Fast bi-monocular Visual Odometry using Factor Graph Sparsification

César Debeunne[1], Joan Vallvé[2] , Alex Torres[3]and Damien Vivet[1]

***Abstract*— Visual navigation has become a standard in robotic applications with the emergence of robust and versatile algorithms. In particular, Visual Odometry (VO) has proven to be the most reliable navigation solution for space missions to estimate an unmanned vehicle's motion and state. Lava Tubes exploration is one of the recent challenges in this field of applied robotics. VO in this scenario requires more robustness to poor lighting conditions while keeping a low computational cost. We propose investigating an indirect bi-monocular VO based on sliding-window optimization in such a context. It focuses on maintaining the sparsity of the problem while keeping the information of the marginalized frames to reduce the computational burden. Different sparse graph topologies are studied to encode information from the past and are evaluated on accuracy and computation load. The best method retained is then compared to state-of-the-art systems on real data under extreme illumination conditions and reaches similar accuracy results at a lower computational cost.**

I. Introduction

Navigation in an unstructured environment is one of the main challenges for robotics applied to space exploration. Indeed, neither global positioning nor online human supervision is available on extraterrestrial surfaces to ensure safe navigation. A robot must only rely on robust navigation solutions based on embedded sensors. Mars Exploration Rover (MER) was the first mission that involved a Visual Odometry (VO) system for navigation using a stereo rig called NavCam [1]. This method was the most reliable solution onboard to estimate the robot displacement, as slippage often corrupted odometers. Nowadays, vision navigation systems are standard for spatial missions, as demonstrated recently with the Lander Vision System (LVS), which successfully provided the position of Mars 2020 during the Entry, Descent and Landing phase [2]. However, new challenges are at stake with the recent discovery of lava tubes on Mars and the Moon. These areas, shielded from spatial radiation and impacts, might host potential extraterrestrial human bases. Robotic exploration of these caves is thus necessary for a preliminary investigation and requires advanced navigation techniques.

Simultaneous Localization and Mapping (SLAM) in cave environments has been widely studied on Earth, mostly using high power consuming Light Detection and Ranging (LiDAR) devices and/or with offline solutions. To solve this problem in a spatial context, we must address the power limitation using passive sensors such as cameras. Therefore, we propose to discuss visual navigation in an underground context while minimizing the computational load, critical for spatial applications, as much as possible.

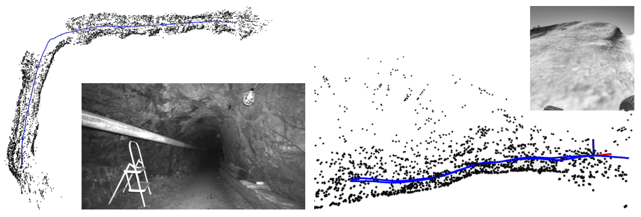

(a) (b)

Fig. 1: Our VO produced sparse maps and robot trajectories on real images from the OIVIO dataset (a) and on simulated planetary surfaces with fisheye by Gazebo (b).

VO systems have reached a high level of maturity with versatile and robust algorithms like ORB-SLAM3 [3]. However, there is a need to reach similar performances with limited computational resources. In [4], we introduced a low computational cost front-end design. In the present paper, we investigate how to limit computations on the back end of our bi-monocular VO by maintaining the size of the problem bounded while ensuring its sparsity, retaining most of the information of older measurements. To do so, when the boundary of our sliding window is reached, we proceed in two steps: we marginalize the variables that are no longer optimized, and we approximate the resulting prior into a set of sparse factors. This method, as detailed in [5], allows us to reduce the computational load of the sliding window optimization while keeping a fair amount of information from the past. We also propose to study the different factor graph topologies that can be implemented in the sparsification step. The contributions of this paper are as follows:

- A light VO system (Fig. 1) that minimizes the information loss.
- A detailed guideline about marginalization and sparsification for VO. Many implementation details of marginalization are often omitted in SLAM papers.
- A proposal and an experimental study of new factor graph topologies applied to a landmark-based SLAM with simulated image sequences for fair analysis.
- Our system is compared with state-of-the-art algorithms on real data recorded in challenging illumination conditions and reaches comparable accuracy results at a lower

*This work was supported by the CNES, Occitanie Region and the Consolidated Research Group RAIG (2021 SGR 00510) of the Departament de Recerca i Universitats de la Generalitat de Catalunya.

[1]César Debeunne and Damien Vivet are with ISAE-SUPAERO, University of Toulouse, France `firstname.lastname@isae-supaero.fr`

[2]Joan Vallvé is with the Institut de Robòtica i Informàtica Industrial (IRI), CSIC-UPC, Barcelona, Spain, `jvallve@iri.upc.edu`

[3]Alex Torres is with the CNES, France `alex.torres@cnes.fr`

computational cost.

