Fast Nonlinear Approximation of Pose Graph Node Marginalization

Duy-Nguyen Ta, Nandan Banerjee, Stephen Eick, Scott Lenser and Mario E. Munich

***Abstract*— We present a fast nonlinear approximation method for marginalizing out nodes on pose graphs for long-term simultaneous localization, mapping, and navigation. Our approximation preserves the pose graph structure to leverage the rich literature of pose graphs and optimization schemes. By re-parameterizing from absolute- to relative-pose spaces, our method does not suffer from the choice of linearization points as in previous works. We then join our approximation process with a scaled version of the recently-demoted pose-composition approach. Our approach eschews the expenses of many state-of-the-art convex optimization schemes through our efficient and simple $O(N^2)$ implementation for a given known topology of the approximate subgraph. We demonstrate its speed and near optimality in practice by comparing against state-of-the-art techniques on popular datasets.**

I. INTRODUCTION

Resource management for long-term simultaneous localization and mapping (SLAM) is a critical hurdle to overcome when developing robust perception methods for autonomous navigation [3]. While filtering and sliding-window SLAM methods provide a low memory footprint [27], linearization errors accumulated over the unbounded time horizon make these techniques unsuitable for long-term navigation tasks. On the other hand, optimally-stable methods such as batch and incremental smoothing [9], [19] require memory and computational resources too vast for most commercial robotic systems.

Systems continuously operating across long time periods compromise between data persistence and minimum processing requirements by efficiently sparsifying their data representation while retaining as much information as possible. Graph sparsification has long been actively explored in SLAM research, with early work attempting to minimize information loss resulting from edge removal [18], [28], [12], [13], [29]. More recent work explores efficient node marginalization approximation (either coupling or decoupling with edge sparsification) by searching for summarizing factors and virtual measurements minimizing the Kullback-Leiber Divergence (KLD) with respect to the original distribution [16], [5], [4], [23], [22], [11], [25]. However, these state-of-the-art methods struggle to strike a balance between accuracy, sparsity, and computability.

In this paper, we present a fast and near-optimal nonlinear approximation method to sparsify a *pose graph* while maintaining its core structure. A pose graph is a graph-based SLAM representation, where nodes are diachronic, unknown variables of robot poses and edges represent pairwise, relative pose measurements between nodes. Modern methods such as GLCs [4] or Cliquey Subgraphs [22] utilize various factors to approximate post-marginalization graph information. Our improved method preserves the pose graph structure to leverage its well-studied properties [6], [26], [7] and fast optimization schemes [10], [14], [24], [30].

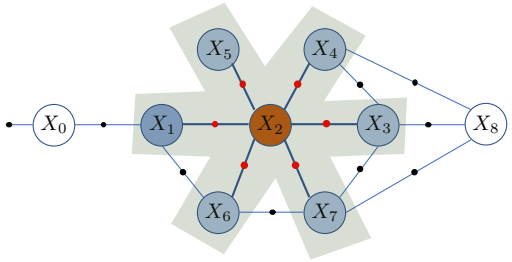

Fig. 1. A pose graph example. Each binary factor (dots) represents the relative pose likelihood function. Marginalizing out X_2 only involves its neighboring factors (red) and nodes inside the shaded area.

Preserving the pose graph pairwise structure after node marginalization is a nontrivial problem due to the resultant dense factor correlating neighboring nodes [20]. Since the pose variable manifolds and relative-pose measurement function are nonlinear, the dense factor is normally approximated by a linear Gaussian distribution. To retain pose graph structure, this dense Gaussian factor is again approximated with a set of pairwise nonlinear relative pose factors, the linearization of which must match the marginal Gaussian as close as possible. However, computing this set of nonlinear factors is challenging because the Jacobians of the linearized relative-pose measurement function have a very specific form and depend on unknown future linearization points.

State-of-the-art pose graph sparsification methods suffer from either accuracy or time-performance problems. [4], [16] and [22] have noted the pose composition approach used in [10] is indeed fast but suffers from double counting, resulting in an overly-confident graph post-marginalization. Conversely, the recent Nonlinear Factor Recovery (NFR) method [23], [22] produces the optimal approximation but with substantial performance overhead due to a convex optimization procedure on the space of symmetric positive semi-definite covariance matrices, making NFR unsuitable for resource-constrained robotics platforms. Although closed-form solutions for NFR exist under certain conditions, those cases are too simple to be good approximations in practice.

Our proposed nonlinear approximation method can both achieve near-optimal results and attain best-in-class speed and simplicity due to its pose composition implementation.

The authors are with iRobot Corporation, Bedford, MA 01730, USA. {duynguyen, stephen.eick}@gatech.edu, {nbanerjee, slenser, mmunich}@irobot.com

The specific advantages of our method and our key contributions in this paper are the following:

- We introduce a new re-parameterization view for the stated problem using variable elimination on factor graphs with hard constraint factors. This exactly preserves graph information since no new information is being introduced to the original graph.
- Through re-parameterization of the problem from absolute to relative poses, we avoid choosing either local or global linearization points as in previous works [4], [11], [22]. Also, our choice of re-parameterization makes our method *independent* of unknown future linearization points.
- Using the relative-pose re-parameterization, we show the connection of the pose composition method with the well-known product-of-marginals approximation and prove the technique's optimality in full-rank target subgraph topologies, e.g. tree structures.
- We propose a fast solution based on spanning trees of the target subgraph for the final approximation step to find an independent Gaussian approximation minimizing the KLD for a degenerate low-rank Gaussian distribution.
- We connect our method with the traditional pose composition approach used in [10] to enjoy its powerful yet simple implementation and $O(N^2)$ complexity, with N being the number of edges in the subgraph which approximates the original dense marginal factor.

