

# Factor descent optimization for sparsification in graph SLAM

Joan Vallvé, Joan Solà, Juan Andrade-Cetto

**Abstract**—In the context of graph-based simultaneous localization and mapping, node pruning consists in removing a subset of nodes from the graph, while keeping the graph’s information content as close as possible to the original. One often tackles this problem locally by isolating the Markov blanket sub-graph of a node, marginalizing this node and sparsifying the dense result. It means computing an approximation with a new set of factors. For a given approximation topology, the factors’ mean and covariance that best approximate the original distribution can be obtained through minimization of the Kullback-Liebler divergence. For simple topologies such as Chow-Liu trees, there is a closed form for the optimal solution. However, a tree is oftentimes too sparse to explain some graphs. More complex topologies require nonlinear iterative optimization. In the present paper we propose Factor Descent, a new iterative optimization method to sparsify the dense result of node marginalization, which works by iterating factor by factor. We also provide a thorough comparison of our approach with state-of-the-art methods in real world datasets with regards to the obtained solution and convergence rates.

## I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is the problem of building a representation of the environment while getting localized in it. Without any strategy to face it, the longer the experiment, the larger the problem to solve. Efforts to reduce resource demands have been focused mainly in two directions: by facing the computational complexity of the algorithms, or by tackling the problem size. Even though several improvements have been made in the first direction [1, 2], the later is still of concern, as the solution is always linked to the length of the experiment. This claims for suboptimal strategies that reduce problem size but still keep as much information as possible.

Several SLAM methods include mechanisms to limit problem size growth. One of the simplest approaches consists in considering a temporal or spatial window and discarding the landmarks and/or poses that lie outside this window. This implies giving up loop closures, such as in visual and visual-inertial odometry methods [3, 4]. In Pose SLAM [5], new observations and robot poses are only added to the problem if their entropy-based information content is significant. In contrast, in [6] a hierarchical graph structure is devised. The graphs higher up in the hierarchy represent marginalized sub-graphs and if the error is low enough, only part of the problem is solved. Johannsson et al. [7] propose a reduced pose graph that only grows with the size of the environment being mapped by marginalizing poses with regards to distance. Chouldhary et al. [8] on the other hand propose an information-based reduced landmark SLAM system. The method proposes a trade off between

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain. {jvallve, jsola, cetto}@iri.upc.edu.

This work has been supported by the Spanish Ministry of Economy and Competitiveness under Project ROBINSTRUCT (TIN2014-58178-R), by the EU H2020 Project LOGIMATIC (H2020-Galileo-2015-1-687534) and by the María de Maeztu Seal of Excellence to IRI MDM-2016-0656.

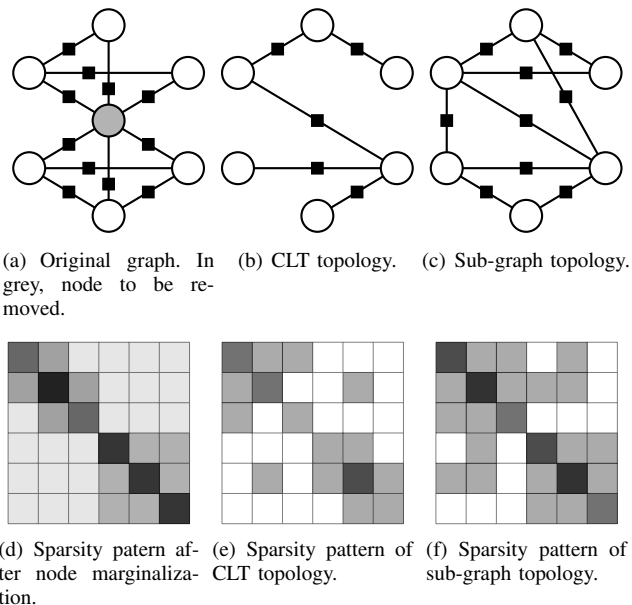


Fig. 1. Tree topology can be too sparse to accurately approximate the dense distribution resulting of a node marginalization.

memory footprint and accuracy using an entropy-based cost function to decide which landmarks should be discarded.

One important characteristic of SLAM is sparsity: the network of geometrical constraints corresponding to sensor measurements between robot trajectory and/or environment representation is only a small subset of all the possibilities. Graph-SLAM methods take profit of this sparsity to speed up the computation of the optimal solution.

Moreover, SLAM is a non-linear problem that is faced by linearizing it. The capability of relinearization greatly improves the accuracy of the solution.

Normally, the only way of reducing the problem size without loss of information is marginalization. However, marginalization induces fill-in, increasing computational cost, and does not allow for relinearization, deriving in accuracy loss.

Several works have been published with methods for finding the best sparse and relinearizable approximation of the dense and not-relinearizable result of a marginalization. This is known as sparsification, and is the focus of the present paper. Kretzschmar and Stachniss [9] present an information-theoretic compression method for pose graph SLAM that selects the nodes containing the most informative laser scans. They find the subset of measurements that maximize the mutual information of the map for that subset. More recently, [10]–[12] approach the problem by finding the sparse approximation that minimizes the Kullback-Liebler divergence (KLD) with the dense distribution resulting from the node marginalization. While there is a closed form for the simplest topology, *e.g.* the Chow-Liu tree (CLT), an iterative optimization is needed for richer topologies.