II. Related Work

A. Cave Exploration

SLAM in underground conditions has been widely studied. For instance, Zlot *et al.* [6] proposed a large-scale SLAM for mapping a 17 km long underground mine with a 2D LiDAR and an industrial Inertial Measurement Unit (IMU) . Recently, the DARPA Subterranean challenge enabled a major demonstration of state-of-the-art multi-robot SLAM for underground exploration. However, all the proposed solutions were LiDAR-centric, implementing complex sensor fusion schemes and using powerful embedded computers. The conclusive survey [7] highlighted the need for research in low-cost navigation solutions with vision-based SLAM.

B. Visual SLAM

Visual SLAM or odometry (*i.e.*, without loop closure) has been widely studied in the past decade. Among state-of-the-art, two types of methods have emerged. On the one hand, feature-based (or indirect) methods rely on partial information about an image, like salient points or edges and estimate motion by minimizing geometric errors. Feature-based SLAM has been tackled with both filtering-based [8], and optimization-based methods [9]. But filtering methods have limitations in accuracy and robustness [10] while full batch optimization methods quickly become computationally intractable. The current state-of-the-art systems have been inspired by Structure From Motion algorithms and take advantage of both filtering and smoothing methods. This is the case of ORB-SLAM3 [3] that performs Bundle-Adjustement (BA) on a sliding window with a fixed number of frames. The system described in this paper is inspired by such methods, with contributions in information sparsification and computational cost.

On the other hand, direct methods perform tracking of pixels and pose estimation on the whole image by minimizing photometric errors. This is the case of LSD-SLAM [11], which produces semi-dense maps, and DSO [12], a similar algorithm using a sparse set of keypoints. Hybrid methods exist, like SVO [13] which initializes a direct estimator using a coarse feature-based pose estimation. This algorithm show impressive results for cameras with tedious photometric calibration on challenging scenarios for both modalities.

C. Marginalization and Sparsification in SLAM

We propose in this paper a complete guideline for marginalization and sparsification in a VO. Marginalization was first introduced for a spatial application in [14] to limit the information loss of a sliding window filter and maintain constant computational complexity. It is applied in the Visual Inertial Odometry (VIO) framework VINS-MONO [15] based on the tight fusion of pre-integrated IMU deltas and visual features. But marginalization impacts the sparse structure of the problem by introducing correlations between variables *i.e.*, adding "fill-in" in the information matrix of the SLAM. The sparsity of the SLAM problem is an important feature that allows the use of sparse linear algebra for efficient solving as in CERES [16], g2o [17], or ISAM2 [18].

Approximating a dense distribution with sparse factors has been studied first in the pose-SLAM community. For example, Generic Linear Constraint (GLC) [19] is a method that translates the marginalization into a set of linearized factors. Mazuran *et al.* [5] then proposed Non-Linear Factor Recovery (NFR) to approximate the dense distribution with custom non-linear factors that minimize the Kullback-Leibler divergence (KLD) w.r.t. the dense prior. It has been used for global mapping in [20] to transfer information from local VIO to global map optimization. NFR is also applied to a VIO in [21] to turn the dense prior into absolute factors for the robot and IMU states and pose to landmark factors for visual features. The proposed sparse topology was arbitrarily chosen because of its similarity in terms of information matrix entries with the dense prior. To our knowledge, this paper is the first to present a similar work on a VO with information analysis on the sparse topology design.

III. Notation

In this paper, the pose of a camera in the world frame is referred to as ${}^{w}\mathbf{T}_c \in SE(3)$. We parameterize landmark j with its 3D position in the world frame $\mathbf{l}_j^w \in \mathbb{R}^3$. We can compute the coordinates of the j-th landmark in the i-th camera frame with $\mathbf{l}_j^{c_i} = {}^{w}\mathbf{T}_{c_i}^{-1}\mathbf{l}_j^w = {}^{c_i}\mathbf{T}_w\mathbf{l}_j^w$. We note the projection function of a camera $\pi : \mathbb{R}^3 \longrightarrow \mathbb{R}^2$ that maps a 3D point in the camera frame in the 2D image $\mathbf{p}_{i,j} = \pi(\mathbf{l}_j^{c_i})$. We note abusively $\pi^{-1} : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ the function that computes the bearing vector of a given pixel $\mathbf{p}_{i,j}$ as $\pi^{-1}(\mathbf{p}_{i,j})$. We work with a set of keyframes in a sliding window W and each keyframe K_i has a set of map points L_i that can be seen from a set C_i of cameras. The extrinsic transformations between the set of cameras are assumed to be known. We concatenate all landmarks and keyframe poses in a state vector $\mathbf{x}$. Matrices are denoted with capital letters or greek capital letters when they have a probabilistic interpretation (*e.g.* Λ and Ω are information matrices and Σ is a covariance). Block of a matrix on a given subset of variables is noted with a subscript in parenthesis.

