Generic Factor-Based Node Marginalization and Edge Sparsification for Pose-Graph SLAM

Nicholas Carlevaris-Bianco and Ryan M. Eustice

***Abstract*—This paper reports on a factor-based method for node marginalization in simultaneous localization and mapping (SLAM) pose-graphs. Node marginalization in a pose-graph induces fill-in and leads to computational challenges in performing inference. The proposed method is able to produce a new set of constraints over the elimination clique that can represent either the true marginalization, or a sparse approximation of the true marginalization using a Chow-Liu tree. The proposed algorithm improves upon existing methods in two key ways: First, it is not limited to strictly full-state relative-pose constraints and works equally well with other low-rank constraints such as those produced by monocular vision. Second, the new factors are produced in a way that accounts for measurement correlation, a problem ignored in other methods that rely upon measurement composition. We evaluate the proposed method over several real-world SLAM graphs and show that it outperforms other state-of-the-art methods in terms of Kullback-Leibler divergence.**

I. Introduction

Pose-graph simultaneous localization and mapping (SLAM) [1]–[5] has been demonstrated successfully over a wide variety of applications. Unfortunately, the standard pose-graph formulation is not ideal for long-term applications as the size of the graph grows with time and spatial extent—even if a robot is working in a finite region (since it must continue to add nodes and measurements to the graph in order to stay localized).

This paper seeks to address this challenge by developing a new method that allows one to remove nodes and factors from the graph, thereby reducing inference complexity and allowing for graph maintainability. Our proposed algorithm is designed so that it meets the following criteria:

- The algorithm produces a new set of independent factors using the current graph factors as input. The method does not require the full linearized information matrix as input.
- The algorithm is able to produce constraints that can represent exact node marginalization, as well as constraints that can represent a sparse Chou-Liu tree approximation of the dense marginal.
- The algorithm works equally well with non full-state constraints. Constraints with lower degree of freedom (DOF) than full state (e.g., bearing-only, range-only and partial state constraints) are handled under the same framework as full-state constraints, without special consideration.
- The new factors are produced in a way that does not double count measurement information. As we will show in §II, methods based on the pairwise composition of measurements produce pairwise constraints that are not independent, which leads to inconsistency in the graph.
- The computational complexity of the algorithm is dependent only on the number of nodes and factors in the elimination clique, not on the size of the graph beyond the clique.
- The algorithm does not require committing to a world-frame linearization point, rather, the new factors are parametrized in such a way as to use a local linearization that is valid independent of the global reference frame. This allows for the exploitation of methods that relinearize during optimization (e.g., [1], [2], [5]).

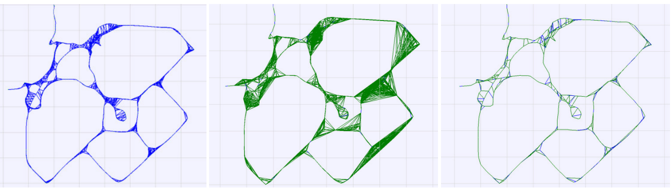

(a) Original graph (b) Dense GLC (c) Sparse GLC

Fig. 1: Depiction of dense-exact and sparse-approximate generic linear constraint (GLC) node removal for the Duderstat SLAM pose-graph. 33.3% of nodes from the original graph are removed. Green links represent new GLC constraints.

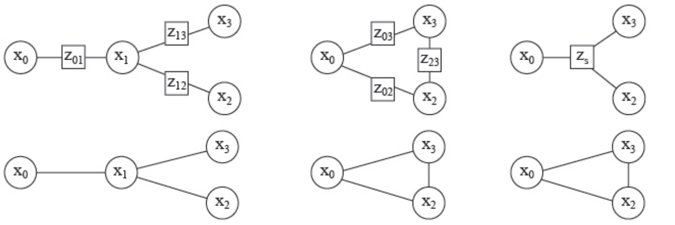
Methods that seek to slow the rate of growth of the pose-graph exist. In [6], an information-theoretic approach is used to add only non-redundant nodes and highly-informative measurements to the graph. Similarly, [7] induces new constraints between existing nodes when possible, instead of adding new nodes to the graph. In this formulation the number of nodes grows only with spatial extent, not with mapping duration—though the number of factors and connectivity density within the graph remain unbounded.

Methods that work directly on the linearized information matrix (best suited for filtering-based SLAM solutions) include [8]–[10]. In [8], weak links between nodes are removed to enforce sparsity. Unfortunately, this removal method causes the resulting estimate to be overconfident [11]. In [9], odometry links are removed in order to enforce sparsity in feature-based SLAM. Recently, [10] proposed an optimization-based method that minimizes the Kullback-Leibler divergence (KLD) of the

This work was supported in part by the National Science Foundation under award IIS-0746455, and in part by the Office of Naval Research under award N00014-12-1-0092.

N. Carlevaris-Bianco is with the Department of Electrical Engineering & Computer Science, University of Michigan, Ann Arbor, MI 48109, USA carlevar@umich.edu

R. Eustice is with the Department of Naval Architecture & Marine Engineering, University of Michigan, Ann Arbor, MI 48109, USA eustice@umich.edu



(a) Original graph (b) Composition (c) Marginalization

Fig. 2: Measurement composition vs. marginalization. The top row shows the factor graph; bottom row shows its Markov random field.

information matrix while enforcing a sparsity pattern and the requirement that the estimated information is conservative. This method performs favorably in comparison with [9] and [8], but requires a large matrix inversion in order to formulate the optimization problem, which limits its online utility.

For full-state constraints (i.e., 3- or 6-DOF relative-pose constraints depending on application) it is possible to compose constraints and their associated uncertainty. The basic composition functions for compounding and inversion are reported in [12]. Measurement composition is used in [13]–[15] in order to produce a new set of constraints when nodes are removed. In [13], all composed constraints are kept, causing fill-in within the graph. In order to preserve sparsity, a subset of the composed edges are pruned in [14] using a heuristic based on node degree. In [15], composed-edge removal is guided by a Chow-Liu tree calculated over the conditional information of the elimination clique.