In the majority of cases, and as we reveal in the experimental section, a tree topology is too simple to accurately approximate the dense result of a node marginalization (see Fig.1). Hence, in this paper we focus on iterative optimization for sparsification. We introduce Factor Descent optimization for sparsification. Given a non-dense factor topology, we iteratively optimize each of the factors leaving fixed the rest. For each factor, we compute its parameters (mean and information matrix) that minimize the KLD given the rest of topology factors' parameters.

The paper is organized as follows. The next section includes the problem formulation and existing methods. Section III presents our novel factor descent method. Section IV presents the results, and conclusions and future work are exposed in the last section.

## II. NODE REMOVAL AND SPARSIFICATION IN GRAPH SLAM

Graph-based SLAM methods represent the problem as a set of variables (nodes) and a set of geometrical constraints (factors). The state  $\mathbf{x}$  includes nodes representing poses of the vehicle along its trajectory and/or some map representation. Each factor expresses the discrepancy or error  $\mathbf{e}$  between a measurement  $\mathbf{z}$  and its expectation,

$$\mathbf{e}(\mathbf{x}) = h(\mathbf{x}) - \mathbf{z} + \mathbf{v}, \quad \mathbf{v} \sim \mathcal{N}(0, \mathbf{\Omega}^{-1}) \quad (1)$$

being  $h(\mathbf{x})$  the sensor's measurement model and  $\mathbf{\Omega}$  the information matrix of the measurement Gaussian noise  $\mathbf{v}$ .

The problem is solved iteratively by minimizing the Mahalanobis squared norm of all linearized errors

$$\Delta \mathbf{x}^* = \arg \min_{\Delta \mathbf{x}} \sum_k \|h_k(\mathbf{x}) - \mathbf{z}_k + \mathbf{J}_k \Delta \mathbf{x}\|_{\mathbf{\Omega}_k^{-1}}^2 \quad (2)$$

being  $\mathbf{x}$  the state estimate at the current iteration, and  $\mathbf{J}_k$  the Jacobian of the  $k$ -th measurement.<sup>1</sup> Imposing null derivative of the cost in (2) w.r.t  $\Delta \mathbf{x}$ , the optimal step  $\Delta \mathbf{x}^*$  is found and used to update the estimate. Current methods for solving for  $\Delta \mathbf{x}^*$  use Cholesky [2, 14, 15] or QR [1, 16, 17] matrix factorizations. Important speed-ups are obtained with incremental methods [1, 2, 15, 17], which update the problem directly on the factorized matrix.

Reducing the problem size in graph SLAM is usually approached in two steps: node marginalization and sparsification (see Fig. 2). These two stages do not necessarily have to be immediately consecutive, and the second one can be postponed depending on computational availability [11].

Having selected a node to prune (Fig. 2.a), the process is faced locally. A local problem around the node (Fig. 2.b) is defined by cropping the node's Markov blanket (all nodes at distance 1) and all its intra-factors (the factors involving only nodes in the Markov blanket). Optionally, this cropped problem can be solved. Then, all factors can be relinearized using the new solution before proceeding, yielding slightly better results especially in on-line cases [12].

After that, marginalization of the selected node is performed via Schur complement. This marginalization can be

<sup>1</sup>In case of manifolds, (1) and the squared Mahalanobis norm in (2) become  $\mathbf{e}(\mathbf{x}) = h(\mathbf{x}) \ominus \mathbf{z} \oplus \mathbf{v}$  and  $\|h_k(\mathbf{x}) \ominus \mathbf{z}_k + \mathbf{J}_k \Delta \mathbf{x}\|_{\mathbf{\Omega}_k^{-1}}^2$ , respectively, with  $\mathbf{J}_k = \partial(h_k(\mathbf{x}) \ominus \mathbf{z}_k) / \partial \Delta \mathbf{x}$ . The  $\oplus$  and  $\ominus$  are the addition and subtraction operators on the manifold, as described in [13].

understood as adding a dense factor (Fig. 2.c) that substitutes all intra-factors that involve the removed node. This new dense factor has no measurement model associated to; hence, its error cannot be re-evaluated, and re-linearization is not possible.

The goal of the sparsification process is to approximate the dense distribution  $p(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , resulting from node marginalization, with a sparse distribution  $q(\mathbf{x}) \sim \mathcal{N}(\check{\boldsymbol{\mu}}, \check{\boldsymbol{\Sigma}})$  defined by a new set of (relinearizable) factors (Fig. 2.d). This is usually split in two phases: building a topology (i.e. define a set of factors with their measurement model) and computing their mean and information that best approximate the original distribution.

### A. Topology

The topology defines the arrangement between the Markov blanket nodes and the new set of factors, each factor with a measurement model. Typically, the factors are made up of relative measurements between pairs of nodes. The simplest topology using relative measurements is a spanning tree. The Chow-Liu tree (CLT) defines a tree topology with factors between the most correlated nodes (i.e. the ones with most mutual information).

