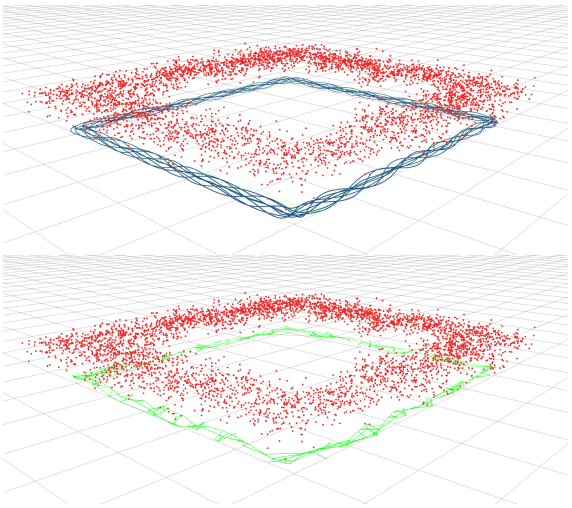

Fig. 3: Top: groundtruth trajectory (blue) and landmark points (red). Bottom: sparsified graph (green). Note that the pose-landmark constraints are not visualized. The size of background grids is $10m \times 10m$ each.

When the oldest pose is marginalized from the filter, we remove the corresponding node from the small graph and add it to the backend graph with its factors. In this way, only camera measurements, roll-pitch and relative pose factors are added to the backend, and the information about unobservable directions always stays in the small factor graph, which naturally preserves the observability pattern.

B. Loop Closure Management

We do not handle loop constraints explicitly because the visibility of the same set of landmarks naturally establishes the connection and corrects the drift. Nevertheless, we do a coarse PGO to avoid local minima caused by suboptimal initial guesses if it essentially contradicts the loop constraint. We argue that this has no probabilistic effects.

Despite the simplicity of handling the case of rediscovering previous landmarks, special care should be taken when marginalizing a pose at the crossroads of loop closure. This is because the PnP problem solving for a relative pose between the two loop poses can be ill-posed in the Markov blanket.

VI. Experiments and Results

A. Simulation

In this section, we explore the relationship between optimizing the sparsified graph's computation cost and the resulting pose estimation accuracy via simulation [35] with configuration as listed in Tab.I. The simulator provides a direct ID of each landmark, making loop closure much easier. The "Full-SLAM" method means a backend graph without removing anything. The test trajectory is generated around a $50m \times 50m$ square area and is perturbed with some smooth rotation and translation to avoid exactly repeating previous laps. The ground truth trajectory and the final sparsified graph of poses is shown in Fig.3. We run 10 laps around the area, generating about 2km of data. We admit that the duration is not extremely long, but according to experiment results, the trend has been revealed in the test period (either converged or growing without bound), see Fig.4.

The results show that sparsification can bound the error of state estimation while significantly more resource efficient than the full SLAM approach. Interestingly, the VIO frontend has a slightly better RMSE before loop closing than "full SLAM". We think this is caused by the information loss in the small adaptation graph, and the relinearization of landmarks also shifts the mean state values slightly away from the filter estimate. The "sparsified" one is slightly better, maybe due to marginalizing unstable poses.

B. Real-world dataset

Similar to Basalt [10], we also test on EuRoC [36] dataset. We select four baseline backend methods apart from VIO,

TABLE II: RMS ATE of the estimated trajectory in meters on EuRoC dataset, and the sparsity information. Note that all the methods except VIO runs periodically during the trajectory, and only the result of the last optimization (i.e., over the whole sequence) is reported. The proposed method only uses poses remained in the graph for calculating RMSE, but the graph is sparsified online following each optimization.

Methods [+]	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02
Mono-VIO	**0.07**	0.24	0.12	0.18	0.45	**0.05**	0.07	0.07	0.13	0.07
Simple loop	0.12	0.28	0.59	0.27	0.66	0.34	0.20	0.24	0.14	0.14
Weighted loop	**0.07**	0.24	0.12	0.18	0.45	0.06	0.07	0.08	0.13	0.07
Dense pairwise constraint	**0.07**	0.24	0.11	0.18	0.45	**0.05**	0.06	0.07	0.13	0.07
Full SLAM	**0.07**	**0.22**	**0.10**	**0.17**	0.47	**0.05**	**0.04**	0.07	**0.10**	**0.06**
Proposed	0.08	0.24	0.13	**0.17**	**0.43**	0.06	0.06	**0.06**	**0.10**	**0.06**
Final pose number after sparsif.	310	472	357	277	283	360	376	505	281	472
Total camera poses [*]	2768	2243	2295	1600	1834	2784	1595	1989	2160	2244

[+] The number of loop closures affects improvements from global optimization.
[*] Due to IMU initialization, etc, this may not equal to the total frames of images in the dataset.