These composition-based methods meet many of the aforementioned design criteria. They produce a new set of factors using the existing factors as input, the computational complexity is only dependent on the number of nodes and factors in the elimination clique, and the new factors can be re-linearized during subsequent optimization. However, as we show in §II, pairwise measurement composition is not marginalization, and yields inconsistent estimates in all but the simplest of graph topologies (since the composed pairwise constraints are assumed to be independent, which they are not). Additionally, it is not uncommon for a graph to be composed of many different types of non-full-state constraints, such as bearing-only, range-only and other partial-state constraints. In these heterogeneous cases, measurement composition quickly becomes complicated as the constraint composition rules for all possible pairs of measurement types must be defined.

The remainder of this paper is outlined as follows: In Section II we discuss the pitfalls associated with the use of measurement composition for node removal. Our proposed method is then described in Section III and experimentally evaluated in Section IV. Finally, Sections V and VI offer a discussion and concluding remarks.

II. Pairwise Composition $\neq$ Marginalization

Consider the simple pose-graph depicted in Fig. 2(a) where we show both its factor graph and Markov random field (MRF) representations. Suppose that we wish to marginalize node $\mathbf{x}_1$. Following [14], [15], and using the composition notation of [12], we can compose the pairwise measurements to produce the graph depicted in Fig. 2(b) as follows,

$$\begin{aligned}\mathbf{z}'_{02} &= h_1(\mathbf{z}_{01}, \mathbf{z}_{12}) = \mathbf{z}_{01} \oplus \mathbf{z}_{12},\\ \mathbf{z}'_{03} &= h_2(\mathbf{z}_{01}, \mathbf{z}_{13}) = \mathbf{z}_{01} \oplus \mathbf{z}_{13},\\ \mathbf{z}'_{23} &= h_3(\mathbf{z}_{12}, \mathbf{z}_{13}) = \ominus\mathbf{z}_{12} \oplus \mathbf{z}_{13}.\end{aligned} \tag{1}$$

These composed measurements are meant to capture the fully connected graph topology that develops in the elimination clique once $\mathbf{x}_1$ has been marginalized. In [14], [15], this composition graph forms the conceptual basis from which their link sparsification method then acts to prune edges and produce a sparsely connected graph. The problem with this composition is that the pairwise edges/factors in Fig. 2(b) are assumed to be independent, which they are not.

It should be clear that the composed measurements in (1) are correlated, as $\mathbf{z}'_{02}$, $\mathbf{z}'_{03}$ and $\mathbf{z}'_{23}$ share common information (e.g., $\mathbf{z}'_{02}$ and $\mathbf{z}'_{03}$ both share $\mathbf{z}_{01}$ as input), yet, if we treat these factors as strictly pairwise, we are unable to capture this correlation. Now consider instead a stacked measurement model defined as

$$\mathbf{z}_s = \begin{bmatrix}\mathbf{z}'_{02}\\ \mathbf{z}'_{03}\\ \mathbf{z}'_{23}\end{bmatrix} = h\left(\begin{bmatrix}\mathbf{z}_{01}\\ \mathbf{z}_{12}\\ \mathbf{z}_{13}\end{bmatrix}\right) = \begin{bmatrix}\mathbf{z}_{01} \oplus \mathbf{z}_{12}\\ \mathbf{z}_{01} \oplus \mathbf{z}_{13}\\ \ominus\mathbf{z}_{12} \oplus \mathbf{z}_{13}\end{bmatrix}. \tag{2}$$

Its first-order uncertainty is given as

$$\Sigma_s = \mathrm{H}\begin{bmatrix}\Sigma_{01} & 0 & 0\\ 0 & \Sigma_{12} & 0\\ 0 & 0 & \Sigma_{13}\end{bmatrix}\mathrm{H}^\top,$$

where

$$\mathrm{H} = \begin{bmatrix}\frac{\partial \mathbf{z}'_{02}}{\partial \mathbf{z}_{01}} & \frac{\partial \mathbf{z}'_{02}}{\partial \mathbf{z}_{12}} & 0\\ \frac{\partial \mathbf{z}'_{03}}{\partial \mathbf{z}_{01}} & 0 & \frac{\partial \mathbf{z}'_{03}}{\partial \mathbf{z}_{13}}\\ 0 & \frac{\partial \mathbf{z}'_{23}}{\partial \mathbf{z}_{12}} & \frac{\partial \mathbf{z}'_{23}}{\partial \mathbf{z}_{13}}\end{bmatrix}.$$

The joint composition in (2) produces the factor graph shown in Fig. 2(c), where in this formulation we see that Σ_s captures the correlation between the compounded measurements. In order to do this, it requires a trinary factor with support including all three variables,

$$\mathbf{z}_s = \begin{bmatrix}\mathbf{z}'_{02}\\ \mathbf{z}'_{03}\\ \mathbf{z}'_{23}\end{bmatrix} = f\left(\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_2\\ \mathbf{x}_3\end{bmatrix}\right) = \begin{bmatrix}\ominus\mathbf{x}_0 \oplus \mathbf{x}_2\\ \ominus\mathbf{x}_0 \oplus \mathbf{x}_3\\ \ominus\mathbf{x}_2 \oplus \mathbf{x}_3\end{bmatrix} + \mathbf{w}, \tag{3}$$

where $\mathbf{w} \sim \mathcal{N}(0, \Sigma_s)$.

It is this inability of pairwise factors to capture correlation between composed measurements that causes simple compounding to be wrong. Note that the graphs in Fig. 2(b) and Fig. 2(c) have the same Markov network representation and information matrix sparsity pattern. The difference between the binary and trinary factorization is only made explicit in the factor graph representation. The two observations: *(i)* that composed measurements are often correlated, and *(ii)* that representing the potential of an elimination clique with n nodes requires n-nary factors, will prove important in the remainder of the paper.

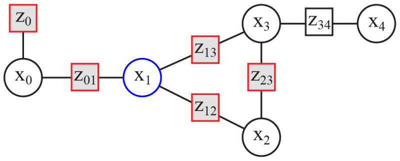


Fig. 3: Sample factor graph where node $\mathbf{x}_1$ is to be marginalized. Here $\mathbf{X}_m = [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$. The factors $\mathbf{Z}_m = [\mathbf{z}_0, \mathbf{z}_{01}, \mathbf{z}_{12}, \mathbf{z}_{23}, \mathbf{z}_{13}]$ (highlighted in red) are those included in calculating the target information, Λ_t.