II. Problem formulation

A. Pose graph preliminary

A pose graph (Fig. 1) is a specific type of factor graph [9] with exactly one type of node to represent robot poses and one type of pairwise edge factors (save for a special prior factor on the first node) to represent relative pose measurements between the nodes. Begin with a set of robot poses $X = \{X_i\}_{i=0..M}$ on a Special Euclidean group $SE(n)$ [2] and *independently* distributed relative pose measurements $\bar{Z} = \{\bar{z}_{ij} \in SE(n)\}$ between the poses. The resulting pose graph represents the factorization of the posterior probability

$$p(X_0, \dots, X_M|\bar{Z}) \propto p(X_0) \prod_{ij} p(\bar{z}_{ij}|X_i, X_j), \quad (1)$$

which is maximized to find the maximum a posteriori (MAP) estimate for the given pose graph. The relative pose measurements are commonly assumed to be Gaussian-distributed, and the likelihood function corresponding to each pairwise factor in the pose graph takes the form

$$p(\bar{z}_{ij}|\, X_i, X_j) \propto \exp(-\frac{1}{2} \left\|\log(\bar{z}_{ij}^{-1}(X_i^{-1}X_j))\right\|^2_{\Sigma_{ij}}), \quad (2)$$

where $\log(\cdot)$ is the mapping between the $SE(n)$ manifold to its $\mathfrak{se}(n)$ Lie algebra vector space.

Optimizing (1) is equivalent to minimizing its negative log, which is a nonlinear least-squares problem. Standard techniques to compute the MAP estimate linearize at the current guessed solution, solve the linear least-squares problem, and update the estimate until it converges [9].

The linearization of (1) is a factor graph of Gaussian factors. Using the exponential map representation $X_i = \mathring{X}_i \exp(x_i)$—where $\mathring{X}_i$ is the current linearization point, $\exp(\cdot)$ the exponential map from $\mathfrak{se}(n)$ to $SE(n)$, and $x_i \in \mathfrak{se}(n)$—the linearization of (2) is equivalent to a Gaussian factor of the form [2]

$$p(\bar{z}_{ij}|\, x_i, x_j) \propto \exp(-\frac{1}{2} \left\|-\mathrm{Ad}_{\bar{z}_{ij}^{-1}} x_i + x_j - e_{ij}\right\|^2_{\Sigma_{ij}}), \quad (3)$$

where $e_{ij} = \log(\bar{z}_{ij}^{-1}(\mathring{X}_i^{-1}\mathring{X}_j))$ is the error vector and the Jacobian $\mathrm{Ad}_{\bar{z}}$ is the Adjoint map matrix of $\bar{z}$ on $SE(n)$ [17].

B. Node marginalization

Marginalizing out a node X_k from the graph only involves operations on directly-connected factors [20] (see Fig. 1). This is demonstrated by

$$\int_{X_k} p(X_0, \dots, X_M|\bar{Z}) \propto p(X_0) \prod_{i\neq k, j\neq k} p(\bar{z}_{ij}|\, X_i, X_j) \times \left(\int_{X_k} \prod_{X_l \in NB(X_k)} p(\bar{z}_{kl}|\, X_k, X_l) \right), \quad (4)$$

where $NB(X_k)$ is the set of X_k's neighboring nodes and $p(\bar{z}_{kl}|\, X_k, X_l)$ are its neighboring factors. Marginalizing out X_k from its neighboring subgraph is equivalent to a variable elimination process [20], refactoring the product of the neighboring factors into a conditional distribution on X_k and a new factor $f(\cdot)$ on its neighbors

$$\prod_{X_l \in NB(X_k)} p(\bar{z}_{kl}|\, X_k, X_l) = p(X_k|NB(X_k)) f(NB(X_k)).$$

For tractability reasons, the marginalization is performed on a linearized version of the graph, resulting in a dense Gaussian factor $f(NB(x_k))$ linearly approximating the ideal nonlinear marginal factor at the chosen linearization point.

This marginalization only involves neighboring factors of a node. Therefore, what follows in this paper simplifies the index notation by focusing on marginalizing out X_0 with its set of N neighbors $X_{1..N} = NB(X_0) = \{X_1, \dots, X_N\}$ without losing generality. We are only interested in the neighboring subgraph of X_0 (Fig. 2) which corresponds to the density subgraph

$$f(X_{0..N}) \triangleq \prod_{i=1..N} p(\bar{z}_{0i}|X_0, X_i) \quad (5)$$

with its corresponding linearized graph

$$f(x_{0..N}) \triangleq \prod_{i=1..N} p(\bar{z}_{0i}|x_0, x_i), \quad (6)$$

where $p(\bar{z}_{0i}|X_0, X_i)$ and $p(\bar{z}_{0i}|x_0, x_i)$ are defined in (2) and (3), respectively.

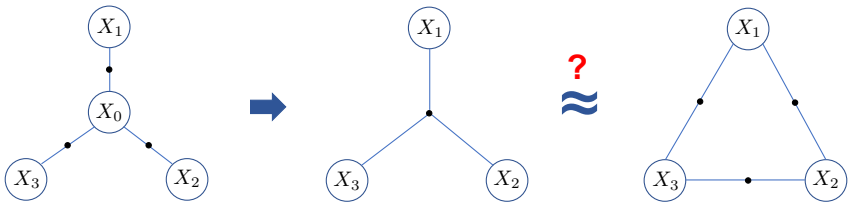

Fig. 2. Nonlinear Factor Approximation problem. Left: The neighboring subgraph of interest to marginalize out X_0. Middle: The graph after marginalizing out X_0. Right: The desired pose graph that approximates the marginalization result with pairwise factors.

C. Nonlinear factor approximation

The Gaussian factor $f(x_{1..N})$ resulting from marginalizing out X_0 requires large computational resources and breaks the pose graph structure. Recent work attempts to reduce its density [4], [16], maintain the pose graph structure [10], or implement some combination of the two [22], [11].

