A Partial Sparsification Scheme for Visual-Inertial Odometry

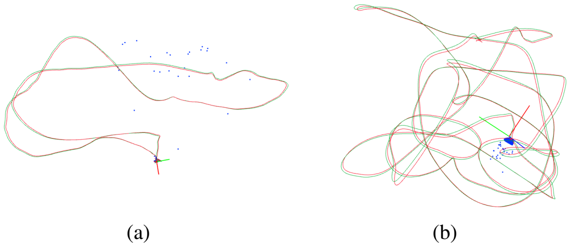
Zhikai Zhu[1,2] and Wei Wang[1,2]

***Abstract*— In this paper, we present a partial sparsification scheme for the marginalization of visual inertial odometry (VIO) systems. Sliding window optimization is widely used in VIO systems to guarantee constant complexity by optimizing over a set of recent states and marginalizing out past ones. The marginalization step introduces fill-in between variables incident to the marginalized ones, and most VIO systems discard measurements targeted at active landmark points to maintain sparsity of the marginalized information matrix, at the expense of potential information loss. The scheme is to first retain the dense prior from the marginalization excluding visual measurements, followed by a dense marginalization step that connects landmarks. The dense marginalization prior is then partially sparsified to extract pseudo factors that maintain the overall sparsity while minimizing the information loss. The proposed scheme is tested on public datasets and achieves appreciable results compared with several state-of-the-art approaches. The test also demonstrates that our scheme is applicable to real-time operations.**

I. INTRODUCTION

State estimation is a crucial module of autonomous robots. Both the robustness and accuracy of a state estimator are essential for the motion planning and control of robots in challenging scenarios. Due to the often limited computation resources on board, designing and implementing a real-time whilst optimal state estimator remains a key research focus in the field of robotics.

Vision-only systems have been widely used for the localization of autonomous robots in indoor and GPS-denied environments [1]–[5]. The robustness of vision-only systems, however, may often be compromised by illumination change, motion blur or textureless area. The lack of an accurate motion model further deteriorates the performance of the visual tracking, resulting in suboptimal state estimation accuracy. To tackle the aforementioned problems, much attention has recently been given to assisting the vision system with a low-cost inertial measurement unit (IMU) due to their complementary nature [6]. Vision systems provide exteroceptive information for long-term navigation, while IMUs provide interoceptive information for short-term motion update. The combined system is typically termed as a Visual Inertial Odometry (VIO) system. VIO systems are either based on filtering [7]–[9] or graph optimization [6], [10]–[13]. The filtering-based approach achieves higher efficiency at the expense of less accuracy compared with graph optimization. Sliding window optimization [6], [12], [13] is proposed to make a tradeoff between efficiency and accuracy by only optimizing over a fixed-sized set of recent states and marginalizing out past observations and states. There are, however, several known drawbacks of such approach. Marginalization of past states fixes the linearization points, thus the results no longer represent the original nonlinear optimization. Marginalization typically causes fill-in between the variables related to the marginalized states, which turns the sparse information matrix into a dense marginalization prior and thus increasing the computation burden of the VIO system.


(a)

(b)

Fig. 1. The VIO estimates with our proposed scheme are aligned with the ground-truth trajectories of the (**a**)MH_02 and (**b**)V1_01 sequences in the EuRoC datasets.

To address the issues listed above, we propose a partial sparsification scheme for the marginalization of sliding window VIO. The proposed scheme retains the dense marginalization prior by first marginalizing out the host landmarks and the keyframe pose excluding visual measurements, then carries another marginalization step including the visual cues and implements a partial sparsification step extracting nonlinear factors from the resultant information matrix, thus preserving the sparse nature of the VIO optimization. Our main contributions are:

- We propose a novel sparsification scheme that retains the dense prior connecting only keyframe poses and frame states in the sliding window and extracts landmark-to-pose factors from a full-sized marginalization prior which also includes landmarks.
- We compare our proposed scheme with the current state-of-the-art VIO systems on the EuRoC visual inertial datasets [14] to prove its effectiveness (see Fig. 1).
- We perform a run-time analysis of our proposed method to demonstrate that it is applicable to real-time operations.

II. RELATED WORK

VIO systems can be generally categorized into two types. The first type is loosely coupled sensor fusion [15], [16],

[1] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China.