III. Method

The proposed method consists of two main parts. First, we compute the information induced by marginalization over the elimination clique. We refer to this information matrix as the target information, Λ_t. Second, we use Λ_t to compute either *(i)* an exact n-nary factor that produces an equivalent potential over the elimination clique (in the case of dense node removal), or *(ii)* a sparse set of factors that best approximate the true distribution over the elimination clique using a Chow-Liu tree (in the case of sparsified node removal). Having computed this new set of factors, we can simply remove the marginalization node from the graph and replace its surrounding factors with the newly computed set.

A. Building the target information

The first step in the algorithm is to correctly identify the target information, Λ_t. Letting $\mathbf{X}_m \subset \mathbf{X}$ be the subset of nodes including the node to be removed and the nodes in its Markov blanket, and letting $\mathbf{Z}_m \subset \mathbf{Z}$ be the subset of measurement factors that only depend on the nodes in $\mathbf{X}_m$ we consider the distribution $p(\mathbf{X}_m|\mathbf{Z}_m) \sim \mathcal{N}^{-1}\big(\boldsymbol{\eta}_m, \Lambda_m\big)$. From Λ_m we can then compute the desired target information, Λ_t, by marginalizing out the elimination node using the standard Schur-complement form. For example, in the graph shown in Fig. 3, to eliminate node $\mathbf{x}_1$ we would first calculate Λ_m using the standard information-form measurement update equations [8], [11] as

$$\begin{aligned}\Lambda_m = \quad & \mathrm{H}_0^\top \Lambda_0 \mathrm{H}_0 + \mathrm{H}_{01}^\top \Lambda_{01} \mathrm{H}_{01} + \mathrm{H}_{12}^\top \Lambda_{12} \mathrm{H}_{12} \\ & + \mathrm{H}_{23}^\top \Lambda_{23} \mathrm{H}_{23} + \mathrm{H}_{13}^\top \Lambda_{13} \mathrm{H}_{13},\end{aligned}$$

where H_{ij} are the Jacobians of the observation models for measurements $\mathbf{z}_{ij}$ with information matrices Λ_{ij}, and then compute the target information as

$$\Lambda_t = \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\alpha\beta}^\top,$$

where $\Lambda_{\alpha\alpha}$, $\Lambda_{\alpha\beta}$ and $\Lambda_{\beta\beta}$ are the required sub-blocks of Λ_m with $\alpha = [\mathbf{x}_0, \mathbf{x}_2, \mathbf{x}_3]$ and $\beta = [\mathbf{x}_1]$. Note that, though this example only contains unary and binary factors, general n-nary factors are equally acceptable.

The key observation when identifying the target information is that, for a given linearization point, a single n-nary factor can recreate the potential induced by the original pairwise factors by adding the same information (i.e., Λ_m) to the graph. Moreover, because marginalization only affects the information matrix blocks corresponding to nodes *within* the elimination clique, an n-nary factor that adds the information contained in Λ_t to the graph will induce the same potential in the graph as true node marginalization at the given linearization point.

Note that the target information, Λ_t, *is not* the conditional distribution of the elimination clique given the rest of the nodes, i.e., $p(\mathbf{x}_0, \mathbf{x}_2, \mathbf{x}_3|\mathbf{x}_4, \mathbf{Z})$, nor is it the marginal distribution of the elimination clique, i.e., $p(\mathbf{x}_0, \mathbf{x}_2, \mathbf{x}_3|\mathbf{Z})$. Using either of these distributions as the target information results in a wrong estimate as information will be double counted when the n-nary factor is reinserted into the graph.

It is also important to note that the constraints in $\mathbf{Z}_m$ may be purely relative and/or low-rank (e.g., bearing or range-only) and, therefore, may not fully constrain $p(\mathbf{X}_m|\mathbf{Z}_m)$. This can cause Λ_t to be singular. Additionally, some of Λ_t's block-diagonal elements may also be singular. This will require special consideration in subsequent sections.

B. Generic linear constraints

Having defined a method for calculating the target information, Λ_t, we now seek to produce an n-nary factor that captures the same potential. We refer to this new n-nary factor as a generic linear constraint (GLC). Letting $\mathbf{x}_c$ denote a stacked vector of the variables within the elimination clique after node removal, we begin by considering an observation model that directly observes $\mathbf{x}_c$ with a measurement uncertainty that is defined by the target information:

$$\mathbf{z} = \mathbf{x}_c + \mathbf{w} \ \text{ where } \ \mathbf{w} \sim \mathcal{N}^{-1}\big(\mathbf{0}, \Lambda_t\big). \tag{4}$$

Setting the measurement value, $\mathbf{z}$, equal to the current linearization point, $\hat{\mathbf{x}}_c$, induces the desired potential in the graph. Unfortunately, the target information, Λ_t, may not be full rank, which is problematic for optimization methods that rely upon a square root factorization of the measurement information matrix [1], [5]. We can, however, use principle component analysis to transform the measurement to a lower-dimensional representation that is full rank.

We know that Λ_t will be a real, symmetric, positive semi-definite matrix due to the nature of its construction. In general then, it has an eigen-decomposition given by

$$\Lambda_t = \begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_q\end{bmatrix} \begin{bmatrix}\lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_q\end{bmatrix} \begin{bmatrix}\mathbf{u}_1^\top \\ \vdots \\ \mathbf{u}_q^\top\end{bmatrix} = \mathrm{U}\mathrm{D}\mathrm{U}^\top, \tag{5}$$

where U is a $p \times q$ matrix, D is a $q \times q$ matrix, p is the dimension of Λ_t, and $q = \mathrm{rank}(\Lambda_t)$. Letting $\mathrm{G} = \mathrm{D}^{\frac{1}{2}}\mathrm{U}^\top$ allows us to write a transformed observation model,

$$\mathbf{z}_{glc} = \mathrm{G}\mathbf{z} = \mathrm{G}\mathbf{x}_c + \mathbf{w}' \ \text{ where } \ \mathbf{w}' \sim \mathcal{N}^{-1}\big(\mathbf{0}, \Lambda'\big). \tag{6}$$