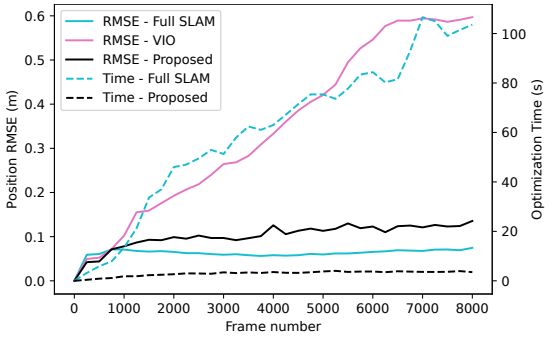

Fig. 4: Optimization time consumption and accuracy over time, compared with full SLAM and VIO. Batch optimization is done every 250 frames (about once every 1/4 lap).

including

1) Simple loop [37], [38], which uses identity information matrix in all consecutive and loop relative constraints.
2) Weighted loop [10]: the information of each relative pose measurement is from the adaptation factor graph, which has more accurate measure of uncertainty.
3) Dense pairwise constraint [12], [39]: add pairwise constraints for every possible pair, either in local window or in loop closure. Initially for LiDAR SLAM.
4) Full SLAM: use all the factors from the adaptation graph (Sec.V-A) without any approximation.

The RMSE results are shown in Tab.II. The full SLAM is generally more accurate than all other methods, while the sparsification one follows next. The simple loop method performs the worst due to incorrect weighting of factors.

We find that limited loops detected do not significantly increase the estimation accuracy. The same phenomenon is noticed in [10], which also has limited improvement over their VIO with their mapper. This could be a dataset characteristic. To be rigorous about the measurement model, we only generate descriptors at landmark points, lowering the chance of finding a loop. Looking at loop records, we find V2_01 has the most loop points and, therefore, the best improvement compared to VIO, while in MH_05, there is only one loop with a pose time difference larger than $3s$.

The final poses after sparsification require $2-5s$ for a run of optimization. Sparsifying a landmark's Markov blanket takes as low as $2ms$, and parallelization can accelerate batch processing. Overall, the sparsification can be done online, and doing periodic optimization is acceptable.

C. Discussion

Our VIO+Backend example acts as a prototype to prove the effectiveness of the proposed backend. We expect it to limit computation cost and bound error, which is verified in the experiments.

The sparsification policy is currently user-defined, and we only use an observation density-based strategy. This policy also affects performance, but it is beyond the scope of this paper and will be investigated in the future. Also, modern feature descriptors will be used for loop detection, and the detected points will be linked to poses with hindsight since our two-step sparsification naturally separates landmark operations from intra-pose operations.

VII. Conclusion

This paper proposes a new sparsification-based backend suitable for long-term general feature-based SLAM. The proposed method focuses on efficient and principled operations to sparsify pose-landmark graphs in a two-step manner. To demonstrate the effectiveness of our backend, we adapt a filter-based mono VIO to our backend. The results show that our sparsified backend can curb computation costs and has similar error performance as full SLAM.

Due to the space limitation, we have to leave more detailed experimental investigations (e.g., direct localization accuracy on the graph, time decomposition) to future reports. Also, aspects like the marginalization strategy, sparse topology calculation, minimal requirements for new Jacobian construction, and the influence of correlated measurements (e.g., estimating intrinsics online) will be thoroughly studied in our future works. More learning-based modules for change detection, robust relocalization, and landmark parameterization will be integrated into our subsequent research.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, p. 1309–1332, Dec. 2016.

[2] R. B. Sousa, H. M. Sobreira, and A. P. Moreira, "A systematic literature review on long-term localization and mapping for mobile robots," *Journal of Field Robotics*, vol. 40, no. 5, p. 1245–1322, Aug. 2023.

[3] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, p. 3565–3572.

[4] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Proc. Robot. Sci. Syst.*, vol. 2, no. 9, 2014.

[5] C. Campos, R. Elvira, J. J. Rodriguez, J. M. Montiel, and J. D. Tardos, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, p. 1874–1890, 2021.

[6] 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, p. 1004–1020, Aug. 2018.

[7] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic node removal for factor-graph slam," *IEEE Transactions on Robotics*, vol. 30, no. 6, p. 1371–1385, Dec. 2014.