To maintain the pose graph structure, we must approximate $f(x_{1..N})$ with a target subgraph of *nonlinear* relative-pose factors $q(\bar{z}_{ij}|\, X_i, X_j)$ of the form in (2), where (X_i, X_j) are pairs of X_0's neighbors connected in the subgraph. Specifically, we find the means $\bar{z}_{ij}$ and covariances Σ_{ij} of the nonlinear factors $q(\bar{z}_{ij}|\, X_i, X_j)$ in the target subgraph to minimize the KLD such that

$$\begin{aligned} \bar{z}^*_{ij}, \Sigma^*_{ij} &= \underset{\bar{z}_{ij}, \Sigma_{ij}}{\text{argmin}} KL(f(x_{1..N}) \parallel q(x_{1..N})) \\ &= \underset{\bar{z}_{ij}, \Sigma_{ij}}{\text{argmin}} KL(f(x_{1..N}) \parallel \prod_{ij \in K} q(\bar{z}_{ij}|\, x_i, x_j)), \end{aligned} \quad (7)$$

where $q(x_{1..N})$ is the linearized version of the nonlinear subgraph $q(X_{1..N}) = \prod_{ij \in K} q(\bar{z}_{ij}|\, X_i, X_j)$ and $K = \{(i,j)\}_{i,j \in 1..N}$ is the target subgraph topology. Our method is generic to the target subgraph's topology, ranging from the sparsest Chow-Liu tree [8] to the fully-dense clique. Different strategies to choose a topology have been discussed extensively in previous work [4], [22].

In this KLD minimization problem, solving for the optimal covariance Σ^*_{ij} of the nonlinear factor $q(\bar{z}_{ij}|\, X_i, X_j)$ is challenging for several reasons. First, the covariance of the Gaussian factor $f(x_{1..N})$ may take any arbitrary form, whereas the covariance of its approximation, the linearized subgraph $q(x_{1..N})$, is constrained in a subspace of special Jacobian matrices of its Gaussian factors in (3)—specifically, the Adjoint and the Identity matrices. The covariances Σ^*_{ij} belong to a non-Euclidian space of symmetric positive semi-definite matrices, so convex optimization techniques are needed to minimize (7) [22].

III. Re-parameterizations

We consider each relative-pose measurement between X_i and X_j as a random variable Z_{ij}. Each nonlinear binary factor $q(\bar{z}_{ij}|X_i, X_j)$ between X_i and X_j is equivalent to a Gaussian distribution $p(Z_{ij})$ on Z_{ij} with mean $\bar{z}_{ij}$ and covariance Σ_{ij}. Hence, the original density $f(X_{0..N})$ before marginalizing out X_0 in (5) is equivalent to the distribution $\prod_{i=1..N} p(Z_{0i})$ on the relative-pose random variables Z_{0i}'s, where

$$Z_{0i} = X_0^{-1} X_i \quad (8)$$

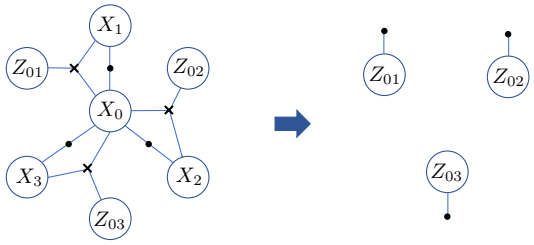

Fig. 3. Reparameterization from absolute to relative poses. Left: The original graph augmented with relative pose random variables and hard constraints (cross) representing their relation with the corresponding absolute poses in Eq. (8). Right: The equivalent independent distribution on relative poses after eliminating all absolute poses.

and $p(Z_{0i}) = \mathcal{N}(\bar{z}_{0i}, \Sigma_{0i})$ according to (2).

This re-parameterization step can be done explicitly by adding special factors representing the hard constraint (8) to the original X_0's neighboring subgraph (Fig. 3). As an exact equivalence, hard constraint factors are degenerate delta distributions carrying no new information, so adding them to the graph does not alter the total amount of information. We then eliminate all absolute poses $X_0, \ldots, X_N$ to obtain the equivalent independent distribution $\prod_i p(Z_{0i})$ defined on the relative poses. The resulting graph of relative poses (Fig. 3-right) includes unary factors that are exactly equivalent to the constraints in the original absolute graph in Fig. 2-left.

With the relative-pose parameterization, our non-linear factor approximation problem becomes *finding an independent distribution of relative poses* $\prod_{ij \in K} q(Z_{ij})$ *corresponding to the target absolute-pose subgraph topology* K *that best approximates the distribution* $\prod_i p(Z_{0i})$, which itself is equivalent to the original distribution in (5).

We follow NFR to approximate the mean of the target distribution as $\bar{z}_{ij} = X_i^{*-1} X_j^*$, where X_i^*'s the current MAP estimate. To solve for the covariance Σ_{ij}, our strategy is to first find the joint distribution $p(\{Z_{ij}\}_{ij \in K})$ of the new relative poses Z_{ij}'s corresponding to the target subgraph topology K and approximate it with an independent distribution $\prod_{ij \in K} q(Z_{ij})$.

To find the joint distribution $p(\{Z_{ij}\}_{ij \in K})$, we note that

$$Z_{ij} = Z_{0i}^{-1} Z_{0j}. \quad (9)$$

Therefore, we can use the same technique as before to augment the relative-pose graph in Fig. 3-right with new variables Z_{ij} and hard constraints representing (9) (Fig. 4-left). After eliminating all variables Z_{0i}'s, the result is a dense factor on Z_{ij}'s representing their joint distribution

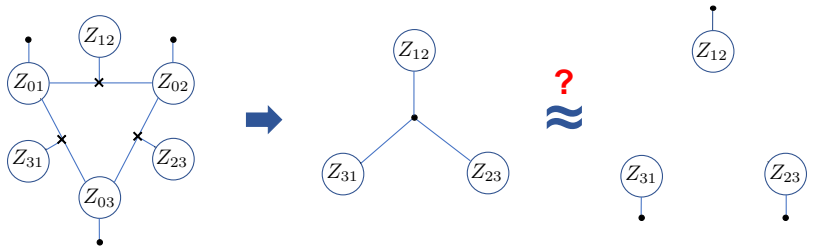

Fig. 4. **Nonlinear Factor Approximation problem in relative-pose space.** Left: The original distribution augmented with relative poses of the target subgraph topology and hard constraints. Middle: The distribution of the new edge measurements after marginalizing out the original measurements. Right: The desired independent approximate distribution of the new edge measurements, which is equivalent to our target pose graph in Fig.2-right.