Using the pseudo-inverse [16], $\Lambda_t^+ = \mathrm{U}\mathrm{D}^{-1}\mathrm{U}^\top$, and noting that $\mathrm{U}^\top\mathrm{U} = \mathrm{I}_{q\times q}$, we find that

$$\Lambda' = (\mathrm{G}\Lambda_t^+\mathrm{G}^\top)^{-1} = (\mathrm{D}^{\frac{1}{2}}\mathrm{U}^\top(\mathrm{U}\mathrm{D}^{-1}\mathrm{U}^\top)\mathrm{U}\mathrm{D}^{\frac{1}{2}})^{-1} = \mathrm{I}_{q\times q}.$$

This GLC factor will contribute the desired target information back to the graph, i.e.,

$$\mathrm{G}^{\top}\Lambda'\mathrm{G} = \mathrm{G}^{\top}\mathrm{I}_{q\times q}\mathrm{G} = \Lambda_t,$$

but is itself non-singular. This is the key advantage of the proposed GLC method; it automatically determines the appropriate measurement rank such that Λ' is $q \times q$ and invertible, and G is an $p \times q$ new observation model that maps the p-dimensional state to the q-dimensional measurement.

C. Avoiding world-frame linearization in GLC

At this point, the GLC method still fails to meet our initial design criteria because it linearizes the potential with respect to the state variables in the *world-frame*. This may be acceptable in applications where a good world-frame linearization point is known prior to marginalization; however, in general, a more tenable assumption is that a good linearization point exists for the local *relative-frame* transforms between nodes within the elimination clique.

To adapt GLC so that it only locally linearizes the relative transformations between variables in the elimination clique, we first define a "root-shift" function that maps its world-frame coordinates, $\mathbf{x}_c$, to relative-frame coordinates, $\mathbf{x}_r$. Letting $\mathbf{x}^i_j$ denote the j^{th} pose in the i^{th} frame, the root-shift function for $\mathbf{x}_c$ becomes

$$\mathbf{x}_r = \begin{bmatrix} \mathbf{x}^1_w \\ \mathbf{x}^1_2 \\ \vdots \\ \mathbf{x}^1_n \end{bmatrix} = r\left(\mathbf{x}_c\right) = r\left(\begin{bmatrix} \mathbf{x}^w_1 \\ \mathbf{x}^w_2 \\ \vdots \\ \mathbf{x}^w_n \end{bmatrix}\right) = \begin{bmatrix} \ominus\mathbf{x}^w_1 \\ \ominus\mathbf{x}^w_1 \oplus \mathbf{x}^w_2 \\ \vdots \\ \ominus\mathbf{x}^w_1 \oplus \mathbf{x}^w_n \end{bmatrix}. \quad (7)$$

In this function the first node is arbitrarily chosen as the root of all relative transforms. The inclusion of the inverse of the root pose, $\mathbf{x}^1_w$, is important as it ensures that the Jacobian of the root-shift operation, R, is invertible, and allows for the representation of target information that is not purely relative.

To derive, instead of starting with a direct observation of the state variables, as in (4), we instead start with their root-shifted relative transforms,

$$\mathbf{z}_r = \mathbf{x}_r + \mathbf{w}_r \quad \text{where} \quad \mathbf{w}_r \sim \mathcal{N}^{-1}\big(\mathbf{0}, \Lambda_{t_r}\big). \qquad (8)$$

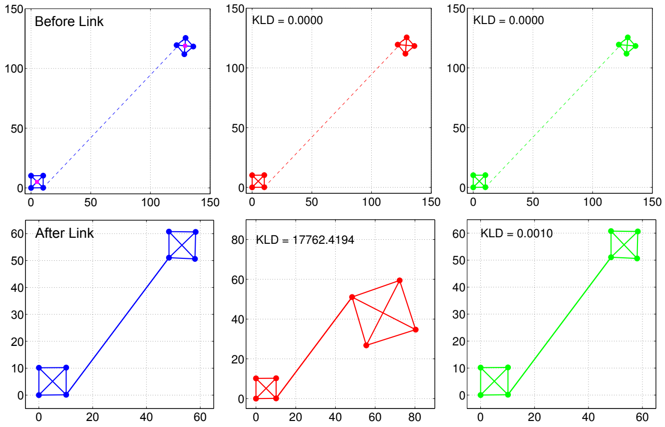
Here, the root-shifted target information, Λ_{t_r}, is calculated using the fact that the root-shift Jacobian, R, is invertible,

$$\Lambda_{t_r} = \mathrm{R}^{-\top}\Lambda_t\mathrm{R}^{-1}. \qquad (9)$$

Like the original target information, the root-shifted target information, Λ_{t_r}, will also be low-rank. Following the same principal component analysis procedure as before, we perform the low-rank eigen-decomposition $\Lambda_{t_r} = \mathrm{U}_r\mathrm{D}_r\mathrm{U}_r^{\top}$, which yields a new observation model,

$$\mathbf{z}_{glc_r} = \mathrm{G}_r r(\mathbf{x}_c) + \mathbf{w}'_r \quad \text{where} \quad \mathbf{w}'_r \sim \mathcal{N}^{-1}\big(\mathbf{0}, \Lambda'_r\big), \qquad (10)$$

where $\mathrm{G}_r = \mathrm{D}_r^{\frac{1}{2}}\mathrm{U}_r^{\top}$, and measurement information $\Lambda'_r = \mathrm{I}_{q\times q}$. Using the root-shifted linearization point to compute the measurement value, $\mathbf{z}_{glc_r} = \mathrm{G}_r r(\hat{\mathbf{x}}_c)$, will again induce the desired potential in the graph. Now, however, the advantage is that the GLC factor embeds the linearized constraint within a relative coordinate frame defined by the clique, as opposed to an absolute coordinate world-frame. Fig. 4 demonstrates this benefit.



(a) Truth (b) World-frame GLCs (c) Root-shifted GLCs

Fig. 4: Demonstration of root-shifted vs. world-frame GLC factors. Depicted is a simple graph (a) that is initially constructed with two well-connected clusters connected by a highly-uncertain and inaccurate link. The center (magenta) node in each cluster is removed inducing a GLC factor over each cluster. Subsequently, a second measurement is then added between the two clusters, correcting the world-frame location of the upper-right cluster. After adding the strong inter-cluster constraint, the graph with the world-frame linearized GLCs fails to converge to the correct optima (b), while the graph with root-shifted GLCs does (c). The Kullback-Leibler divergence from the true marginalization is displayed for each test.