[8] 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, p. 50–72, Jan. 2016.

[9] 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*, 2013, p. 54–61.

[10] V. Usenko, N. Demmel, D. Schubert, J. Stuckler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, p. 422–429, Apr. 2020.

[11] X. Liu, Z. Liu, F. Kong, and F. Zhang, "Large-scale lidar consistent mapping using hierarchical lidar bundle adjustment," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, p. 1523–1530, Mar. 2023.

[12] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally consistent 3d lidar mapping with gpu-accelerated gicp matching cost factors," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, p. 8591–8598, Oct. 2021.

[13] M. Mazuran, T. Gian Diego, S. Luciano, and W. Burgard, "Nonlinear graph sparsification for slam," in *Robotics: Science and Systems X*, 2014.

[14] J. Vallve, J. Sola, and J. Andrade-Cetto, "Graph slam sparsification with populated topologies using factor descent optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, p. 1322–1329, Apr. 2018.

[15] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess, "Information sparsification in visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, p. 1146–1153.

[16] C. Debeunne, J. Vallvé, A. Torres, and D. Vivet, "Fast bi-monocular visual odometry using factor graph sparsification," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit (MI), United States, 2023.

[17] L. Paull, G. Huang, M. Seto, and J. J. Leonard, "Communication-constrained multi-auv cooperative slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015, p. 509–516.

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

[19] K. Eckenhoff, L. Paull, and G. Huang, "Decoupled, consistent node removal and edge sparsification for graph-based slam," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, p. 3275–3282.

[20] J. Vallvé, J. Solà, and J. Andrade-Cetto, "Pose-graph slam sparsification using factor descent," *Robotics and Autonomous Systems*, vol. 119, p. 108–118, Sept. 2019.

[21] N. Carlevaris-Bianco and R. M. Eustice, "Conservative edge sparsification for graph slam node removal," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014, p. 854–860.

[22] J. Vial, H. Durrant-Whyte, and T. Bailey, "Conservative sparsification for efficient and consistent approximate estimation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, 2011, p. 886–893.

[23] K. Khosoussi, M. Giamou, G. S. Sukhatme, S. Huang, G. Dissanayake, and J. P. How, "Reliable graphs for slam," *The International Journal of Robotics Research*, vol. 38, no. 2–3, p. 260–298, Mar. 2019.

[24] K. J. Doherty, D. M. Rosen, and J. J. Leonard, "Spectral measurement sparsification for pose-graph slam," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, 2022, p. 01–08.

[25] G. Kurz, M. Holoch, and P. Biber, "Geometry-based graph pruning for lifelong slam," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, 2021, p. 3313–3320.

[26] S. Choudhary, V. Indelman, H. I. Christensen, and F. Dellaert, "Information-based reduced landmark slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015, p. 4620–4627.

[27] Y. Zhao and P. A. Vela, "Good feature matching: Toward accurate, robust vo/vslam with low latency," *IEEE Transactions on Robotics*, vol. 36, no. 3, p. 657–675, June 2020.

[28] G. D. Tipaldi, D. Meyer-Delius, and W. Burgard, "Lifelong localization in changing environments," *The International Journal of Robotics Research*, vol. 32, no. 14, p. 1662–1678, Dec. 2013.

[29] 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)*, Prague, Czech Republic, 2021, p. 3305–3312.

[30] V. Ila, L. Polok, M. Solony, and P. Svoboda, "Slam++ -a highly efficient and temporally scalable incremental slam framework," *International Journal of Robotics Research*, vol. 36, no. 2, p. 210–230, Feb. 2017.

[31] A. Adkins, T. Chen, and J. Biswas, "Obvi-slam: Long-term object-visual slam," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, p. 2909–2916, Mar. 2024.

[32] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic lidar fusion: Dense map-centric continuous-time slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, p. 1206–1213.

[33] C. Park, P. Moghadam, J. Williams, S. Kim, S. Sridharan, and C. Fookes, "Elasticity meets continuous-time: Map-centric dense 3d lidar slam," *IEEE Transactions on Robotics*, vol. 38, no. 2, p. 978–997, Apr. 2022.

[34] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, June 2021, p. 7206–7215.

[35] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, p. 4666–4672.

[36] 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*, 2016.

[37] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv*, 2019.

[38] R. Mur-Artal and J. D. Tardos, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, p. 796–803, Apr. 2017.

[39] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized gicp for fast and accurate 3d point cloud registration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, p. 11054–11059.