IV. Visual Odometry

This section describes the two main components of our bi-monocular VO algorithm. Indeed, two cameras are used here without any stereo rectification. The front-end deals with feature extraction, association and filtering, and the back-end controls the sliding window optimization.

A. Front-End

For each frame, we recover 2D point features by using a pyramidal Kanade Lucas Tracker (KLT). To improve the convergence of the KLT tracker, it is initialized with predicted pixel values for each 3D point from a constant velocity model. We provide for each frame an estimate of the pose using P3P [22] in a RANSAC fashion as a first outlier filtering. Then, a second filtering of the tracked features is done through an epipolar plane check that is independent of

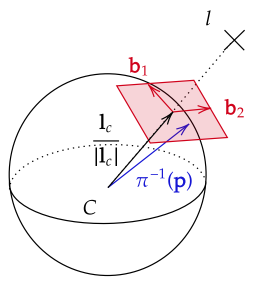


Fig. 2: Illustration of the visual residual.

the camera model. The pose is finally refined with a single-frame BA.

Keyframes are voted if the average parallax of the current set of features goes over a threshold to avoid redundant information. In this case, the tracked 2D points are associated on the second camera with a KLT and landmarks are triangulated. A BA is performed on current landmarks only, with a robust Huber norm to detect and eliminate spurious variables. This three-step outlier filtering ensures we only have inliers for smoothing. Finally, the keyframe is re-populated using the FAST keypoint detector coupled with a bucketing strategy. For further details, we refer the reader to [4].

B. Back-End

The back-end consists of a fixed lag smoothing optimizer. For a general formulation of our problem, a factor graph paradigm is used to define the optimization problem. $\mathcal{G}$ denotes the factor graph of our sliding window problem and is represented in figure 3. It contains the visual factors of every landmark $\mathbf{l}_j^w$ that is observed from every keyframe K_i in the window W through the set of cameras C_i. It also contains the prior factor on the subset of variables $\mathbf{x}_P$. The sliding window optimization consists in finding the optimal state $\mathbf{x}^*$ that minimizes the summation of the norm of the residuals, that are the errors of all factors weighted by their noises Σ_k

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left(\sum_{k \in \mathcal{G}} \|\mathbf{e}_k(\mathbf{x})\|^2_{\Sigma_k} \right). \tag{1}$$

We use the CERES library to solve this non-linear optimization problem, and the Jacobians were analytically derived according to [23]. In the following, visual and dense prior factors are introduced.

1) Visual Factor: The visual factor formulation is inspired by [15]. It is based on landmark projection on the unit sphere and doesn't require any camera model inversion, as represented in figure 2. The error of a visual observation of the landmark $\mathbf{l}_j^w$ from the i-th camera is

$$\mathbf{e}_V({}^w\mathbf{T}_{c_i}, \mathbf{l}_j^w) = [\mathbf{b}_1 \ \mathbf{b}_2]^T \left(\frac{\mathbf{l}_j^{c_i}}{|\mathbf{l}_j^{c_i}|} - \pi^{-1}(\mathbf{p}_{i,j}) \right), \tag{2}$$

with $\mathbf{b}_1$ and $\mathbf{b}_2$ arbitrary tangent vectors to the unit sphere around the projection point. The covariance of a visual observation is necessary to compute the associated residual; we set it to $\Sigma_V = I_2$.

2) Marginalization Factor: This VO adopts a marginalization scheme that generates a prior on the landmarks previously linked to the marginalized poses. This prior is dense in this section but will be made sparse in the next one. When the sliding window reaches its bound, the last frame and all the lonely landmarks (*i.e.* that are not linked to other frames) are marginalized. A dense prior is computed on all the landmarks that are linked to the last frame and not lonely.