D. Sparse approximate node removal

Exact node marginalization causes dense fill-in. As the number of marginalized nodes increases, this dense fill-in can quickly reduce the graph's sparsity and greatly increase the computational complexity of optimizing the graph [1], [5]. In [15], Kretzschmar and Stachniss insightfully propose the use of a Chow-Liu tree (CLT) [17] to approximate the individual elimination cliques as sparse tree structures.

The CLT approximates a joint distribution as the product of pairwise conditional distributions,

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_n) \approx p(\mathbf{x}_1)\prod_{i=2}^{n} p(\mathbf{x}_i|\mathbf{x}_{\mathrm{p}(i)}), \qquad (11)$$

where $\mathbf{x}_1$ is the root variable of the CLT and $\mathbf{x}_{\mathrm{p}(i)}$ is the parent of $\mathbf{x}_i$. The pairwise conditional distributions are selected such that the KLD between the original distribution and the CLT approximation is minimized. To construct it, the maximum spanning tree over all possible pairwise mutual information pairings is found (Fig. 5), where the mutual information between two Gaussian random vectors,

$$p(\mathbf{x}_i, \mathbf{x}_j) \sim \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_j \end{bmatrix}, \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix}\Big) \equiv \mathcal{N}^{-1}\Big(\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\eta}_j \end{bmatrix}, \begin{bmatrix} \Lambda_{ii} & \Lambda_{ij} \\ \Lambda_{ji} & \Lambda_{jj} \end{bmatrix}\Big), \qquad (12)$$

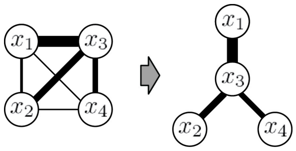


Fig. 5: Illustration of the Chow-Liu tree approximation. The magnitude of mutual information between variables is indicated by line thickness. The original distribution $p(x_1, x_2, x_3, x_4)$ (left), is approximated as $p(x_1)p(x_3|x_1)p(x_2|x_3)p(x_4|x_3)$ (right).

is given by [18]

$$I(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2} \log \left(\frac{|\Lambda_{ii}|}{|\Lambda_{ii} - \Lambda_{ij}\Lambda_{jj}^{-1}\Lambda_{ji}|} \right). \tag{13}$$

Like [15], we can apply the CLT approximation to sparsify our n-nary GLC factors; however, our implementation of CLT-based sparsification differs in a few subtle, yet important, ways. In [15], the maximum mutual information spanning tree is computed over the conditional distribution of the elimination clique given the remainder of the graph. This tree is then used to guide which edges should be composed and which edges should be excluded. This is not ideal for two reasons. First, the conditional distribution of the elimination clique is not the distribution that we wish to reproduce by our new factors (see §III-A). Second, pairwise measurement composition fails to track the proper correlation (see §II).

We address these issues by computing the CLT distribution (11) from the target information, Λ_t, which is the distribution that we wish to approximate, and then represent the CLT's unary and binary potentials as GLC factors.

1) CLT factors: To start, let's first consider CLT binary potentials, $p(\mathbf{x}_i|\mathbf{x}_{\mathrm{p}(i)})$, and in the following use $\mathbf{x}_j = \mathbf{x}_{\mathrm{p}(i)}$ as shorthand for the parent node of $\mathbf{x}_i$. We note that the target-information-derived joint marginal, $p_t(\mathbf{x}_i, \mathbf{x}_j)$, can be computed from Λ_t and written as in (12).[1] From this joint marginal, we can then easily write the desired conditional, $p_t(\mathbf{x}_i|\mathbf{x}_j) = \mathcal{N}\big(\boldsymbol{\mu}_{i|j}, \Sigma_{i|j}\big) \equiv \mathcal{N}^{-1}\big(\boldsymbol{\eta}_{i|j}, \Lambda_{i|j}\big)$, and express it as a constraint as

$$\mathbf{e} = \mathbf{x}_i - \boldsymbol{\mu}_{i|j} = \mathbf{x}_i - \Lambda_{ii}^{-1}(\boldsymbol{\eta}_i - \Lambda_{ij}\mathbf{x}_j), \tag{14}$$

where $\mathbf{e} \sim \mathcal{N}^{-1}\big(\mathbf{0}, \Lambda_{i|j}\big)$, and with Jacobian,

$$\mathrm{E} = \begin{bmatrix} \frac{\partial \mathbf{e}}{\partial \mathbf{x}_i} & \frac{\partial \mathbf{e}}{\partial \mathbf{x}_j} \end{bmatrix} = \begin{bmatrix} \mathrm{I} & \Lambda_{ii}^{-1}\Lambda_{ij} \end{bmatrix}. \tag{15}$$

Therefore, using the standard information-form measurement update, we see that this constraint adds information

$$\mathrm{E}^\top \Lambda_{i|j} \mathrm{E}, \tag{16}$$

where $\Lambda_{i|j}$ is simply Λ_{ii}.

Treating (16) as the input target information, we can produce an equivalent GLC factor for this binary potential using the techniques described in §III-B and §III-C. Similarly, the CLT's root unary potential, $p_t(\mathbf{x}_1)$, can also be implemented as a GLC factor by using the target-information-derived marginal information, Λ_{11}, and the sames techniques.

2) Pseudo-inverse: As discussed in Section III-A, the target information, Λ_t, is generally low rank. This is problematic for the joint marginal (12) and conditioning (14)–(15) calculations used to compute the CLT, as matrix inversions are required. To address this issue, in place of the inverse we use the generalized- or pseudo-inverse [16, §10.5], which can be calculated via an eigen-decomposition for real, symmetric, positive semi-definite matrices. For full-rank matrices the pseudo-inverse produces the same result as the true inverse, while for low rank matrices it remains well defined. Calculating the pseudo-inverse numerically requires defining a tolerance below which eigenvalues are considered to be zero. We found that our results are fairly insensitive to this tolerance and that automatically calculating the numerical tolerance using the machine epsilon produced good results. In our experiments we use $\epsilon \times n \times \lambda_{max}$ (the product of the machine epsilon, the size of the matrix, and the maximum eigenvalue) as the numerical tolerance.