$p(\{Z_{ij}\}_{ij\in K})$ (Fig. 4-middle). As discussed, the variable elimination process occurs on the linearized version of the graph, and its result is a dense Gaussian factor $p(\{z_{ij}\}_{ij\in K})$ on the tangent vectors $z_{ij} \in \mathfrak{se}(n)$ of Z_{ij}. Linearizing the hard constraint (9) results in

$$z_{ij} = -\mathrm{Ad}_{Z_{0j}^{-1}} z_{0i} + z_{0j}. \tag{10}$$

At the linear level, the problem is mathematically equivalent to an affine transform of a Gaussian distribution $y = Az$, where $y = [\{z_{ij}\}_{ij\in K}]^T$ and $z = [z_{01} \dots z_{0N}]^T$ are vectors created by stacking all corresponding tangent vectors from the relative-pose graph, and the matrix A is determined from (10). This shows that the covariance matrix of the joint density $p(y) \triangleq p(\{z_{ij}\}_{ij\in K})$ is simply $\Sigma_{joint} = A\Sigma_0 A^T$, where $\Sigma_0 = \mathrm{diag}(\Sigma_{01}, \dots, \Sigma_{0N})$ is the block diagonal matrix of the covariances of the measurements $\{Z_{0i}\}_{i=1..N}$.

IV. Approximation

After obtaining the joint density distribution $p(\{z_{ij}\}_{ij\in K})$, the final step is to approximate it with an independent Gaussian distribution $\prod_{ij\in K} q(z_{ij})$. Each component $q(z_{ij})$ provides the covariance Σ_{ij} to produce the desired edge measurements (Fig. 4-right).

A. Full-rank Gaussian Approximation and the Connection with Pose Composition Method

When the distribution $p(\{z_{ij}\}_{ij\in K})$ has full rank, the best-approximating independent distribution becomes the cummulative product of each variable's marginal distributions:

$$p(\{z_{ij}\}_{ij\in K}) \approx \prod_{ij\in K} p(z_{ij}),$$

where $p(z_{ij}) \triangleq \int p(\{z_{ij}\}_{ij\in K}) \delta z_{lk, lk\neq ij}$.

Hence, the optimal component $q(z_{ij})$ in this case is simply the marginal distribution $p(z_{ij})$. This has been proven more generally in [1], Theorem 1, which approximates a Gaussian distribution with a tree structure.

As we have previously discussed, computing the marginal distribution $p(z_{ij})$ from the joint distribution $p(\{z_{ij}\}_{ij\in K})$ is possible using variable elimination. However, the variable elimination process possesses $O(N^3)$ worst-case time complexity, rendering it unsuitable for many situations.

Fortunately, a faster and simpler method to compute $p(z_{ij})$ exists through the recently-demoted pose composition approach. We first notice from the augmented graph in Fig. 4-left that each new variable Z_{ij} only depends on Z_{0i} and Z_{0j}. In fact, the joint density of that graph can be factorized as:

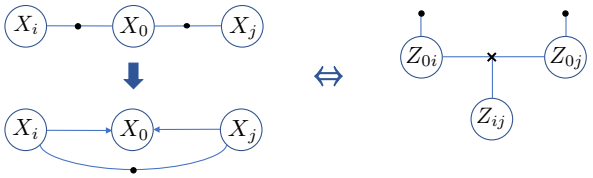

Fig. 5. Marginalization on a simple graph with two edges (left) and the equivalent relative-pose representation of the process.

$$\begin{aligned} p(\{Z_{0i}\}_{1..N}, \{Z_{ij}\}_K) &= \prod_{i=1}^{N} p(Z_{0i}) \prod_{ij\in K} p(Z_{ij}|Z_{0i}, Z_{0j}) \\ &= p(Z_{ij}|Z_{0i}, Z_{0j}) p(Z_{0i}) p(Z_{0j}) \\ &\times \prod_{u\neq i,j} p(Z_{0u}) \prod_{uv\neq ij} p(Z_{uv}|Z_{0u}, Z_{0v}) \end{aligned}$$

Marginalizing out all variables u, v uncorrelated with Z_{ij} results in a simple subgraph of only three factors $p(Z_{ij}|Z_{0i}, Z_{0j})$, $p(Z_{0i})$ and $p(Z_{0j})$, in which the first one represents the hard constraint in (9). Remarkably, this subgraph of relative poses is exactly equivalent to the one used to marginalize out X_0 from the graph of two edges (X_i, X_0) and (X_0, X_j) to obtain the measurement of the new edge (X_i, X_j) as shown in Fig. 5. Due to the hard constraint, the marginal distribution of Z_{ij} can be easily obtained by composing the distributions of Z_{0i}^{-1} and Z_{0j}. The absolute-graph equivalent view is that we obtain the measurement of the new edge (i, j) by composing together the measurements of two edges $(i, 0)$ and $(0, j)$. Formulas to compose two distributions of poses have been studied extensively in literature, and recent work achieves up to second-order accuracy on $SE(3)$ ([2], Section III).

In summary, we see the pose composition approach used in [10] essentially computes the relative-pose marginal distributions corresponding to the new edges of the target subgraph topology. If the joint distribution of relative poses is full-rank, this method guarantees the optimal solution.