To compute this prior, one needs to focus on the Markov Blanket (MB) of the last frame, that is, all the variables that are linked to the last frame via a factor (Fig. 3). The information matrix and the gradient of the problem on this MB are then computed

$$\Lambda = \sum_{k \in MB} J_k^T \Omega_k J_k, \tag{3}$$

$$\mathbf{g} = \sum_{k \in MB} J_k^T \Omega_k \mathbf{e}_k, \tag{4}$$

with $\mathbf{e}_k$ the error of the k-th factor, J_k its Jacobian w.r.t. the MB part of the state and $\Omega_k = \Sigma_k^{-1}$ the information matrix of the noise. In this operation, we consider all the factors in the MB, which are here: dense prior and visual factors. We note m, the subset of variables that will be marginalized and u, the variables of the MB that are not marginalized. We then use the Schur Complement to compute the information matrix and the gradient of the reduced problem, such as

$$\Lambda_P = \Lambda_{(u,u)} - \Lambda_{(u,m)} \Lambda^{-1}_{(m,m)} \Lambda^T_{(u,m)}, \tag{5}$$

$$\mathbf{g}_P = \mathbf{g}_{(u)} - \Lambda_{(u,m)} \Lambda^{-1}_{(m,m)} \mathbf{g}_{(m)}. \tag{6}$$

To translate this into a prior factor, we must compute the associated Jacobian J_P, the error $\mathbf{e}_P$ and the covariance Σ_P. However, we need to invert Λ_P to compute the prior but it can be rank deficient. To circumvent this, we use the rank revealing eigen decomposition to work on the subspace of variables where Λ_P can be inverted. We note U the eigenvectors that correspond to non zero eigenvalues and D the corresponding diagonal matrix of strictly positive eigenvalues. We can then compute

$$J_P = U^T, \tag{7}$$

$$\hat{\mathbf{e}}_P = D^{-1} U^T \mathbf{g}_P, \tag{8}$$

$$\Sigma_P = D^{-1}. \tag{9}$$

The prior will then have a lower or equal dimension than the number of parameters kept in the marginalization process. After removing the marginalized variables and the factors involving them, relinearization is not possible anymore. Instead, the remaining prior factor is linear, with its Jacobian evaluated at the state where the marginalization was performed. During optimization, the error is then updated using a first-order approximation. If we note $\hat{\mathbf{x}}_P$ the values of the states involved in the prior at marginalization time it gives

$$\mathbf{e}_P(\mathbf{x}_P) = \hat{\mathbf{e}}_P + J_P(\mathbf{x}_P - \hat{\mathbf{x}}_P). \tag{10}$$

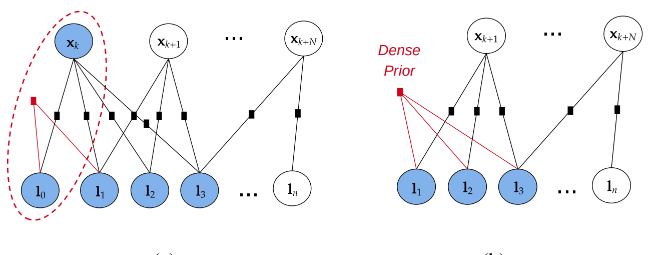


Fig. 3: (a) represents the full-size sliding window, the variables that need to be marginalized (dotted red) and the current prior (red). The states in the MB of this marginalization clique are represented in blue. (b) represents the effect of marginalization on the factor graph: the dense prior factor is updated.

V. Sparsification of the Dense Prior

The system described in the previous section is using a dense prior factor that affects the sparsity of the SLAM problem. To circumvent that, as proposed by [5] and [19], we approximate this dense prior factor as sparse factors.

A. Factor Recovery

We assume that we have defined a graph topology $\mathcal{T}$ that is a set of factors. We want to recover them *i.e.*, finding their error functions $\mathbf{e}_i$ and their information matrix Ω_i. We also define

$$J_S = \begin{bmatrix} \vdots \\ J_i \\ \vdots \end{bmatrix} \quad \Omega_S = \begin{bmatrix} \ddots & & 0 \\ & \Omega_i & \\ 0 & & \ddots \end{bmatrix},$$

where J_S is the stacked Jacobian of all the new factors and Ω_S all the information matrices of the new factors stored in a diagonal matrix. The sparse distribution is then defined by $\Lambda_S = J_S^T \Omega_S J_S$. These values have to be computed so that the new distribution $q(P) \sim \mathcal{N}(\boldsymbol{\mu}_S, \Sigma_S = \Lambda_S^{-1})$ is the most similar to the dense one $p(P) \sim \mathcal{N}(\boldsymbol{\mu}_P, \Sigma_P = \Lambda_P^{-1})$. The Kullback-Leibler Divergence (KLD) is usually computed to determine how much a distribution diverges from another

$$D_{KL}(p||q) = \frac{1}{2}\left(\mathrm{tr}(\Lambda_S \Sigma_P) - \ln|\Lambda_S \Sigma_P| + \|\boldsymbol{\mu}_S - \boldsymbol{\mu}_P\|^2_{\Sigma_S} - d\right) \tag{11}$$

where d is the dimension of the problem. The KLD between p and q has to be minimized for an optimal approximation. First, to set the mahalanobis term to zero, one need $\boldsymbol{\mu}_P = \boldsymbol{\mu}_S$ that is true when for each factor $\mathbf{e}_i(\boldsymbol{\mu}_P) = 0$. Then, according to Mazuran *et al.*, finding all the Ω_i that minimizes (11) is a convex problem whose global optimum can be computed. In the general case, a closed-form solution doesn't exist. Iterative methods like Interior Point or Factor Descent [24] can be used to reach the optimum numerically. However, such algorithms require too much computational load for our application.