3) Pinning: When calculating the pairwise mutual information, the determinants of both the conditional and marginal information matrices in (13) must be non-zero, which is again problematic because these matrices are generally low-rank as calculated from the target information, Λ_t. It has been proposed to consider the product of the non-zero eigenvalues as a pseudo-determinant [16], [19] when working with singular, multivariate, Gaussian distributions. Like the pseudo-inverse, this requires determining zero eigenvalues numerically. Experimentally, however, we found that in some cases the numerical instability in the pseudo-determinant's reliance on the numerical tolerance results in the edges being sorted incorrectly. This results in a non-optimal structure when the maximum mutual information spanning tree is built and, therefore, a slightly higher KLD from the true marginalization in some graphs.

Instead, we recognize that the CLT's construction requires only the ability to *sort* pairwise links by their relative mutual information (13), and not the actual value of their mutual information. A method that slightly modifies the input matrix so that its determinant is non-zero, without greatly affecting the relative ordering of the edges, would also be acceptable. Along these lines we approximate the determinant of a singular matrix using

$$|\Lambda| \approx |\Lambda + \alpha \mathrm{I}|. \tag{17}$$

This can be thought of as applying a low-certainty prior on the distribution, and we therefore refer to it as "pinning".[2] Experimentally we found the quality of the results to be less sensitive to the value of α than the numerical epsilon in the pseudo-determinant. We, therefore, elected to use pinning with $\alpha = 1$ in our experiments when evaluating the determinants in the pairwise mutual information (13).

[1] In this section, when we refer to marginal and conditional distributions, they are with respect to the target information, Λ_t, *not* with respect to the distribution represented by the full graph. See [3] for a summary of marginalization and conditioning operations for Gaussian variables.

[2] This is related to the derivation of the pseudo-determinant in [19], which uses a similar form in the limit as $\alpha \to 0$.

TABLE I: Experimental Datasets

Dataset	Robot	Factor Types	# Nodes	# Factors	Λ % NZ
Duderstadt Center	Segway	6-DOF odometry, 6-DOF laser scan-matching	552	1,774	1.12%
EECS Building	Segway	6-DOF odometry, 6-DOF laser scan-matching	611	2,134	1.20%
USS Saratoga	HAUV	6-DOF odometry, **5-DOF monocular-vision**, 1-DOF depth	1,513	5,433	0.35%

TABLE II: Experimental Results

		Dense GLC				**Sparse GLC**				
% Nodes Removed		25.0 %	33.3 %	50.0 %	66.6 %	25.0 %	33.3 %	50.0 %	66.6 %	75.0 %
Duderstadt Center	KLD	2.481	0.515	0.009	1.710E-8	7.295	5.793	4.219	5.855	5.939
EECS Building	KLD	7.630	3.882	1.679E-8	8.207E-8	13.170	11.944	7.313	11.204	16.540
USS Saratoga	KLD	108.040	77.000	0.708	2.682	113.481	94.836	10.216	3.907	1.837
Duderstadt Center	time (ms/node)	5.2	17.4	78.5	3.2E4	10.1	10.0	7.4	7.3	7.0
EECS Building	time (ms/node)	840.8	538.1	4.6E4	6.0E4	78.5	49.9	16.4	11.7	15.2
USS Saratoga	time (ms/node)	6.4	1.1E3	4.5E3	1.2E4	7.3	7.2	6.0	4.9	4.4

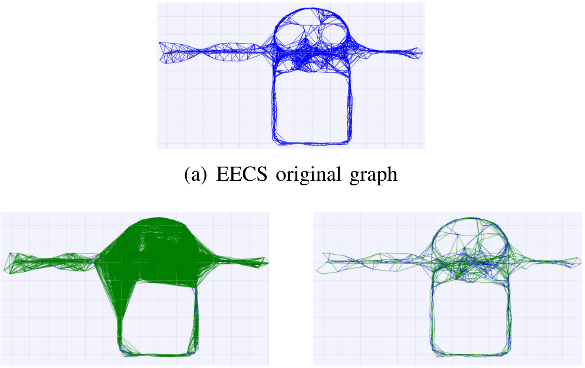
		Dense Pairwise Measurement Composition				**Sparse Pairwise Measurement Composition**				
% Nodes Removed		25.0 %	33.3 %	50.0 %	66.6 %	25.0 %	33.3 %	50.0 %	66.6 %	75.0 %
Duderstadt Center	KLD	204.200	473.213	3.544E4	N/A	7.198	7.176	23.461	24.399	158.209
EECS Building	KLD	9.984E4	2.871E4	N/A	N/A	19.282	13.188	33.840	414.370	411.717

E. Computational complexity

The core operations that GLC relies on, in and of themselves, are computationally expensive. The CLT approximation has a complexity of $\mathcal{O}(m^2 \log m)$, where m is the number of nodes. Matrix operations on the information matrix with n variables, including the eigen-decomposition, matrix multiplication, and inversion operations, have a complexity of $\mathcal{O}(n^3)$. Fortunately, the input size for these operations is limited to the number of nodes within the elimination clique, which in a SLAM pose-graph is controlled by the perceptual radius. In general, the number of nodes and variables in an elimination clique is much less than the total number of nodes in the full-graph, which makes GLC's calculations easily feasible.

IV. Experimental Results

The proposed algorithm was implemented using iSAM [5], [20] as the underlying optimization engine. The code is available for download within the iSAM repository [21]. For comparison, a dense measurement composing method as described in §II, and a sparse measurement composing method based upon CLT-guided node removal, as proposed in [15], were also implemented. For evaluation we use three SLAM pose-graphs: The first two graphs were built using data from a Segway ground robot equipped with a Velodyne HDL-32E laser scanner as the primary sensing modality. The third graph was produced by a Hovering Autonomous Underwater Vehicle (HAUV) performing monocular SLAM for autonomous ship hull inspection [22]. These graphs are characterized in Table I and a depiction of each are shown in Fig. 1, Fig. 6 and Fig. 7. In the following experiments, the original full graph is first optimized using iSAM. Then the different node removal algorithms are each performed to remove a percentage of nodes evenly spaced throughout the trajectory. Finally, the graphs are again optimized in iSAM. For each experiment the true marginal distribution is recovered by obtaining the linearized information matrix about the optimization point and performing Schur complement marginalization, which provides a ground-truth distribution.