However, even taking the maximum mutual informative factors, a tree topology is usually too sparse to approximate the original distribution. For this reason, the so-called sub-graph topology departs from the CLT and adds (a few) more factors, also based on the nodes' mutual information [12]. Alternatively, the cliquy topology [12] takes the CLT and converts pairs of independent factors into one single factor by correlating them.

Differently to the CLT-based methods, a  $\ell_1$ -regularized KLD minimization can be used to compute the topology that will encode the most information [11].

### B. Sparsification through KLD minimization

Given the topology, we want to compute its factors' means  $\check{\mathbf{z}}_k$  and information  $\check{\mathbf{\Omega}}_k$  that minimize the KLD between the dense  $p(\mathbf{x})$  and sparse  $q(\mathbf{x})$  distributions. This can be posed as

$$D_{KL} = \frac{1}{2} \left( \langle \check{\boldsymbol{\Lambda}}, \boldsymbol{\Sigma} \rangle - \ln |\check{\boldsymbol{\Lambda}} \boldsymbol{\Sigma}| + \|\check{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_{\check{\boldsymbol{\Lambda}}^{-1}}^2 - d \right), \quad (3)$$

where  $\langle \cdot, \cdot \rangle$  denotes the matrix inner product and  $\check{\boldsymbol{\Lambda}} = \check{\boldsymbol{\Sigma}}^{-1}$  is the information matrix of  $q(\mathbf{x})$ .

This expression can be minimized as follows. The dimension  $d$  of both distributions and  $\boldsymbol{\Sigma}$  are constant w.r.t the information of each measurement  $\check{\mathbf{\Omega}}_k$ . The squared norm term  $\|\check{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_{\check{\boldsymbol{\Lambda}}}^2$  is null if the means of all measurements are set using the dense distribution mean  $\check{\mathbf{z}}_k = h_k(\boldsymbol{\mu})$ . Then, introducing the block diagonal matrix  $\check{\boldsymbol{\Omega}} = \text{diag}(\check{\mathbf{\Omega}}_1 \dots \check{\mathbf{\Omega}}_k \dots)$  containing all new factors' information matrices, and the Jacobian  $\check{\mathbf{J}} = [\check{\mathbf{J}}_1^T \dots \check{\mathbf{J}}_k^T \dots]^T$  stacking all new factors' Jacobians, the sparse information matrix of the approximate distribution is  $\check{\boldsymbol{\Lambda}} = \check{\mathbf{J}}^T \check{\boldsymbol{\Omega}} \check{\mathbf{J}}$ . Considering the above, the factors' information that minimize the KLD in (3), can be written as the constrained problem

$$\begin{aligned} \check{\boldsymbol{\Omega}}^* &= \arg \min_{\check{\boldsymbol{\Omega}}} \langle \check{\mathbf{J}}^T \check{\boldsymbol{\Omega}} \check{\mathbf{J}}, \boldsymbol{\Sigma} \rangle - \ln |\check{\mathbf{J}}^T \check{\boldsymbol{\Omega}} \check{\mathbf{J}}| \\ &\text{s.t. } \check{\boldsymbol{\Omega}} \in \mathcal{D}, \check{\boldsymbol{\Omega}} \succ 0 \end{aligned} \quad (4)$$

where  $\mathcal{D}$  refers to the set of block-diagonal matrices.

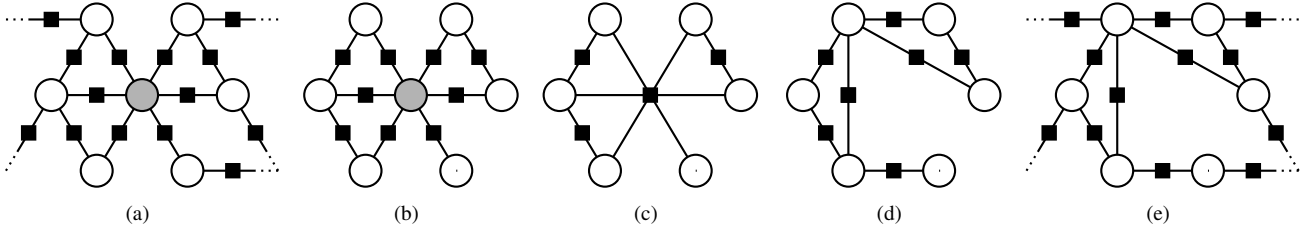


Fig. 2. Example of a pruning sequence of actions. The removed node is depicted in grey. (a) Initial graph. (b) Markov blanket and intra-factors are kept. (c) Node marginalization produces a dense factor (the central factor). (d) Sparsification computes an approximation with a set of new factors. (e) Substitution of the sparse approximation into the initial graph.

In some cases such as dense problems with only relative measurements, the dense problem has a rank-deficient information matrix  $\mathbf{\Lambda}$ , and the covariance matrix  $\mathbf{\Sigma}$  is not defined. In that case, we can apply a projection  $\mathbf{\Lambda} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$  such that  $\mathbf{D}$  is invertible. Then, all formulation derived from (4) holds by substituting

$$\begin{aligned} \check{\mathbf{J}} &\mapsto \check{\mathbf{J}}\mathbf{U} \\ \mathbf{\Sigma} &\mapsto \mathbf{D}^{-1}. \end{aligned} \quad (5)$$

This projection can be obtained by re-parametrizing the problem to relative poses w.r.t an arbitrarily chosen node [10, 11] or using a rank-revealing eigen decomposition [12].