B. Degenerate Gaussian Approximation

Unfortunately, $p(\{z_{ij}\}_{ij\in K})$ is not typically full rank, therefore the above does not always hold. The relative-pose graph is only full-rank if the topology K of its equivalent absolute-pose subgraph is a tree. This can be observed by choosing a spanning tree of the absolute-pose graph. Any edge (i,j) not belonging to this spanning tree forms a cycle

$$i \to j \to t_1 \to t_2 \to \cdots \to t_k \to i,$$

where $\{(j,t_1),\{(t_l,t_{l+1})\}_{l=1..k-1},(t_k,i)\}$ are edges in the spanning tree involved in the cycle. The property indicates the relative pose Z_{ij}, corresponding to the non-spanning-tree edge (i,j), is fully determined by composing other poses corresponding to spanning-tree edges in this cycle: $Z_{ij} = Z_{it_k}Z_{t_kt_{k-1}}\dots Z_{t_1j}$. From this, the tangent vector z_{ij} is related to other vectors in the cycle via a hard constraint, so $p(\{z_{ij}\}_{ij\in K})$ is ***not*** full rank. The product of marginal distributions $\prod_{ij\in K} p(z_{ij})$ is no longer the best approximation of the rank-deficient density $p(\{z_{ij}\}_{ij\in K})$ because it double-counts information. In summary, this result indicates the pose composition method falls short.

Our method to approximate the degenerate Gaussian distribution $p(\{z_{ij}\}_{ij\in K})$ is based on the above analysis of spanning trees. Let G be the number of spanning trees in the absolute-pose target subgraph, and $S=\{S_1,S_2,\dots,S_G\}$ the set of all spanning trees, where S_k is the set of all relative-pose variables z_{ij} corresponding to edges on the k^{th} spanning tree, and $S_{\setminus k}$ the set of variables not in S_k. The orignal distribution can be factorized and approximated as:

$$p(\{z_{ij}\}_{ij\in K}) = p(S_k)p(S_{\setminus k}|S_k) = p(S_k) \approx \prod_{z_{ij}\in S_k} p(z_{ij}).$$

for any tree $k \in 1..G$. This is possible since the conditional $p(S_{\setminus k}|S_k)$ is a degenerate delta distribution, as each variable in $S_{\setminus k}$ does not belong to the tree S_k, forms a cycle with other tree edges, and can be fully determined by them. Furthermore, the variables in S_k cannot be fully determined from any others in the set, so $p(S_k)$ has full rank and the best approximation of $p(S_k)$ is the product of the marginals.

This is not the best approximation, however, because we could use the other non-tree variables to capture more correlation information of variables on the tree. Our idea to capture more correlation information is to consider all possible spanning trees of the absolute-pose target subgraph. We first note that the original distribution can be duplicated G times:

$$p(\{z_{ij}\}_{ij\in K}) = \prod_{k=1}^{G} {p_k}^{\alpha_k}(\{z_{ij}\}_{ij\in K}),$$

where each p_k is exactly equivalent to $p(\{z_{ij}\}_{ij\in K})$, and α_k is a weight term such that $\sum_k \alpha_k = 1$. We then approximate each term p_k as a product of marginals of variables on the tree S_k:

$$p(\{z_{ij}\}_{ij\in K}) = \prod_{k=1}^{G} p^{\alpha_k}(S_k) \approx \prod_{k=1}^{G}\left(\prod_{z_{ij}\in S_k} p^{\alpha_k}(z_{ij})\right). \tag{11}$$

This approximation includes all possible relative-pose variables and promises to capture more correlation information in the graph than using any single tree.

The weights α_k allow us to capture additional information in the approximation. Our idea is that trees whose marginal approximation contains more information are better than those with less information and should be assigned more weight. As a heuristic, we use the total trace of information matrices of all variables in the tree to quantify how much information the tree approximation captures. Let λ_{ij} be the trace of variable z_{ij}'s information matrix. The weight for each tree is chosen as:

$$\alpha_k = \frac{\Lambda_k}{\Lambda_{all}},$$

where $\Lambda_k \triangleq \sum_{z_{ij}\in S_k}\lambda_{ij}$ is the approximate total information of the tree S_k and $\Lambda_{all} \triangleq \sum_{l=1}^{G}\left(\sum_{z_{ij}\in S_l}\lambda_{ij}\right)$ is the approximate total information of all trees.

Let H_{ij} be the number of spanning trees containing an edge (i,j). Another way to compute the total information of all trees is $\Lambda_{all} = \sum_{ij} H_{ij}\lambda_{ij}$. After some manipulation, the approximation formula in (11) can be written as a product of weighted marginals of all variables in the relative pose graph:

$$\prod_{k=1}^{G}\left(\prod_{z_{ij}\in S_k} p^{\alpha_k}(z_{ij})\right) = \prod_{ij} p^{\beta_{ij}}(z_{ij}), \tag{12}$$

where $\beta_{ij} = \frac{\Lambda_{all}-\Lambda_{\setminus ij}}{\Lambda_{all}}$ and $\Lambda_{\setminus ij}$ denotes the total sum of information of all trees that do not contain (i,j). In special cases where all trees capture the same amount of information, $\alpha_k = 1/G$, and the marginal weights are $\beta_{ij} = H_{ij}/G$.

Still requiring elucidation are G—the number of spanning trees in the absolute-pose target subgraph—and H_{ij}—the number of spanning trees containing the edge (i,j). For general subgraph topologies, these numbers can be computed in $O(N^3)$ time complexity using Kirchhoff's matrix tree theorem [15]. Moreover, closed-form formulas for G and H_{ij} exist for specific topologies. For example, a circular topology provides $H_{ij} = N-1$ and $G = N$, while complete graphs provide $H_{ij} = 2N^{N-3}$ and $G = N^{N-2}$ (also known as Cayley's formula [15]). For these two topologies, $\Lambda_{\setminus ij}$ can be efficiently computed as $\Lambda_{\setminus ij} = G_{\setminus ij}\sum_{kl\neq ij}\lambda_{ij}$, where $G_{\setminus ij}$ is the number of spanning trees in the graph after the edge (i,j) is removed. This result leads to the following:

$$\beta_{ij} = \begin{cases} 1 - \frac{\sum_{kl\neq ij}\lambda_{ij}}{(N-1)\sum_{kl}\lambda_{kl}} & \text{for circular topology,} \\ 1 - \frac{1}{2}\left(\frac{N-1}{N}\right)^{N-3}\frac{\sum_{kl\neq ij}\lambda_{ij}}{\sum_{kl}\lambda_{kl}} & \text{for dense topology.} \end{cases}$$

In summary, due to the exponential form of the Gaussian $p(z_{ij})$, the approximate covariance of each new edge (i,j) in the target absolute-pose subgraph is determined by computing the marginal distribution $p(z_{ij})$ using the pose composition formula, then weighting the marginal covariance matrix by a factor of $1/\beta_{ij}$.