Corresponding author: Wei Wang (`wei.wang@ia.ac.cn`)

where an independent visual odometry system is aided by the IMU. The second type is tightly coupled sensor fusion [6], [12], [13], [17], where the IMU and visual measurements are jointly optimized. The tightly coupled VIO can be further categorized by their backend algorithms. They are either filtering-based [9] or optimization-based [6], [10]–[13]. The filtering-based approach is computationally efficient, yet suffers from linearization errors as the linearization points of the state transition model and measurement model cannot be changed. The graph optimization approach formulates VIO as a nonlinear optimization problem and the optimal states can be iteratively solved by standard nonlinear optimization techniques. The graph optimization approach is more computationally demanding, yet the sparse nature of the accumulated information matrix reduces the size of the matrix to be inverted to that of only those frame states'. The scale of the graph optimization, however, may go unbounded with time as more frame states and landmark positions are included. To attain real-time performance the problem can be reduced to a sliding window optimization over a computationally manageable size of variables by marginalizing past states and only optimizing over a set of most recent states. Efficient open-source solvers such as g2o [18] and Ceres [19] can be applied to implement real-time VIO.

Marginalization of past states may cause inaccuracy as the linearization points of past states are fixed and the prior can no longer represent the original nonlinear optimization problem. Moreover, marginalization causes fill-in among all the variables related to the marginalized states, which turns the sparse information matrix into a dense marginalization prior and thus significantly increasing the computation burden of the VIO system. The common approach to avoid fill-in is discarding all the keypoints and observations observed in marginalized keyframes. This approach inevitably reduces the information content of the sliding window since the observations targeted at marginalized keyframes are directly dropped, and the solution to the sliding window optimization is therefore no longer optimal. On the contrary, our sparsification scheme allows the inclusion of observations targeting at marginalized keyframes in marginalization and retains both the dense prior connecting frame states and the sparse pattern of the overall information matrix to minimize the information loss in marginalization without any decrease in efficiency.

Given the popularity of graph-based optimization, numerous efforts have been made towards reducing the number of nodes in the graph, while minimizing the approximation error and maintaining the sparse pattern of the information matrix. Most of the existing literature on information sparsification focus on pose-graph node removal. Kretzschmar et al. [20] employed the Chow-Liu Tree Approximation [21] to sparsify the Markov blanket of the marginalized nodes. Calevaris-Bianco et al. introduced Generic Linear Constraint (GLC) factors [22] to approximate the information matrix of the Markov blanket. Maruzan et al. introduced Nonlinear Factor Recovery (NFR) [23] to use specified nonlinear factors to approximate the dense prior by minimizing Kullback-Leibler divergence (KLD). Hsiung et al. [24] utilizes NFR to achieve a sparse marginalization prior, yet their experimental results on datasets are all generated offline, which fails to demonstrate the claimed accuracy in real time. Our sparsification scheme follows a greedy selection method to extract nonlinear landmark-to-pose factors in terms of mutual information, after retaining the dense prior connecting keyframe poses and frame states. We show that our partial sparsification scheme achieves appreciable performance on the EuRoC visual inertial datasets [14] and is applicable to real-time operations.

III. Problem formulation

The sliding window VIO optimizes over a set of keyframe poses, frame states and landmark positions

$$X = \{x_k, x_f, x_l\}, \tag{1}$$

where x_k consist of poses for m consecutive keyframes, x_f consist of IMU states for n most recent frames and x_l consist of landmark positions. The IMU state f_i is defined as:

$$f_i = \{\xi_i^{\mathrm{T}}, \mathbf{v}_i^{\mathrm{T}}, \mathbf{b}_i^{\mathrm{T}}\}^{\mathrm{T}}, \tag{2}$$

where $\xi_i \in \mathbb{R}^6$ is the pose, $\mathbf{v}_i \in \mathbb{R}^3$ the velocity and $\mathbf{b}_i \in \mathbb{R}^6$ the IMU biases. The landmark position l_i is parameterized as follows:

$$l_i = \{u, v, \lambda_i\}^{\mathrm{T}}, \tag{3}$$

where $\{u, v\}^{\mathrm{T}}$ is the unit unit-length direction vector of the landmark point in the keyframe where it was observed for the first time[13] and λ_i the inverse distance[4]. The sliding window VIO is formulated as a bundle adjustment problem and it minimizes the sum of the weighted norm of all measurement residuals and the prior:

$$\begin{aligned} \underset{X}{min}\{r_M + \sum_{(i,j)\in I} r_{ij}^{\mathrm{T}}\Sigma_{ij}^{-1}r_{ij} \\ + \sum_{l\in L, t\in mea(l)} \rho(r_{lt}^{\mathrm{T}}\Sigma_{lt}^{-1}r_{lt})\}, \end{aligned} \tag{4}$$

where r_M is the marginalization prior, I the set of pairs of frames connected by IMU factors, r_{ij} the corresponding factors, L the set of all landmark points, $mea(l)$ the set of targeting frames of point l, r_{lt} the reprojection errors and Σ the corresponding covariances. The Huber norm [25] is defined as

$$\rho(r) = \begin{cases} r & r \leqslant 1 \\ 2\sqrt{r} - 1 & r > 1 \end{cases}. \tag{5}$$

The reprojection error r_{lt} is defined as:

$$r_{lt} = z_{lt} - \pi(T_t^{-1}T_h \frac{1}{\lambda_l}\pi^{-1}(u, v)), \tag{6}$$

where π is the camera projection model, T_t the pose of the target frame, T_h the pose of the host frame and z_{lt} the camera measurement.

The second type of residual is the IMU factor. To avoid repeated IMU reintegration in the iterative optimization, we

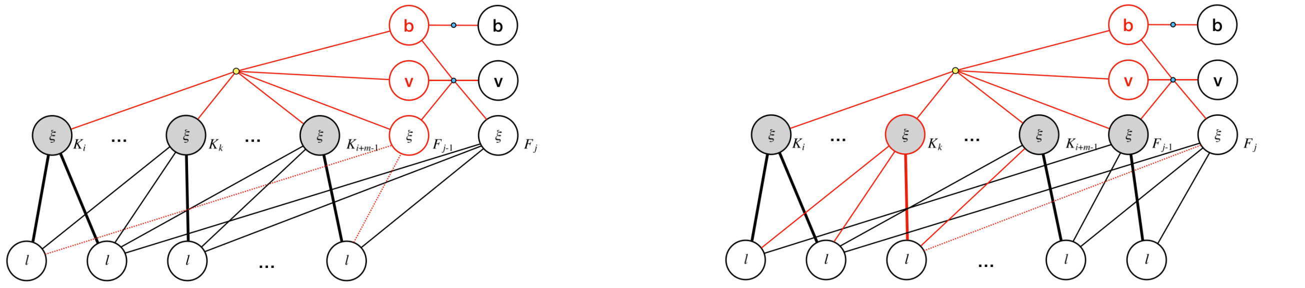


Fig. 2. The proposed marginalization scheme for sliding window VIO. The blue dots represent the IMU factors, the yellow dots represent the marginalization prior connecting all the keyframe poses and the last frame state, the camera reprojection errors are represented by black straight lines, among them the bold lines indicate that the points are host by the connected keyframes. The variables to be marginalized are in red circle, while the red straight lines denote the factors to be included in the marginalization, and the red dotted lines denote those to be excluded. Two possible scenarios are: (**a**) the last frame F_{j-1} is a regular frame and we simply discard the reprojection errors targeting at F_{j-1} before marginalizing out the frame state, (**b**) the last frame F_{j-1} is a keyframe and we first marginalize the velocity and biases of this frame then marginalize out one older keyframe K_k in the prior.

follow [26] and preintegrate a number of consecutive IMU measurements into one IMU factor $\Delta\mathbf{f} = (\Delta\mathbf{R}, \Delta\mathbf{v}, \Delta\mathbf{p})$. $\Delta\mathbf{f}$ is initialized as $(\mathbf{I}, 0, 0)^{\mathrm{T}}$ and for each IMU measurement (ω_t, a_t) at time t in $[t_i, t_j]$ updated as follows:

$$\Delta\mathbf{R}_{t+1} = \Delta\mathbf{R}_t Exp(\omega_t \Delta t), \tag{7}$$

$$\Delta\mathbf{v}_{t+1} = \Delta\mathbf{v}_t + \Delta\mathbf{R}_t a_t \Delta t, \tag{8}$$

$$\Delta\mathbf{p}_{t+1} = \Delta\mathbf{p}_t + \Delta\mathbf{v}_t \Delta t + \frac{1}{2}\Delta\mathbf{R}_t a_t \Delta t^2. \tag{9}$$

Exp is the composition of the hat operator ($\mathbb{R}^3 \rightarrow \mathfrak{so}(3)$) and the matrix exponential ($\mathfrak{so}(3) \rightarrow SO(3)$) and maps rotation vectors to their corresponding rotation matrices.