For specific topologies, a closed-form solution can be computed. This is the case when the stacked Jacobian J_S is invertible. Such topologies are limited in terms of correlations and information encoding but they can be computed fast. Following [5], the solution can be obtained by

$$\Omega_i^* = (J_i \Sigma_P J_i^T)^{-1}, \forall i. \tag{12}$$

This paper details the implementation of two topologies whose solution to the KLD minimization can be reached with (12). These topologies on a toy example are represented in figure 4.

B. Absolute Landmark topology

The Absolute Landmark topology is only made of unary factors that observe the position of the landmarks of the MB in the world frame. This topology discards any correlation between landmarks, but it encodes absolute information.

The error and the Jacobian for the i-th factor are straightforward in this case

$$\mathbf{e}_U(\mathbf{l}_i^w) = \mathbf{l}_i^w - \hat{\mathbf{l}}_i^w, \; J_U = I_3,$$

with $\hat{\mathbf{l}}_i^w$ the landmark position at marginalization time. The stacked Jacobian is then the identity matrix of the dimension of our prior, (12) can be applied to recover Ω_i^*.

C. Landmark Tree topology

This topology contains a unary factor, that is necessary to ensure that the stacked Jacobian J_S is invertible by observing the absolute position of a landmark. The rest is made of relative translation factors between landmarks. The error and Jacobian of such a factor involving landmarks i and j are given by

$$\mathbf{e}_T(\mathbf{l}_i^w, \mathbf{l}_j^w) = (\mathbf{l}_i^w - \mathbf{l}_j^w) - (\hat{\mathbf{l}}_i^w - \hat{\mathbf{l}}_j^w), \; J_T = \begin{bmatrix} I_3 & -I_3 \end{bmatrix}.$$

The landmark that has a unary factor is the one in P with the lowest entropy

$$H(\mathbf{l}_i^w) = \log((2\pi e)^{\frac{n}{2}} |\Sigma_{P(ii)}|), \tag{13}$$

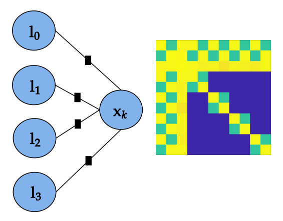
with n the dimension of the state.

1) Mutual Information Landmark Tree: To choose the correct landmark tree, we can apply the Chow-Liu tree (CLT) algorithm. It computes the Mutual Information (MI) between all pairs of landmarks

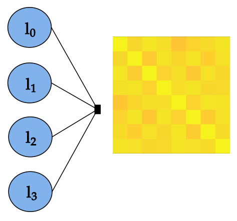
$$I(\mathbf{l}_i^w, \mathbf{l}_j^w) = \log \frac{|\Sigma_{P(ii)}||\Sigma_{P(jj)}|}{\begin{vmatrix} \Sigma_{P(ii)} & \Sigma_{P(ij)} \\ \Sigma_{P(ji)} & \Sigma_{P(jj)} \end{vmatrix}}, \tag{14}$$

and returns the spanning tree of the most correlated landmarks. This procedure ensures that we have the most informative topology possible. However, this decomposition strategy is computationally expensive: each mutual information derivation requires computing determinants.

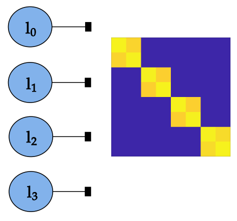
2) Off Landmark Tree: Off-diagonal entries of the dense information matrix can only be explained by a factor between the variables involved. In [24], the determinant of the off-diagonal blocks of Λ_P is used instead of the MI in CLT algorithm to build a topology. We explored which is the best metric for evaluating the importance of the off diagonal block including determinant, maximum absolute value, summation of absolute values and trace of absolute values. The trace



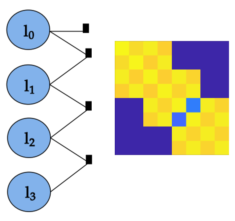
(a)



(b)



(c)



(d)