### C. Sparsification in closed form

Certain topologies admit a closed form solution to (4). When  $\check{\mathbf{J}}$  is invertible, imposing null derivative of (4) w.r.t. all factor information matrices yields

$$\check{\mathbf{\Omega}}_k = (\check{\mathbf{J}}_k \mathbf{\Sigma} \check{\mathbf{J}}_k^\top)^{-1}. \quad (6)$$

This is the case of the tree topology in SLAM of relative measurements, using a projection as (5). However, and as has been said, this topology can be too sparse to accurately approximate the exact dense distribution. Also, (6) holds for the cliquy topology. However, it can carry convergence issues to the SLAM problem solution [12]. Moreover, the cliquy forms break the homogeneity of factors, which is valuable in many cases.

### D. Sparsification via iterative optimization

Other topologies with non-invertible Jacobian  $\check{\mathbf{J}}$  do not admit a closed form solution and the problem (4) has to be solved using iterative optimization. The state-of-the-art literature proposes two different optimization algorithms: Interior Point (IP) and Limited-memory Projected Quasi-Newton (PQN) [18]. IP includes the constraint of positive definiteness of the solution in the cost function

$$\langle \check{\mathbf{J}}^\top \check{\mathbf{\Omega}} \check{\mathbf{J}}, \mathbf{\Sigma} \rangle - \ln |\check{\mathbf{J}}^\top \check{\mathbf{\Omega}} \check{\mathbf{J}}| - \rho \ln |\check{\mathbf{\Omega}}|. \quad (7)$$

The log barrier parameter  $\rho$  is iteratively decreased towards 0. It requires an initial guess for  $\check{\mathbf{\Omega}}$  that strictly accomplishes the positive definite restriction, such as the identity matrix [12]. IP has quadratic convergence, but requires the (costly) computation of the Hessian and gradient of (7). A stricter constraint can be applied in (7) instead of the log barrier term to guarantee the conservativeness:  $\rho \ln |\mathbf{\Lambda} - \check{\mathbf{J}}^\top \check{\mathbf{\Omega}} \check{\mathbf{J}}|$  [11].

On the contrary, PQN does not require the computation of the Hessian (it still needs the gradient), nor a feasible initial guess. The positive definiteness constraint is accomplished

through the projection  $\mathcal{P}(\check{\mathbf{\Omega}})$  onto the positive semi-definite subspace, by setting all negative eigenvalues to zero,

$$\mathcal{P}(\check{\mathbf{\Omega}}) = \mathbf{V} \text{diag}(\max\{0, \lambda_i\}) \mathbf{V}^\top, \quad (8)$$

being  $\check{\mathbf{\Omega}} = \mathbf{V} \text{diag}(\lambda_i) \mathbf{V}^\top$  the eigen decomposition. PQN has a slower convergence than IP, but it can be initialized closer to the optimal solution using an initial guess based on the off-diagonal blocks of the dense information matrix, as proposed in [19],

$$\check{\mathbf{\Omega}}_k = \mathbf{J}_{k_1}^{-\top} \mathbf{\Lambda}_{k_1, k_2} \mathbf{J}_{k_2}^{-1} \quad (9)$$

being  $k_1, k_2$  the two nodes involved in the factor  $k$  (so  $\mathbf{J}_{k_1}, \mathbf{J}_{k_2}$  are the non-zero blocks of  $\mathbf{J}_k$ ) and being  $\mathbf{\Lambda}_{k_1, k_2}$  the off-diagonal block corresponding to the involved nodes. Such initial guess is normally not symmetric nor positive semi-definite, and one usually takes its closest symmetric positive semi-definite approximation [20]. Since this may result in a semi-definite positive guess (or close), it cannot be used in the IP method.

## III. FACTOR DESCENT OPTIMIZATION FOR SPARSIFICATION

We propose Factor Descent sparsification (FD), a novel optimization approach for solving (4) that takes inspiration in coordinate descent optimization. FD is a cyclic block-coordinate descent method; each step of the cycle consists in solving for a (small) block of variables (those defining one factor's information matrix  $\check{\mathbf{\Omega}}_i$ ) while fixing the rest.

Consider a given topology and an initial guess  $\check{\mathbf{\Omega}}$ . The derivative of (4) w.r.t the  $i$ -th factor's information matrix  $\check{\mathbf{\Omega}}_i$  is

$$\frac{\partial D_{KL}}{\partial \check{\mathbf{\Omega}}_i} = \check{\mathbf{J}}_i \mathbf{\Sigma} \check{\mathbf{J}}_i^\top - \check{\mathbf{J}}_i (\check{\mathbf{Y}}_i + \check{\mathbf{J}}_i^\top \check{\mathbf{\Omega}}_i \check{\mathbf{J}}_i)^{-1} \check{\mathbf{J}}_i^\top, \quad (10)$$

where  $\check{\mathbf{Y}}_i$  is the information matrix of the problem considering only the rest of factors,

$$\check{\mathbf{Y}}_i = \sum_{j \neq i} \check{\mathbf{J}}_j^\top \check{\mathbf{\Omega}}_j \check{\mathbf{J}}_j. \quad (11)$$

Imposing null derivative and applying the Woodbury matrix identity twice,<sup>2</sup> we get the closed form,

$$\check{\mathbf{\Omega}}_i = \underbrace{(\check{\mathbf{J}}_i \mathbf{\Sigma} \check{\mathbf{J}}_i^\top)^{-1}}_{\Phi_i} - (\check{\mathbf{J}}_i \check{\mathbf{Y}}_i^{-1} \check{\mathbf{J}}_i^\top)^{-1}. \quad (12)$$

This is the optimal  $i$ -th factor's information matrix in terms of KLD if the rest of factors are fixed.