Algorithm 1 Nonlinear approximation of marginalization

Input: A node X_0 to marginalize out, pose measurements $(\bar{z}_{0i}, \Sigma_{0i})_{i=1..N}$ relative to its neighbors $\{X_i\}_{i=1..N}$, and a target subgraph topology K.

Output: The distributions of the new relative pose measurements Z_{ij} for each edge $(i,j) \in K$ in the subgraph that best approximate the result of marginalizing out X_0.

1: **for** each edge $(i,j) \in K$: **do**

2: Compute the mean:

$$\bar{z}_{ij} = X_i^{*-1} X_j^*$$

3: Compute the marginal covariance:

$$\Sigma_{ij} = \mathrm{Ad}_{z_{ij}^{-1}} \Sigma_{0i} \mathrm{Ad}_{z_{ij}^{-1}}^T + \Sigma_{0j}$$

4: Weight the covariance by a scale factor:

$$\Sigma_{ij} \leftarrow \frac{1}{\beta_{ij}} \Sigma_{ij}$$

5: **end for**

V. Experiments

A. Verification on Simple Graphs

We first verify our algorithm on a graph topology consisting of only three nodes and two edges (Fig. 5). The result is equivalent to the standard pose composition technique (the formula for the optimal covariance is shown in Algorithm 1, Line 3). Interestingly, an example in [22] shows the technique does not optimally minimize the KL divergence between the linear marginal factor and the approximate nonlinear relative pose factor. Their experiment computed different pose covariances when using pose composition and NFR. Since NFR is known optimal (as it directly minimizes the KL divergence), the pose composition approach is shown to be non-optimal.

We decided to revisit this experiment. In formulas (28)-(29) in [22], the authors used two relative pose measurements

$$\begin{aligned} \bar{z}_{10} &= \begin{bmatrix} 0 & 0 & \pi/2 \end{bmatrix}^T \\ \bar{z}_{02} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{13}$$

with covariances of

$$\Sigma_{10} = \Sigma_{02} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}. \tag{14}$$

Our results found their claim inaccurate. Using the exponential map representation of $SE(2)$, both the pose composition formula and the NFR method give the equivalent optimal answer

$$\Sigma_{12} = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 8 & 4 \\ 0 & 4 & 4 \end{bmatrix}, \tag{15}$$

reclaiming optimality of the pose composition approach for this simple case.

We next try our method on a 3-neighbor subgraph with ground-truth poses (x, y, θ) at $X_0 = (0,0,0)$, $X_1 = (-1, 1, 3\pi/4)$, $X_2 = (1, 1, \pi/4)$, and $X_3 = (1, -1, -\pi/4)$, and covariance matrices as in the previous example (14). Fig. 6 plots our KL divergence results with different values of the inverse Sigma's scale factor $s \triangleq \beta_{ij}$. Since all spanning trees capture the same amount of information, we expect a good scale factor at $s = H_{ij}/G = 2/3$. Fig. 6 confirms that the minimal KLD value is attained when $s = 2/3$, therefore this result is the optimal solution in the space of scaled marginal covariance matrices.

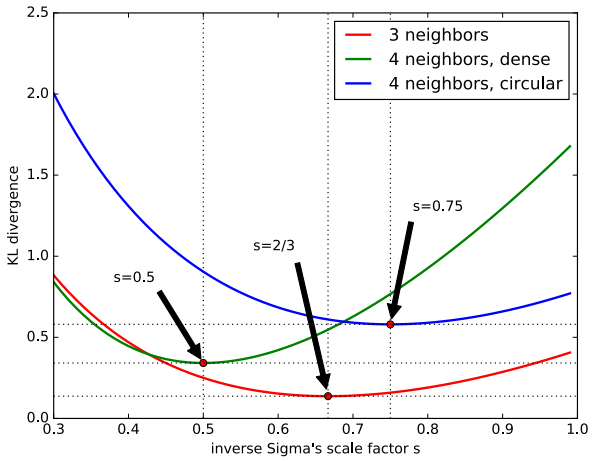

Fig. 6. KL divergences for experiments with simple subgraphs

Our experiments to approximate a 4-neighbor subgraph with different target topologies also confirmed this finding. We used the same ground-truth poses and covariances from the previous 3-neighbor graph experiment while adding an additional ground-truth pose $X_4 = (-1, -1, -3\pi/4)$ with the same covariance. As shown in Fig. 6, $s \triangleq \beta_{ij} = 0.5$ for the dense target topology. This matches the theoretical expectation of Cayley's formula with $s = H_{ij}/G$, where $H_{ij} = 2 \times 4^{4-3} = 8$ and $G = 4^{4-2} = 16$. Computing the result for circular topology also confirms the theoretical formula with $s = H_{ij}/G = 3/4$.

B. On Large Datasets

We tested our method on five different datasets and compared it with NFR results from [22] and the pose composition method (PC) without scale factors from [10]. Table I shows our results on the datasets with different target subgraph topologies (Tree, Circular, and Dense) and sparsity levels in a full batch process. The testing process first constructs and optimizes the full graph. Once complete, all spatially redundant nodes are sequentially marginalized. Chow-Liu trees are selected for the Tree topology. For the Circular topology, circular subgraphs were constructed following the order of indices of the neighboring nodes. For proper comparison with our method, NFR was run without considering intra factors connecting neighbor nodes, as those are not relevant with respect to the marginalization formula (4).