Fig. 4: This 2D toy problem illustrates the structure of our problem through the marginalization and sparsification steps with factor graphs and information matrices. The matrices' cells are displayed with a color map that represents 0 in marine blue. The state $\mathbf{x}_k \in SE(2)$, the landmarks have 2D world coordinates and the observation model is simply the 2D relative position of the landmark in the robot frame. This is equivalent in 3D to stereo observations. (a) represents the MB of $\mathbf{x}_k$ and its factors before marginalization. (b) shows the dense distribution after the marginalization of $\mathbf{x}_k$. (c) illustrates the absolute topology with only unary factors on landmarks, and so does (d) with the landmark tree topology. In both cases, the structure of the problem has changed, but it maintains a certain sparsity.

resulted to be the most efficient and provided the most informative topology

$$I_{\text{off}}(\mathbf{l}_i^w, \mathbf{l}_j^w) = \text{abs}(\text{tr}(\Lambda_{P(ij)})). \tag{15}$$

The CLT algorithm was adapted using this metric instead of the MI to return a landmark tree. A comparison between these two methods is provided in the next section.

D. Invertibility of Λ_P

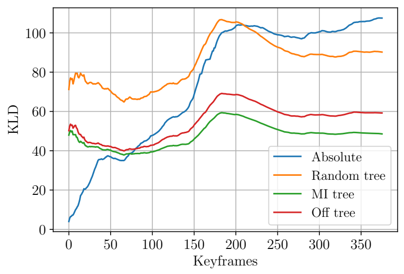
To ensure that our problem is invertible, we set a confident absolute pose prior in the beginning of the experiment and we only include observable landmarks (*i.e.* with more than one factor) in our marginalization scheme. However, even if it never occurred during the presented experiments, Λ_P may not invertible. In this case, we apply the rank revealing decomposition as in [5] and project the Jacobians of our sparse factors. In this case, (12) is not the closed-form solution of the KLD minimization anymore. We use this result as an approximation of the KLD minimization as iterative methods would be computationally prohibitive.

E. Reuse of dense prior

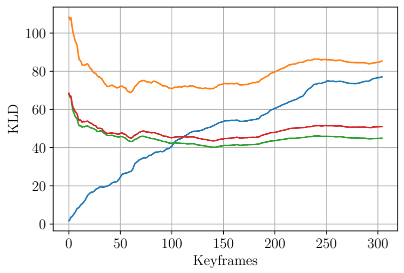
The marginalization (3) involves all factors in the MB. When sparsification is performed, the recovered factors will be eventually marginalized as well but the information that could not be encoded in the sparse set of recovered factors will be lost. To limit information loss, we propose to keep the dense prior factor obtained after the marginalization step in memory. Later when marginalizing again, this stored dense prior can be used instead of the sparse set of factors to compute (3). Doing this, the information loss due to sparsification is not propagated through time. Experimental validation of this choice is provided in the next section.

VI. Experiments

We first compare the different sparse topologies on simulated video sequences from Gazebo to limit the influence of noisy data in our results. Then, to validate the accuracy of our system on real trajectories under poor illumination conditions, we produced results on the OIVIO dataset [25]. The software has been developed in C++, and all experiments have been performed on a desktop station equipped with an Intel Core i7, 3.2 GHz clock rate.



(a)



(b)

Fig. 5: Plot of the average KLD for each keyframe on (a) the turn-back trajectory and (b) the straight line trajectory for all the topologies studied.

A. Topology study on a simulated dataset

We implemented four sparse topologies in our VO framework, and we studied them experimentally on a simulated dataset. The dataset is recorded with a pair of fisheye cameras on a planetary surface on the simulator Gazebo. Two scenarios were tested: a roundabout trajectory with many changes in the scene and many landmarks to marginalize, and a straight-line trajectory with fewer changes in the local map. After each marginalization step, the prior is sparsified with four different topologies and their KLD w.r.t. the dense distribution is calculated. This metric was chosen to determine which topology is the closest to the original prior, this is a more direct indicator than accuracy metrics.

	Turn Back					Straight Line				
Topology	Dense	Absolute	Random tree	MI tree	Off tree	Dense	Absolute	Random tree	MI tree	Off tree
KLD optim (ms)	0	0.54	0.51	1.63	1.00	0	0.28	0.28	1.11	0.56
BA optim (ms)	6.50	4.88	4.92	5.30	4.98	6.02	4.5	4.75	4.80	4.96