After introducing the preintegration technique, the IMU factor given the measurement $\Delta\tilde{\mathbf{f}} = (\Delta\tilde{\mathbf{R}}, \Delta\tilde{\mathbf{v}}, \Delta\tilde{\mathbf{p}})$ is then defined as:

$$r_{\Delta\mathbf{R}} = Log(\Delta\tilde{\mathbf{R}}^{\mathrm{T}}\mathbf{R}_i^{\mathrm{T}}\mathbf{R}_j), \tag{10}$$

$$r_{\Delta\mathbf{v}} = \mathbf{R}_i^{\mathrm{T}}(\mathbf{v}_j - \mathbf{v}_i - g\Delta t_{ij}) - \Delta\tilde{\mathbf{v}}, \tag{11}$$

$$r_{\Delta\mathbf{p}} = \mathbf{R}_i^{\mathrm{T}}(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) - \Delta\tilde{\mathbf{p}}, \tag{12}$$

where Log is the inverse of Exp and maps rotation matrices to their corresponding rotation vectors, g is the gravity vector and $\Delta t_{ij} = t_j - t_i$.

The covariance matrices of IMU factors can be recursively calculated by the error dynamics and the readers can refer to [26] for more detailed derivation of IMU preintegration.

The overall minimizing function can be rearranged as $f(X) = r(X)^{\mathrm{T}}\Sigma^{-1}r(X)$, and the optimization step first calculates the Jacobian of $r(X)$:

$$J_r(X) = \lim_{\delta X \rightarrow 0} \frac{r(X \oplus \delta X) \ominus r(X)}{\delta X}, \tag{13}$$

and the Gauss-Newton update is calculated as:

$$\delta X = -(J_r(X)^{\mathrm{T}}\Sigma^{-1}J_r(X))^{-1}J_r(X)^{\mathrm{T}}\Sigma^{-1}r(X), \tag{14}$$

after which the state is updated as:

$$X = X \oplus \delta X. \tag{15}$$

The calculation of Jacobian and state update is carried iteratively to solve for the optimal estimate.

IV. Marginalization with Partial Spasification

Marginalization of past states is necessary as the sliding window VIO only optimizes over a window of recent states to ensure constant complexity. We first define the Markov blanket as the collection of state variables that are incident to the marginalized variables. The marginalization is implemented using Schur complement and only on the Markov blanket of the variables to be marginalized. Define H as the information matrix of the Markov blanket and b the information vector, x the variables in the Markov blanket and can be further split into $x = [x_m^{\mathrm{T}}, x_r^{\mathrm{T}}]^{\mathrm{T}}$, where x_m corresponds to the variables to be marginalized and x_r the remaining variables in the Markov blanket. H and b can be split as:

$$H = \begin{bmatrix} H_{mm} & H_{mr} \\ H_{rm} & H_{rr} \end{bmatrix}, b = \begin{bmatrix} b_m \\ b_r \end{bmatrix}, \tag{16}$$

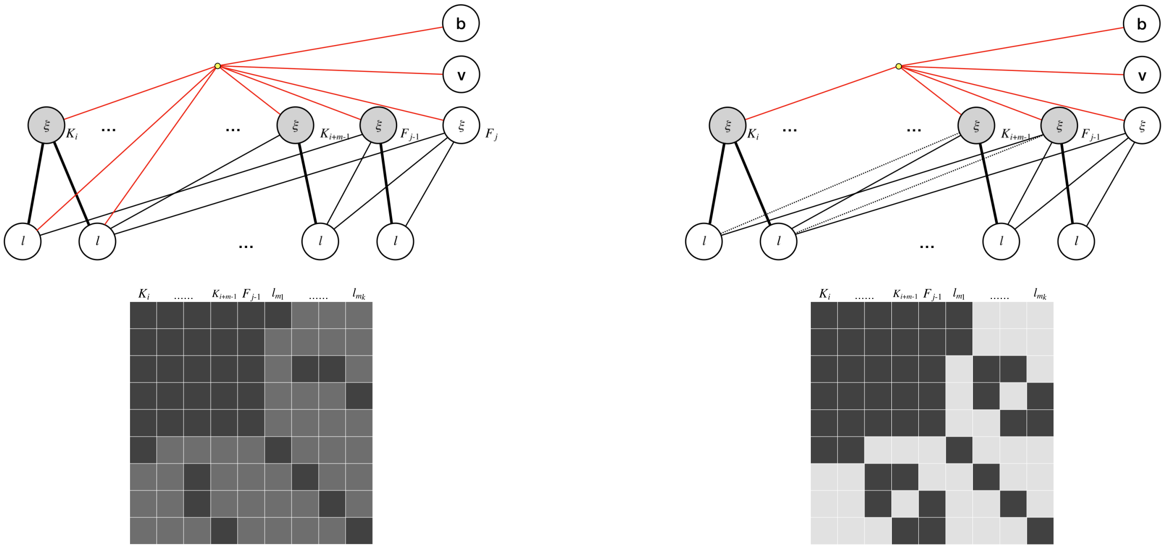
The Schur complement on H and b is implemented as follows:

$$H_M = H_{rr} - H_{rm}H_{mm}^{-1}H_{mr}, \tag{17}$$

$$b_M = b_r - H_{rm}H_{mm}^{-1}b_m. \tag{18}$$

H_M and b_M now represent the marginalization prior and describe the distribution of the remaining variables in the Markov blanket.

It is important to note that the original information matrix of the nonlinear optimization (the matrix to be inverted on the right hand side of (14)) is derived from the IMU factors, reprojection errors and the prior from last marginalization. The IMU factors introduce nonzero entries between the frame states they connect and the reprojection errors introduce nonzero entries between landmarks and target frames. The nonlinear optimization needs to invert the information matrix to solve for the increments of optimizing variables, and the inversion can be carried out in an efficient way by using Schur complement and solving first for the states and poses variables then the landmarks. The speedup in the optimization is due to the sparse pattern of the information



(a)

(b)

Fig. 3. The proposed partial sparsification scheme for sliding window VIO. The upper subgraphs for each subfigure represent the factor graph, while the lower ones are the visualization of the corresponding information matrices from the marginalization prior and selective factors. (**a**): The marginalization prior before partial sparsification (represented by the yellow dot) is densely connected by keyframe poses, the last frame state and landmark points targeted at the marginalized keyframe. (**b**): The marginalization prior after partial sparsification is only connected by keyframe poses and the last frame state. The extracted pseudo factors are represented by the black dotted lines. Note that the information matrix in (**b**) is constructed by both the new prior and the pseudo factors, and only the dense submatrix in the upper left of it corresponds to the new marginalization prior.

matrix and if the marginalization prior somehow destroys the overall sparsity of the information matrix, the compuation complexity of the optimization will significantly increase.

A. *Marginalization*

It is worthwhile noting that marginalization causes fill-in between variables in the Markov blanket, therefore the marginalization strategy needs to be carefully designed to maintain the sparsity pattern of the information matrix.

When a new frame enters the sliding window, our marginalization step first inspects the last frame in the window. If it is not a keyframe as shown in Fig. 2a, we discard all its visual measurements and marginalize out the frame, such marginalization only causes fill-in between poses and states and does not destroy the overall sparsity. If it is a keyframe as shown in Fig. 2b, we first marginalize the velocity and biases of this frame and then marginalize out one older keyframe in the prior. We first discard the visual measurements from the points host by the keyframe to active frames in the window, as the linearization points of those active frames would be fixed otherwise, then marginalize out the landmark points host by the keyframe. The Markov blanket now only contains other keyframe poses connected by the previous marginalization prior and the points host by other keyframes targeting at the to-be-marginalized keyframe. Current state-of-the-art VIO systems like VINS-MONO [13] and OKVIS [6] directly discard those visual measurements to maintain sparsity at the expense of possible information loss. We follow this strategy to obtain a dense prior H_d constraining only the keyframe poses and the last frame state in the sliding window. To further minimize information loss, we implement another marginalization including those visual measurements which results in a full-sized dense marginalization prior H_o connecting also landmarks, the full-sized prior is then partially sparsified to guarantee optimization efficiency.

B. *Partial Sparsification*

The inclusion of visual measurements in the marginalization results in a dense prior H_o (see Fig. 3a) represented by a Gaussian $p_o(X) \sim N(\mu_o, \Sigma_o)$, μ_o the current estimate and $\Sigma_o = H_o^{-1}$ the covariance matrix of the variables in Markov blanket. We implement partial sparsification to sparsify the information matrix H_t while minimizing information loss.