TABLE I
SPARSIFICATION RESULTS ON LARGE DATASETS.
KLDS ARE COMPUTED IN $SE(2)$ SPACE.

Dataset	Approach		Sparsity (50%)		Sparsity (66.6%)		Sparsity (75%)		Sparsity (80%)	
	Topology	Method	KLD	Time (s)	KLD	Time (s)	KLD	Time (s)	KLD	Time (s)
Duderstadt (2D) 545 nodes, 1800 edges	Tree	NFR	9.84	0.08	4.00	0.08	4.79	0.08	2.69	0.09
		PC	9.84	0.12	4.00	0.14	4.79	0.16	2.69	0.17
		Ours	9.84	0.11	4.00	0.14	4.79	0.17	2.69	0.16
	Circular	NFR	12.04	2.84	7.37	3.81	8.98	3.98	6.83	14.93
		PC	**11.39**	**0.17**	**5.97**	**0.21**	**6.90**	**0.22**	**4.80**	**0.31**
		Ours	12.45	**0.16**	8.06	**0.20**	9.78	**0.23**	7.80	**0.23**
	Dense	NFR	NA	NA	NA	NA	NA	NA	NA	NA
		PC	25.63	**0.41**	31.40	**0.55**	32.64	**0.60**	38.06	**0.68**
		Ours	**16.07**	**0.42**	**11.09**	**0.53**	**11.08**	**0.59**	**10.35**	**0.62**
EECS (2D) 615 nodes, 2134 edges	Tree	NFR	22.33	0.14	13.19	0.15	7.27	0.23	7.03	0.29
		PC	22.33	0.11	13.18	0.18	7.27	0.19	7.02	0.21
		Ours	22.33	0.11	13.18	0.15	7.27	0.19	7.02	0.20
	Circular	NFR	26.09	6.44	22.30	11.46	17.74	13.62	18.36	14.12
		PC	**25.05**	**0.17**	**20.35**	**0.26**	**15.40**	**0.29**	**16.02**	**0.34**
		Ours	26.81	**0.17**	24.00	**0.24**	19.73	**0.28**	20.72	**0.31**
	Dense	NFR	NA	NA	NA	NA	NA	NA	NA	NA
		PC	52.15	0.69	53.05	1.24	43.61	1.38	47.13	1.56
		Ours	**35.55**	**0.60**	**26.31**	**1.10**	**16.34**	**1.27**	**16.45**	**1.38**
Intel 943 nodes, 1833 edges	Tree	NFR	64.37	0.09	46.68	0.10	39.18	0.12	36.48	0.12
		PC	64.39	0.07	46.68	0.11	39.18	0.13	36.48	0.15
		Ours	64.39	0.06	46.68	0.10	39.18	0.12	36.48	0.14
	Circular	NFR	**50.52**	7.79	**41.78**	11.52	**35.78**	8.02	**36.10**	7.94
		PC	60.98	**0.10**	51.93	**0.16**	45.69	**0.19**	49.73	**0.21**
		Ours	50.78	**0.10**	43.58	**0.15**	39.40	**0.17**	41.23	**0.20**
	Dense	NFR	NA	NA	NA	NA	NA	NA	NA	NA
		PC	240.36	**0.27**	305.73	**0.49**	325.39	**0.64**	307.70	**0.66**
		Ours	**119.60**	**0.25**	**114.34**	**0.43**	**96.50**	**0.53**	**80.51**	**0.59**
Killian 5489 nodes, 7626 edges	Tree	NFR	130.90	0.43	127.29	0.42	69.40	0.44	99.80	0.45
		PC	131.16	0.13	127.38	0.31	69.46	0.33	99.82	0.38
		Ours	131.16	0.13	127.38	0.29	69.46	0.39	99.82	0.37
	Circular	NFR	**82.98**	5.98	**89.16**	13.35	91.17	15.35	**110.71**	19.32
		PC	133.30	**0.23**	250.90	**0.51**	296.07	**0.62**	313.22	**0.71**
		Ours	83.02	**0.30**	91.56	**0.48**	**81.47**	**0.63**	119.33	**0.70**
	Dense	NFR	NA	NA	NA	NA	NA	NA	NA	NA
		PC	170.24	0.35	899.22	0.98	747.96	1.04	1252.48	1.44
		Ours	**92.68**	**0.26**	**234.65**	**0.93**	**155.12**	**0.99**	**208.87**	**1.43**
Manhattan 3500 nodes, 5596 edges	Tree	NFR	380.54	0.36	221.48	0.38	171.03	0.39	146.22	0.39
		PC	380.62	0.12	221.77	0.21	171.11	0.25	146.40	0.31
		Ours	380.62	0.12	221.77	0.19	171.11	0.24	146.40	0.29
	Circular	NFR	**237.04**	11.43	**187.70**	16.66	**227.56**	21.12	**211.52**	22.85
		PC	307.62	**0.25**	281.45	**0.44**	292.23	**0.58**	270.15	**0.70**
		Ours	238.44	**0.21**	201.22	**0.40**	251.41	**0.53**	243.62	**0.62**
	Dense	NFR	NA	NA	NA	NA	NA	NA	NA	NA
		PC	893.26	**0.49**	1150.00	**0.90**	1390.25	**1.35**	1355.58	**1.64**
		Ours	**379.13**	**0.39**	**341.32**	**0.88**	**343.76**	**1.25**	**277.03**	**1.47**

All experiments are performed and timed on a MacbookPro11,3-Mid 2014 with a 2.8 GHz Core i7 CPU and 16 GB of RAM. In each experiment, the sparse graph after marginalization is compared with the full original distribution using KLDs *computed in* $\mathfrak{se}(2)$. This is notable since NFR computes KLDs in g2o's (x, y, θ) tangent space representation [21], whereas our method operates on the Lie algebra $\mathfrak{se}(2)$. For a fair comparison, we convert the covariances of NFR's input graphs and its resulting graphs to $\mathfrak{se}(2)$ and compute the KLDs in that space. We also run PC and our method on the $\mathfrak{se}(2)$ version of the input graphs.