TABLE I: Run time analysis on the simulated dataset for all the topologies

	Ours, Off tree (Bi-mono)			DSO (Mono)			VINS-Fusion (Bi-mono)			ORBSLAM3 (Bi-mono)		
Scenario	ATE(m)	RTE(m)	dt(ms)	ATE(m)	RTE(m)	dt(ms)	ATE(m)	RTE(m)	dt(ms)	ATE(m)	RTE(m)	dt(ms)
MN 015 GV1	**0.08**	**0.03**	**9.9**	0.36	0.08	44	0.16	0.04	22	0.11	0.04	25
MN 050 GV1	0.13	0.05	**9.9**	0.75	0.07	52	0.25	**0.04**	21	**0.12**	**0.04**	26
MN 100 GV1	**0.14**	**0.04**	**9.4**	1.18	0.07	57	0.16	0.05	21	**0.14**	0.05	25
MN 015 GV2	**0.10**	**0.03**	**9.5**	0.38	0.05	46	0.18	**0.03**	24	0.11	**0.03**	25
MN 050 GV2	**0.10**	**0.03**	**9.7**	0.58	0.05	56	0.12	**0.03**	23	**0.10**	**0.03**	25
MN 100 GV2	**0.09**	0.04	**9.4**	0.4	0.05	62	**0.09**	**0.03**	24	0.10	**0.03**	25
TN 015 GV1	0.17	0.07	**11**	0.46	0.09	43	0.21	0.06	21	**0.12**	**0.04**	24
TN 050 GV1	0.18	**0.05**	**10**	1. 16	0.07	51	0.28	**0.05**	21	**0.15**	**0.05**	25
TN 100 GV1	0.15	0.08	**9.8**	0.53	0.07	53	0.19	**0.05**	21	**0.10**	**0.05**	24

TABLE II: Performances on OIVIO Dataset: the best results for each metric are displayed in bold.

By default, the dense factor is used in the sliding window optimization. The absolute topology is computed, as well as three different landmark trees. The random tree is the naive landmark tree with arbitrary links, the MI tree is built with a CLT, and the Off tree is using the trace of off-diagonal blocks in Λ_P.

The comparison of the average KLD of each topology w.r.t. the dense distribution on the two simulated trajectories is available in figure 5. The absolute topology seems to better represent the dense distribution at the beginning of the course, while the most informative landmark trees work better in the long run. We offer an interpretation of this phenomenon. In the beginning, the initial pose prior is responsible for most of the information in the dense prior, and such information is more accurately encoded in absolute factors. As the experiment goes by, the relative position of the landmarks is better encoded in the problem with visual factors. Such information is more completely represented by landmark-to-landmark factors. Then, the absolute topology is less and less accurate as more relative information is propagated in the prior. We also observe that the random tree is logically markedly less representative than the two other similar topologies as it randomly links landmarks that can be poorly correlated. It seems that the off-diagonal topology is not notably less informative than the MI one while it diminishes the computational load, as shown in table I.

The run time analysis was performed by running the VO with each factor topology on the two trajectories. We evaluated the average sparsification time (including the topology building and the factor recovery) and run time of the sliding window optimization on the whole trajectory. The sparse topologies reduce greatly the optimization time compared to the dense one. The absolute topology and the random tree are computed the fastest since they do not calculate information metrics, only factor recovery. The MI tree is more costly than the off-diagonal topology by a non-negligible factor: it is two times slower on the straight-line trajectory. In conclusion, the off-landmark tree topology offers a good trade-off between computational load and similarity to the dense distribution.

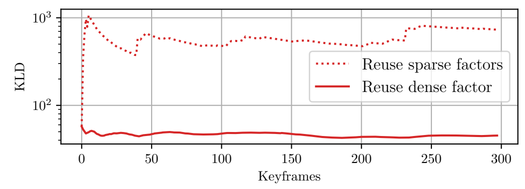


Fig. 6: Comparison between the reuse of the dense prior and the use of sparse factors to compute Λ on the straight line trajectory.

B. *Reuse of previous dense prior factor*

We experimentally demonstrate the choice of reusing the prior using the dense factor instead of the set of approximated sparse factors as described in V-E. The two marginalization and sparsification methods were run in parallel on the straight-line trajectory with the off diagonal topology. The KLD w.r.t. the original dense distribution (*i.e.*, the one obtained with dense prior propagation) is computed in both cases in figure 6. The benefits of reusing the previous dense factor instead of the sparse set of factors while marginalizing again is significant from the second keyframe. This happens because even though the sparsification could not encode all the information of the marginalized observations, this information can be recovered when marginalizing. If the dense prior is not reused, the information not encoded during sparsification is lost and discarded forever.