We first define our pseudo Gaussian distribution as $p_p(X_t) \sim N(\mu_p, \Sigma_p)$ and minimize the Kullback-Leibler divergence (KLD) between the pseudo distribution and the original distribution:

$$D_{KL}(p_o(X)|p_p(X)) = \frac{1}{2}(\langle \Sigma_p^{-1}, \Sigma_o \rangle - logdet(\Sigma_p^{-1}\Sigma_o) + \left\| \Sigma_p^{-\frac{1}{2}}(\mu_p - \mu_o) \right\|^2 - d), \quad (19)$$

where d is a constant that can be neglected in the following optimization.

$p_p(X)$ is termed as pseudo distribution as it represents pseudo factors that we will now define to enforce sparsity. By inspection of the information matrix structure excluding the prior we notice that the submatrix corresponding to the keyframe poses and frame states is dense, thus we need not actually design sparse factors for them. Somehow we still define a pseudo factor r_p connecting all the keyframe poses and the last state and do not explicitly specify its structure for now. To enforce the general sparsity of the information matrix we define the following pseudo factor between poses and landmarks:

$$r_l = h(T_t^{-1} T_h \frac{1}{\lambda_l} \pi^{-1}(u, v)), \tag{20}$$

where h transforms the homogenous coordinate into the three-dimensional Euclidean coordinate.

The pseudo factor r_l can be interpreted as a three-dimensional measurement of landmark points in the target frames, while the reprojection error defined in (6) can be interpreted as a two-dimensional measurement of such. It follows that the pseudo distribution $p_p(X)$ induced by r_p and r_l maintains the sparse pattern of the overall information matrix, as the accumulated information matrix from those factors is similar to that from the reprojection factors in terms of sparsity pattern (see Fig. 3b), and the efficiency of the optimization is not compromised.

We then follow NFR [23] to recover the measurement z_i and information matrix H_i for the ith pseudo factor r_i. Choose $z_i = r_i(\mu_o)$ induces $\mu_p = \mu_o$ and the third term in (19) vanishes. The Jacobians and information matrices of the pseudo factors are stacked as:

$$J = \begin{bmatrix} \vdots \\ J_i \\ \vdots \end{bmatrix}, H = \begin{bmatrix} \ddots & & \\ & H_i & \\ & & \ddots \end{bmatrix}, \tag{21}$$

where J_i and H_i are respectively the Jacobian and information matrix of the ith pseudo factor r_i. It follows that $H_p = \Sigma_p^{-1}$ can be written as $H_p = J^T H J$, and can be substituted into (19) as:

$$\begin{aligned} D_{KL}(p_o(X)|p_p(X)) = \frac{1}{2}((\langle J^T H J, \Sigma_o \rangle \\ - logdet(J^T H J). \end{aligned} \tag{22}$$

Note that (22) is a convex MAXDET problem [27] and a closed form solution exists if J is full-rank and invertible [23] from which H_i can be solved as:

$$H_i = (\{J \Sigma_o J^T\}_i)^{-1}, \tag{23}$$

where $\{\}_i$ denotes the ith diagonal submatrix. The optimality can be proved as in [23] by calculating the gradient of the objective function (22) with respect to each block H_i on the diagonal of H:

$$\begin{aligned} \frac{\partial D_{KL}}{\partial H_i} &= \{J(\Sigma_o - (J^T H J)^{-1}) J^T\}_i \\ &= \{J \Sigma_o J^T - H^{-1}\}_i. \end{aligned} \tag{24}$$

Note that by now the structure of r_p is not defined, yet we can still safely recover the measurements and information matrices of r_l, as r_p is constrained to only connect keyframe poses and frame states, and the corresponding Jacobian does not have any non-zero entries related to landmarks. As shown in Fig. 2a the marginalization prior connects all the keyframe poses and the last active frame state in the sliding window, which suggests that the information matrix before marginalization has a fixed-size dense submatrix corresponding to those variables. It is obvious then that there is actually no need to construct the pseudo factor r_p to sparsify the subbmatrix related to keyframe poses and frame state variables in H_o, as the accumulation of a sparse matrix and a dense one is still dense.

We thus propose to construct the new marginalization prior by first marginalizing out the landmark points host by the keyframe, then the keyframe pose itself excluding all the visual measurements. The resultant prior H_d is a dense matrix constraining all the keyframe poses and the last frame state. This way we can avoid designing the topology of r_p and calculating the new information matrix of the pseudo factor, which is optimal only in the sense of the particularly assigned topology.