As shown in Table I, the KLDs of all three methods on Tree topology subgraphs are nearly identical. This result confirms the optimality of the pose composition method on Tree topology subgraphs, as previously discussed in Section IV-A. On the Circular topology, our method is slightly less optimal than NFR, whereas PC marginally outperforms NFR on small graphs (Duderstardt and EECS) but significantly underperforms on large graphs (Killian and Manhattan). We theorize the better-than-optimal performance of PC on small graphs stems from nonlinearity errors in the g2o-to-$\mathfrak{se}(2)$ conversion, which linearly approximates and only remains

accurate for small rotations. Finally, our method significantly outperforms PC on the Dense topology. Curiously, NFR never finishes computing on the Dense topology within our 30 minute time ceiling. We note that [22] does not include NFR results for the Dense topology.

In summary, the simplicity of our pose composition method provides speedup over NFR by greater than an order of magnitude for the Circular and Dense topologies. Our method performs especially well on the Dense topology, which NFR fails to finish in a reasonable amount of time. Although our spanning tree analysis in Section V-A helps explain optimal scale factors in simple graphs for various target topologies, our information-based scaling heuristics require more generalized optimality, despite achieving significantly better results than PC. Our approach yields better results for the Dense topology over the Tree topology on Killian and Manhattan, but only with a graph sparsity level of 50%. The result does not hold for higher sparsity levels. The optimal scaling factor still awaits discovery.

VI. CONCLUSIONS AND FUTURE WORK

We have demonstrated a fast and near optimal method for the nonlinear approximation of node marginalization on pose graphs based on the simple pose composition approach. Our relative-pose reparameterization view helps avoid the issue of linearization points and re-establishes the optimality of pose composition in full-rank tree target subgraph topology. We have also contributed a method to approximate a degenerate low-rank Gaussian distribution as a scaled version of the pose composition method. Due to the pose composition approach, our method is efficient and simple enough to implement on resource-constrained robot platforms. Although our method is fast and nearly optimal as demonstrated in the experiments, a method achieving optimality while being suitable for resource-constrained robotics platforms still remains an open and interesting research problem.

VII. ACKNOWLEDGMENTS

We would like to thank Mladen Mazuran for his NFR source code and datasets used in our experiments as well as fruitful discussions regarding the parameterization and optimality of the pose composition approach, Philip Fong for introducing us to this interesting problem, and Christopher Jones for his support on this paper.

REFERENCES

[1] Francis R Bach and Michael I Jordan. Beyond independent components: trees and clusters. *Journal of Machine Learning Research*, 4(Dec):1205–1233, 2003.

[2] Timothy D Barfoot and Paul T Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[3] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[4] Nicholas Carlevaris-Bianco and Ryan M. Eustice. Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. pages 5728–5735, 2013.

[5] Nicholas Carlevaris-Bianco and Ryan M Eustice. Long-term simultaneous localization and mapping with generic linear constraint node removal. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1034–1041. IEEE, 2013.

[6] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Transactions on Robotics*, 32(3):545–565, June 2016.

[7] Luca Carlone, Rosario Aragues, José A Castellanos, and Basilio Bona. A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research*, 33(7):965–987, 2014.

[8] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. 14(3):462–467, May 1968.

[9] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. 25(12):1181–1203, Dec 2006.

[10] Ethan Eade, Philip Fong, and Mario E Munich. Monocular graph slam with complexity reduction. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3017–3024. IEEE, 2010.

[11] Kevin Eckenhoff, Liam Paull, and Guoquan Huang. Decoupled, consistent node removal and edge sparsification for graph-based slam. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016.

[12] R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: Insights into sparsification. pages 3281–3288, Aug 2005.

[13] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. 22(6):1100–1114, Dec 2006.

[14] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. Jun 2007.

[15] John Michael Harris, Jeffry L Hirst, and Michael J Mossinghoff. *Combinatorics and graph theory*, volume 2. Springer, 2008.

[16] Guoquan Huang, Michael Kaess, and John J Leonard. Consistent sparsification for graph optimization. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 150–157. IEEE, 2013.

[17] Arieh Iserles, Hans Z Munthe-Kaas, Syvert P Nørsett, and Antonella Zanna. Lie-group methods. *Acta Numerica 2000*, 9:215–365, 2000.

[18] S.J. Julier. A sparse weight Kalman filter approach to simultaneous localisation and map building. volume 3, pages 1251 – 1256, 2001.

[19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. 31:217–236, Feb 2012.

[20] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[22] Mladen Mazuran, Wolfram Burgard, and Gian Diego Tipaldi. Nonlinear factor recovery for long-term slam. *The International Journal of Robotics Research*, 35(1-3):50–72, 2016.

[23] Mladen Mazuran, Gian Diego Tipaldi, Luciano Spinello, and Wolfram Burgard. Nonlinear graph sparsification for slam. In *Proceedings of Robotics: Science and Systems (RSS)*, pages 1–8, 2014.

[24] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. pages 2262–2269, May 2006.

[25] Liam Paull, Guoquan Huang, and John Leonard. A unified resource-constrained framework for graph slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016.

[26] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. A certifiably correct algorithm for synchronization over the special euclidean group. *arXiv preprint arXiv:1611.00128*, 2016.

[27] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular slam: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664. IEEE, 2010.

[28] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. 23(7-8):693–716, 2004.

[29] Matthew R. Walter, Ryan M. Eustice, and John J. Leonard. Exactly sparse extended information filters for feature-based SLAM. 26(4):335–359, April 2007.

[30] Jiacheng Wang and Edwin Olson. Robust pose graph optimization using stochastic gradient descent. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4284–4289. IEEE, 2014.