C. *Validation on the OIVIO dataset*

This dataset was recorded on a rover, in dark environments like mines or tunnels, with onboard illumination. The ground truth comes from a laser tracker, and three illumination levels were tested (1500, 5000 and 10000 lumens). Our system is compared to three state-of-the-art Visual SLAM algorithms: DSO [12], VINS-Fusion [15] and ORB-SLAM3 [3]. DSO is one of the references in direct VO which is based on photometric error minimization. It reaches real-time operation by sampling pixels with high-intensity gradients instead of using the whole image for motion estimation. Its precise photometric model takes into account vignetting and inverse response functions of the pixels whose data are provided in the OIVIO dataset. Note that the comparison is not really fair as DSO is a monocular algorithm where the scale is unknown. VINS-Fusion is an extension of VINS-Mono for a wider range of sensor setups: the bi-mono configuration is tested here. The comparison is interesting as it uses a front-end based on feature tracking and it performs the naive marginalization approach in its optimizer. ORB-SLAM3 is considered the state-of-the-art Visual SLAM system, its indirect approach is based on descriptor matching with ORB features and it provides long-term data association with loop closure and relocalization. Our solution is using sparsification of the dense prior with the dense factor reusing strategy and the Off tree topology.

We computed the Absolute Trajectory Error (ATE) in meters, the Relative Translation Error (RTE) in meters and the average run time per frame (dt) in milliseconds as metrics in table II. DSO is less accurate than indirect methods on this benchmark. Direct methods may suffer from extreme lighting conditions due to onboard illumination. Our method performs slightly better than VINS-Fusion and similarly to ORB-SLAM3 on this dataset. Our solution runs at 100Hz on average on our setup, being the lightest solution here. The two main differences between our algorithm and VINS-Fusion are the careful design of our front-end [4] that may explain the higher accuracy and the sparsification of the prior that leads to a faster run-time. ORB-SLAM3 seems to perform better on the tunnel scenarios. The latter shows the strongest illumination changes, which may impact more a solution based on tracking than ORB-SLAM3 which is based on descriptor matching.

VII. Conclusion

This paper describes a VO algorithm that offers a compromise between accuracy and computational complexity via information sparsification. In our sliding window approach, we approximate the dense prior into a set of sparse factors for optimization but keep it in memory for information propagation. The topology of the sparse prior was selected following a careful experimental study on simulated data. Our method achieves state-of-the-art accuracy at a lower run time on a public dataset that exhibits extreme illumination conditions that may be found in Lava Tubes. Our future work will focus on acquiring our own dataset with fisheye cameras and generating traversability information with a 3D mesh of our sparse map.

References

[1] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proceedings, IEEE Aerospace Conference*, vol. 5, 2002, pp. 2025–2036.

[2] A. Johnson, S. Aaron, H. Ansari, C. Bergh, H. Bourdu, J. Butler, J. Chang, R. Cheng, Y. Cheng, K. Clark, D. Clouse, R. Donnelly, K. Gostelow, W. Jay, M. Jordan, S. Mohan, J. Montgomery, J. Morrison, S. Schroeder, and J. Zheng, "Mars 2020 lander vision system flight performance," in *AIAA SciTech 2022 Forum*, 2022, pp. 1214–1234.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. 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.

[4] C. Debeunne, D. Vivet, and A. Torres, "Design of a bi-monocular Visual Odometry System for Lava Tubes exploration," in *PNARUDE Workshop 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[5] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term SLAM," *The International Journal of Robotics Research*, vol. 35, pp. 50–72, 2015.

[6] R. Zlot and M. Bosse, "Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine," in *Field and service robotics*, 2014, pp. 479–493.

[7] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, L. Nogueira, M. Palieri, P. Petráček, M. Petrlík, A. Reinke, V. Krátký, S. Zhao, A.-a. Agha-mohammadi, K. Alexis, C. Heckman, K. Khosoussi, N. Kottege, B. Morrell, M. Hutter, F. Pauling, F. Pomerleau, M. Saska, S. Scherer, R. Siegwart, J. L. Williams, and L. Carlone, "Present and future of SLAM in extreme underground environments," 2022. [Online]. Available: https://arxiv.org/abs/2208.01787

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[9] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, p. 652—659.

[10] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2657–2664.

[11] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, 2014, p. 834–849.

[12] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, 2017, pp. 611–625.

[13] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation*, 2014, pp. 15–22.

[14] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.

[15] 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.

[16] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[17] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[18] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3281–3288.

[19] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic Node Removal for Factor-Graph SLAM," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1371–1385, 2014.

[20] 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, pp. 422–429, 2020.

[21] 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*, 2018, pp. 1146–1153.

[22] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2969–2976.

[23] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," 2018.

[24] J. S. Joan Vallvé and J. Andrade-Cetto, "Pose-graph slam sparsification using factor descent," *Robotics and Autonomous Systems*, vol. 119, pp. 108–118, 2019.

[25] M. Kasper, S. McGuire, and C. Heckman, "A Benchmark for Visual-Inertial Odometry Systems Employing Onboard Illumination," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.