(a) EECS original graph

(b) Dense GLC

(c) Sparse GLC

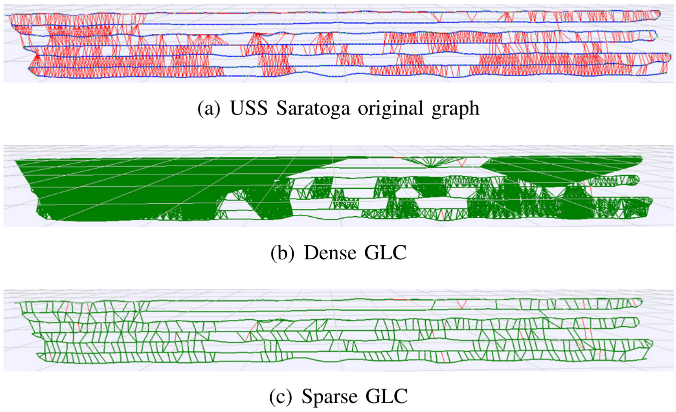
Fig. 6: Comparison of the original graph, dense-exact GLC, and sparse-approximate GLC node removal for the EECS graph. 33.3% of original nodes are removed. Blue links represent 6-DOF constraints, and green links represent new GLC constraints.

A summary of our results are provided in Table II, which shows the KL-divergence from the true marginalization and average computation time per node removed, as an increasing percentage of nodes are removed from the graph. Results for dense-exact and sparse-approximate GLC are provided for all three graphs, while results for dense and sparse-approximate measurement composition are provided only for the Duderstadt and EECS datasets. The Saratoga graph is excluded as it contains 5-DOF monocular relative-pose constraints for which measurement composition is ill-defined.

A. Dense GLC node removal

We first consider the results for our method when performing exact node removal with dense fill-in. Visual depictions of the resulting dense GLC graphs for the Duderstadt, EECS, and Saratoga datasets are shown in Fig. 1(b), Fig. 6(b), and Fig. 7(b), respectively.

To put GLC's KLD values from Table II into perspective, we look at the case with the highest KLD, which is the Saratoga graph with 25% of nodes removed (i.e., KLD = 108.040). Under these conditions the reconstructed graph has



(a) USS Saratoga original graph

(b) Dense GLC

(c) Sparse GLC

Fig. 7: Comparison of the original graph, dense-exact GLC, and sparse-approximate GLC node removal for the Saratoga graph. 50% of original nodes are removed. Blue, red and green links represent 6-DOF constraints, 5-DOF monocular vision constraints, and new GLC constraints, respectively.

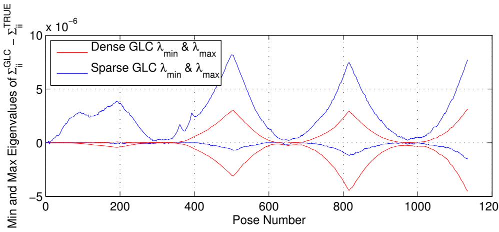
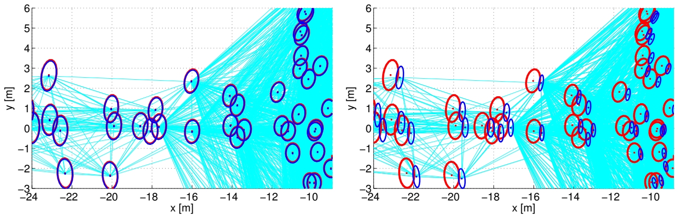


Fig. 8: Accuracy of GLC-derived marginals for the Saratoga data set with 25% of nodes removed. The min and max eigenvalues of the difference between the GLC marginals and the true marginals for each pose are plotted for both dense and sparse GLC. Note the eigenvalue scale is $\mathcal{O}(10^{-6})$.

a mean squared error in translation and rotation of $0.64\,\mathrm{mm}$ and $1.2\,\mathrm{mrad}$, respectively, when compared to the original baseline pose-graph SLAM result. To more systematically investigate the accuracy of GLC's marginal pose uncertainties, a plot of the minimum and maximum eigenvalues of the difference between the GLC marginals and the true marginals, $\Sigma_{ii}^{\mathsf{GLC}} - \Sigma_{ii}^{\mathsf{TRUE}}$, is shown in Fig. 8. In the ideal case all eigenvalues of this difference will be zero, indicating perfect agreement between GLC and the true marginalization. Eigenvalues larger than zero indicate conservative estimates while those less than zero indicate over-confidence. For dense GLC we see that these eigenvalues are on the order of 10^{-6} in magnitude, indicating excellent agreement between GLC and the true marginalization.

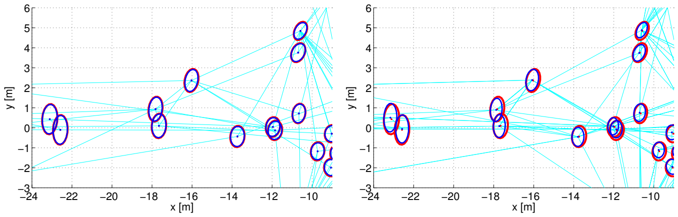
Considering the results for dense measurement composition, Table II shows that it performs quite poorly—as more nodes are removed, the KLD increases. This is because dense pairwise measurement composition fails to properly track the correlation that develops between composed measurements (as demonstrated in §II); thus, the higher the connectivity in the graph, the more measurement information gets double counted when compounding. This results in overconfidence as well as a shift in the optimal mean (Fig. 9(b)). In fact, for both the EECS and Duderstadt graphs, a point was reached where the constraints were so overconfident due to node removal that the iSAM optimization diverged—these cases are labeled as "N/A" in Table II.



(a) Dense GLC

(b) Dense measurement composition

Fig. 9: Sample 3-σ uncertainty ellipses for the EECS graph with 33.3% node removal using dense GLC and dense measurement composition. The true marginalization uncertainties are shown in red.