The whole process is thus termed as partial sparsification as we only extract sparse pseudo factors between robot poses and landmark points (see the sparse subblock in the upper right and lower left part of the information matrix in Fig. 3b) by processing the full-sized marginalization prior including visual cues (see the dense information matrix in Fig. 3a), while constructing the dense prior H_d connecting the keyframe poses and the last frame state from the marginalization not considering any reprojection errors. The marginalization prior after partial sparsification (see the dense subblock in the upper left part of the information matrix in Fig. 3b), albeit a still dense matrix, remains fix-sized and thus the optimization complexity thereafter is not increased.

C. Greedy selection of pseudo factors

The sliding window includes several keyframes and for each landmark point in the marginalization prior we need to designate which keyframe to be connected by its respective pseudo factor. For this we propose a greedy selection criteria to maximize the overall mutual information between variables.

Given the overall covariance matrix Σ which can be calculated as the inverse of the information matrix of the Markov blanket before partial sparsification, the mutual information between two variables x_i and x_j can be directly calculated as in [23]:

$$I(x_i, x_j) = \frac{1}{2} log \frac{det\Sigma_{ii} det\Sigma_{jj}}{det \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix}}, \tag{25}$$

we then choose the target keyframe k_i of the pseudo factor connecting landmark l_i as:

$$k_i = \arg \max_{k_j \in x_k} I(k_j, l_i), \tag{26}$$

this way the sum of mutual information between the variables connected by the pseudo factors r_l is maximized, and the selection method is implemented in combination with NFR to minimize the information loss of the partial sparsification.

TABLE I

RMS ATE (METER) OF THE ESTIMATES ON THE EUROC DATASET FOR SEVERAL DIFFERENT METHODS

Sequence	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02
Proposed	0.09	**0.05**	**0.073**	0.13	**0.10**	0.05	0.056	**0.082**	0.040	**0.066**
OKVIS	0.33	0.37	0.25	0.27	0.39	0.09	0.14	0.21	0.09	0.17
VINS-MONO	0.15	0.15	0.22	0.32	0.30	0.08	0.11	0.18	0.08	0.16
BASALT	0.09	0.06	0.076	**0.11**	0.12	**0.04**	**0.055**	0.084	**0.037**	0.072
ROVIO	0.35	0.36	0.45	0.92	1.11	0.13	0.16	0.17	0.22	0.39
VIO with full information sparsification in [24]	**0.06**	0.06	0.10	0.24	0.19	0.06	0.09	0.26	0.08	0.21

V. EXPERIMENTAL RESULTS

To validate our partial sparsification scheme, we implement our method using the same visual frontend as in BASALT [28], and evaluate it on the EuRoC [14] visual inertial datasets. The EuRoC datasets are collected by sensors on a micro aerial vehicle (MAV), including synchronized 20Hz stereo images and 200Hz IMU measurements and ground truth. We first align the estimates with the ground-truth trajectories and calculate the root mean square (RMS) of the absolute trajectory error (ATE). All experiments are run on an Ubuntu desktop with 2.2GHz 4-core Intel i7.

We compare our results with several state-of-the-art VIO systems utilizing two different backend methods: OKVIS [6], VINS-MONO [13] without loop closure, and BASALT [28] which are based on sliding window optimization and ROVIO [29] which implements iterated extended Kalman filter (IEKF). For comparison we also include the VIO system proposed by Hsiung et al. [24] which implements full information sparsification.

The evaluation results as shown in Table. I demonstrate that our proposed method achieves competitive accuracy compared with the existing methods mentioned above. Further comparison with the VIO system in [24] shows that our method achieves inferior results only in one out of ten sequences and outperforms it in the rest. The visualization of the VIO estimates aligned with the ground-truth trajectories of the EuRoC datasets is shown in Fig .1, and Fig. 4 shows the absolute pose error (APE) with respect to the translation part of our estimate generated by evo [30].

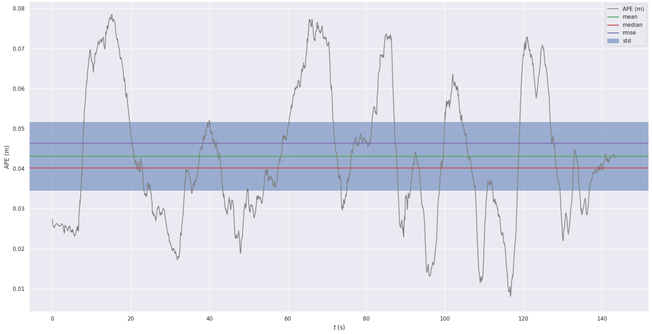


Fig. 4. The APE with respect to the translation part of our estimate of the V1_01 sequence, the RMSE is around 0.46m and the maximum below 0.08m.