Descent of the full KLD cost of (4) is achieved factor by factor, and hence the Factor Descent name. This can be

<sup>2</sup>applying the Woodbury matrix identity forwards and backwards from  $D(A+BCD)^{-1}B = D(A^{-1} - A^{-1}B(C^{-1} + DA^{-1}C)^{-1}DA^{-1})B = DA^{-1}B - DA^{-1}B(C^{-1} + DA^{-1}C)^{-1}DA^{-1}B = ((DA^{-1}B)^{-1} + C)^{-1}$

---

**Algorithm 1** Factor descent sparsification

---

**Input:** Dense mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$ , topology  $\mathcal{Z}$ 

// Precompute constant variables

**for**  $\mathbf{z}_i \in \mathcal{Z}$  **do** $\mathbf{J}_i \leftarrow \text{evaluateJacobian}(\mathbf{z}_i, \boldsymbol{\mu})$  $\boldsymbol{\Phi}_i \leftarrow (\check{\mathbf{J}}_i \boldsymbol{\Sigma} \check{\mathbf{J}}_i^\top)^{-1}$ **end for** $i := 1$ **while not**  $\text{endConditions}()$  **do**

// Compute the information of the rest of factors

 $\check{\mathbf{Y}}_i \leftarrow \sum_{j \neq i} \check{\mathbf{J}}_j^\top \check{\boldsymbol{\Omega}}_j \check{\mathbf{J}}_j$ //  $i$ -th factor descent $\check{\boldsymbol{\Omega}}_i \leftarrow \boldsymbol{\Phi}_i - (\check{\mathbf{J}}_i \check{\mathbf{Y}}_i^{-1} \check{\mathbf{J}}_i^\top)^{-1}$ 

// Ensure positive semi-definite solution

**if**  $\check{\boldsymbol{\Omega}}_i \prec 0$  **then** $\mathbf{V}, \boldsymbol{\lambda} \leftarrow \text{eigenDecomposition}(\check{\boldsymbol{\Omega}}_i)$  $\check{\boldsymbol{\Omega}}_i \leftarrow \mathbf{V} \text{diag}(\max(0, \boldsymbol{\lambda})) \mathbf{V}^\top$ **end if**

// Cycle for all factors

 $i++$ **if**  $i > N$  **then** $i := 1$ **end if****end while**

---

iterated as many times as desired. While the second term of (12) should be computed at each iteration, the first one  $\boldsymbol{\Phi}_i$  is constant for each factor and should be computed only once. The method is described in algorithm 1. The first term  $\boldsymbol{\Phi}_i$  can be interpreted as the information of the dense exact distribution projected in the measurement space of the  $i$ -th factor. Analogously, the second term is the projection of the information of the rest of the factors onto the measurement space of the  $i$ -th factor.

We want to emphasize that (12) is a generalization of (6). The conditions in which (6) is applicable are the same in which the second term in (12) is null. For example, in the tree topology, the projection of the information of the rest of the factors to each factor's measurement space is null.

As in other methods, in case of rank-deficient  $\boldsymbol{\Lambda}$ , the method holds using the projection (5). Note that the rest of new factors must be projected too, with  $\check{\mathbf{Y}} \mapsto \mathbf{U}^\top \check{\mathbf{Y}} \mathbf{U}$ .

Since the optimal solution in the factor's subspace is computed in closed form (12), iterations in FD refer to the fact that we iterate on the factors, not on finding the optimal for each factor through repeated linearizations. In other words, a clear benefit of FD is that there is no fitting to any linear or quadratic function. The convergence rate mainly depends on how much the direction to the optimal solution is aligned with the sub-spaces corresponding to each factor (see Fig. 3 for an illustrative example).

#### A. Positive-definiteness

It follows from (12) that  $\check{\boldsymbol{\Omega}}_i$  is positive definite only if

$$(\check{\mathbf{J}}_n \boldsymbol{\Sigma} \check{\mathbf{J}}_n^\top)^{-1} \succ (\check{\mathbf{J}}_n \check{\mathbf{Y}}^{-1} \check{\mathbf{J}}_n^\top)^{-1}. \quad (13)$$

This happens when the projection of the information of the dense distribution to the measurement space is 'bigger' than that of the rest of the factors. A zero eigenvalue of  $\check{\boldsymbol{\Omega}}_i$  implies that the rest of new factors already explain completely the original distribution in some direction. Further, a negative eigenvalue implies that the approximation is not

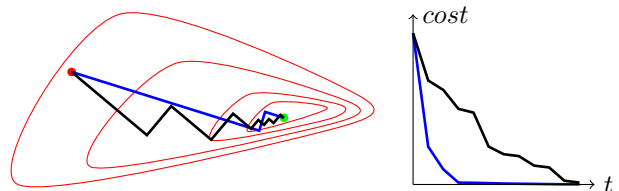


Fig. 3. Examples of convergence of the cyclic coordinate descent optimization for two different alignments of the coordinates w.r.t. the direction to the optimal solution (green dot).

conservative (without considering the  $i$ -th factor) and the optimal factor would subtract this excess of information. After each iteration, we impose positive semi-definite result applying (8).