(a) Sparse GLC

(b) Sparse measurement composition

Fig. 10: Sample 3-σ uncertainty ellipses for the EECS graph with 75% node removal using sparse GLC and sparse measurement composition, as proposed in [15]. The true marginalization uncertainties are shown in red.

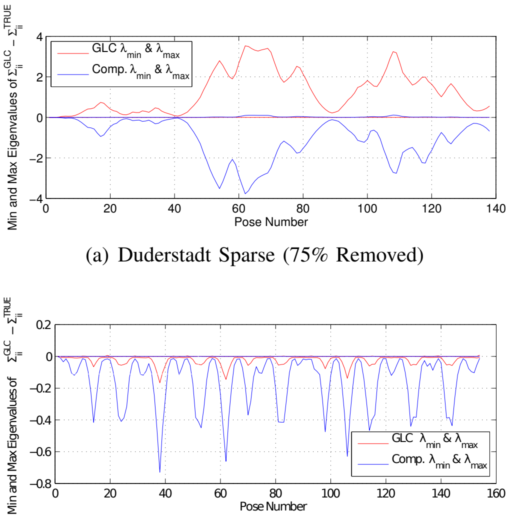
B. *Sparse-approximate GLC node removal*

Next we consider the results for sparse-approximate GLC marginalization. Table II shows that in many instances the KLD for sparse-approximate GLC is only slightly worse than that of dense-exact GLC—indicating that very little graph information is lost due to CLT sparsification. Visual examples for sparsification on the Duderstadt, EECS, and Saratoga graphs are shown in Fig. 1(c), Fig. 6(c) and Fig. 7(c), respectively.

Considering the results for sparse measurement composition, Table II shows that, unlike dense measurement composition, sparse measurement composition performs reasonably well, especially when removing a smaller percentage of nodes. This is because information double counting during measurement composition accumulates to a lesser extent than in the dense case because of sparsification. However, as the percentage of removed nodes increases, we see that sparse measurement composition produces a significantly less accurate and more inconsistent result than sparse GLC (Fig. 11). Visually, however, sparse composition's marginal 3-σ uncertainty ellipses appear to be quite reasonable, as depicted in Fig. 10 for the EECS graph, though, we see that they are less accurate than sparse GLC.

V. Discussion and Future Work

When considering the application of the proposed method, there are a few things to consider, some of which we hope to address in future work. First, when performing GLC, a



(a) Duderstadt Sparse (75% Removed)

(b) EECS Sparse (75% Removed)

Fig. 11: Comparison of marginal distribution accuracy between sparse GLC and sparse measurement composition, as proposed in [15], for the Duderstadt and EECS graphs. The min and max eigenvalues of the difference between the sparsified marginals and the true marginals for each pose are plotted. The results show that sparse GLC consistently outperforms sparse measurement composition in approaching the true marginal.

good linearization point for the relative transforms within the elimination clique must exist. This affects when it is appropriate to remove nodes, especially if performing online node removal. Second, because the target information is low rank, we use "pinning" to compute the mutual information when building the CLT and therefore, cannot guarantee that this yields a minimum KLD from the true distribution (though our experimental results show that we achieve a significantly lower KLD than other state-of-the-art methods). Third, because the CLT approximation itself is not guaranteed to be conservative, we cannot guarantee a conservative estimate when performing sparse approximate GLC node removal.

In fact, our results showed that CLT-based GLC sparse approximation can be either slightly conservative (Fig. 8 and Fig. 11(a)), or slightly over-confident (Fig. 11(b)). While our proposed GLC method avoids inconsistency pitfalls associated with measurement compounding, and accurately recreates the CLT, it may still be slightly overconfident if the CLT approximation is. In this regard, the method proposed in [10], which optimizes the KLD of a sparse distribution while enforcing a consistency constraint, could provide a way forward toward this end.

VI. Conclusions

We presented a factor-based method for node removal in SLAM pose-graphs. This method can be used to alleviate some of the computational challenges in performing inference over long-term pose-graphs by reducing the graph size and density. The proposed method is able to represent either exact marginalization, or a sparse approximation of the true marginalization, in a consistent manner over a heterogeneous collection of constraints. We experimentally evaluated the proposed method over several real-world SLAM graphs and showed that it outperformed other state-of-the-art methods in terms of Kullback-Leibler divergence.

References

[1] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[2] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. Robot. and Automation*, 2006, pp. 2262–2269.

[3] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1100–1114, Dec. 2006.

[4] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1066–1077, 2008.

[5] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[6] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact pose SLAM," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 78–93, Feb. 2010.

[7] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *RSS Workshop on Long-term Operation of Autonomous Robotic Systems in Changing Environments*, 2012.

[8] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7-8, pp. 693–716, Jul.-Aug. 2004.

[9] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *Int. J. Robot. Res.*, vol. 26, no. 4, pp. 335–359, Apr. 2007.

[10] J. Vial, H. Durrant-Whyte, and T. Bailey, "Conservative sparsification for efficient and consistent approximate estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2011, pp. 886–893.

[11] R. Eustice, M. Walter, and J. Leonard, "Sparse extended information filters: insights into sparsification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Aug. 2005, pp. 3281–3288.

[12] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer-Verlag, 1990, pp. 167–193.

[13] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2009, pp. 1156–1163.

[14] E. Eade, P. Fong, and M. E. Munich, "Monocular graph slam with complexity reduction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2010, pp. 3017–3024.

[15] H. Kretzschmar and C. Stachniss, "Information-theoretic compression of pose graphs for laser-based SLAM," *Int. J. Robot. Res.*, vol. 31, pp. 1219–1230, 2012.

[16] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and its Applications*. John Wiley & Sons, 1971.

[17] C. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. on Info. Theory*, vol. 14, pp. 462–467, 1968.

[18] A. Davison, "Active search for real-time vision," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, Oct. 2005, pp. 66–73.

[19] T. P. Minka, "Inferring a gaussian distribution," MIT Media Lab, Tech. Rep., 2001.

[20] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Robot. and Autonmous Syst.*, vol. 57, pp. 1198–1210, Dec. 2009.

[21] M. Kaess, H. Johannsson, and J. Leonard, "Open source implementation of iSAM," http://people.csail.mit.edu/kaess/isam, 2010.

[22] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, Oct. 2012.