TABLE II

RUN-TIME ANALYSIS ON THE EUROC DATASET

Sequence	Optimization (unit: s)	Proposed Marginalization (unit: s)	Marginalization in [24] (unit: s)
MH_01	0.013	0.003	0.248
MH_02	0.013	0.003	0.230
MH_03	0.011	0.003	0.151
MH_04	0.010	0.002	0.088
MH_05	0.009	0.003	0.115
V1_01	0.010	0.004	0.018
V1_02	0.006	0.003	0.020
V1_03	0.006	0.001	0.015
V2_01	0.012	0.003	0.039
V2_02	0.010	0.002	0.039

To ensure that our scheme can be applied for real-time VIO, we next conduct a run-time analysis on the experiments of the EuRoC datasets. As shown by the results in Table. II the time spent on per optimization step varies across different sequences from a minimum of 0.006s to a maximum of 0.013s, while the time consumed by marginalization and partial sparsification for one step is less variable (around 0.001 to 0.004s). The worst case scenario (MH_01) suggests an average of 0.016s for a single backend step and given the update frequency of the stereo camera which is 20 frames per second, it is safe to claim that our scheme can be applied to real-time operations without introducing any unwanted lag. For comparison, we also include in Table. II the time spent on the marginalization step of the VIO system in [24] that implements full information sparsification, which has a maximum of 0.248s per step. The reason for our improvement in the marginalization and sparsification efficiency is twofold. Firstly, our partial sparsification scheme retains the dense prior from the marginalization without visual measurements, and need not recover the pseudo factor connecting the keyframe poses and the last frame state. Secondly, in our implementation the recovery of pseudo factors together with the greedy selection is done in parallelization, since the information matrices of the pseudo factors can be indepedently solved by (23).

VI. CONCLUSIONS

In this paper we present a partial sparsification scheme for the marginalization of sliding window VIO. Our method recovers pseudo factors from the dense full-sized marginalization prior to maintain the overall sparsity of the information matrix while minimizing the information loss, and retains the dense prior connecting keyframe poses and the last frame state in the sliding window. Furthermore, we propose a greedy selection method to assign optimal pseudo factors in order to maximize the mutual information sum. Experimental results illustrate the efficacy of our method and that it is applicable to real-time operations.

Future work includes the fusion of other sensors to achieve higher odometry accuracy and the design and implementation of the corresponding marginalization scheme.

References

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pp. 225–234, IEEE, 2007.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 15–22, IEEE, 2014.

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

[4] 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.

[5] 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.

[6] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[7] 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*, pp. 3565–3572, IEEE, 2007.

[8] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[9] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 298–304, IEEE, 2015.

[10] 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.

[11] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5303–5310, IEEE, 2015.

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

[13] 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.

[14] 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.

[15] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *2012 IEEE international conference on robotics and automation*, pp. 957–964, IEEE, 2012.

[16] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *2013 IEEE/RSJ international conference on intelligent robots and systems*, pp. 3923–3929, IEEE, 2013.

[17] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1885–1892, IEEE, 2016.

[18] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, "g2o: a general framework for (hyper) graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, pp. 9–13, 2011.

[19] S. Agarwal, K. Mierle, *et al.*, "Ceres solver," 2012.

[20] H. Kretzschmar and C. Stachniss, "Information-theoretic compression of pose graphs for laser-based slam," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1219–1230, 2012.

[21] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[22] N. Carlevaris-Bianco and R. M. Eustice, "Generic factor-based node marginalization and edge sparsification for pose-graph slam," in *2013 IEEE International Conference on Robotics and Automation*, pp. 5748–5755, IEEE, 2013.

[23] 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.

[24] 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)*, pp. 1146–1153, IEEE, 2018.

[25] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*, pp. 492–518, Springer, 1992.

[26] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," Georgia Institute of Technology, 2015.

[27] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM journal on matrix analysis and applications*, vol. 19, no. 2, pp. 499–533, 1998.

[28] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, 2019.

[29] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[30] M. Grupp, "evo: Python package for the evaluation of odometry and slam.." `https://github.com/MichaelGrupp/evo`, 2017.