#### B. Initial guess

Since the positive semi-definite constraint is imposed after solving, the initial guess is not required to be in a strictly feasible point. Thus, we can use the off-diagonal blocks based initialization (9).

Alternatively, FD can be used to get an initial guess by just applying the first cycle of (12). During this first cycle, only the previously computed factors must be considered in  $\check{\mathbf{Y}}_i$ , and therefore it can be computed incrementally,

$$\check{\mathbf{Y}}_i = \check{\mathbf{Y}}_{i-1} + \check{\mathbf{J}}_{i-1}^\top \check{\boldsymbol{\Omega}}_{i-1} \check{\mathbf{J}}_{i-1}. \quad (14)$$

## IV. RESULTS

In order to test the performance of the FD sparsification method just presented and to compare it with the state-of-art IP and PQN methods, we implemented all methods in Matlab. To prevent linearization errors to be confused with sparsification inaccuracy, we relinearize the whole SLAM problem and solve it at each new trajectory pose using our own implementation of  $\sqrt{SAM}$  [16]. We implemented the IP method using the gradient and Hessian in [12]. For PQN, we used the authors' Matlab implementation [21]. Finally, we build the sub-graph topology as explained in section II-A with a number of factors doubling that of the tree topology.

#### A. Initial guess and convergence rate

We want to test the different combinations of optimization method and initial guess on several sparsification problems. The methods PQN and FD are combined with three different initial guesses: the identity matrix (Id) as proposed in [12], the one based on the off-diagonal blocks (ODB) as proposed in [19], and our First FD cycle (FFD) of Sec. III-B. The method IP is only combined with Id, as it diverges otherwise. This gives a total of seven combinations.

We executed a SLAM for the Manhattan M3500 dataset [22] with 80% of node removal. To guarantee equal conditions for all the methods under test, all sparsification problems after node marginalization are stored. Then, we compare all methods by solving the stored problems. In all cases, we solve and relinearize the cropped problems before marginalization.

First, we analyze the relation between the type of initial guess used and problem size. Fig. 4 depicts the mean and variance of the KLD between each initial guess and the dense exact distribution, as a function of the sparsification problem size, i.e. the number of nodes in the Markov blanket. As expected, in all cases the smaller the size of the Markov blanket, the better. ODB initialization performs

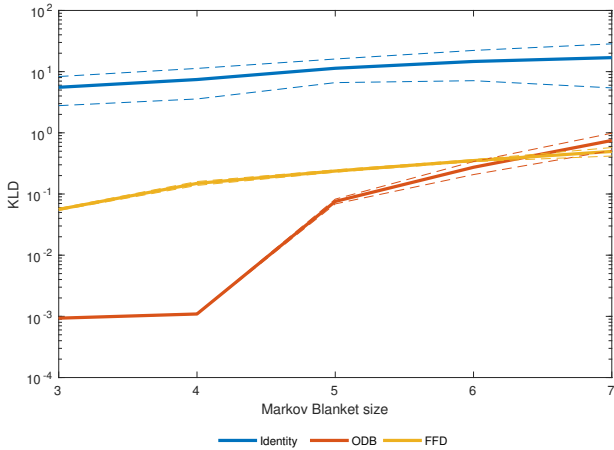


Fig. 4. Mean and variance (dashed line) of all three initial guess KLD vs. the Markov blanket size of for sparsification processes made in the Manhattan experiment with 80% node removal.

better for small Markov blankets whereas the proposed FFD works better for larger problems. The identity initial guess (Id) is significantly inferior than the other two initialization methods (notice the vertical log axis).

We analyze now the convergence rate for all seven combinations of initial guess and methods. In Matlab, the most computationally expensive operations are optimized, and therefore CPU time measures are not reliable. Also, since the size of the sparsification problems is small and independent of the full problem size, the complexity of each method w.r.t. the Markov blanket size is not necessarily representative. Then, the evaluation of the convergence rates is based on the number of optimization iterations.

Fig. 5 depicts the evolution of the mean KLD for all method-initial guess combinations, for Markov blankets of size 3. Although IP converges faster, PQN and FD take profit of a (much) better initial guess at the initial iterations, becoming better alternatives for implementations with low computational resources that do not allow for many iterations. If the number of iterations is not a constraint, IP reaches the best results.

As well as the initial guess, the convergence of the methods depends also on the size of the sparsification problem. We show in Fig. 6 the mean KLD evolution for all problem sizes. While in small problems FD-ODB is the best combination up to 15 iterations, for larger Markov blanket sizes IP becomes the best choice from a smaller amount of iterations.

### B. Application

We tested all methods on three different datasets [22] to evaluate their performance. The typology of the chosen datasets is very different. The Manhattan M3500 sequence is a large problem and, since it is highly connected, it has large Markov blankets. On the contrary, the MIT Killian Court sequence has few loop closures and small Markov blankets. The Intel Research Lab sequence is somewhere in between.

In all cases, the same 80% volume of nodes were marginalized. Since node selection is out of the scope of this paper, we applied the simple strategy of keeping one node every 5.

The following combinations of optimization method and initial guess were tested: IP-Id, PQN-ODB, FD-ODB, FD-FFD. We ran four independent SLAM solutions for each

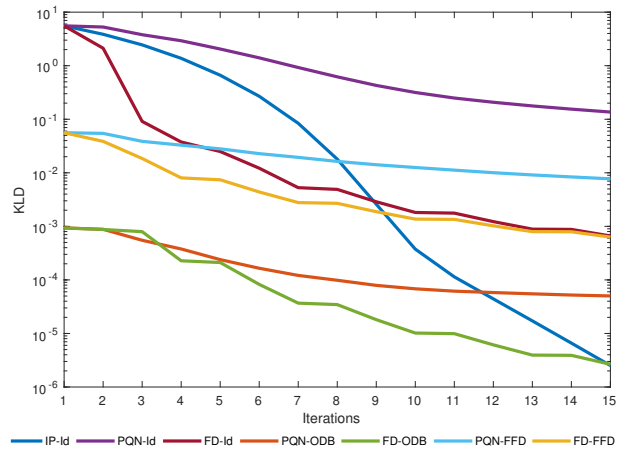


Fig. 5. Mean KLD evolution of all sparsification combinations (methods and initial guesses), for problems with Markov blanket size = 3 in the Manhattan experiment with 80% of node removal.

TABLE I  
FINAL KLD AFTER 80% OF NODE REMOVAL

	Method	Iterations			
		0	5	10	15
Manhattan	IP-Id	1285	110.6	<b>4.967</b>	<b>4.786</b>
	PQN-ODB	<b>7.36</b>	6.424	5.587	5.399
	FD-ODB	<b>7.36</b>	<b>5.469</b>	5.291	4.852
	FD-FFD	81.09	11.46	7.617	5.82
	CLT	25.84	-	-	-
Intel	IP-Id	314.8	85.07	8.083	7.293
	PQN-ODB	<b>18.19</b>	16.33	9.35	8.423
	FD-ODB	<b>18.19</b>	<b>7.962</b>	<b>7.247</b>	7.027
	FD-FFD	29.26	10.47	7.574	<b>6.902</b>
	CLT	27.68	-	-	-
Killian	IP-Id	429.4	210.1	76.73	<b>77.29</b>
	PQN-ODB	78.59	78.07	78.28	78.39
	FD-ODB	78.59	<b>77.21</b>	<b>77.4</b>	77.37
	FD-FFD	<b>70.11</b>	78.37	77.49	77.59
	CLT	82.92	-	-	-

combination fixing different number of optimization iterations: 0, 5, 10 and 15. For greater completeness, we also ran a CLT using the closed form solution.

The baseline for evaluation is the batch optimization of the original SLAM graph without removing any node. Following [12], factors involving previously removed nodes were redirected to the closest existing node. This was also done for the baseline graph in order to evaluate only the sparsification performance.

To evaluate the performance of the different approaches, we computed its KLD with the baseline using (3), but this time evaluating for the whole SLAM problem instead of just the Markov blanket.

Table IV-B shows the final KLDs of all 17 experiments for the three datasets. In the table, KLD for CLT is reported only for zero iterations since it has a closed form solution. The comparison between CLT and all methods evidences the limitation of the tree topology for accurately approximating the dense distribution. For highly connected problems (Manhattan and Intel), the use of a sub-graph topology produces a much smaller KLD. And for sparser cases such as the Killian dataset, the sub-graph topology slightly outperforms CLT. This is not surprising, since the average Markov blanket sizes are small in this case.

Note that, even without any optimization iteration, using

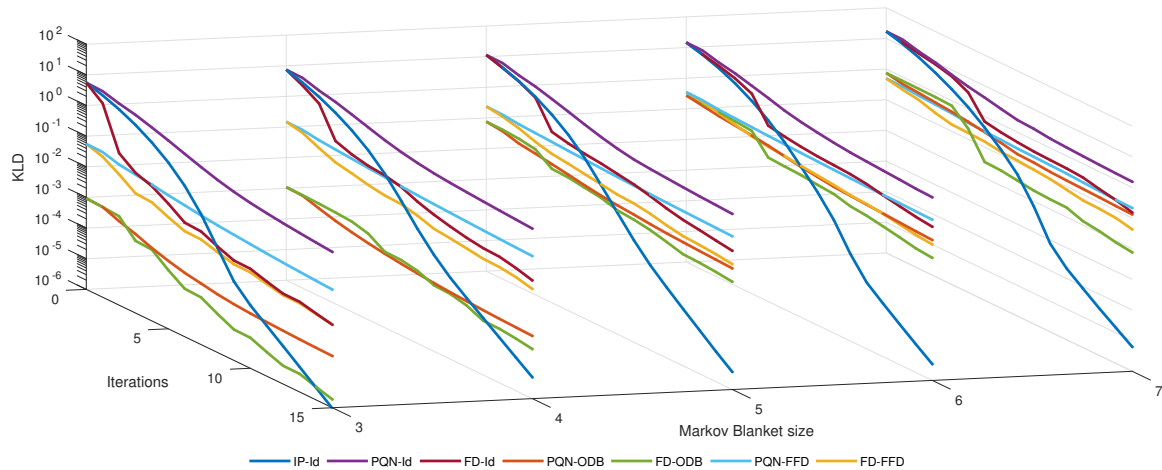


Fig. 6. Mean KLD evolution of all sparsification combinations (methods and initial guesses) for different Markov blanket sizes corresponding to all problems in the Manhattan experiment with 80% of node removal.

the ODB initialization always yields a better approximation than CLT.

The presented results validate the initial hypothesis: the tree topology is too sparse to accurately approximate the dense distribution. A more complex topology can approximate the original graph better without computational burden.

## V. CONCLUSIONS AND FUTURE WORK

Sparsification is a useful mechanism to maintain the SLAM problem bounded. The topology chosen determines the existence of a closed form solution and strongly affects the accuracy of the approximation.

Tree topologies admit a closed form solution but are usually too simple to approximate the original graph. On the contrary, more populated topologies lose the applicability of closed form solution but can encode more information of the original graph.

We presented the novel Factor Descent optimization method for sparsification that provides more accurate approximations with less iterations than other state-of-the-art sparsification methods. When combined with the off-diagonal block initialization, Factor Descent gives the best results in a larger number of situations.

In addition, we observed that the existence of a closed form solution is not a sufficient argument for choosing a tree topology instead a more complex one. Better approximations in terms of KLD can be reached using a more complex topology with few or no optimization iterations.

In future work we consider the implementation in C++ of all analyzed sparsification methods to be able to compare them also with regards to computational time. Also, a non cyclic version of Factor Descent can be explored to further improve convergence.

## REFERENCES

- [1] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robotics Res.*, vol. 31, no. 2, pp. 216–235, 2011.
- [2] V. Ila, L. Polok, M. Solony, and P. Svoboda, "SLAM++-A highly efficient and temporally scalable incremental SLAM framework," *Int. J. Robotics Res.*, vol. 36, no. 2, pp. 210–230, 2017.
- [3] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robotics Res.*, vol. 34, no. 3, pp. 314–334, 2015.
- [4] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robotics Res.*, vol. 32, no. 6, pp. 690–711, 2013.
- [5] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Trans. Robotics*, vol. 26, no. 1, pp. 78–93, Feb. 2010.
- [6] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. IEEE Int. Conf. Robotics Autom.*, Anchorage, May 2010, pp. 273–287.
- [7] H. Johannsson, M. Kaess, M. Fallon, and J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *Proc. IEEE Int. Conf. Robotics Autom.*, Karlsruhe, May 2013, pp. 54–61.
- [8] S. Choudhary, V. Indelman, H. Christensen, and F. Dellaert, "Information-based reduced landmark SLAM," in *Proc. IEEE Int. Conf. Robotics Autom.*, Seattle, May 2015, pp. 4620–4627.
- [9] H. Kretzschmar and C. Stachniss, "Information-theoretic compression of pose graphs for laser-based SLAM," *Int. J. Robotics Res.*, vol. 31, no. 11, pp. 1219–1230, 2012.
- [10] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic node removal for factor-graph SLAM," *IEEE Trans. Robotics*, vol. 30, no. 6, pp. 1371–1385, 2014.
- [11] K. Eickenhoff, L. Paull, and G. Huang, "Decoupled, consistent node removal and edge sparsification for graph-based SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Daejeon, Oct. 2016, pp. 3275–3282.
- [12] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term SLAM," *Int. J. Robotics Res.*, vol. 35, no. 1-3, pp. 50–72, 2016.
- [13] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, 1990, pp. 167–193.
- [14] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g<sup>2</sup>o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robotics Autom.*, Shanghai, May 2011, pp. 3607–3613.
- [15] L. Polok, V. Ila, M. Solony, P. Smrz, and P. Zemcik, "Incremental block Cholesky factorization for nonlinear least squares in robotics," in *Robotics: Science and Systems*, Berlin, Jun. 2013.
- [16] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robotics Res.*, vol. 25, no. 12, pp. 1181–1204, 2006.
- [17] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [18] M. Schmidt, E. Berg, M. Friedlander, and K. Murphy, "Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm," in *Artificial Intelligence and Statistics*, 2009, pp. 456–463.
- [19] M. Mazuran, G. D. Tipaldi, L. Spinello, and W. Burgard, "Nonlinear graph sparsification for SLAM," in *Robotics: Science and Systems*, Berkeley, Jul. 2014, pp. 1–8.
- [20] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear algebra and its applications*, vol. 103, pp. 103–118, 1988.
- [21] M. Schmidt, E. Berg, M. Friedlander, and K. Murphy, "Matlab PQN toolbox," <https://www.cs.ubc.ca/~schmidtm/Software/PQN.html>.
- [22] L. Carlone, <http://www.lucacarlone.com/index.php/resources/datasets>.